Deep Short-Term Weather Elements Forecasting

Siamak Mehrkanoon

Department of Information and Computing Sciences, Utrecht University

Deep Learning and Artificial Intelligence Summer/Winter School (DLAI7), 17-21 July 2023.

Outline:

- ❑ Introduction
- ❑ Overview of recent proposed models
- ❑ Numerical Experiments

Introduction

:

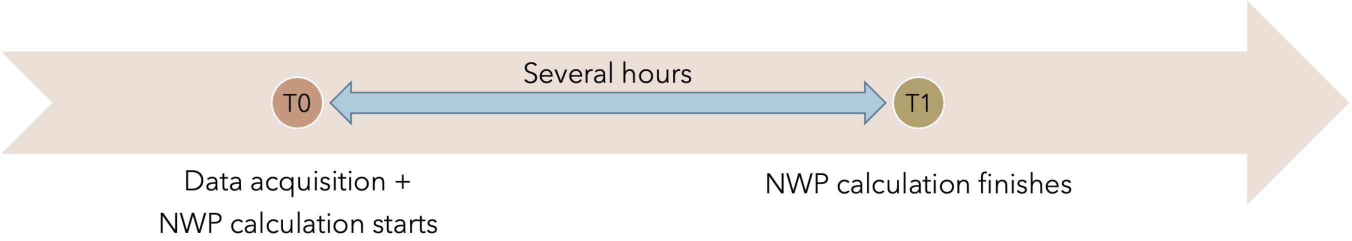
The importance of weather forecasting can be seen in:

- Agriculture and Tourism
- Transportation
- Aviation
- Mining
- Construction
- Sports
- ….

Weather prediction :

❑ Numerical weather prediction (model-driven):

- Uses mathematical models derived from physical principles to predict the weather variables
- Requires immense computing power and time
- The uncertainties in the initial conditions of the governed differential equations (i.e. measurement noise)
- Incomplete understanding of complex atmospheric processes (i.e. process noise)

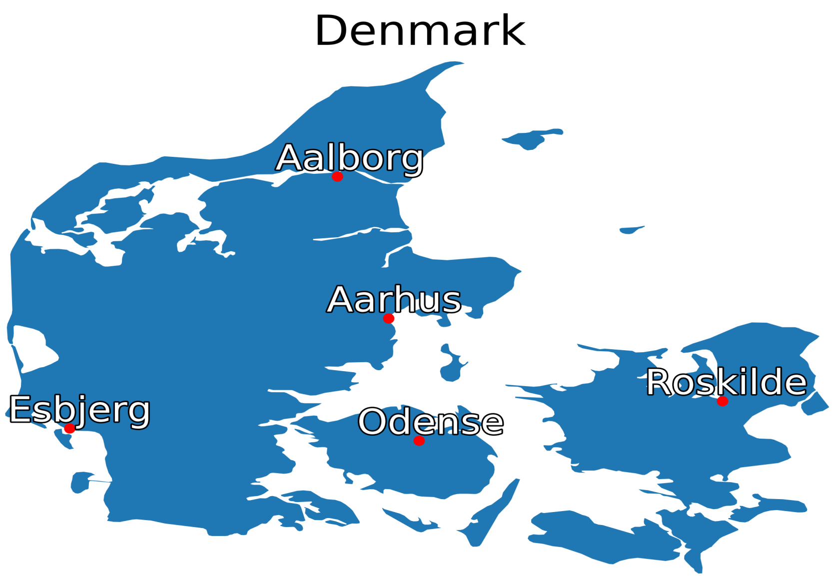


❑ Data driven modelling:

- Learn from spatio-temporal historical weather variables to predict the future of the target variables.

Problem statement :

- Data driven modelling of complex systems.
- Wind speed prediction using spatio-temporal historical records.
- Precipitation nowcasting



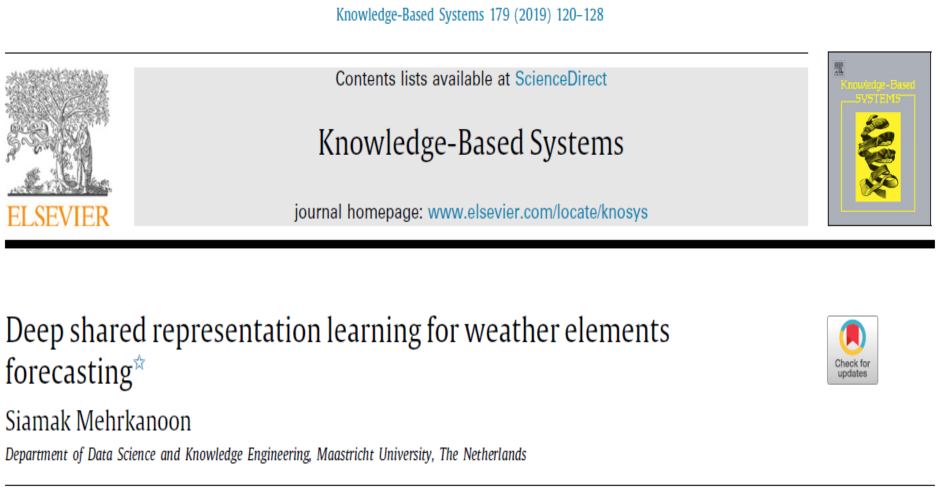
Related works on weather elements forecasting:

- RNN based models
- CNN based models
- Graph based models
- Encoder-Decoder based models

- S. Xingjian et al., Convolutional LSTM network: a machine learning approach for precipitation nowcasting, NIPS 2015, pp. 802-810
- Duan Jikai et al., Short-term wind speed forecasting using recurrent neural networks with error correction, Energy, 2021.
- S. Mehrkanoon, Deep Shared representation learning for weather elements forecasting, Knowledge-based systems, vol. 79, pp. 120-128, 2019.
- K. Trebing and S. Mehrkanoon, Wind speed prediction using multidimensional convolutional neural networks, IEEE-SSCI, vol. 79, pp. 713-720, 2020.
- T. Stanczyk and S. Mehrkanoon, Deep Graph Convolutional Networks for Wind Speed Prediction, In proc. of ESANN, pp. 147-152, 2021.
- S. Mehrkanoon, SmaAt-UNet: Precipitation nowcasting using a small attention-UNet architecture, Pattern Recognition Letters, Vol. 145, Pages 178-186, 2021.
- Jesús García Fernández et al., Deep coastal sea elements forecasting using UNet-based models, Knowledge-Based Systems, vol. 252, pp. 2022.
- Jesús García Fernández et al , Broad-UNet: Multi-scale feature learning for nowcasting tasks, Neural Networks, vol 144, pp. 419-427, 2021.

Let us assume that:

- The number of weather stations is q.
- Total number of weather elements is p.
- $y_j^{s_i}(t)$: The j-th weather element of the i-th station at time t.


Knowledge-Based Systems 179 (2019) 120–128

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Deep shared representation learning for weather elements forecasting☆

Siamak Mehrkanoon

Department of Data Science and Knowledge Engineering, Maastricht University, The Netherlands

For modelling the target $y_j^{s_1}(t)$, one can construct the following regressor vector at time t.

$$y_j^{s_1}(t) = f\underbrace{\begin{pmatrix} y_1^{s_1}(t-1),\ldots,y_p^{s_1}(t-1), \;\; \ldots , \;\; y_1^{s_1}(t-d),\ldots,y_p^{s_1}(t-d), \\ \cdot \\ \cdot \\ \cdot \\ y_1^{s_q}(t-1),\ldots,y_p^{s_q}(t-1), \;\; \ldots , \;\; y_1^{s_q}(t-d),\ldots,y_p^{s_q}(t-d) \end{pmatrix}}_{z(t)} + e(t)$$

$$
\begin{aligned}
z(t) = [&y_1^{s_1}(t-1), \ldots, y_p^{s_1}(t-1), \ldots, y_1^{s_1}(t-d), \ldots, y_p^{s_1}(t-d), \\
&\vdots \\
&y_1^{s_q}(t-1), \ldots, y_p^{s_q}(t-1), \ldots, y_1^{s_q}(t-d), \ldots, y_p^{s_q}(t-d)],
\end{aligned}
$$

$$
\underbrace{\begin{bmatrix} z(t-m) \\ z(t-m+1) \\ \bullet \\ \bullet \\ \bullet \\ z(t) \end{bmatrix}}_{\text{Input space}}
\xrightarrow[f(\cdot)]{\text{nonlinear mapping}}
\underbrace{\begin{bmatrix} y_j^{s_1}(t-m) \\ y_j^{s_1}(t-m+1) \\ \bullet \\ \bullet \\ \bullet \\ y_j^{s_1}(t) \end{bmatrix}}_{\text{Targer space}}
$$

Fig. 1. Nonlinear mapping from input weather data to a target data.

S. Mehrkanoon, Deep Shared representation learning for weather elements forecasting, Knowledge-based systems, vol. 79, pp. 120-128, 2019.

Deep CNN for weather forecasting:

- Exploit the spatio-temporal structure of the input data.

- Each regressor vector is casted into a tensor with (stations,lags,variables) as (height,width,channel).

Knowledge-Based Systems 179 (2019) 120–128

Contents lists available at ScienceDirect

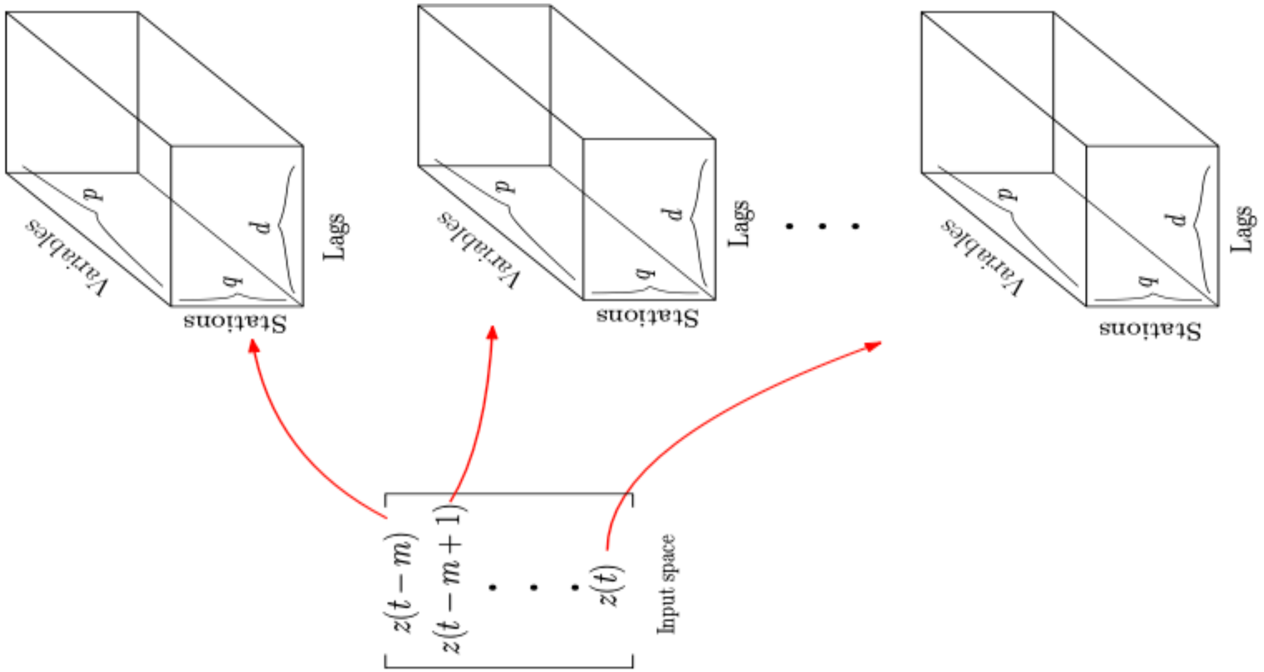
Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Deep shared representation learning for weather elements forecasting☆

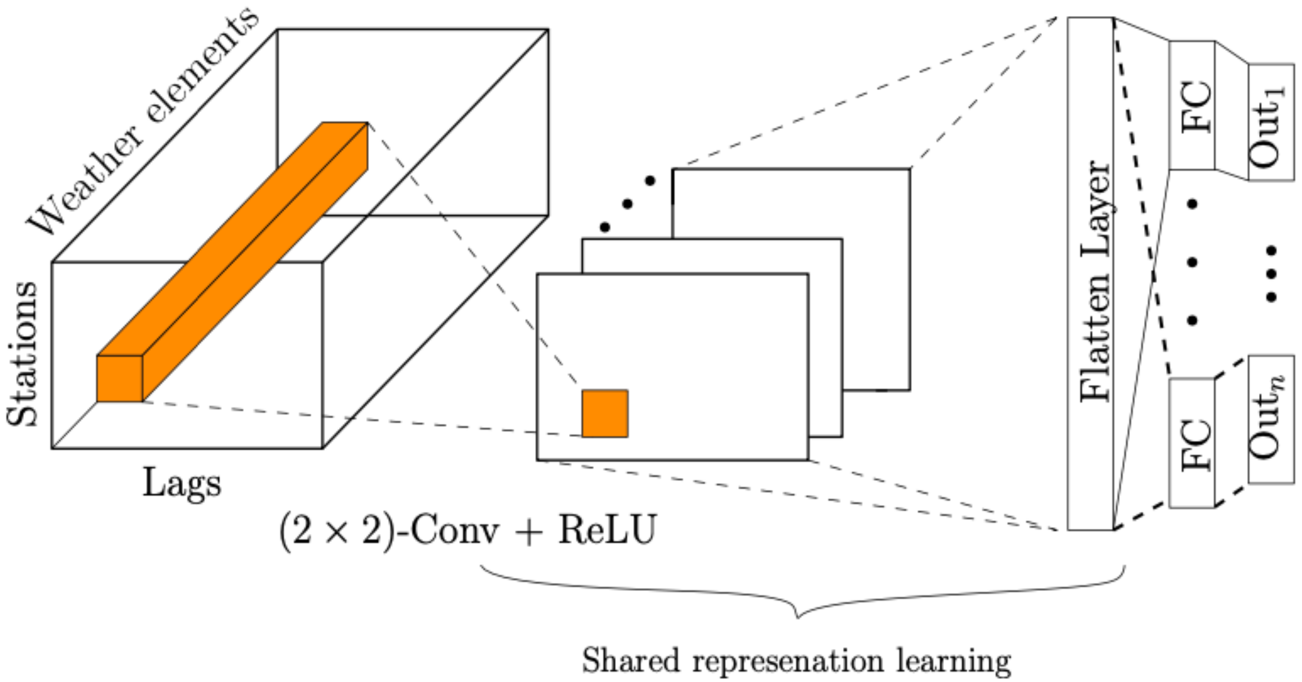
Siamak Mehrkanoon

Department of Data Science and Knowledge Engineering, Maastricht University, The Netherlands



