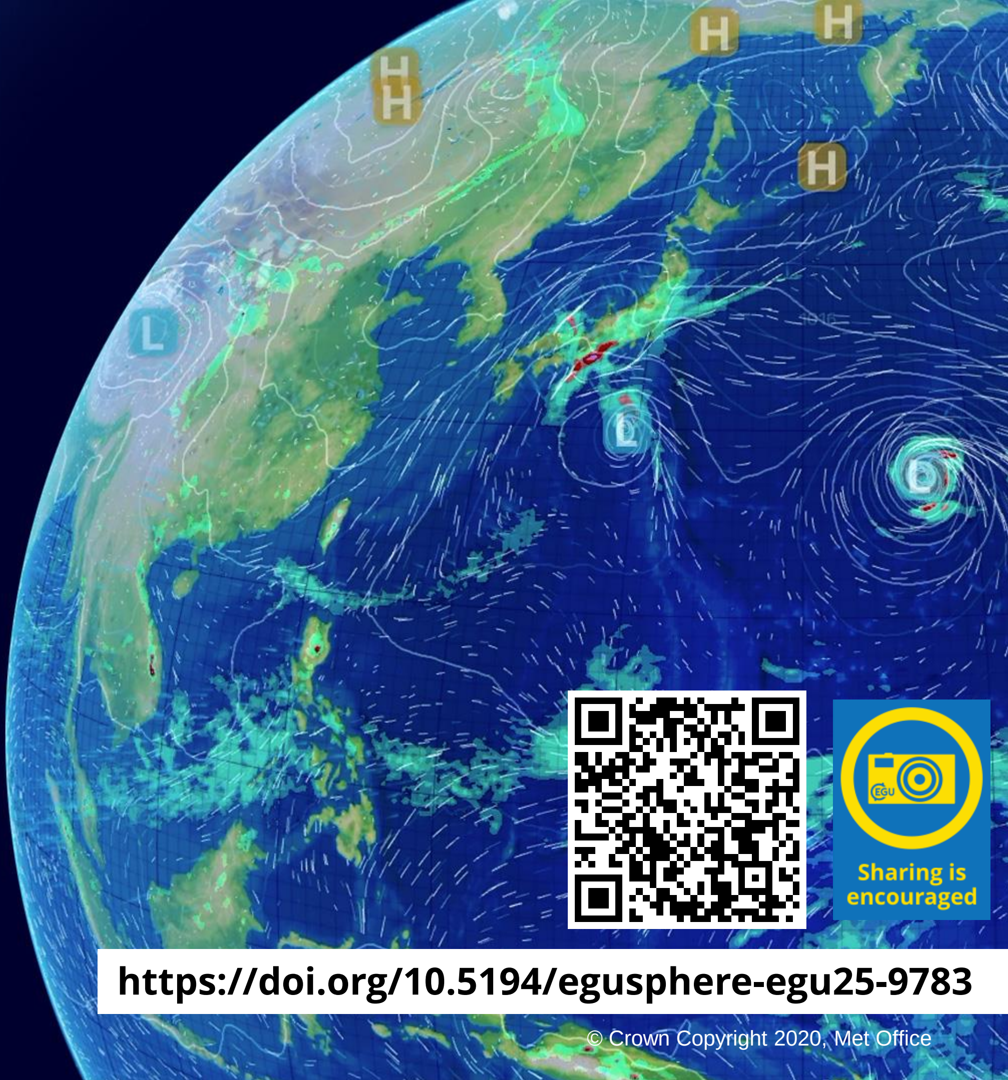
Met Office

A deep learning approach for probabilistic forecasts of cumulonimbus clouds from NWP data

