Comparative Analysis of Prediction Models for Short-Term Forecasting

Marco Di Francesco
Gabriele Franch
Mauro Brunato

Outline

Introduction

Dataset

Models

Results

Introduction

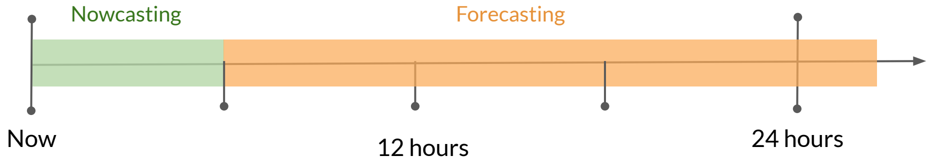
Nowcasting

Nowcasting makes the **prediction up to 6 hours**

Due to global warming extreme events are becoming more likely

Trentino Moena Flood in 2018

Goal is to predict these extreme events



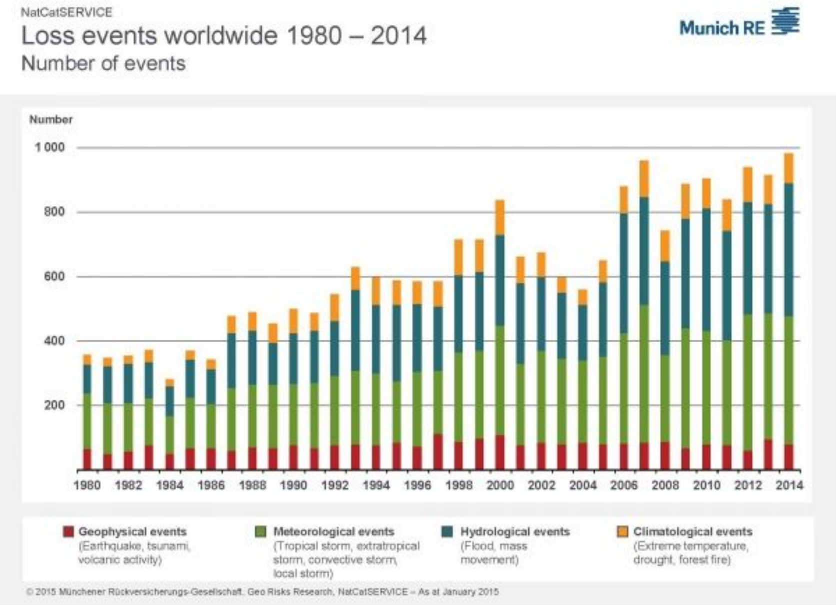
Nowcasting

Nowcasting makes the prediction up to 6 hours

Due to global warming **extreme events** are becoming more likely

Trentino Moena Flood in 2018

Goal is to predict these extreme events



Why nowcasting?

Nowcasting makes the prediction up to 6 hours

Due to global warming extreme events are becoming more likely

Trentino **Moena Flood** in 2018

Goal is to predict these extreme events

Why nowcasting?

Nowcasting makes the prediction up to 6 hours

Due to global warming extreme events are becoming more likely

Trentino Moena Flood in 2018

Goal is to predict these **extreme events**

Dataset

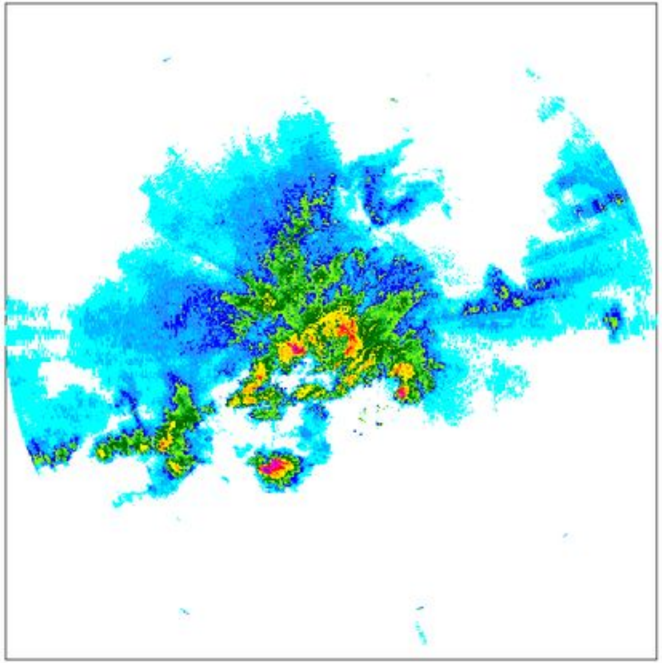
Radar

Radar uses **Radio waves** to create a radar echo map

Radar measures the **reflectivity value** (Z)