2D-CNN based model:

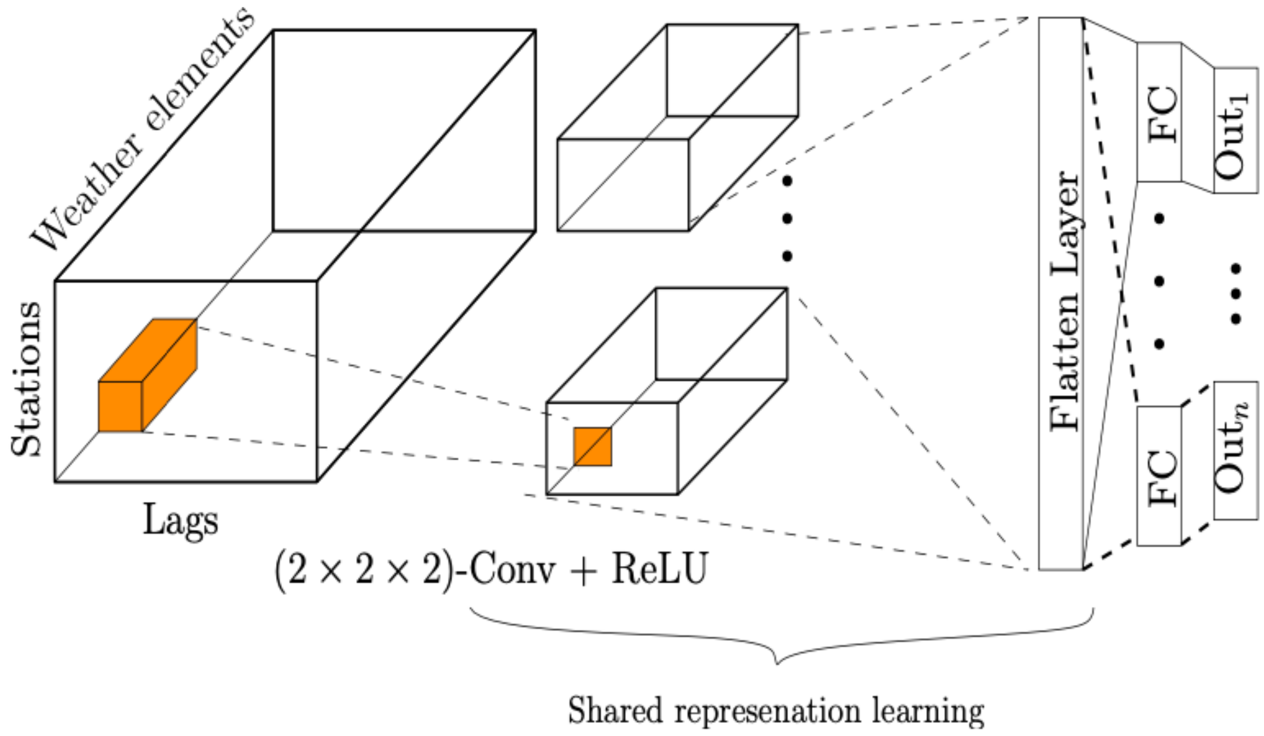
- New shared representations of the data are obtained by convolving the learned kernels over weather stations and lags dimensions.



S. Mehrkanoon, Deep Shared representation learning for weather elements forecasting, Knowledge-based systems, vol. 79, pp. 120-128, 2019.

3D-CNN based model:

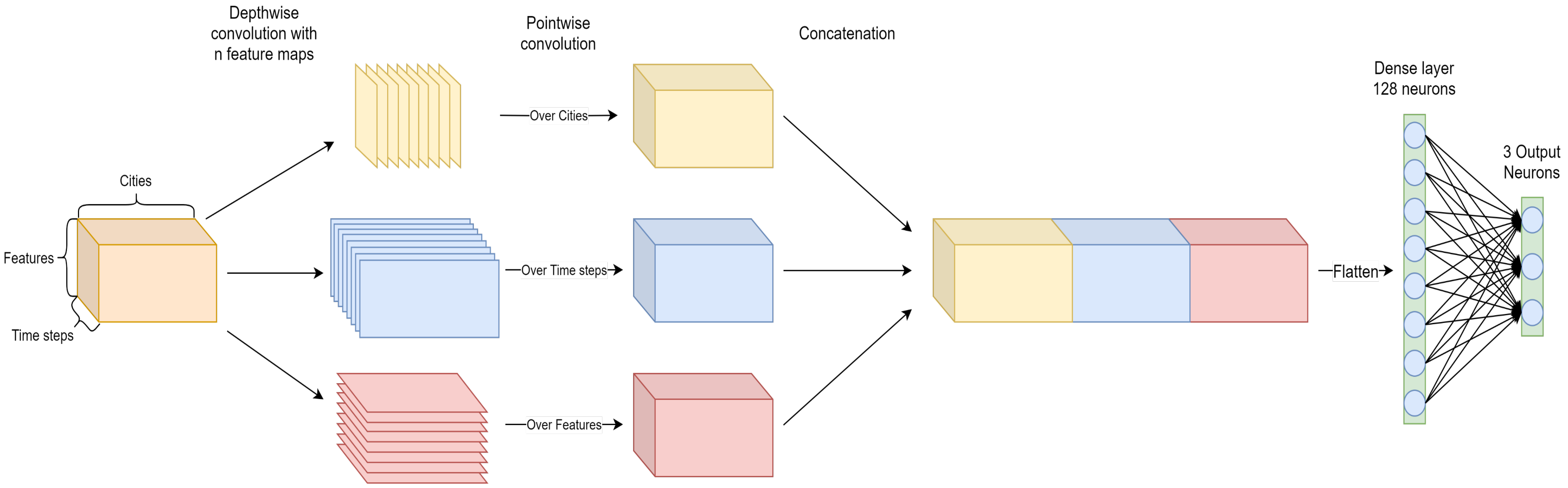
- Three-dimensional filters will be slided along the three dimensions (over weather stations, lags dimensions and weather elements).



S. Mehrkanoon, Deep Shared representation learning for weather elements forecasting, Knowledge-based systems, vol. 79, pp. 120-128, 2019.

Multidimensional model:

Apply **depthwise-separable convolutions (DSC)** to all three input dimensions.



https://github.com/HansBambel/multidim_conv

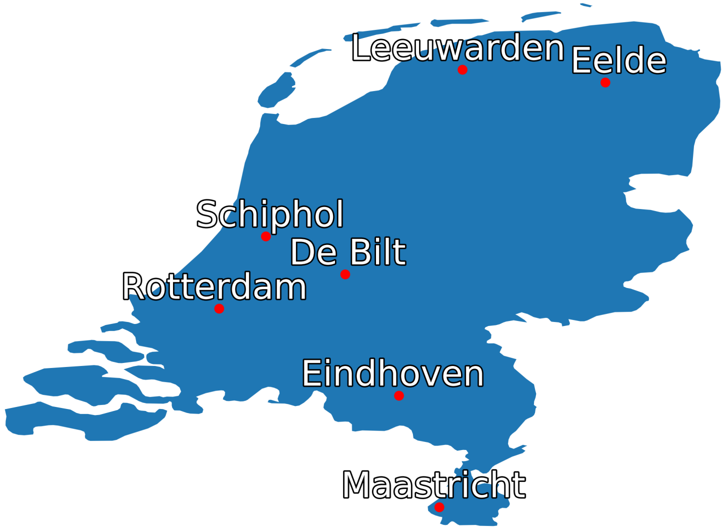
K. Trebing and S. Mehrkanoon, Wind speed prediction using multidimensional convolutional neural networks, IEEE Symposium Series on Computational Intelligence (IEEE-SSCI), vol. 79, pp. 713-720, 2020.

Numerical Experiments: Wind speed

Denmark

- Hourly measurements from 5 cities:
  - Temperature
  - Pressure
  - Wind speed
  - Wind direction
- Training: 2000-2009, Test: 2010
- 96,402 samples
- Sample shape: 5x4x4
- Same dataset as used in [13]
- Predict 6h, 12h, 18h, 24h ahead

Netherlands



- Hourly measurements from 7 cities:
  - Wind speed
  - Wind direction
  - Temperature
  - Dew point
  - Air pressure
  - Rain amount
- Training: 2011-2018, Test: 2019
- 81,000 samples
- Sample shape: 7x6x6
- Predict 1h, 2h, 3h, 4h ahead

Denmark dataset:

Table 5
The MAEs (mean absolute errors) of the proposed models, the NARX and LSTM models for (6 and 12) h ahead wind speed prediction of three stations located in Denmark.

Hours ahead	Station	Method				
		3d-CNN	2d-CNN	1d-CNN	NARX	LSTM
6	Esbjerg	1.40	1.42	1.44	1.59	1.54
	Odense	0.62	0.63	0.63	0.68	0.86
	Roskilde	1.48	1.50	1.52	1.56	1.49
12	Esbjerg	1.71	1.75	1.75	1.81	1.77
	Odense	0.79	0.80	0.82	0.86	1.05
	Roskilde	1.84	1.90	1.92	1.96	1.79

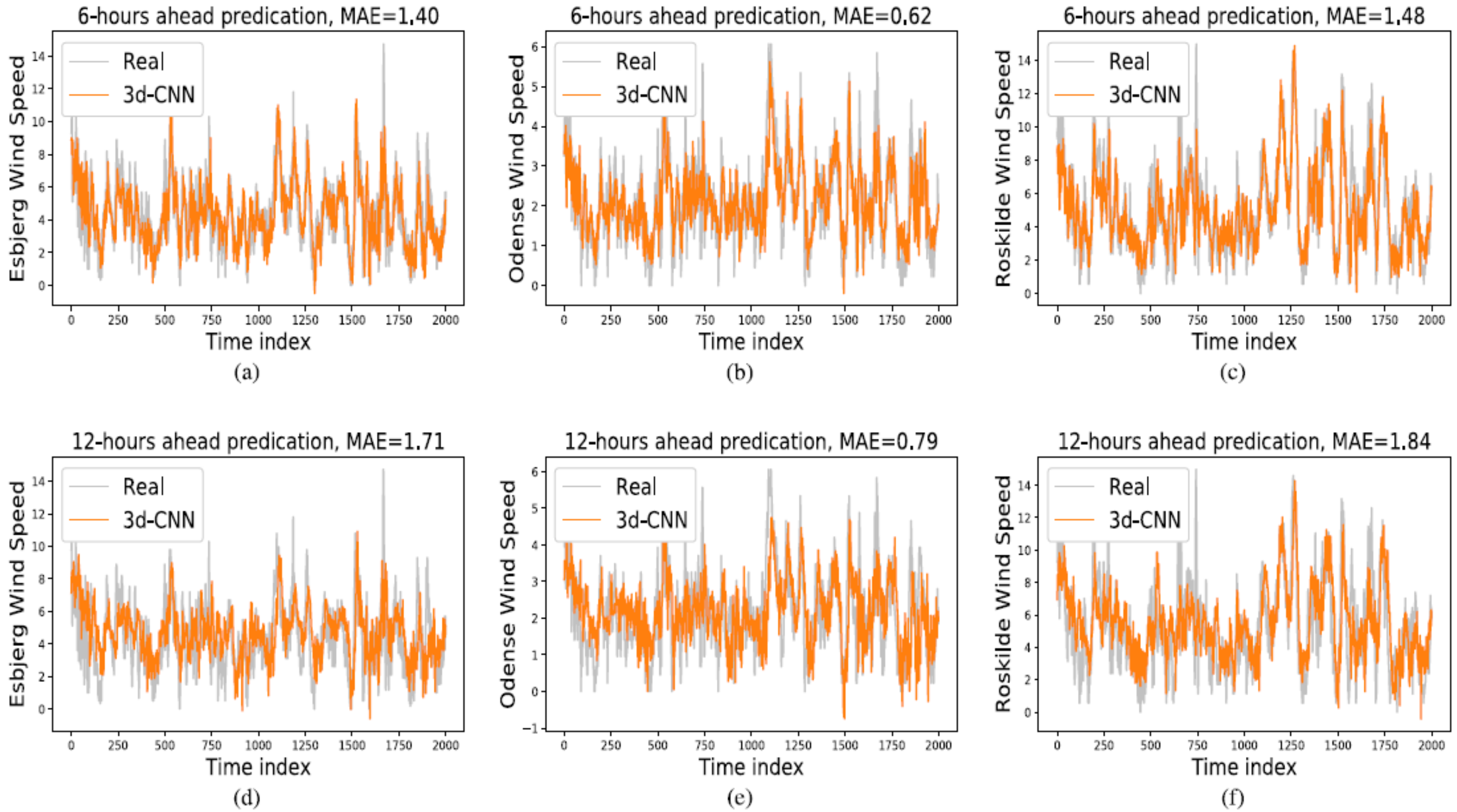
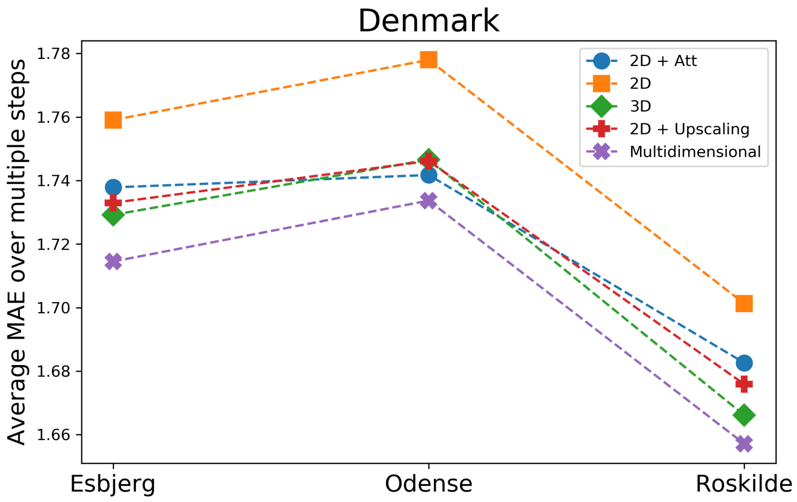


Fig. 12. Illustrations of the obtained forecasts for a subset of test dataset. (a,b,c) The Obtained 6 h ahead wind speed forecasts using the proposed 3d-CNN model for three stations. (e,f,g) The Obtained 12 h ahead wind speed forecasts using the proposed 3d-CNN model for three stations. The reported MAEs correspond to the entire test dataset.