Andrew Creswick
Aviation Application Scientist

EGU, AS1.1 28 April 2025

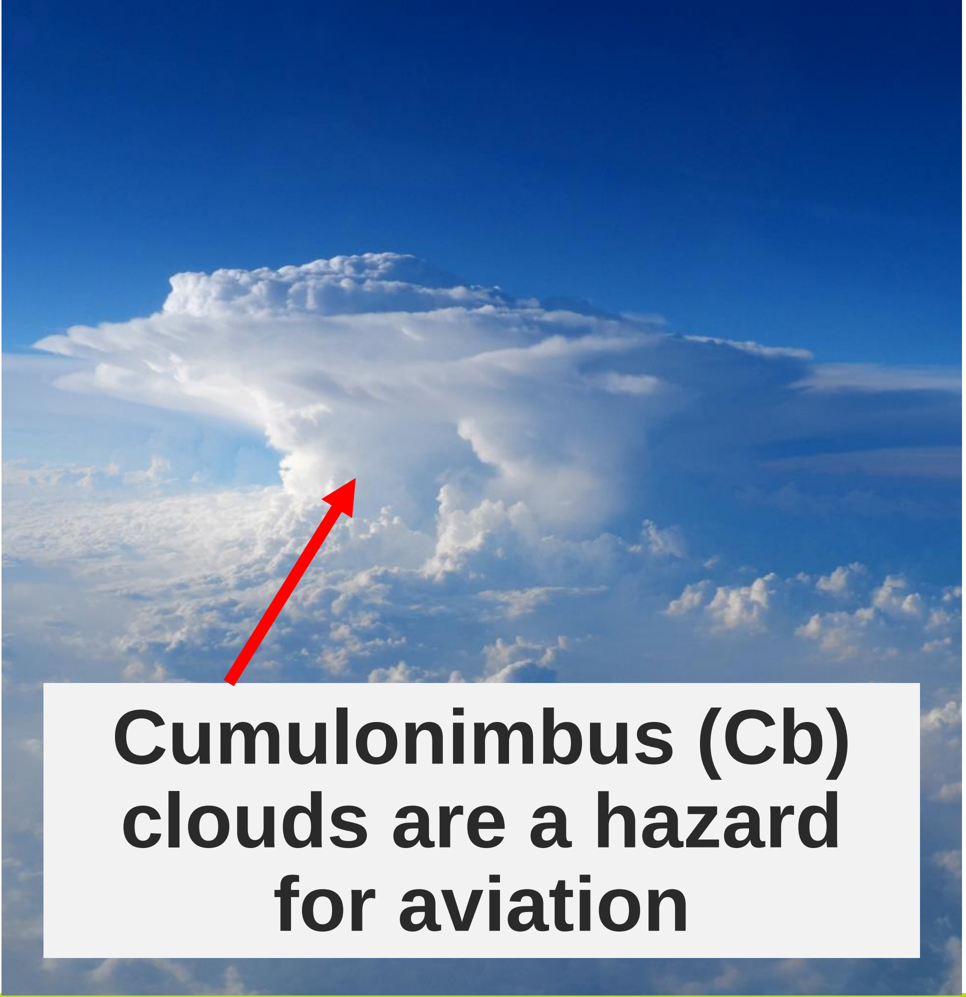


Sharing is encouraged

https://doi.org/10.5194/egusphere-egu25-9783

Met Office

Supplementary Materials



World Area Forecast System

Global meteorological data and hazard forecasts supplied for global civil aviation

Updating to probabilistic output in 2028

… including probabilistic forecasts of Cbs for period T+6 to T+48 in 3 hourly timesteps

Met Office

The Idea

Train a deep learning model to learn what a Cb “looks like” in weather model data.

Photo: Stewart Turner

Semantic image segmentation

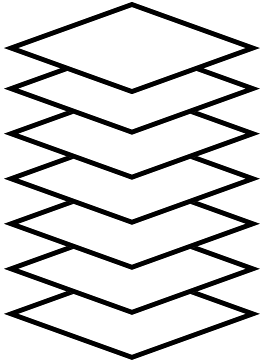
Segmentation
split image into multiple sections

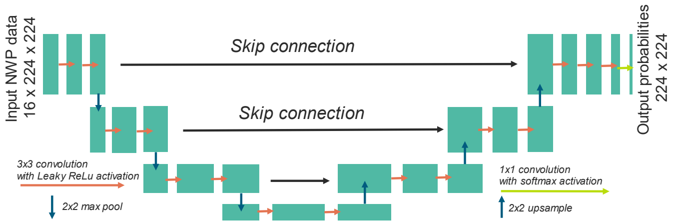
Semantic
attach meaningful label to sections