Reflectivity value is transformed in Decibel (dBZ)

Radar echo map

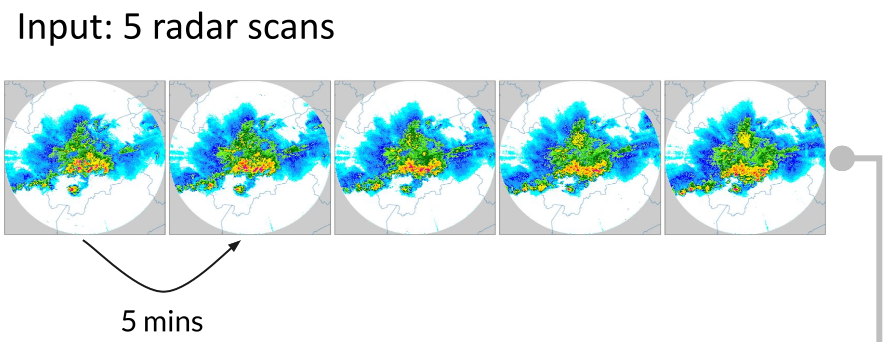


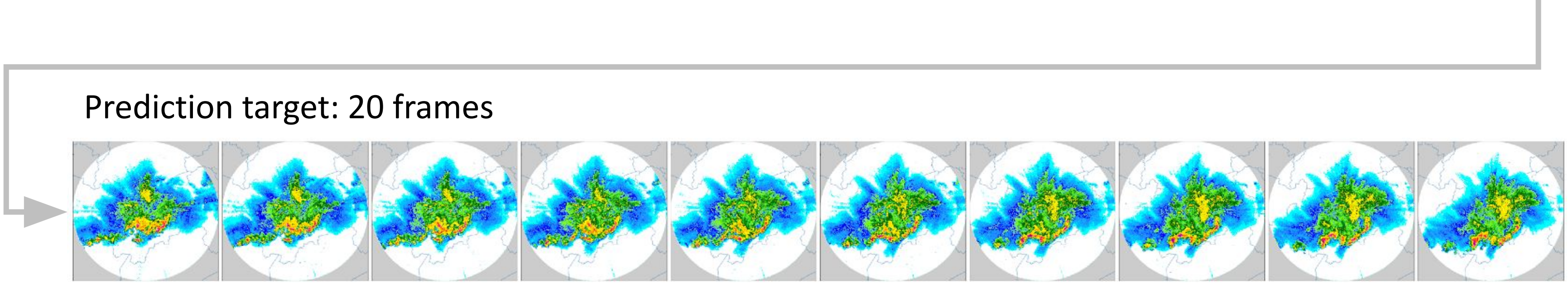
Dataset

MeteoTrentino radar dataset

Image size 480x480px

Window of 25 frames





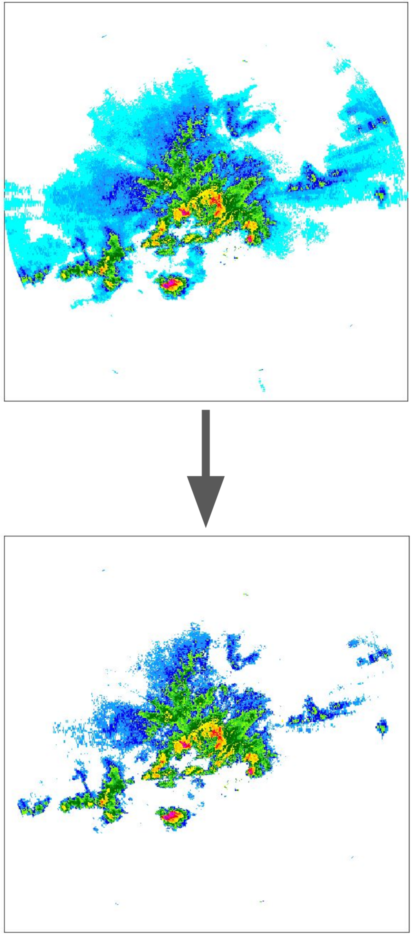
Thresholding

Input threshold → Set pixel value below threshold to 0

- Why? Machine learning models learn faster with less irrelevant information

Output threshold → Used in output image

- If pixel value ≥ threshold → Event happened
- If pixel value < threshold → Event did not happen



Thresholding

Input threshold → Set pixel value below threshold to 0

- Why? Machine learning models learn faster with less irrelevant information

Output threshold → Used in output and image

- If pixel value ≥ threshold → Event happened
- If pixel value < threshold → Event did not happen