S. Mehrkanoon, Deep Shared representation learning for weather elements forecasting, Knowledge-based systems, vol. 79, pp. 120-128, 2019.

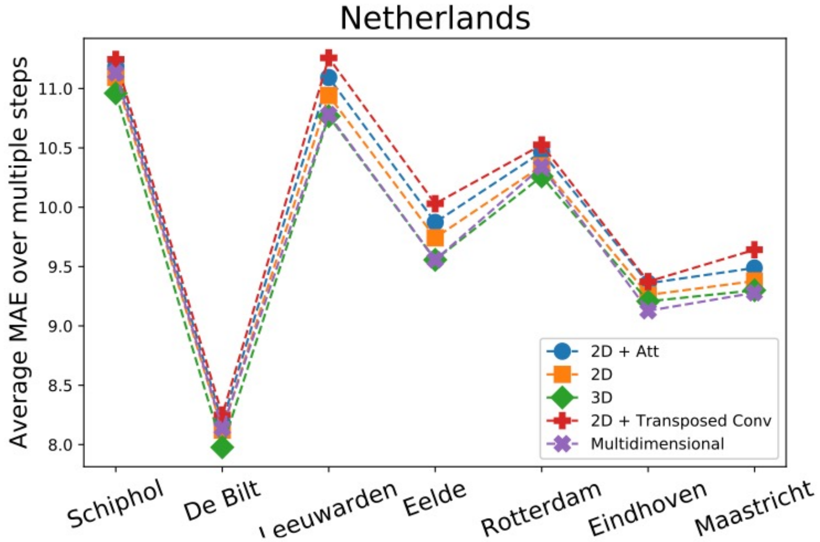
Denmark

Model	MAE				MSE			
	6h	*12h*	*18h*	*24h*	*6h*	*12h*	*18h*	*24h*
Persistence	1.649	2.210	2.309	2.313	4.608	7.929	8.702	8.812
2D	1.304	1.746	1.930	2.004	2.824	5.088	6.120	6.610
2D+Attention	1.313	1.715	1.905	1.950	2.885	4.896	5.933	6.201
2D+Upscaling	1.307	1.723	**1.858**	1.985	2.826	4.931	**5.639**	6.474
3D	1.311	**1.677**	1.908	1.957	2.855	**4.595**	5.958	6.238
Multidimensional	**1.302**	1.706	1.873	**1.925**	**2.804**	4.779	5.773	**6.066**



Netherlands

Model	MAE				MSE			
	1h	*2h*	*3h*	*4h*	*1h*	*2h*	*3h*	*4h*
Persistence	9.55	11.34	12.90	14.37	183.61	246.95	310.38	375.36
2D	8.11	9.17	10.15	11.12	116.89	149.01	181.78	218.49
2D+Attention	**8.08**	9.10	10.11	11.00	**115.96**	147.75	180.66	213.23
2D+Upscaling	8.16	9.07	10.14	**10.85**	117.80	147.21	182.44	208.96
3D	8.17	9.26	10.15	10.93	118.35	151.51	181.35	211.19
Multidimensional	8.12	**9.05**	**9.95**	10.94	116.78	**144.51**	**174.07**	**208.73**



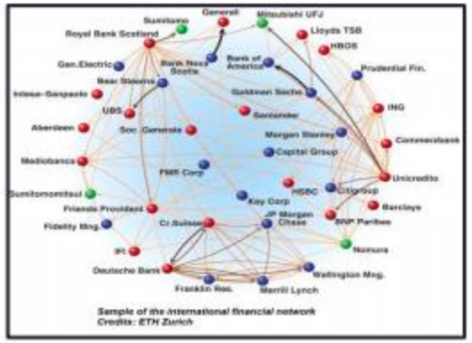
- Multidimensional model outperforms other compared models in several forecasting times

K. Trebing and S. Mehrkanoon, Wind speed prediction using multidimensional convolutional neural networks, IEEE Symposium Series on Computational Intelligence (IEEE-SSCI), vol. 79, pp. 713-720, 2020.

Graph Convolutional Neural Networks

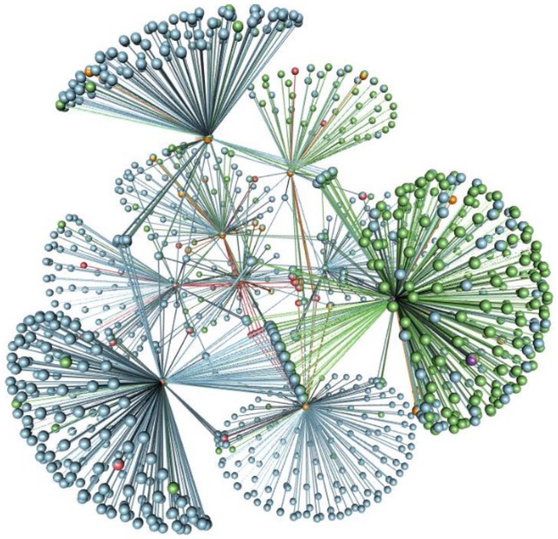
Non-Grid Data:

Finance Networks



Sample of the international financial network
Credits: ETH Zurich

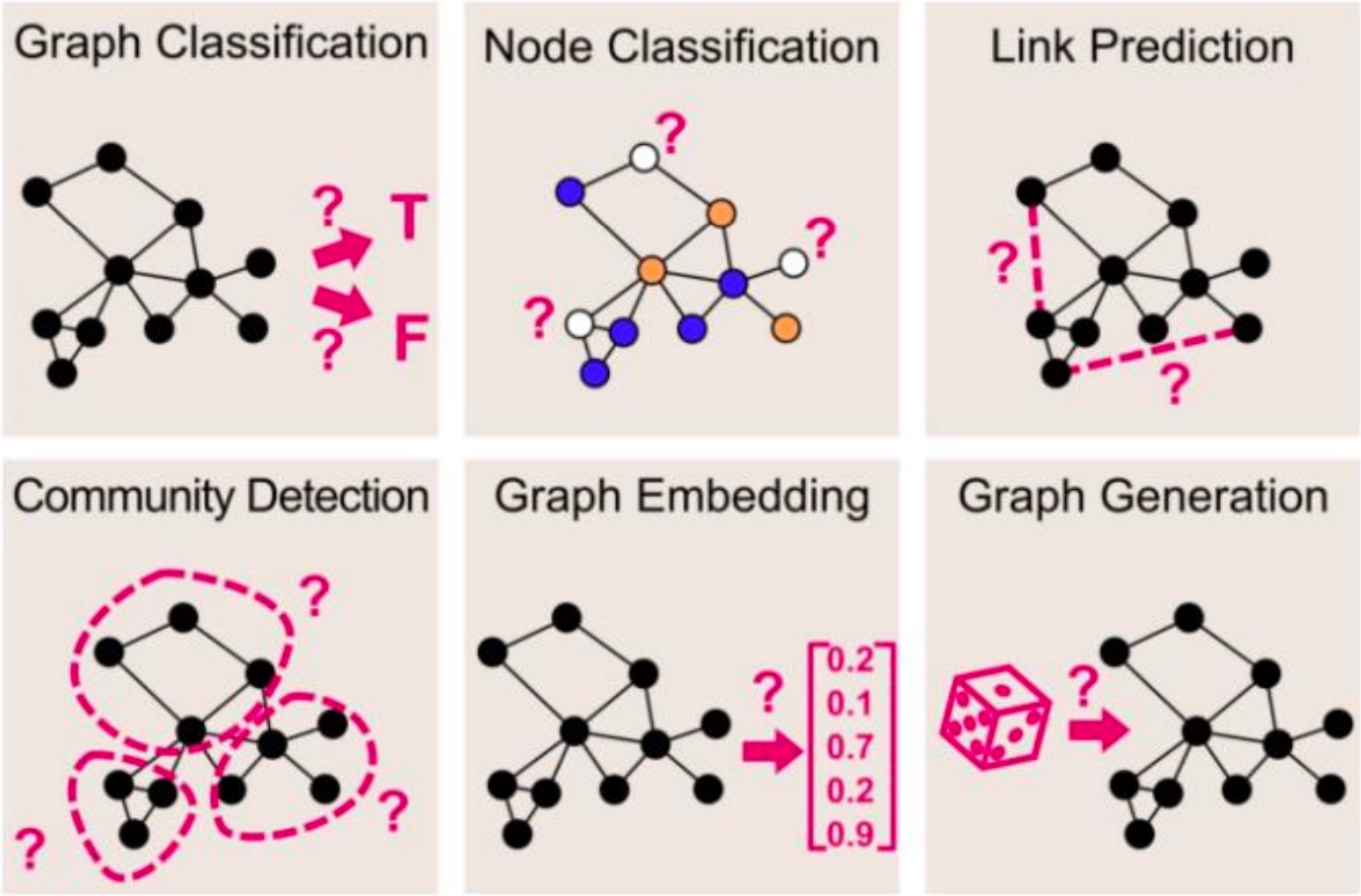
Biology networks



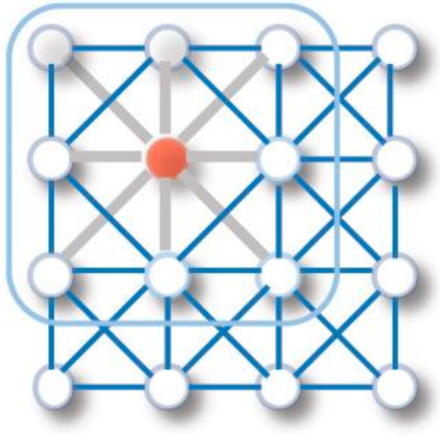
Social networks

Logistic Networks

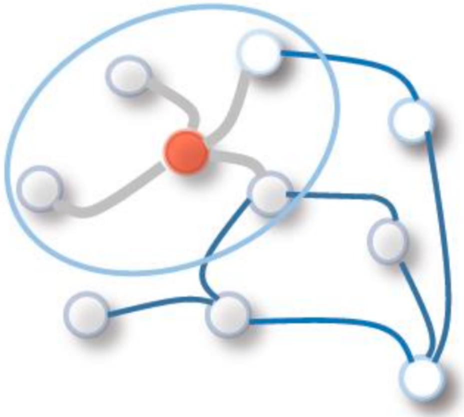
- ❑ Different number of neighbors
- ❑ Various influences (weights)


Graph Classification
T
F
Node Classification
Link Prediction
Community Detection
Graph Embedding
0.2
0.1
0.7
0.2
0.9
Graph Generation

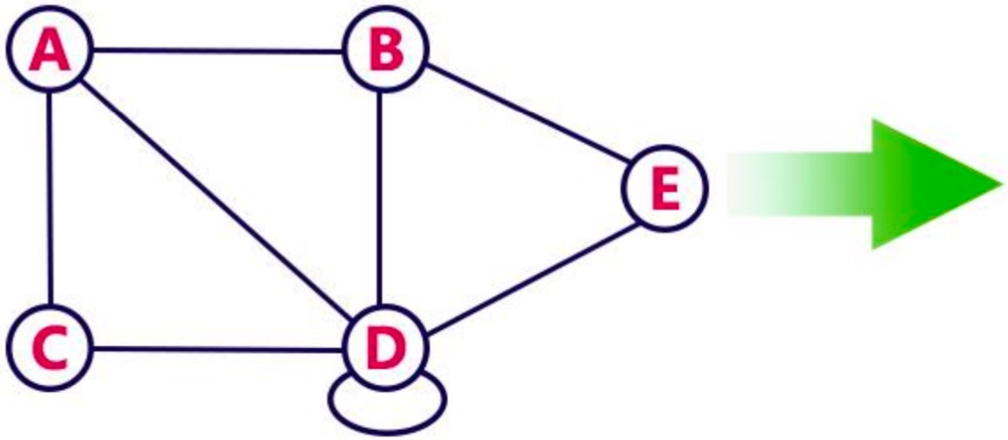
2D Convolution vs. Graph Convolution



The neighbors of a node are ordered and have a fixed size



In a graph, the neighbors of a node are unordered and variable in size



$$\begin{array}{cc} & \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{array}{c} A \\ B \\ C \\ D \\ E \end{array} & \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

Adjacency matrix: A representative description of the graph structure in matrix form.

Graph Structure:

Graph=G(X,A)

X : feature matrix describing the nodes, (NxD)
A : Adjacency matrix (NxN)

❖ Graph convolutional layer can be expressed as:

$$H^{(l+1)} = f(H^{(l)}, A)$$

$$f(H^{(l)}, A) = \sigma\left(AH^{(l)}W^{(l)}\right)$$

with $H^{(0)} = X$

T. Kipf, Thomas and M. Welling, Semi-Supervised Classification with Graph Convolutional Networks, arXiv preprint arXiv:1609.02907, 2016.

Two limitations:

$$f(H^{(l)}, A) = \sigma\left(AH^{(l)}W^{(l)}\right)$$

- ❑ Multiplication with A means that, for every node, we sum up all the feature vectors of all neighboring nodes but not the node itself
- ❑ Matrix A is typically not normalized and therefore the multiplication with A will completely change the scale of the feature vectors.

Solutions:

- Self loop connection: add identity matrix to matrix A
- Normalizing A such that all rows sum to one.

Different ways of normalization:

- $D^{-1}A$
- $D^{-1/2}AD^{-1/2}$

They corresponds to taking the average of neighboring node features

$$f(H^{(l)}, A) = \sigma\left(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

with $\hat{A} = A + I$, where I is the identity matrix and $\hat{D}$ is the diagonal node degree matrix of $\hat{A}$.

T. Kipf, Thomas and M. Welling, Semi-Supervised Classification with Graph Convolutional Networks, arXiv preprint arXiv:1609.02907, 2016.

❑ Compare GCN and simplest fully connected networks:

$$f(H^{(l)}, A) = \sigma\left(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

$$H^{(l+1)} = \sigma\left(H^{(l)}W^{(l)}\right)$$

- Computes a new representation for each vertex which is smoother (more closer to its neighbors) than the original representation.
- Since vertices in the same cluster tend to be densely connected, the smoothing makes their representations similar.
- Thus the subsequent classification task get much easier.