Classify each pixel in an image.
e.g. Vehicle, road, building…

Met Office

Cb prediction = semantic image segmentation





0.1	0.2	0.1
0.2	0.6	0.5
0.1	0.4	0.3

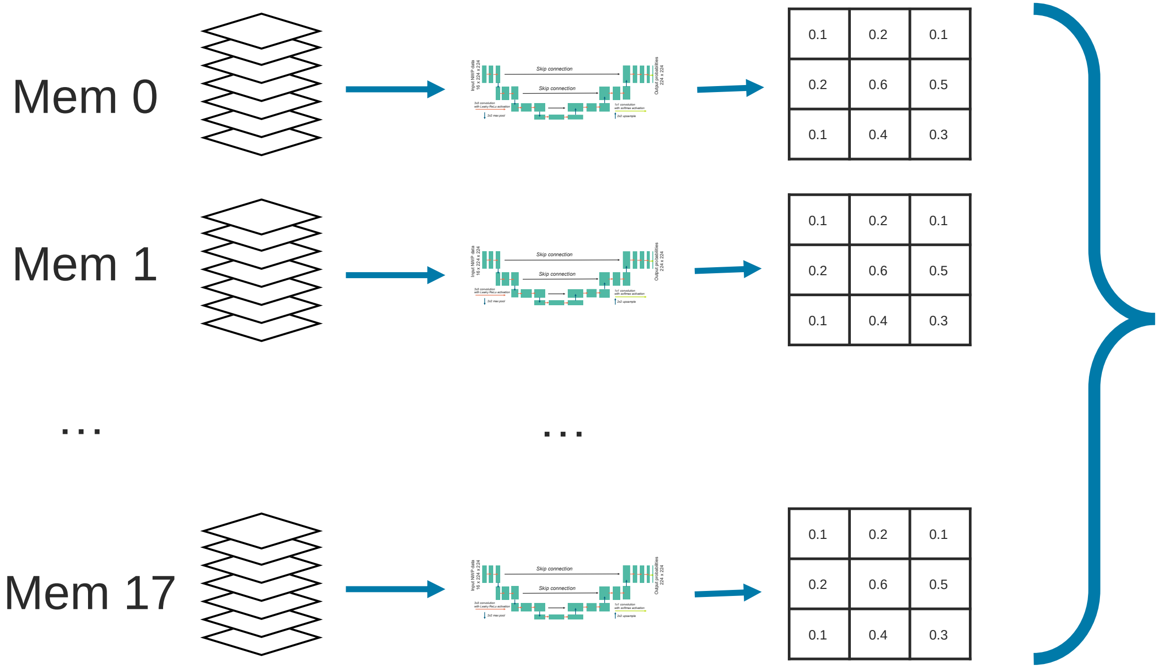
Input image
Set of parameters*
from single
MOGREPS-G member.

Segmentation model
e.g. FCN, U-Net **

Gridded forecast
Each grid point gives probability of classification as Cb

* 16 parameters including 3 flavours of CAPE and CIN, height of equilibrium level, cloud amount, precipitation rate
** U-Net diagram adapted from Ronnenberger et al, 2015; Lagerquist et al, 2021

Generating forecasts from whole ensemble



Take ensemble mean of these probabilities

Met Office

Choosing a Loss Function for model training

Avoid “double-penalty” problem → Apply neighbourhood methods

Cb/No Cb severely imbalanced → Use weighting parameters

Loss Functions Chosen

Brier Score	Binary Cross Entropy
• Probabilistic version of Mean Square Error. • Take max of observed Cbs within neighbourhood.	• Standard classification loss function • Take max of observed Cbs within neighbourhood.
(probabilistic) Fractions Skill Score • Compare fractions of events (forecast and obs) in a neighbourhood.	**Binary Focal Loss** • Extension of Cross Entropy • *Focussing* parameter gives higher weight to correct classification of rarer events. • Aims to deal with data imbalance.