Threshold: 0.5

0	0.3	0.7
0.4	0.5	0.4
0.2	0.6	0.3

Model performance metrics

Contingency table used for comparison of

Observed Event vs Forecast

Performance metrics:

$$FAR = \frac{false\ alarms}{hits + false\ alarms}$$

$$POD = \frac{hits}{hits + misses}$$

$$CSI = \frac{hits}{hits + misses + false\ alarms}$$

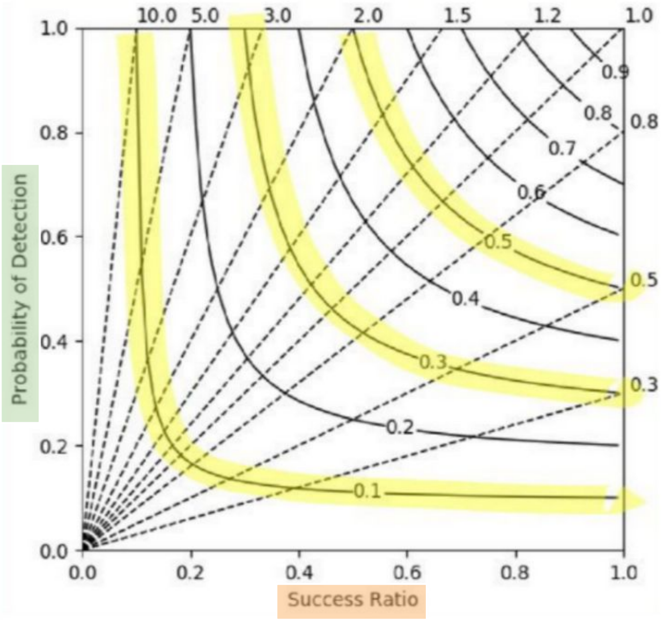
Contingency table		Observed event	
		✅	❌
Forecast	✅	Hit	False alarm
	❌	Miss	Hit

Model performance plot

Roebber plot used to represent the metrics:

- Success Ratio (1-*False alarm rate*)
- Probability of detection
- Critical Success Index

Features the **Conditional Bias**



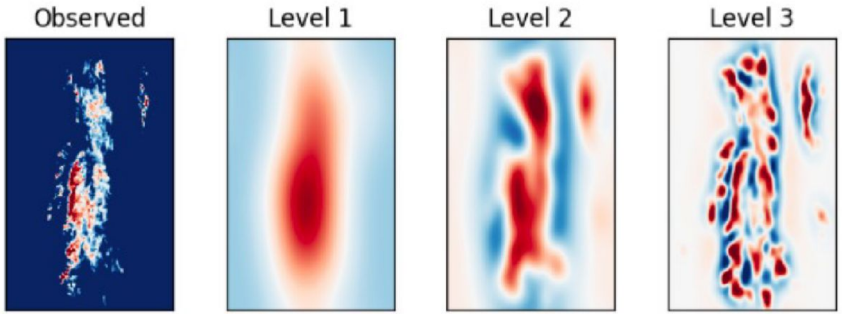
Models

S-PROG

Mathematical method presented in 2003

Decomposes the filed in different level using the fourier transformation decomposition