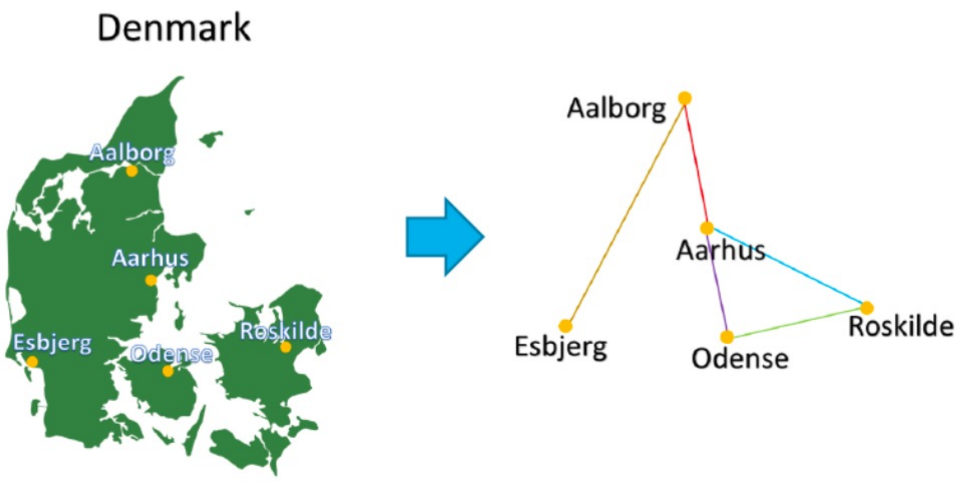
Application: Weather element forecasting

Deep Graph Convolutional Networks for Wind Speed Prediction

https://github.com/tstanczyk95/WeatherGCNet

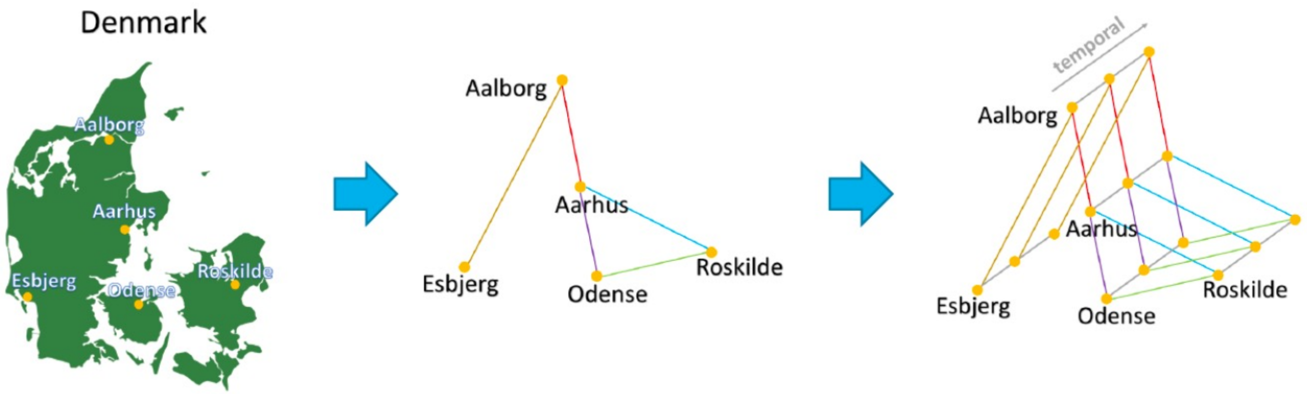
T. Stanczyk and S. Mehrkanoon, Deep Graph Convolutional Networks for Wind Speed Prediction,
In proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), pp. 147-152, 2021.

- Time series data:
  - weather stations
  - weather variable



- Single time step:
  - Each city: a node
  - Node attributes: weather variables (first layer), values encoded by the network (next layers)

- Multiple time steps:
  - A spatio-temporal graph



T. Stanczyk and S. Mehrkanoon, Deep Graph Convolutional Networks for Wind Speed Prediction,
In proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), pp. 147-152, 2021.

Spatial Convolution:

Aggregating information from spatial neighbors of the graph

Temporal Convolution:

Aggregating information from temporal neighbors –next and/or previous time step

T. Stanczyk and S. Mehrkanoon, Deep Graph Convolutional Networks for Wind Speed Prediction,
In proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), pp. 147-152, 2021.

Spatial Convolution:

- Neighborhood information is represented as an **adjacency matrix (A)**
  - **Binarized adjacency matrix**: indicating if there is connection or not
  - **Non-binarized adjacency matrix**: including strength of the connections between nodes

- If you have access to **adjacency matrix (A)** use it and if you don't then learn it!
  - Network learns graph spatial connections.
  - The learned adjacency matrix is not necessarily symmetric then.

T. Stanczyk and S. Mehrkanoon, Deep Graph Convolutional Networks for Wind Speed Prediction,
In proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), pp. 147-152, 2021.

Spatial Convolution:

- Adjacency matrix is further transformed during the training (similar transformations to those applied in Kipfand Welling, 2016):

$$\hat{A} = A + I$$

$$\hat{A} = \frac{\hat{A} - \hat{A}_{min}}{\hat{A}_{max} - \hat{A}_{min}}$$

$$\hat{D}_{ii} = \sum_j \hat{A}_{ij}$$

$$\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}}$$

$$X_{out} = X_{in} (\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}})$$

$$\hat{A} = A + \gamma I$$

Network can decide about the importance of the self-connection, which is originally imposed

T. Stanczyk and S. Mehrkanoon, Deep Graph Convolutional Networks for Wind Speed Prediction,
In proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), pp. 147-152, 2021.

Spatial Convolution:

- -Processing input tensor of a format C x T x V:
  - Where C= #channels (features), T = #time steps, V = #graph vertices (cities)
- -Reshaping the input into a matrix of shape CT x V
- - Matrix multiplication (shown in last slide) followed by reshaping the output matrix back to a tensor
- - Performing 1x1 2D convolution
  - Adding linear combinations of features channel-wise
  - Changing the number of channels

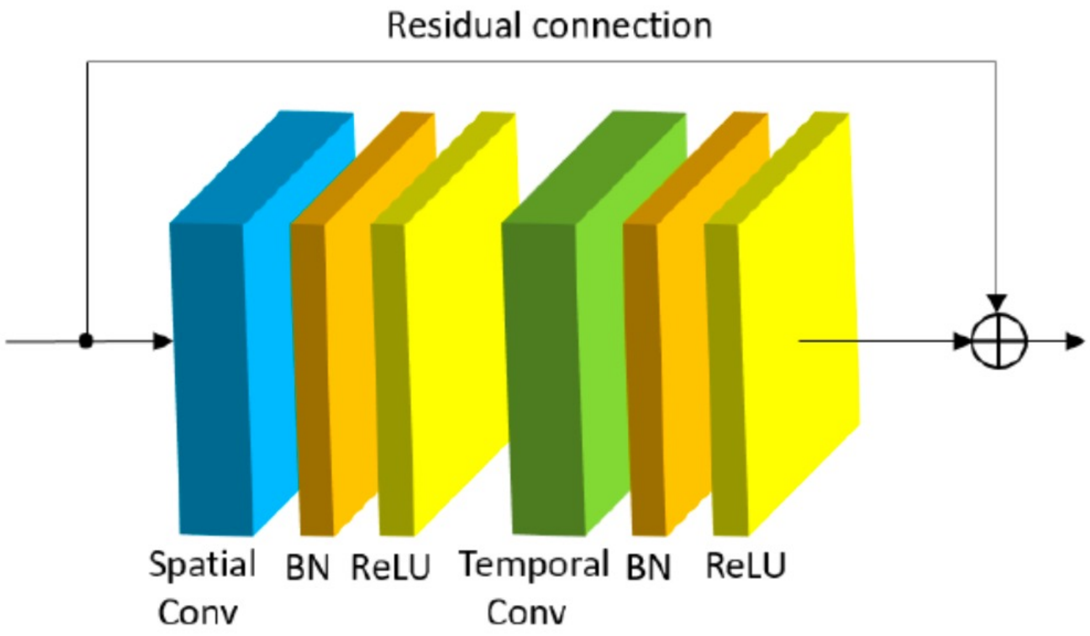
Temporal Convolution:

- Aggregating information from temporal neighbors –next and/or previous time step
- Regular 2D convolution with filter size kx1:
  - Including information from only one node at a time

T. Stanczyk and S. Mehrkanoon, Deep Graph Convolutional Networks for Wind Speed Prediction,
In proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), pp. 147-152, 2021.

Building a Spatio-Temporal block (ST-block)

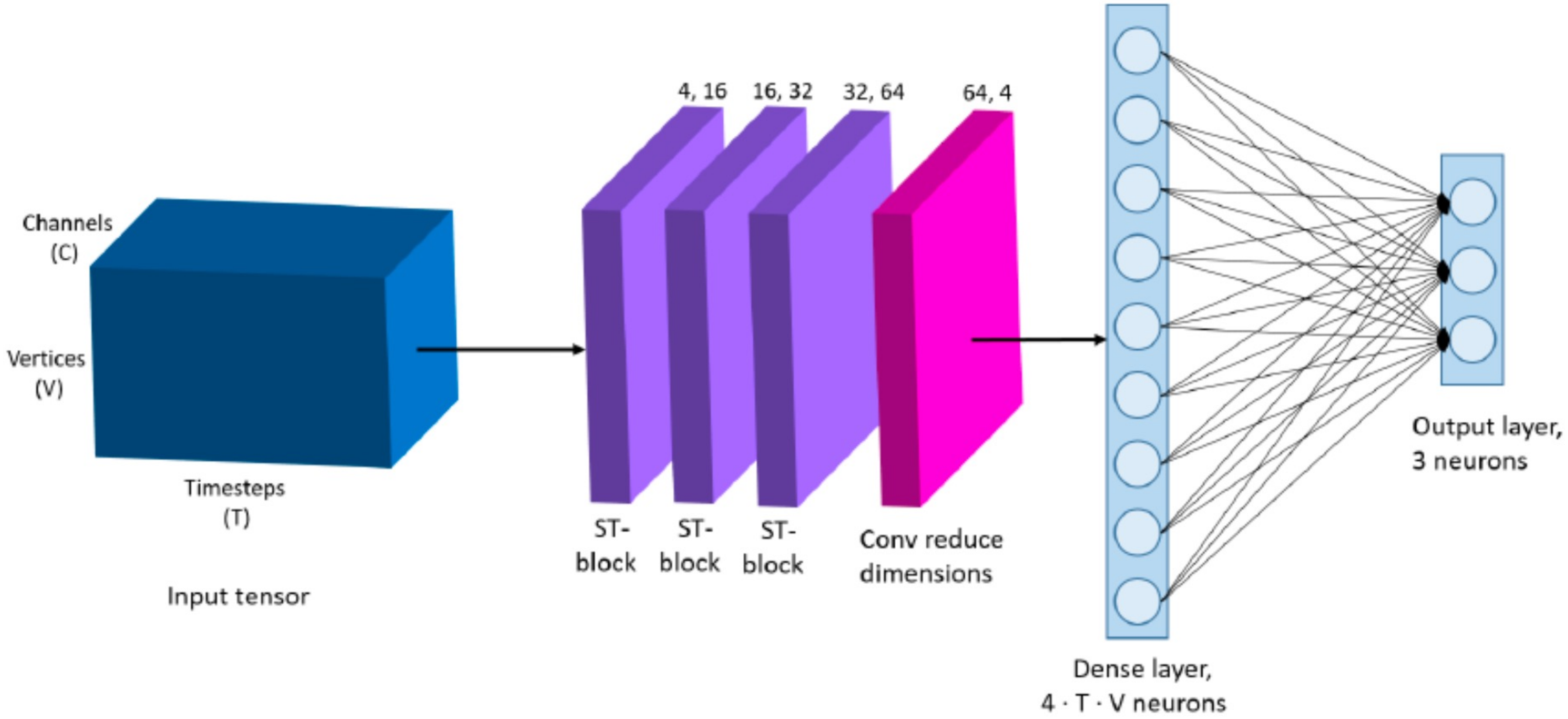
- Spatial graph convolution
- Temporal convolution
- Other elements
  - Batch normalization
  - ReLU activation
  - Residual connection



T. Stanczyk and S. Mehrkanoon, Deep Graph Convolutional Networks for Wind Speed Prediction, In proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), pp. 147-152, 2021.

Referring to the models as:

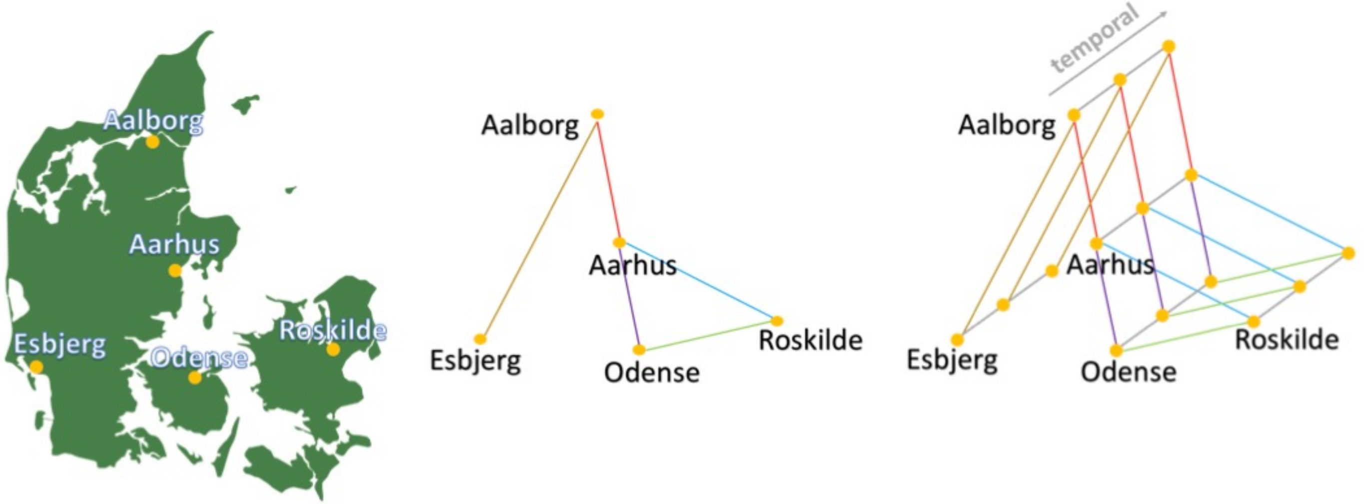
- WeatherGCNet
- WeatherGCNet with γ



https://github.com/tstanczyk95/WeatherGCNet

T. Stanczyk and S. Mehrkanoon, Deep Graph Convolutional Networks for Wind Speed Prediction, In proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), pp. 147-152, 2021.

