Adapting COALITION-4 nowcasting ML model to the NMA's operational context

Claudiu Adam

WeADL 2025 Workshop

The workshop is organized under the umbrella of WinDMiL, project funded by CCCDI-UEFISCDI, project number PN-IV-P7-7.1-PED-2024-0121, within PNCDI IV

Content of the presentation

- Objectives
- Constraints
- Technical framework
- Development and implementation
- Challenges
- Next steps

Objectives

- Adapt and integrate COALITION-4 ML-based nowcasting system for Romania's severe weather warnings
- Create a high-precision system for 0-1 hour predictions of heavy rain, hail, and lightning
- Integrate multiple data sources including radar, satellite imagery, lightning detection, NWP, and Romania's DEM
- Document the adaptation process and publish code on MeteoSwiss GitHub
- Evaluate alternative AI nowcasting models alongside COALITION-4 implementation

Constraints

- Adapting COALITION-4 is more efficient given resource constraints
- Model training limited to 2025-2026 convective seasons data due to FCI availability (next-generation imaging instrument onboard the new MTG satellite), NWP archive limitations, and radar product constraints
- Challenge of capturing extreme weather events due to their underrepresentation in training data
- Geographic adaptation concerns when applying models optimized for one region to Romania's specific topographic and climatic characteristics (solution: model fine-tuning)

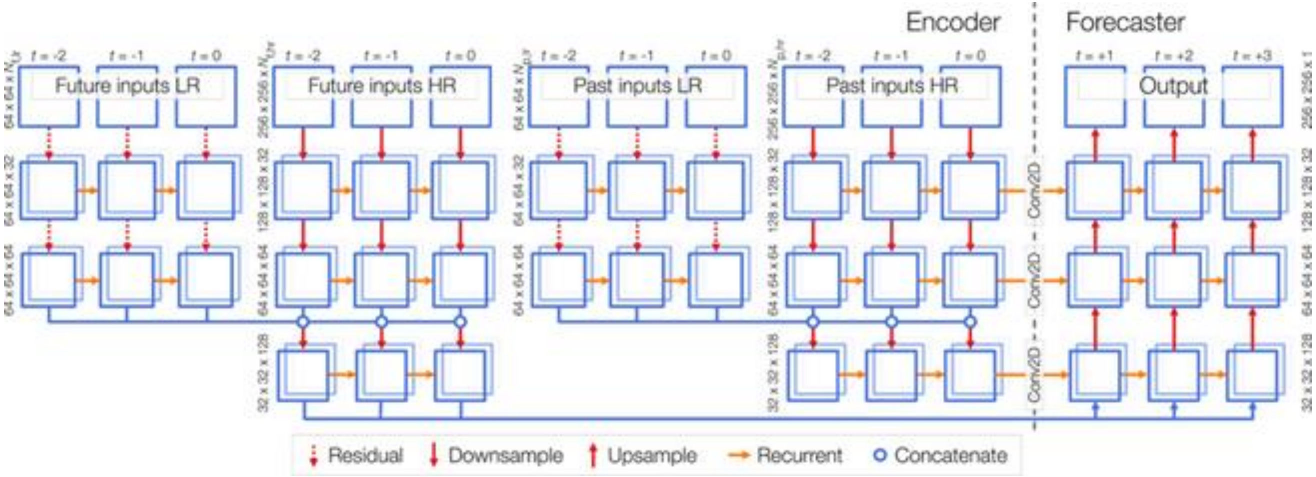
Technical framework (1) – Hardware, software, and model description

- Model uses recurrent-convolutional architecture with encoder-decoder structure for multi-hazard nowcasting (lightning, hail, precipitation)
- Hardware (workstations): Intel Xeon CPU, NVIDIA RTX A6000 GPU, 64/256GB RAM, 2TB SSD storage
- Software: Python 3.12.4, TensorFlow, Xarray, Dask, scikit-learn, CuPy for GPU-accelerated processing
- Data preprocessing leverages GPU acceleration to efficiently process 5-minute temporal resolution data
- Initial model demonstrates performance for short-term forecasts (5-60 mins), with radar data proving most valuable among input sources

Unitatea Executivă pentru Finanțarea Învățământului Superior, a Cercetării, Dezvoltării și Inovării

METEO ROMANIA

Technical framework (1) – Hardware, software, and model description



COALITION-4 architecture

Technical framework (2) – Data sources

- Radar – generated in NetCDF format, Leonardo weather radar network, 7 sites in NMA’s network
- NWP – generated in NetCDF format, ICON model, 2.8 km spatial resolution
- Satellite – NAT files converted to NetCDF files, SEVIRI/FCI data, 0.5 - 2 km spatial resolution
- Lightning – KML files converted to NetCDF files, LINET lightning detection network, 9 sensors in NMA’s network
- DEM – GeoTIFF file converted to NetCDF file, 1 km spatial resolution

Development and implementation (1) – Projection and datasets

- Reconfigured COALITION-4 to use Romania's Stereographic 1970 projection (the code to define the spatial reference system – EPSG:31700) instead of Swiss projection, included a 256 km buffer zone around borders to prevent boundary effects in the proposed patch creation process

- DEM data transformed from GeoTIFF to NetCDF with calculated terrain derivatives from altitude data (east-west and north-south slopes)

- Lightning data converted from KML to NetCDF format, generated binary occurrence maps at 10-minute intervals, density and current-weighted maps at 5-minute intervals

- Acquired 12 satellite products from EUMETSAT Data Store (captured by MSG using the onboard SEVIRI instrument) and processed into standardized NetCDF format