Computes Advection Matrix using semi-lagrangian extrapolation method

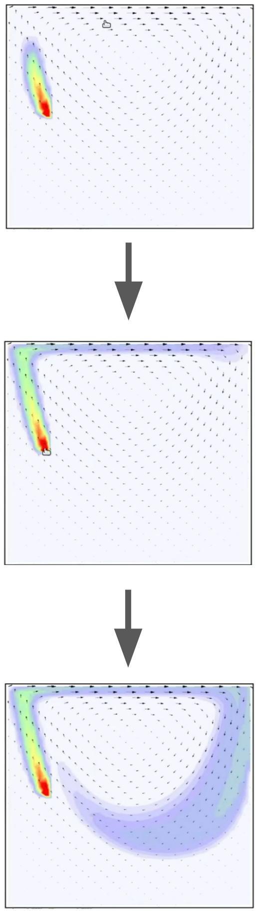


S-PROG

Mathematical method presented in 2003

Decomposes the filed in different level using the fourier transformation decomposition

Computes **Advection Matrix** using semi-lagrangian extrapolation method



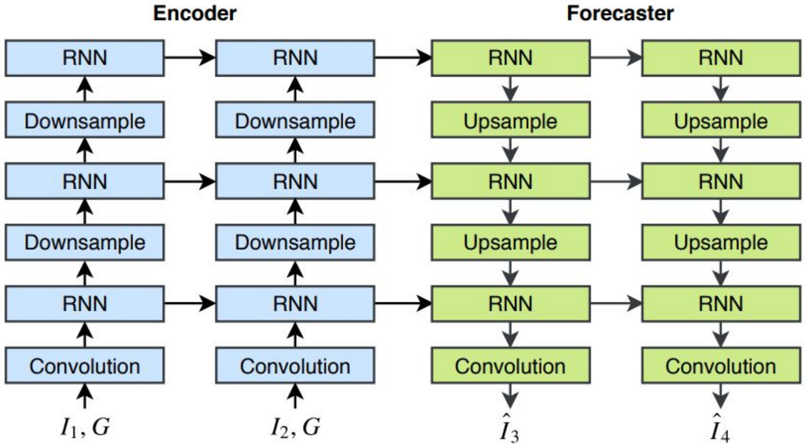
Trajectory GRU

Machine learning model presented in 2017

Uses a **Convolutional Recurrent Neural Network**

Solves the Location-Invariant problem

Uses a Balanced Loss



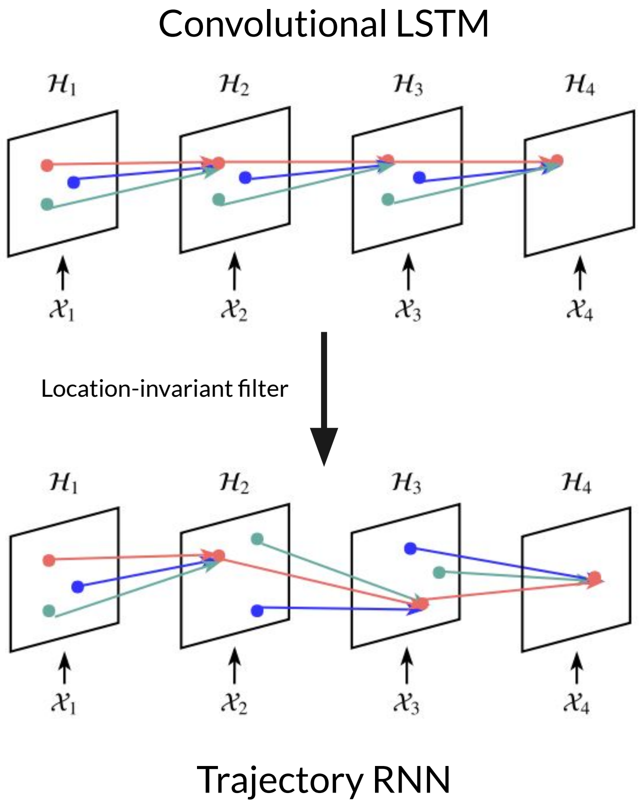
Trajectory GRU

Machine learning model presented in 2017

Uses a Convolutional Recurrent Neural Network

Solves the **Location-Invariant problem**

Uses a Balanced Loss



Trajectory GRU

Machine learning model presented in 2017

Uses a Convolutional Recurrent Neural Network

Solves the Location-Invariant problem

Uses a **Balanced Loss**

Rain Rate (mm/h)	Proportion (%)	Weight
$0 \leq x < 0.5$	90.25	1
$0.5 \leq x < 2$	4.38	1
$2 \leq x < 5$	2.46	2
$5 \leq x < 10$	1.35	5
$10 \leq x < 30$	1.14	10
$30 \leq x$	0.42	30