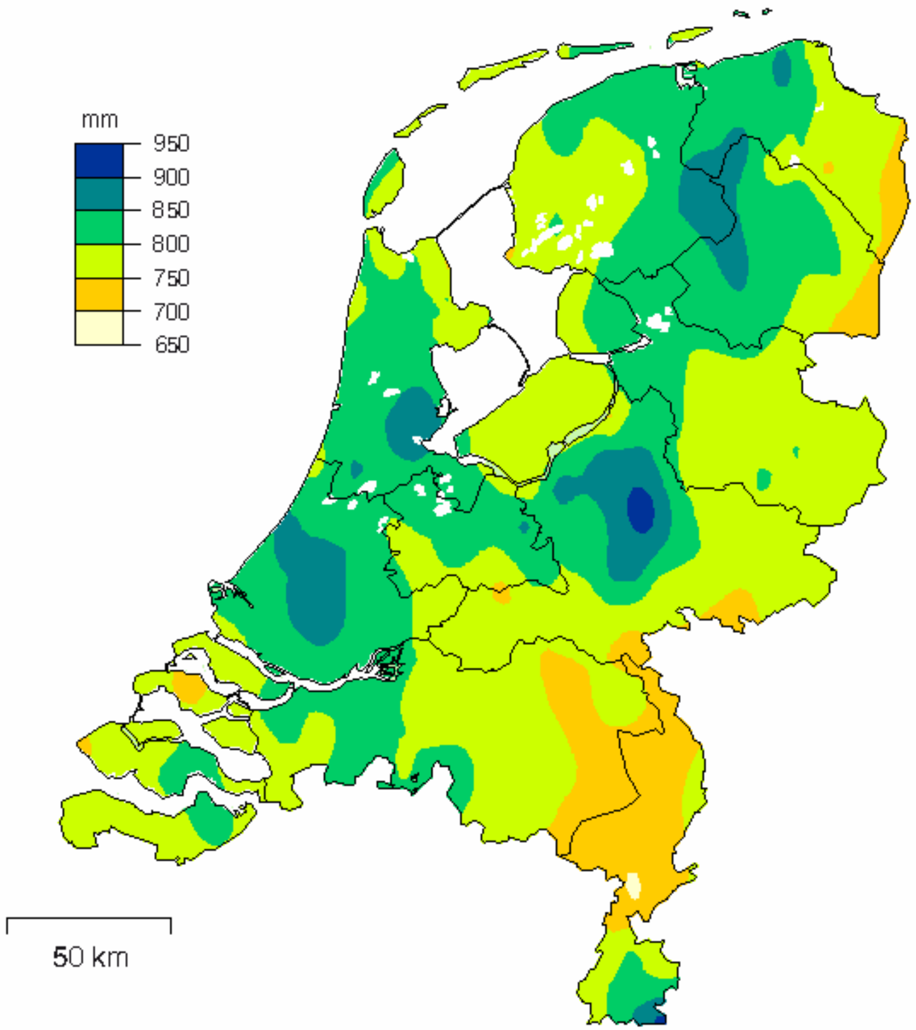
WeatherGCNet:



Model	Denmark				Netherlands				
	6h	12h	18h	24h	2h	4h	6h	8h	10h
2D	1.304	1.746	1.930	2.004	8.18	10.08	12.03	13.15	14.51
2D + Attention	1.313	1.715	1.905	1.950	8.10	10.09	11.83	13.10	14.13
2D + Upscaling	1.307	1.723	1.858	1.985	8.24	10.22	11.83	13.74	14.80
3D	1.311	1.677	1.908	1.957	8.05	10.15	11.93	13.01	14.24
Multidimensional	1.302	1.706	1.873	1.925	8.10	10.03	11.46	12.79	13.81
WeatherGCNet (γ=1)	1.279	1.638	1.777	1.869	<u>7.96</u>	9.97	11.16	<u>12.30</u>	<u>13.33</u>
WeatherGCNet (learnt γ)	<u>1.267</u>	<u>1.616</u>	<u>1.767</u>	<u>1.853</u>	7.97	<u>9.74</u>	<u>10.99</u>	12.44	13.55

T. Stanczyk and S. Mehrkanoon, Deep Graph Convolutional Networks for Wind Speed Prediction,
In proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), pp. 147-152, 2021.

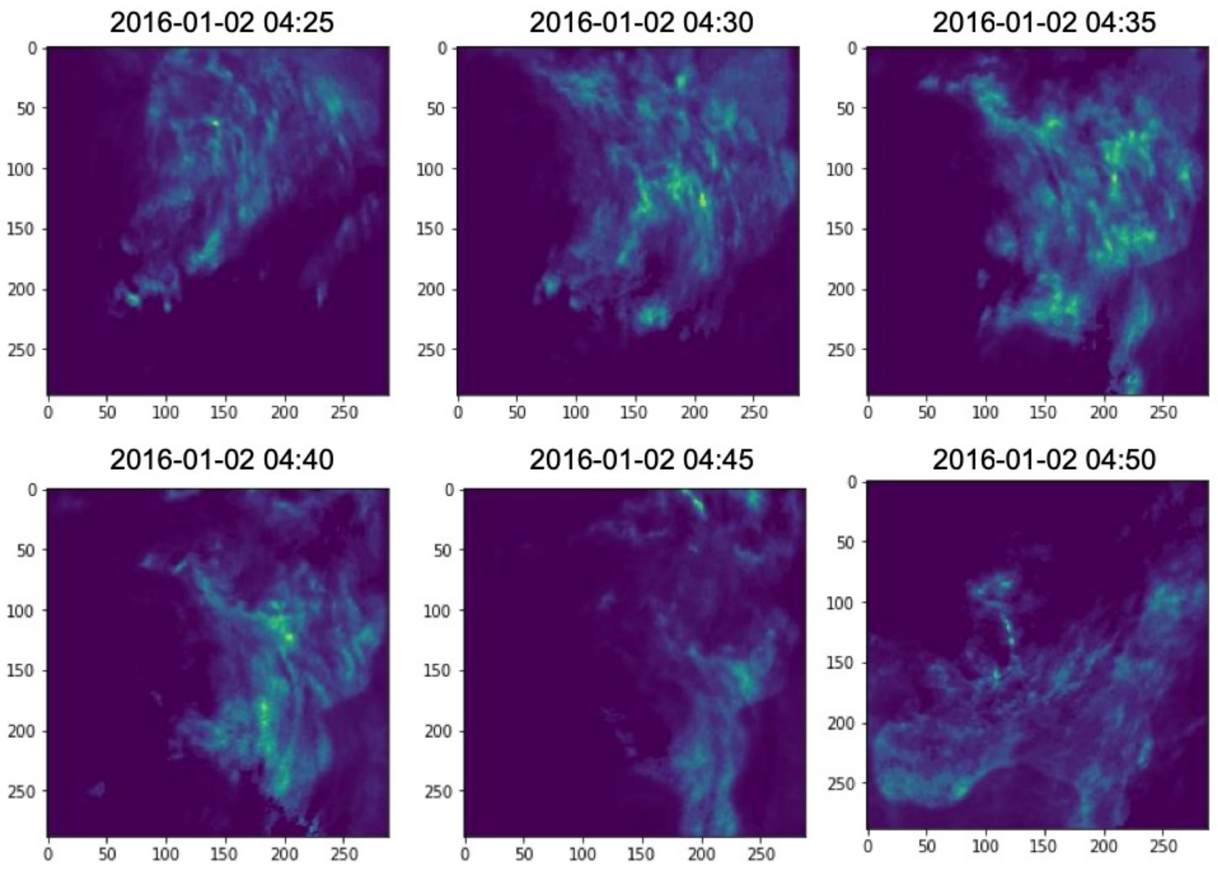
Precipitation Nowcasting



Precipitation maps of Netherlands in 5-minute intervals from 2016-2019.

•Training set: 2016-2018

•Test set: 2019



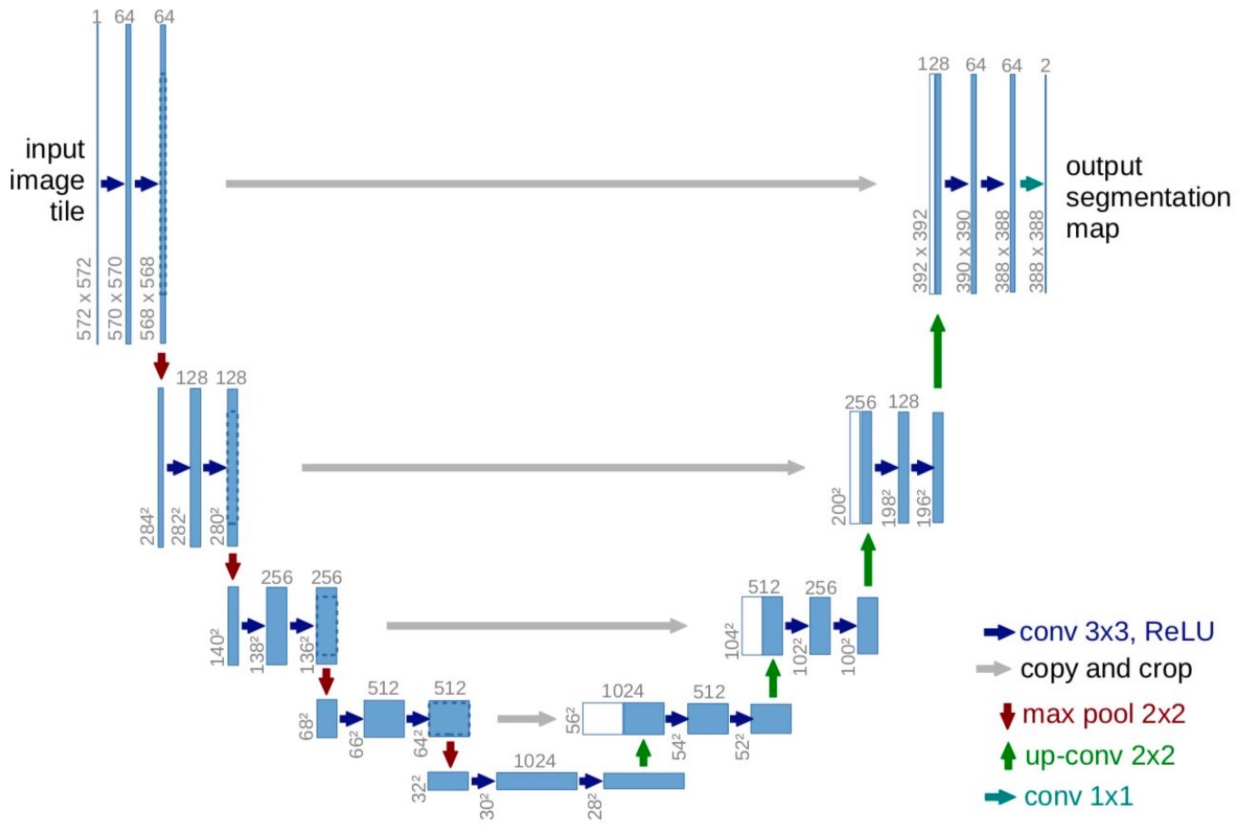
Task:

➢ Use the previous 12 precipitation maps (1 hour) and predict the precipitation map 30 minutes into the future.

S. Mehrkanoon, SmaAt-UNet: Precipitation nowcasting using a small attention-UNet architecture, Pattern Recognition Letters, Vol. 145, Pages 178-186, 2021.

U-Net architecture

- Convolutional autoencoder with middle connections
- Medical image segmentation
- 2-dimensional data
- Extract features and reconstruct segmented image
- Residual connections:
  - Precise localization

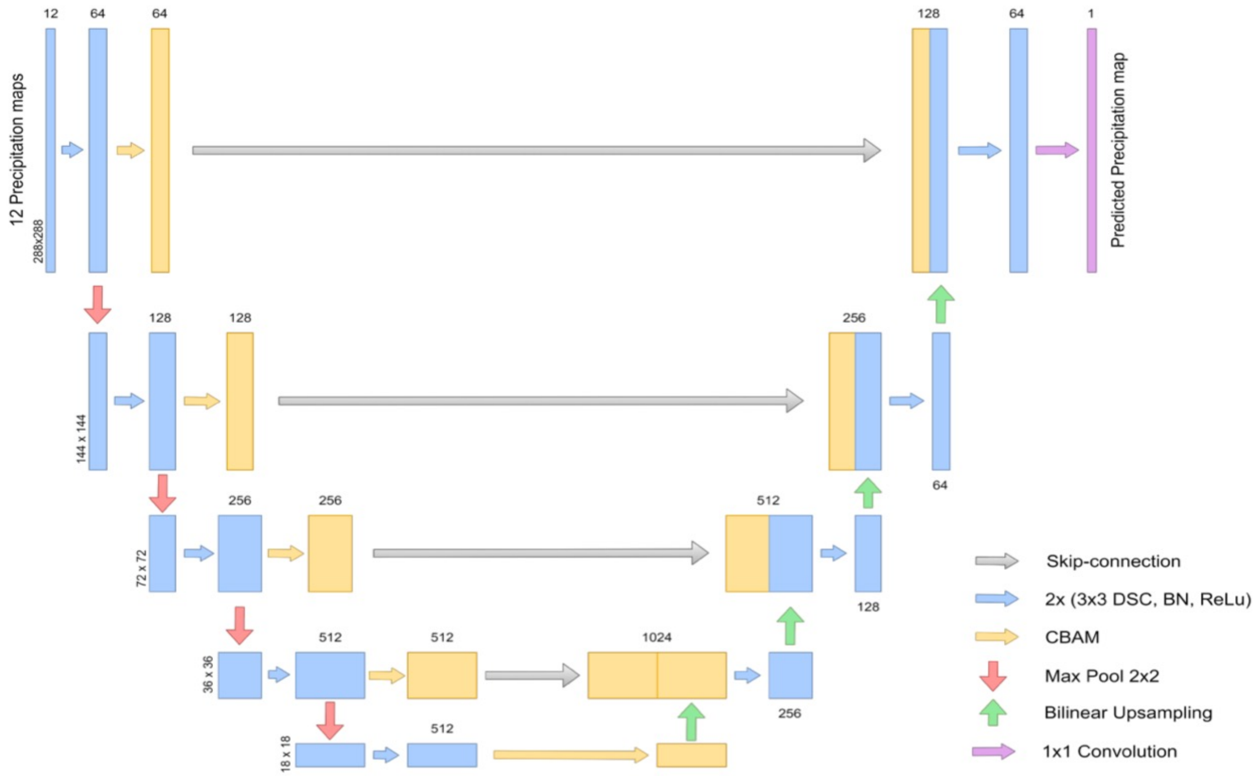


Olaf Ronneberger et al (2015), “U-Net: Convolutional Networks for Biomedical Image Segmentation”

SmaAt-UNet – Small Attention-UNet

UNet with:

- Convolution Block Attention Modules (CBAM)
- The regular convolutions are changed to Depthwise Separable Convolutions to reduce the parameters.