- NMA radar network with seven sites provides data every 5 minutes with multiple products

- Using ICON 2.8km NWP model instead of COSMO due to better regional performance with 10 parameters used by the ML model

Development and implementation (1) – Projection and datasets

Area extent of model domain

Development and implementation (2) – Patch creation and code optimization

- Identifying storm cells in radar data using DBSCAN clustering and extracting 256×256 pixel patches

- Aligning data sources with different resolutions through GPU-accelerated coordinate mapping and standardizing to 32×32 patches (the image on the right side)

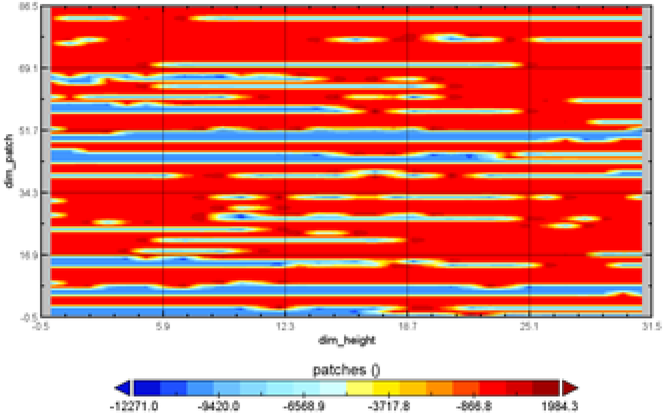
- Accelerated processing through parallel CPU operations and GPU acceleration (~1.5x faster)

- Exploring SVD-based dimensionality reduction and truncated matrix approximations to balance speed and accuracy

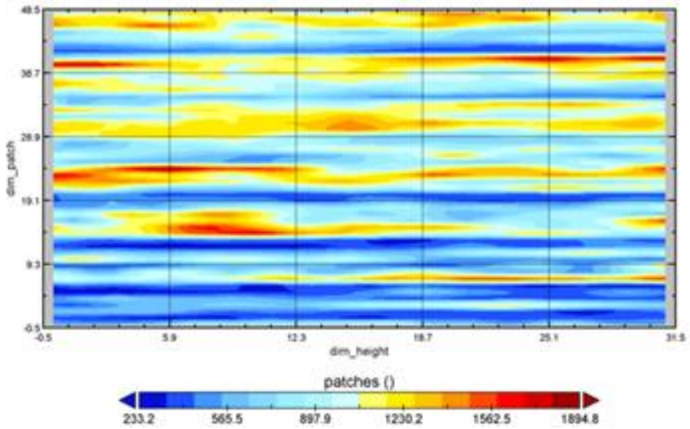
```python
def find_closest_indices_gpu(matrix, coords):
    """
    Find indices of closest points to upper left and lower right coordinates using GPU.

    Parameters:
    matrix: np.array of shape (M, N, 2) where [..., 0] is lat and [..., 1] is lon
    coords: tuple of (ul_lat, ul_lon, lr_lat, lr_lon)

    Returns:
    tuple: ((ul_i, ul_j), (lr_i, lr_j)) - indices for upper left and lower right points
    """
    # Transfer matrix to GPU if not already there
    if not isinstance(matrix, cp.ndarray):
        matrix_gpu = cp.asarray(matrix)
    else:
        matrix_gpu = matrix

    ul_lat, ul_lon, lr_lat, lr_lon = coords

    # Calculate Euclidean distance for both points on GPU
    # For upper left point
    ul_distances = cp.sqrt(
        (matrix_gpu[..., 0] - ul_lat)**2 +
        (matrix_gpu[..., 1] - ul_lon)**2
    )

    # For lower right point
    lr_distances = cp.sqrt(
        (matrix_gpu[..., 0] - lr_lat)**2 +
        (matrix_gpu[..., 1] - lr_lon)**2
    )

    # Get indices of minimum distances
    ul_idx_flat = cp.argmin(ul_distances)
    lr_idx_flat = cp.argmin(lr_distances)

    # Convert flat indices to multidimensional indices
    ul_idx = cp.unravel_index(ul_idx_flat, ul_distances.shape)
    lr_idx = cp.unravel_index(lr_idx_flat, lr_distances.shape)

    # Convert CuPy arrays to Python tuples for return
    ul_idx = tuple(idx.get().item() for idx in ul_idx)
    lr_idx = tuple(idx.get().item() for idx in lr_idx)

    # Return with inversed 1st and 3rd indices (as in the original function)
    return (lr_idx[0], ul_idx[1], ul_idx[0], lr_idx[1])
```

Development and implementation (2) – Patch creation and code optimization

Patches examples



Radar – RZC patches



DEM patches

Challenges

- Converting diverse data formats to standardized NetCDF with consistent spatial resolution across weather products presented a significant challenge
- Planned MSG to MTG transition will require processing pipeline modifications
- Numerical methods (SVD dimensionality reduction, truncated matrix approximations) require balance between computational efficiency and forecast accuracy

Next steps

- Process and integrate remaining radar and NWP products into ML model to obtain first training results (work in progress)
- Fine-tune COALITION-4 to adapt the model to NMA data and Romanian weather patterns
- Analyze product impact to potentially exclude less valuable inputs if performance remains within threshold
- Replace MSG data with newer MTG satellite data in the preprocessing pipeline
- Develop physics-based model to complement existing model in the prediction pipeline
- Evaluate data processing for optimization needs (parallel processing, GPU operations, numerical approximations)