$$B-MSE = \frac{1}{n} * \sum_{n=1}^{20} \sum_{i=1}^{480} \sum_{j=1}^{480} w_{n,i,j} (x_{n,i,j} - \hat{x}_{n,i,j})^2$$

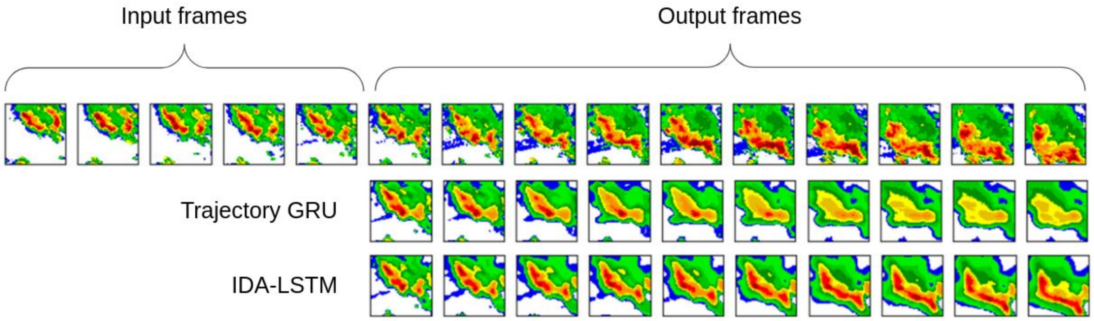
$$B-MAE = \frac{1}{n} * \sum_{n=1}^{20} \sum_{i=1}^{480} \sum_{j=1}^{480} w_{n,i,j} |x_{n,i,j} - \hat{x}_{n,i,j}|$$

IDA-LSTM

Machine learning model presented in 2021

Introduced **Interaction and Dual Attention** mechanisms

Solves the problem of high intensity precipitation

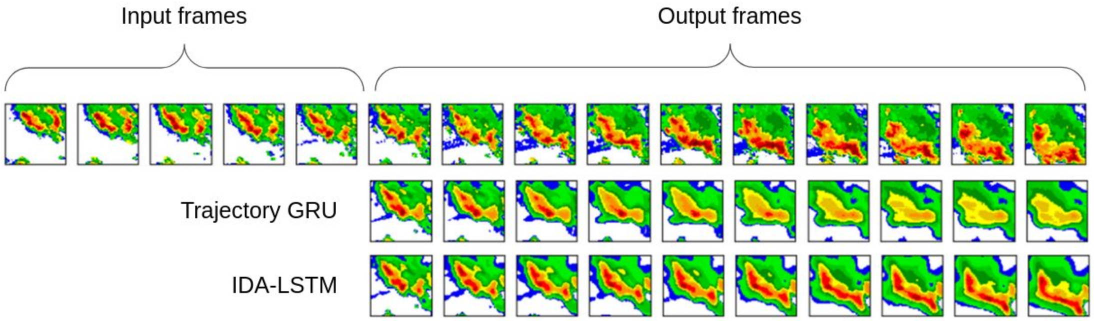


IDA-LSTM

Machine learning model presented in 2021

Introduced Interaction and Dual Attention mechanisms

Solves the **problem of high intensity precipitation**



Results

IDA-LSTM (2021)