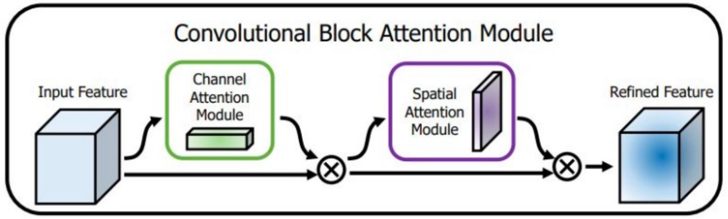
https://github.com/HansBambel/SmaAt-UNet

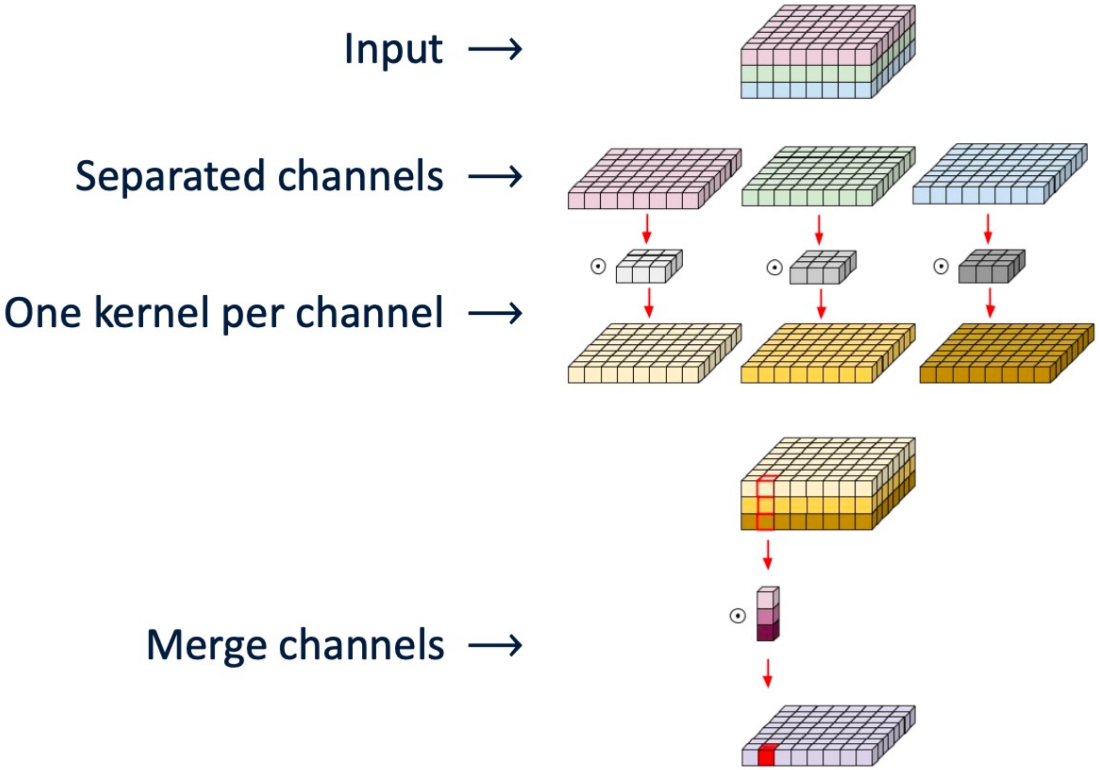
K. Trebing, T Stańczyk, S. Mehrkanoon, SmaAt-UNet: Precipitation nowcasting using a small attention-UNet architecture, Pattern Recognition Letters, Vol. 145, Pages 178-186, 2021.

U-Net variations - SmaAt-UNet

- Depth-wise separable convolution
  - Depth-wise convolution
  - Point-wise convolution

- Convolutional Block Attention Module (CBAM)



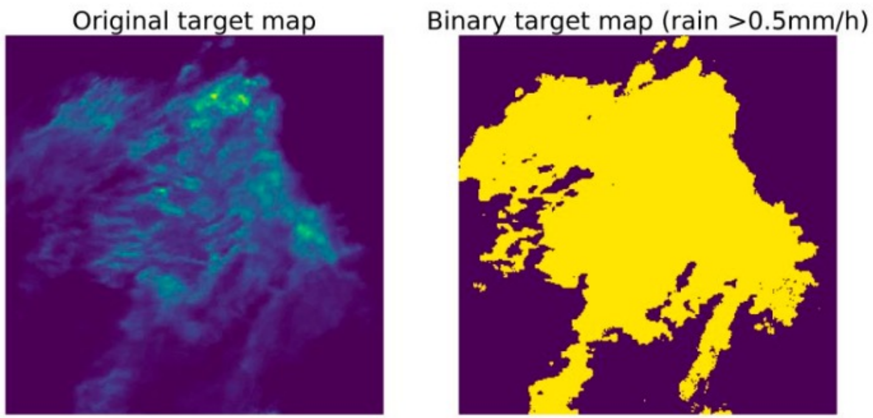


Sanghyun Woo et al (2018), "CBAM: Convolutional Block Attention Module"

François Chollet (2017), "Xception: Deep Learning with Depthwise Separable Convolutions"

Compared models:

- Persistence(Baseline)
- OriginalUNet
- UNet with CBAM
- UNet with DSC
- UNet with CBAM and DCS (SmaAt-Unet)



S. Mehrkanoon, SmaAt-UNet: Precipitation nowcasting using a small attention-UNet architecture, Pattern Recognition Letters, Vol. 145, Pages 178-186, 2021.

MSE, NMSE and scores on rainfall bigger than $0.5mm/h$ indicating rain or no rain on the NL-50 dataset. Best result for that score is in bold. A ↑ indicates that higher values for that score are good whereas a ↓ indicates that lower scores are better.

Model	MSE ↓	NMSE ↓	Accuracy ↑	Precision ↑	Recall ↑	F1 ↑	CSI ↑	FAR ↓	HSS ↑	Model size
Persistence (baseline)	0.0248	847.67	0.756	0.678	0.643	0.660	0.493	0.320	0.235	-
UNet	**0.0122**	416.38	**0.836**	**0.740**	0.855	**0.794**	**0.658**	**0.259**	**0.329**	1×
UNet with CBAM	0.0171	584.46	0.820	0.707	**0.871**	0.780	0.640	0.293	0.315	1.01×
UNet with DSC	0.0127	435.86	0.812	0.700	0.856	0.770	0.626	0.300	0.306	0.23×
SmaAt-UNet	**0.0122**	**416.10**	0.829	0.730	0.850	0.786	0.647	0.270	0.322	0.24×

Number of parameters of the compared models.

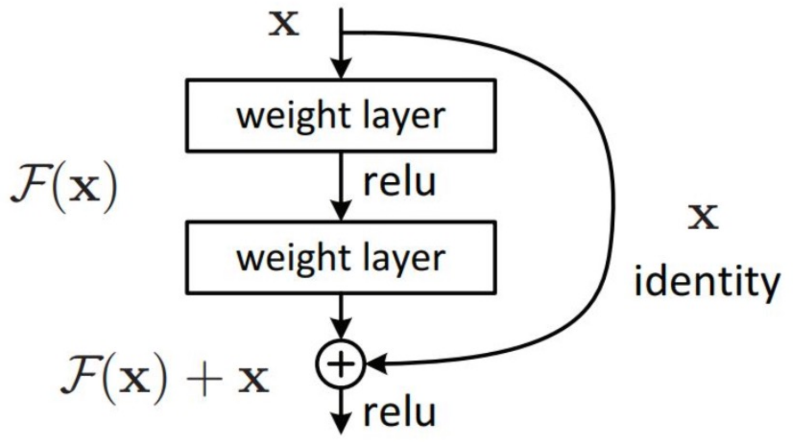
Model	Parameters
UNet	17,272,577
UNet with CBAM	17,428,781
UNet with DSC	3,955,185
SmaAt-UNet	4,111,389

K. Trebing, T Stanczyk, S. Mehrkanoon, SmaAt-UNet: Precipitation nowcasting using a small attention-UNet architecture, Pattern Recognition Letters, Vol. 145, Pages 178-186, 2021.

Skip/Residual connections

Widely used nowadays!

- Appeared to fight the vanishing gradient problem
- Skip some layers and add the main stream's outputs to the skip connection's output
- Feedforward:
  - Keep some of the original input along the network
- Backpropagation:
  - Gradients flow backward easily

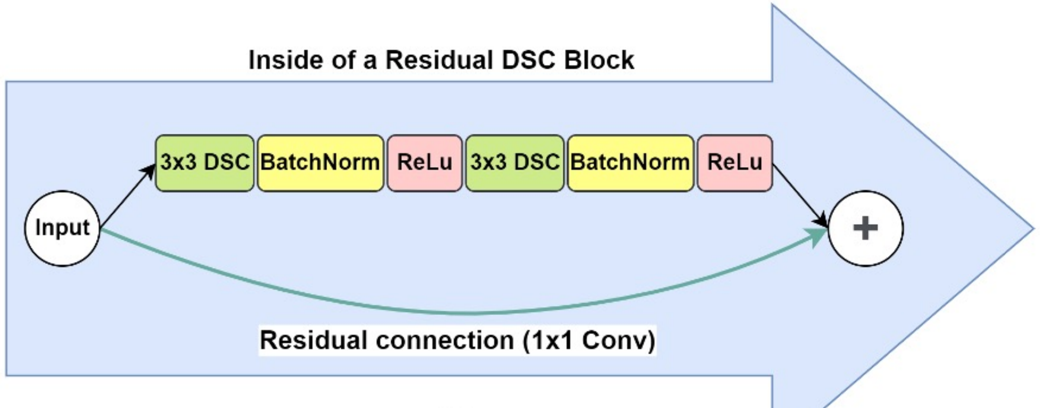


HE, Kaiming, et al (2016), "Deep residual learning for image recognition"

Residual Depthwise Separable Convolution (DSC) Block

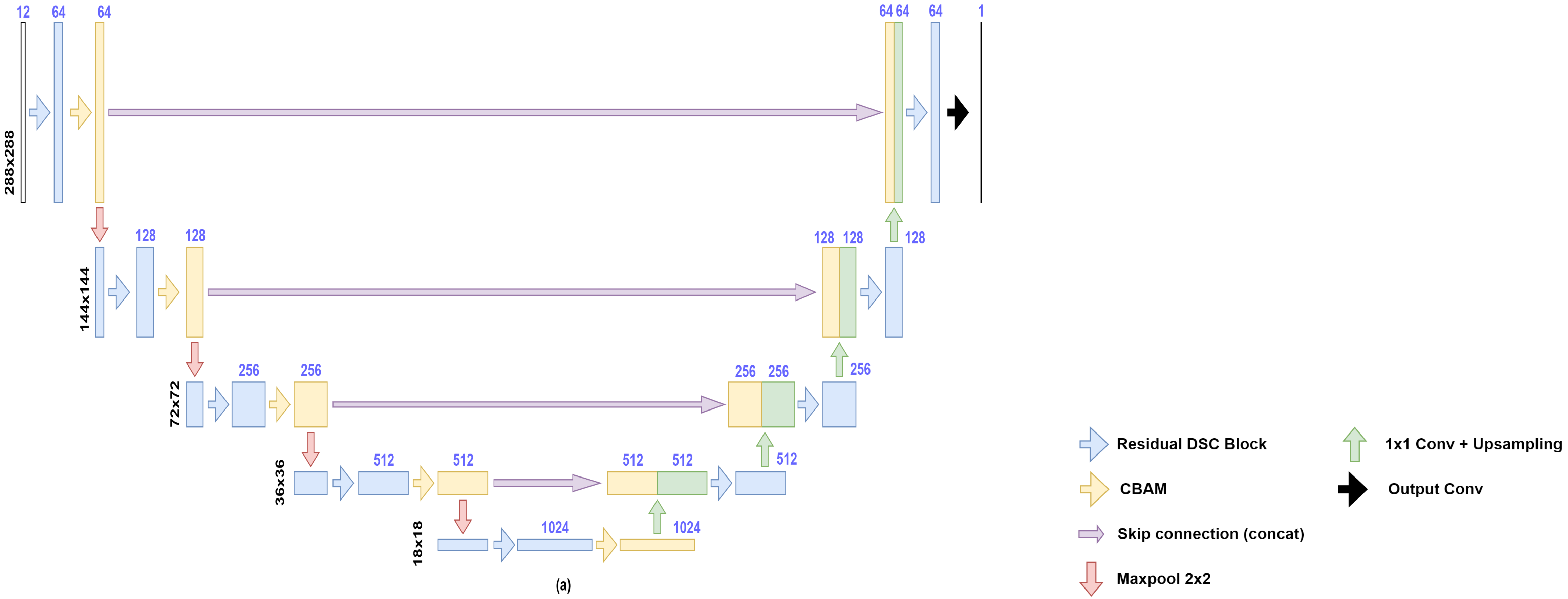
The residual connection is parallel to the regular 3x3 convolutional path.

The output of both path is then summed.