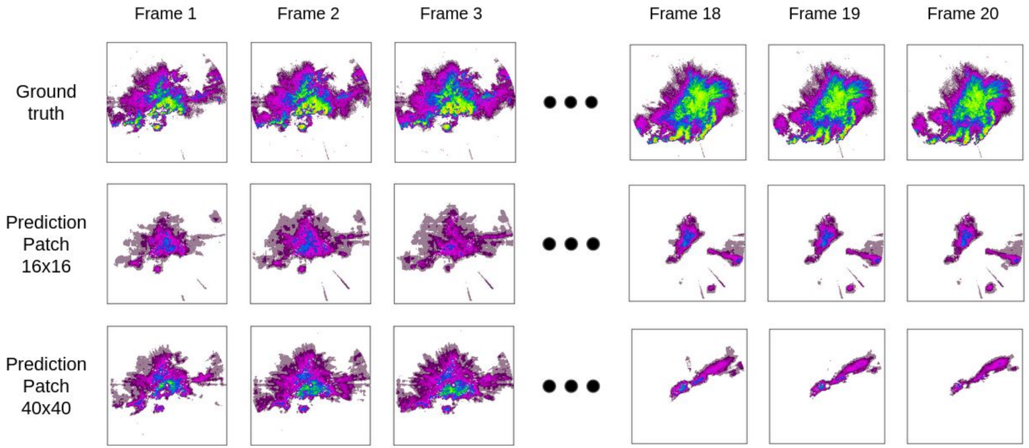
Model tested with patch size 16 and 40

Used a huge amount of video Video Memory (40GB)

Able to learn complex structures

Strongly **underestimated precipitation**

Unable to retain long-term structures



Trajectory GRU (2017)

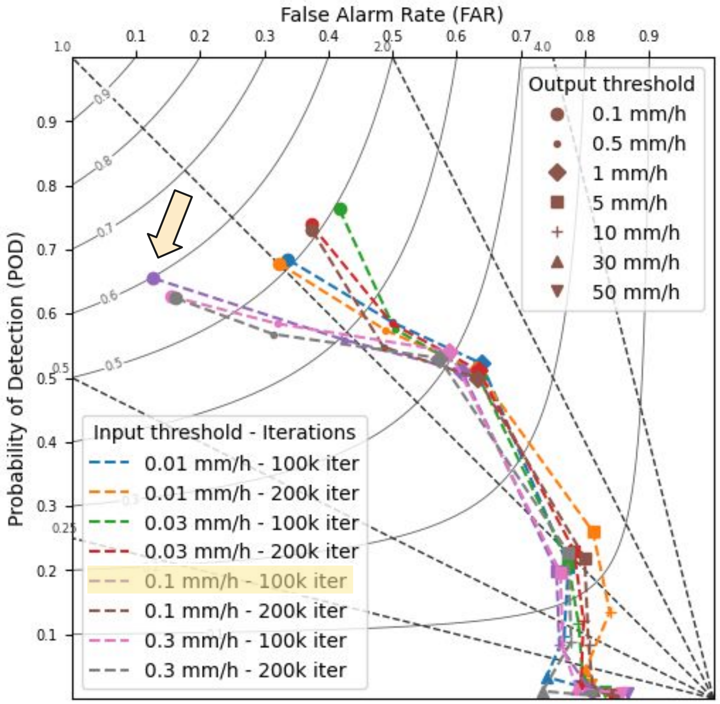
Trained on

- multiple input thresholds

Tested on

- multiple output thresholds
- 100k and 200k model's iterations

Best model found

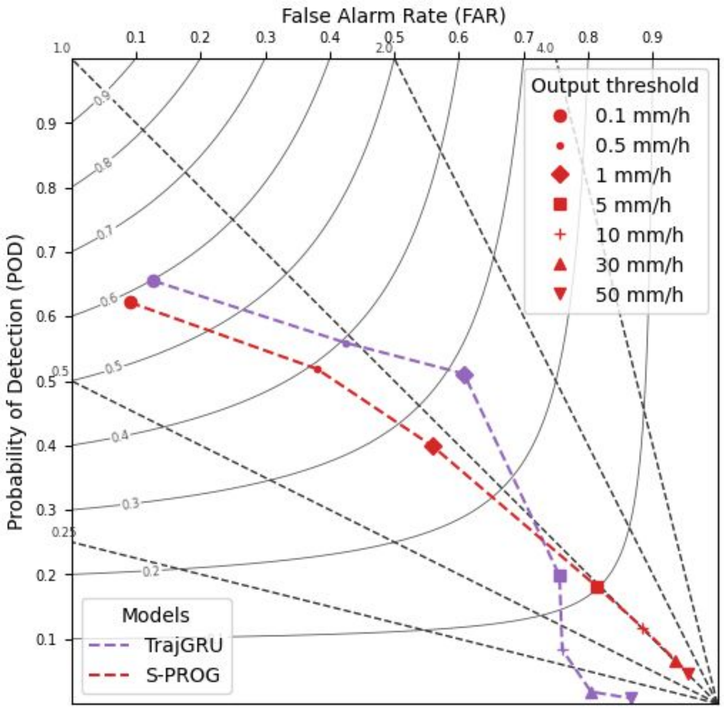


S-PROG (2003)

Tested on multiple output thresholds

The model achieves better results for low values

Keeps a **balanced conditional bias**



Wrap up

S-PROG (2003)

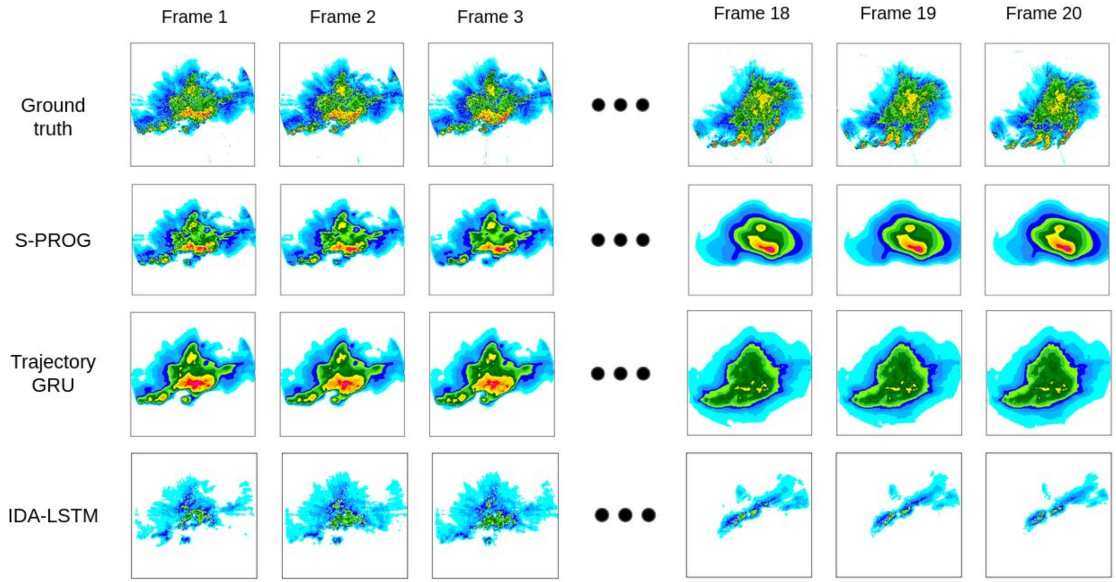
- Able to predict high values
- Unable to get the evolution

Trajectory GRU (2017)

- Good performance in low prec, but no high values

IDA-LSTM (2021)

- Strongly underestimates high and low values



Extra content

Dataset

- Transformation in Decibel
- Iteration vs Epoch
- Dataset window shift
- Dataset split

Models

- S-PROG (2003)
    - Math details
- Trajectory GRU (2017)
    - Location invariant problem
    - Convolution details
    - LR, Optimizer
- IDA-LSTM (2021)
    - Configuration
    - Horrible performance explanation

Thank you!

Trento, 14^{th} March 2022

Dataset - Split

- June 1, 2010, to December 31, 2019
- Split at the end of 2017
- 362,233 total frames
    - 193,611 training
    - 168,622 frames
- Spatial resolution of 500m
- Picture size of 480x480px
- Diameter 240km