(b)

Small Attention Residual UNet (SAR-UNet):

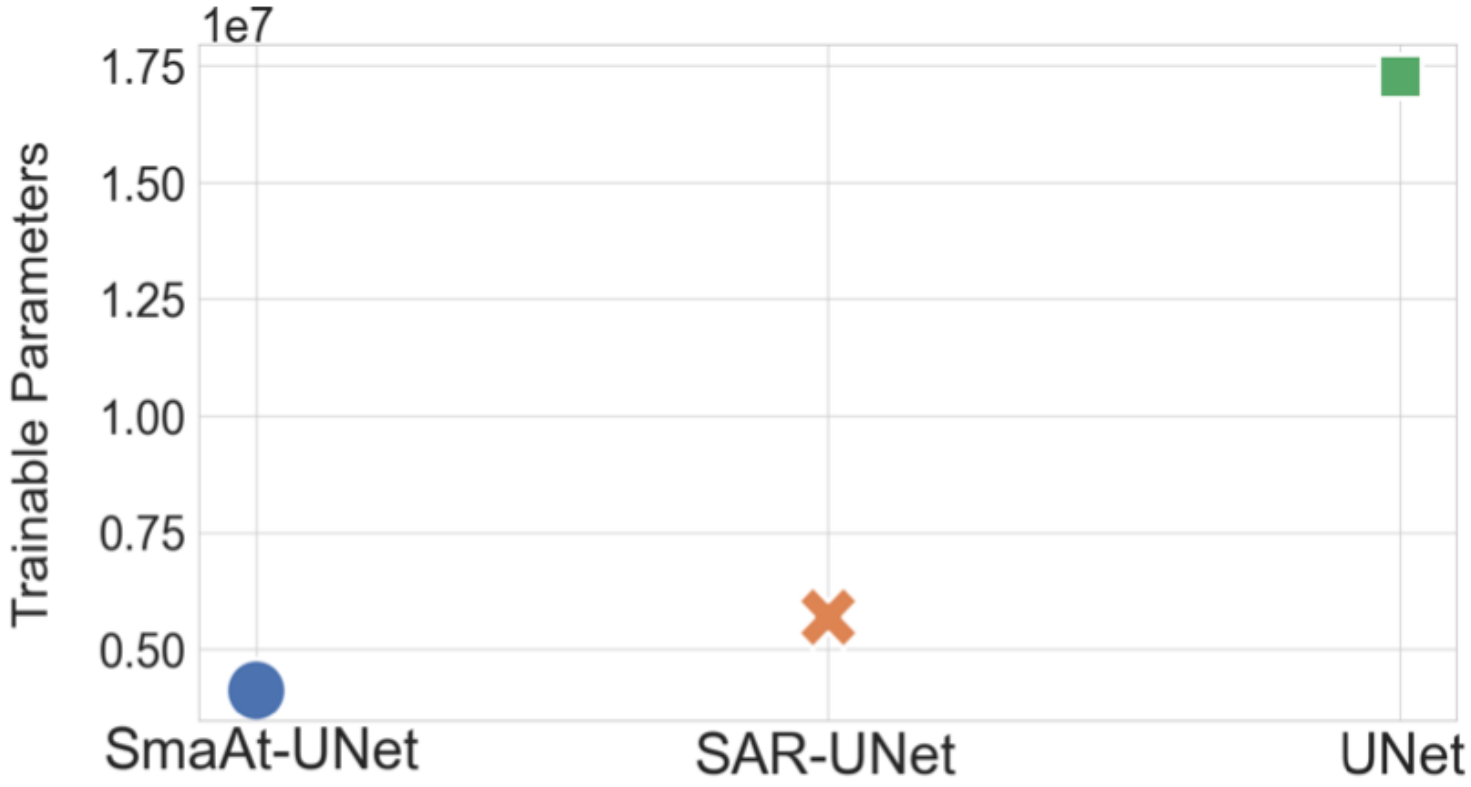


https://github.com/mathieurenault1/SAR-UNet

R Mathieu, S. Mehrkanoon, SAR-UNet: Small Attention Residual UNet for Explainable Nowcasting Tasks, accepted for publication In proc. of IEEE-IJCNN, 2023.

Trainable parameters

Significant reduction of the number of parameters with DSC

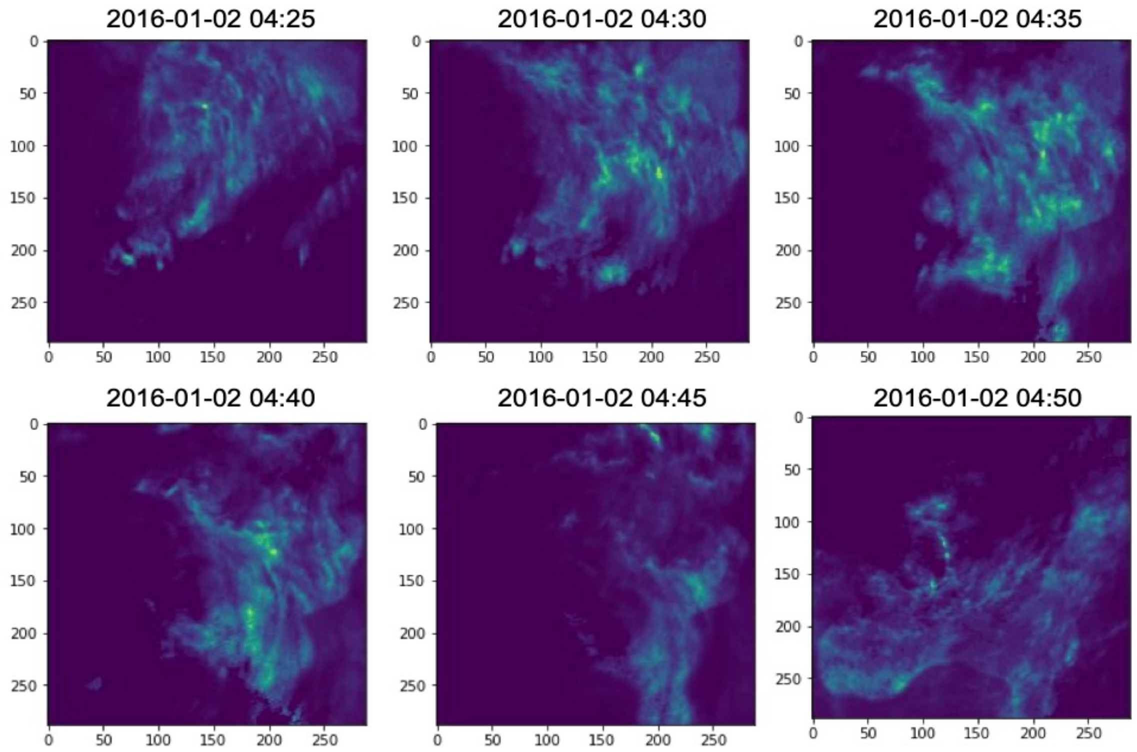


R Mathieu, S. Mehrkanoon, SAR-UNet: Small Attention Residual UNet for Explainable Nowcasting Tasks, accepted for publication In proc. of IEEE-IJCNN, 2023.

Experiments:

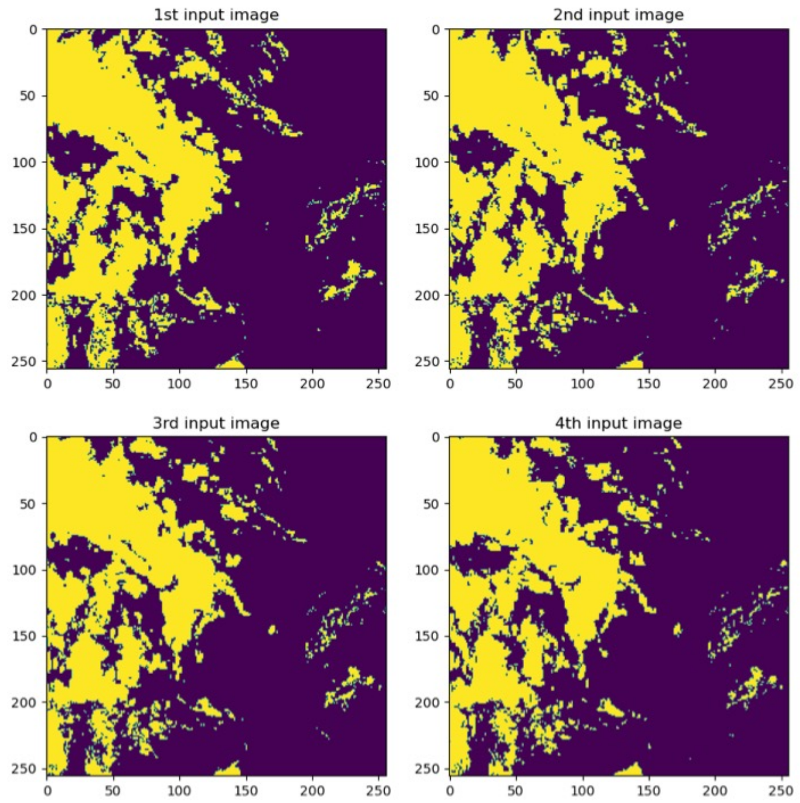
Precipitation nowcasting in the Netherlands:

- Radar images measuring precipitation intensities every 5 minutes.
- 15 different setups:
  - Input data: 30, 60 and 90 minutes (6,12 and 18 images)
  - Minutes ahead: 30, 60, 90, 120 and 180 minutes



Experiments:

- Cloud cover nowcasting in France:
  - Images collected every 15 minutes.
  - Binary value per pixel: 1 for cloud and 0 for no cloud
  - Input data: 60 minutes (4 images)
  - Minutes ahead: predict all images from 15 to 90 minutes (6 images)



Precipitation nowcasting performance

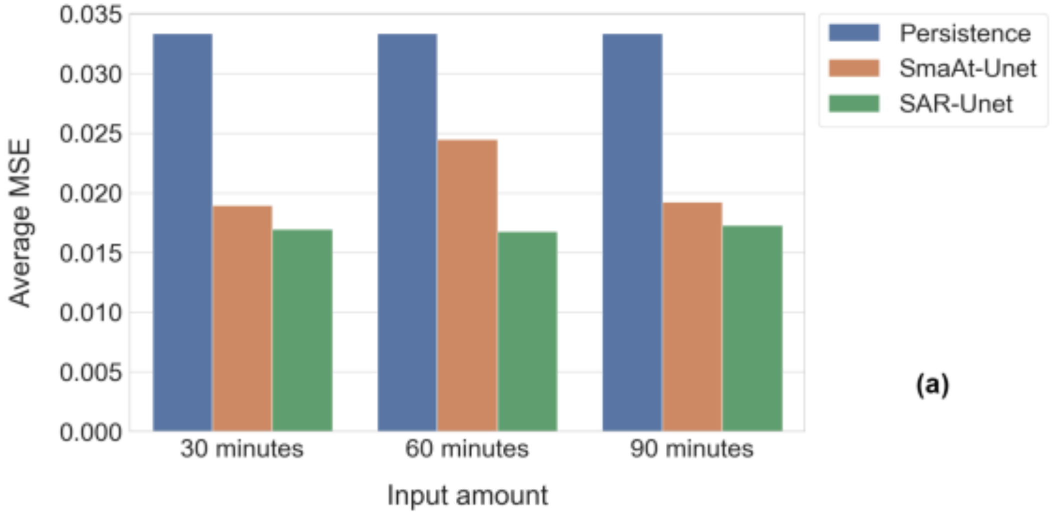
Input amount	*Prediction time*	*Models*	MSE ↓	Precision ↑	Recall ↑	Accuracy ↑	F1 score ↑
60 minutes	30 minutes ahead	Persistence	0,0249	0,678	0,643	0,756	0,66
		SmaAt-UNet	0,0248	0,677	**0,878**	0,801	0,764
		SAR-UNet	**0,0120**	**0,697**	0,868	**0,813**	**0,774**
	60 minutes ahead	Persistence	0,0318	0,603	0,522	0,698	0,56
		SmaAt-UNet	0,0166	0,582	**0,843**	0,719	**0,688**
		SAR-UNet	**0,0163**	**0,623**	0,747	**0,74**	0,679
	90 minutes ahead	Persistence	0,0360	**0,558**	0,43	0,665	0,486
		SmaAt-UNet	0,0314	0,553	0,737	**0,684**	**0,632**
		SAR-UNet	**0,0185**	0,542	**0,747**	0,674	0,628
	120 minutes ahead	Persistence	0,0375	0,532	0,355	0,648	0,426
		SmaAt-UNet	0,0196	**0,54**	0,653	**0,667**	0,591
		SAR-UNet	**0,0176**	0,485	**0,831**	0,613	**0,612**
	180 minutes ahead	Persistence	0,0368	0,478	0,229	**0,624**	0,31
		SmaAt-UNet	0,0302	**0,488**	0,685	0,619	0,57
		SAR-UNet	**0,0196**	0,477	**0,712**	0,607	**0,571**

SAR-Unet slightly outperforms the other methods.

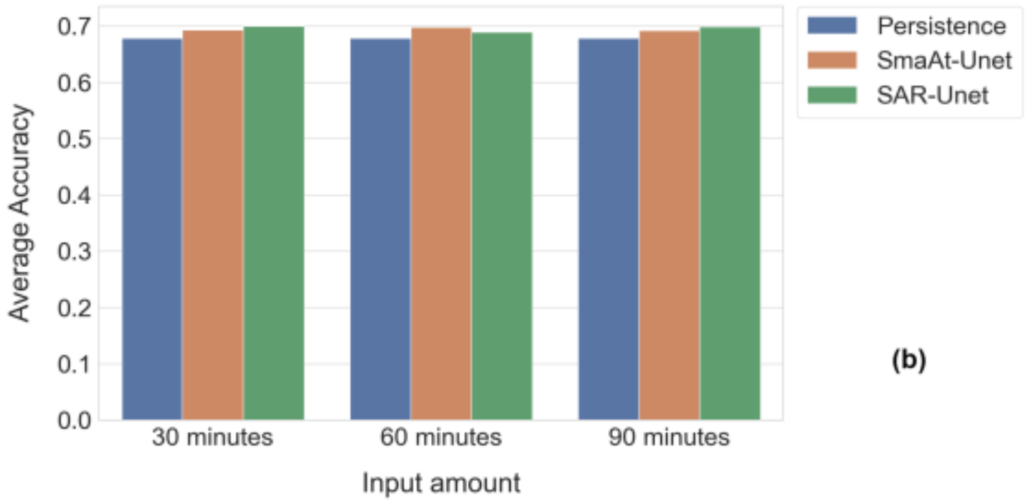
Predicting more minutes ahead reduces the performance.

R Mathieu, S. Mehrkanoon, SAR-UNet: Small Attention Residual UNet for Explainable Nowcasting Tasks, accepted for publication In proc. of IEEE-IJCNN, 2023.

Precipitation nowcasting performance:



(a)



(b)

R Mathieu, S. Mehrkanoon, SAR-UNet: Small Attention Residual UNet for Explainable Nowcasting Tasks, accepted for publication In proc. of IEEE-IJCNN, 2023.