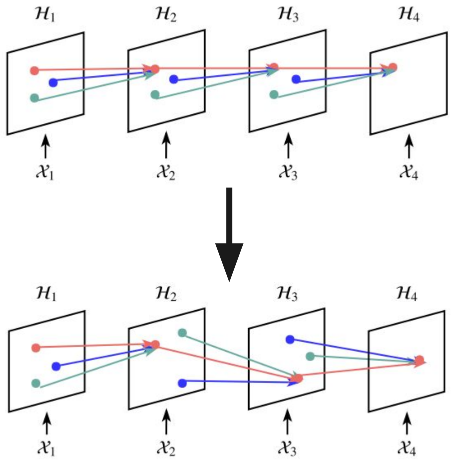
TrajGRU - Location invariant problem

In ConvLSTM **weights are fixed for all the locations**

This model proposes the **Location-invariant filter**

Here **recurrent connections are dynamically determined**

Meaning it considers how an object moves making the prediction (convolution) is moving



TrajGRU - Configuration

- Thresholds
  - Input (mm/h): 0.01, 0.03, 0.1, 0.3
  - Output (mm/h): 0.1, 0.5, 1, 5, 10, 30, 50
- LOC: 4555 lines
- Convolution: stride 2
- Batch size: 2
- Learning rate: 10^{-4}
- Iterations: 200.000
- Saving frequency: 10.000
- Optimizer:
  - Adam
  - Weight_decay: 10^{-6}

- GPU: NVidia GTX 1080, 8GB memory
- Training
  - Iterations: 200.000
  - Time: 96 hours
- Testing
  - Iterations: 80.000
  - Time: 36 hours

IDA-LSTM configuration

- Patch size: 16x16px
- LOC: 7452 lines
- Filter size: 5
- Stride: 1
- Batch size: 1
- Iteration: 80.000
- Loss: L1+L2 loss
- Optimizer:
  - Adam
  - Weight_decay: 10^{-6}

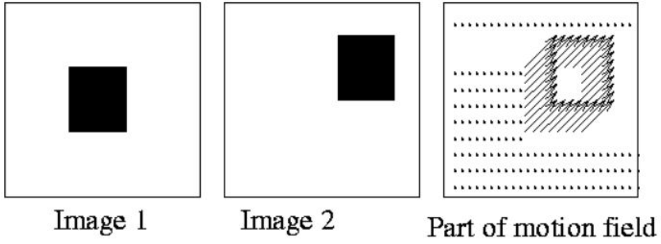
- Train 1:
  - Patch size: 16x16px
  - GPU: NVidia GTX 1080, 8GB memory
  - Iterations: 80.000
  - Full training time: 55 hours
- Train 2
  - Patch size: 40x40px
  - GPU: A100 Tensor Core
  - Iterations: 15.000
  - Full training time: 192 hours

S-PROG math details

Uses Autoregressive model Lag2

Approximates the Lagrangian space to separate

- motion of the field
- temporal evolution field



$$I(x, y, t)$$

$$u = dx/dt$$

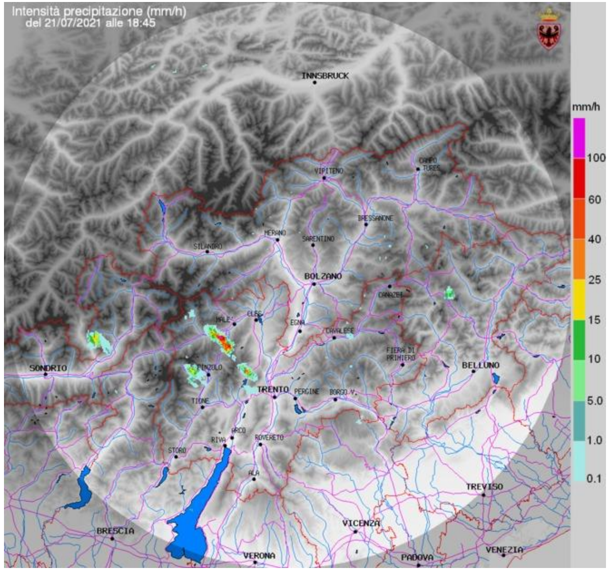
$$v = dy/dt$$

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0$$

Model Deployment

The model was deployed using:

- Docker container
- Server function (Azure function app)
- 1.5 GB ram
- ~23.5 sec to execute



It's going to replace MXNET model in MeteoTrentino website:

https://content.meteotrentino.it/dati-meteo/radar/loop/radar_inc_Temp_N.aspx