Cloud cover nowcasting performance

To obtain the metrics we take the average over the 6 images of the output.

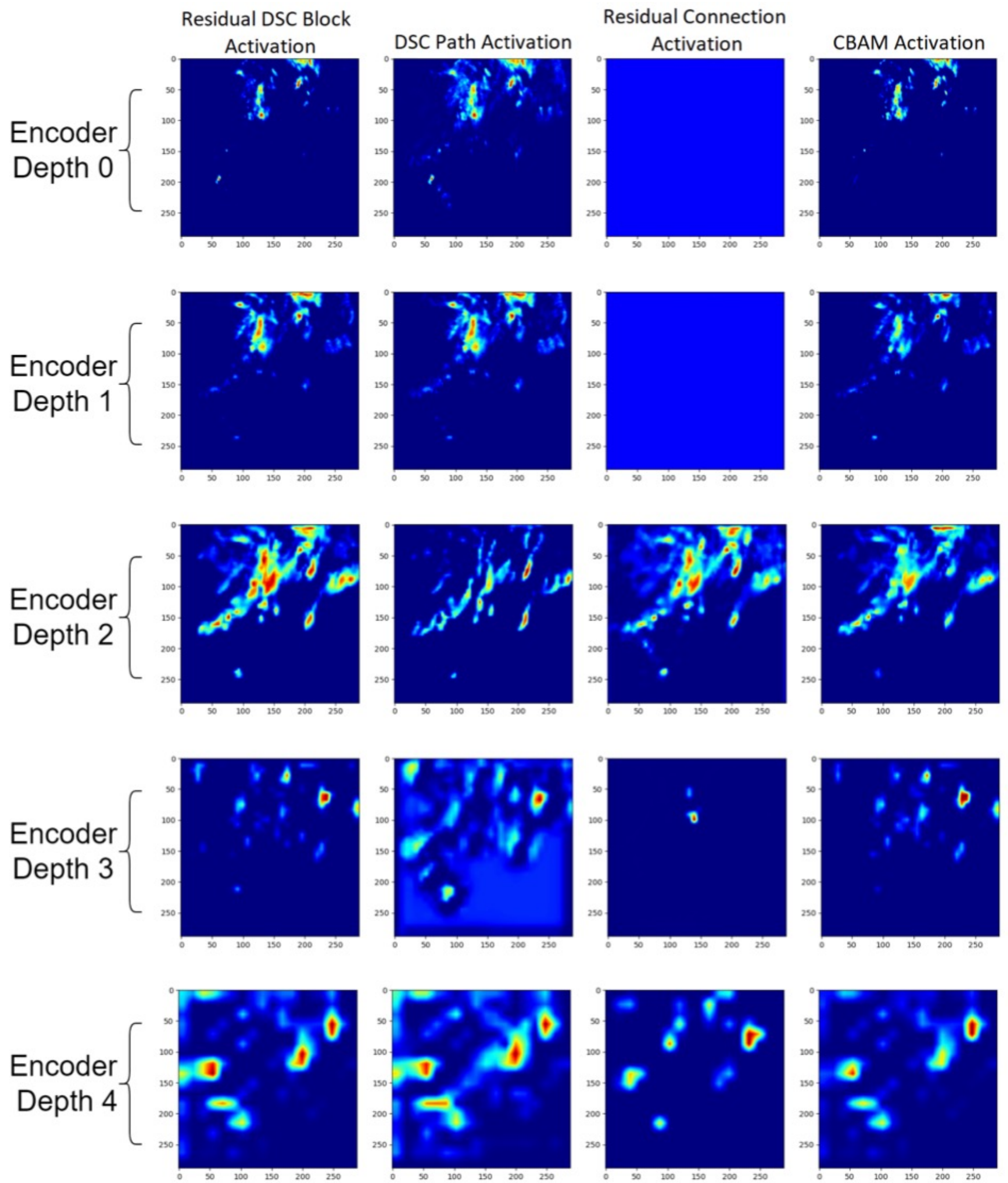
The SAR-Unet is more performant in all metrics considered.

Models	MSE $\downarrow$	Precision $\uparrow$	Recall $\uparrow$	Accuracy $\uparrow$	F1 score $\uparrow$
Persistence	0.1491	0.872	0.872	0.851	0.872
SmaAt-UNet	0.0794	**0.892**	0.921	0.889	0.906
SAR-UNet	**0.0787**	**0.892**	**0.923**	**0.890**	**0.907**

R Mathieu, S. Mehrkanoon, SAR-UNet: Small Attention Residual UNet for Explainable Nowcasting Tasks, accepted for publication In proc. of IEEE-IJCNN, 2023.

Precipitation nowcasting task

- Each row represents a level of the encoder part.
- The columns are different parts of each level:
  - Residual DSC block
  - DSC path
  - Residual Path
  - CBAM

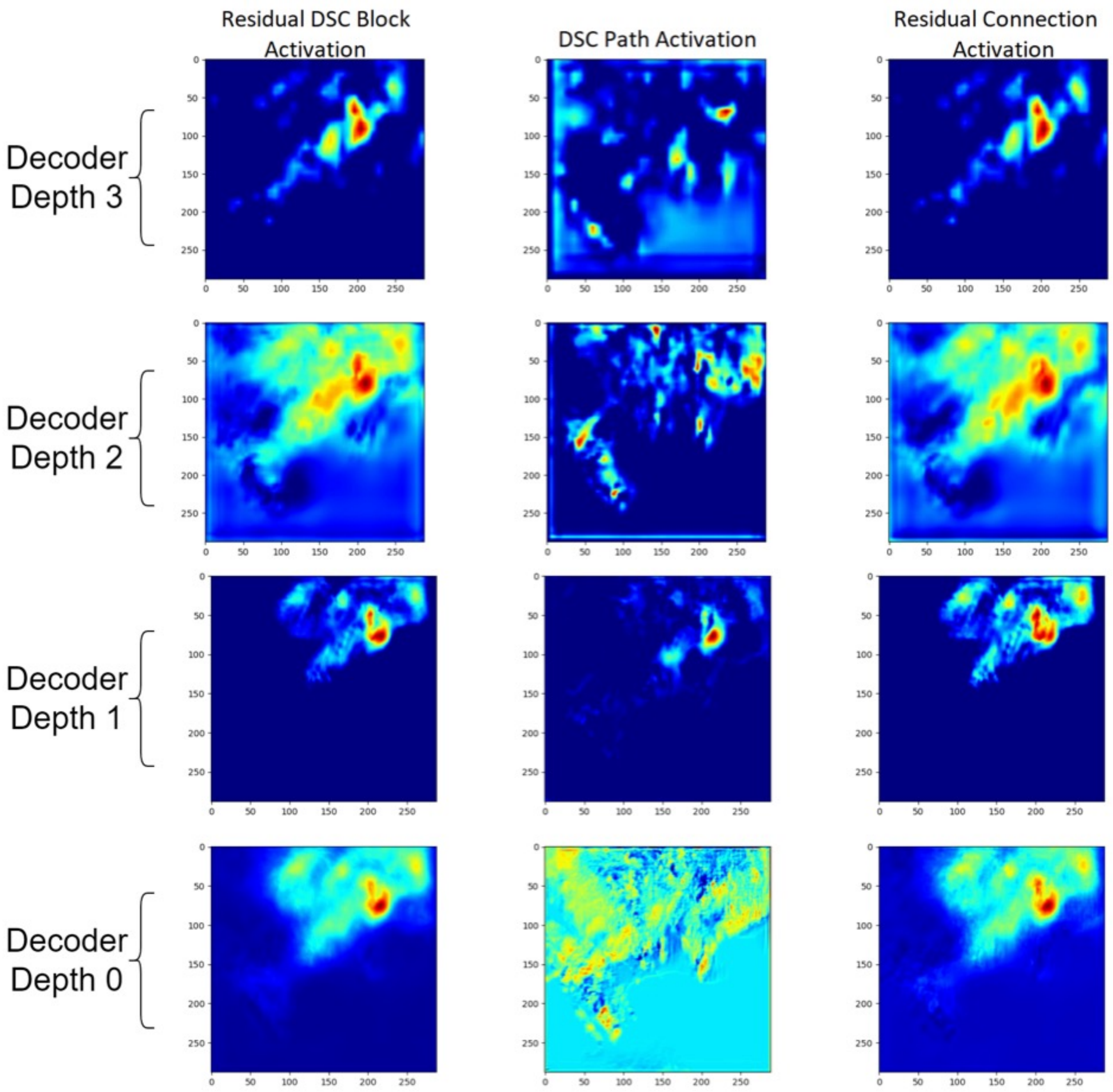


R Mathieu, S. Mehrkanoon, SAR-UNet: Small Attention Residual UNet for Explainable Nowcasting Tasks, accepted for publication In proc. of IEEE-IJCNN, 2023.

Precipitation nowcasting task

- Each row represents a level of the decoder part.
- The columns are different parts of each level:
  - Residual DSC block
  - DSC path
  - Residual Path
  - CBAM

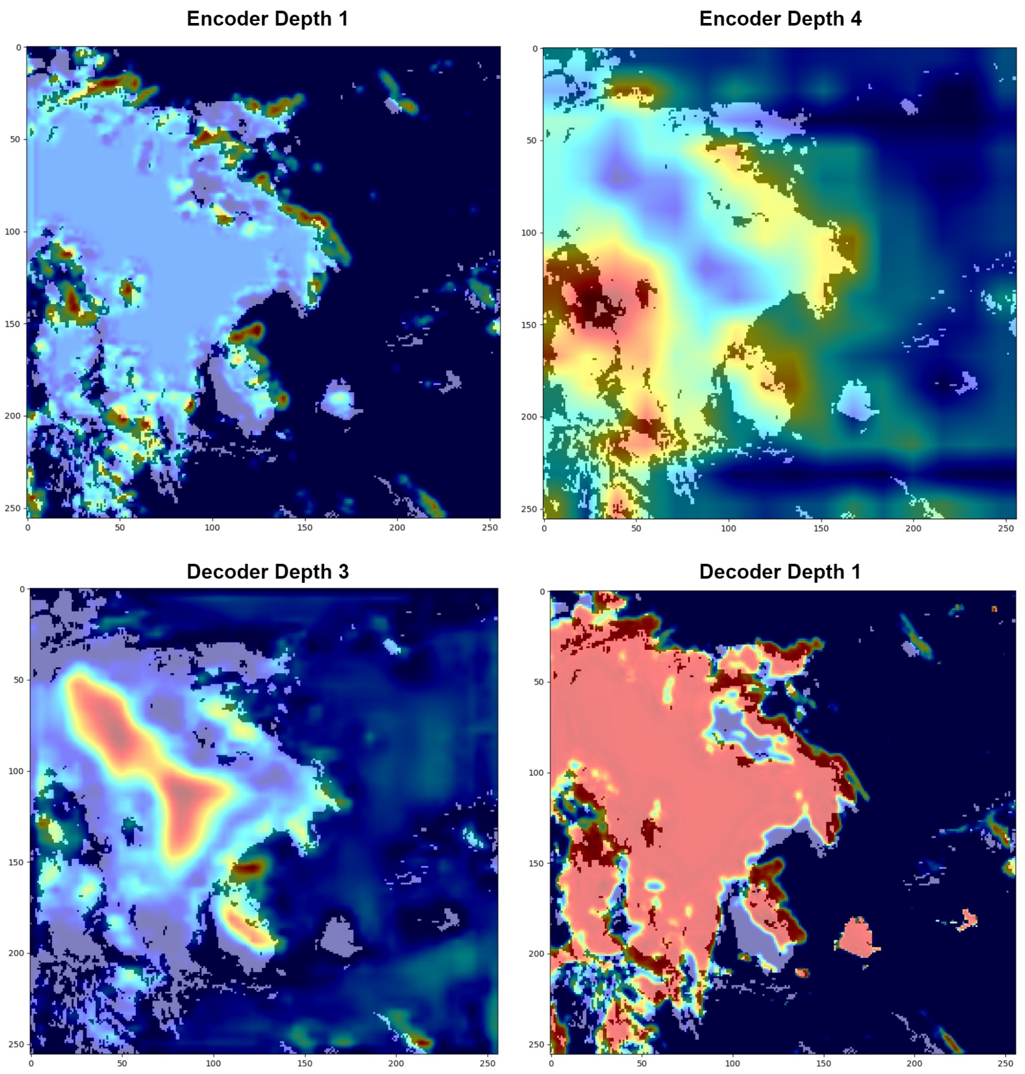
✓ The final three levels of the decoder are much more similar to the prediction made by the network.



R Mathieu, S. Mehrkanoon, SAR-UNet: Small Attention Residual UNet for Explainable Nowcasting Tasks, accepted for publication In proc. of IEEE-IJCNN, 2023.

Cloud cover nowcasting task:

- Four selected levels to summarize the network
- Activation heatmaps of the Residual DSC blocks



- ✓ Encoder paths are activated at the borders between cloud and non-cloud zones.
- ✓ Decoder paths: activation zones are more in the center of the cloudy areas of the image.

R Mathieu, S. Mehrkanoon, SAR-UNet: Small Attention Residual UNet for Explainable Nowcasting Tasks, accepted for publication In proc. of IEEE-IJCNN, 2023.

References:

[1] S. Mehrkanoon, Deep shared representation learning for weather elements forecasting, *Knowledge-Based Systems*, Vol 179, pp. 120-128, Sept 2019.

[2] K. Trebing, T. Stanczyk, S Mehrkanoon, SmaAt-UNet: Precipitation nowcasting using a small attention-UNet architecture, Pattern Recognition Letter, 145, 178-186, 2021.

[3] K. Trebing, S Mehrkanoon, Wind speed prediction using multidimensional convolutional neural networks, IEEE Symposium Series on Computational Intelligence (IEEE-SSCI), vol. 79, pp. 713-720, 2020.

[4] T. Stanczyk and S. Mehrkanoon, Deep Graph Convolutional Networks for Wind Speed Prediction, In proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), pp. 147-152, 2021.

[5] J.G Fernández, IA Abdellaoui, S. Mehrkanoon, Deep coastal sea elements forecasting using UNet-based models, Knowledge-Based Systems, vol. 252, pp. 2022.

[6] J.G.Fernández, S. Mehrkanoon, Broad-UNet: Multi-scale feature learning for nowcasting tasks, Neural Networks, vol 144, pp. 419-427, 2021.

For other related data-driven based models see:

https://sites.google.com/view/siamak-mehrkanoon/projects

Thank you for your attention!