Neighbourhood sizes: 0 (pixelwise), 3 (~ 60km), 5 (~100km)

Experiments

24 models trained and tested.

2 model architectures (UNet + FCN),

4 loss functions,

3 neighbourhood sizes.

3 months of training data – MJJ 2021

T+12 forecasts as input

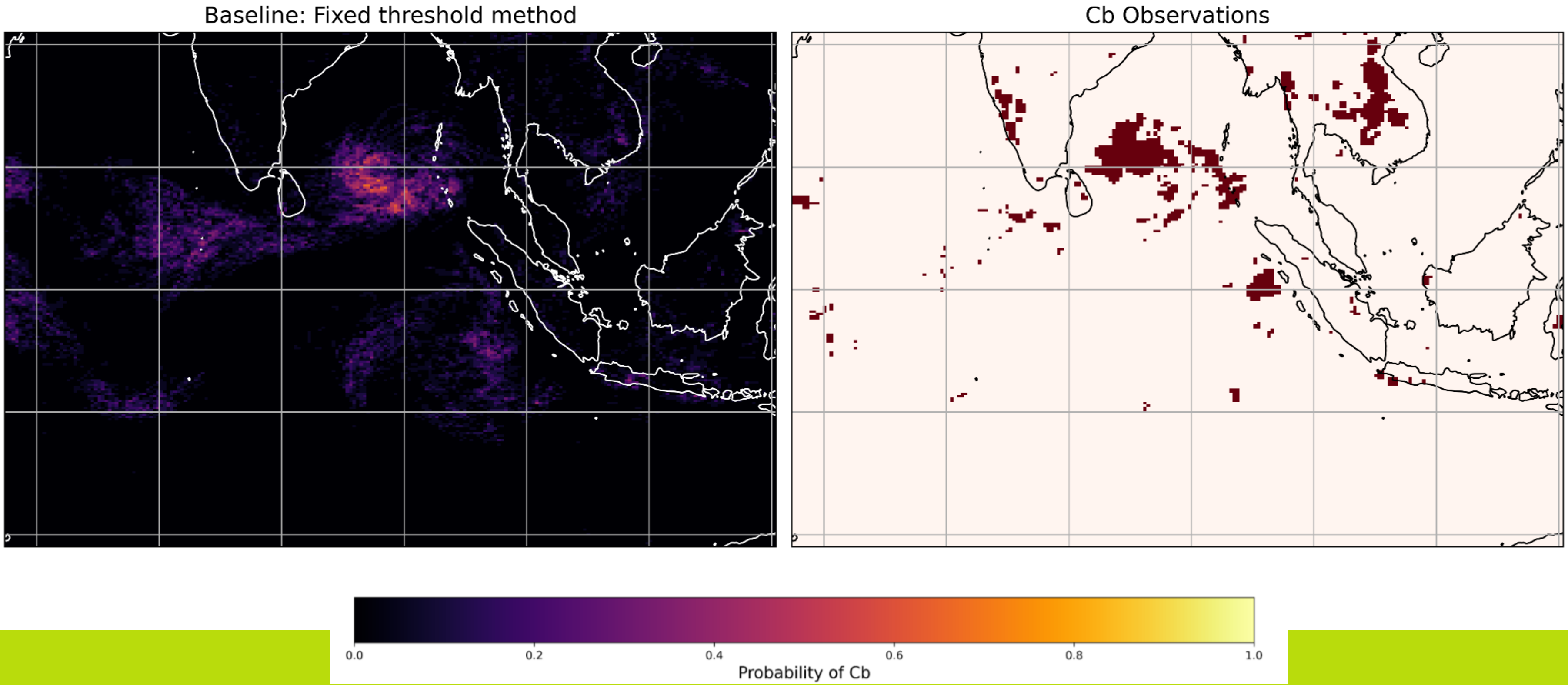
Satellite observations of Cbs as truth.

Only used control member from ensemble

Validation on MJJ 2022, Testing on MJJ 2023

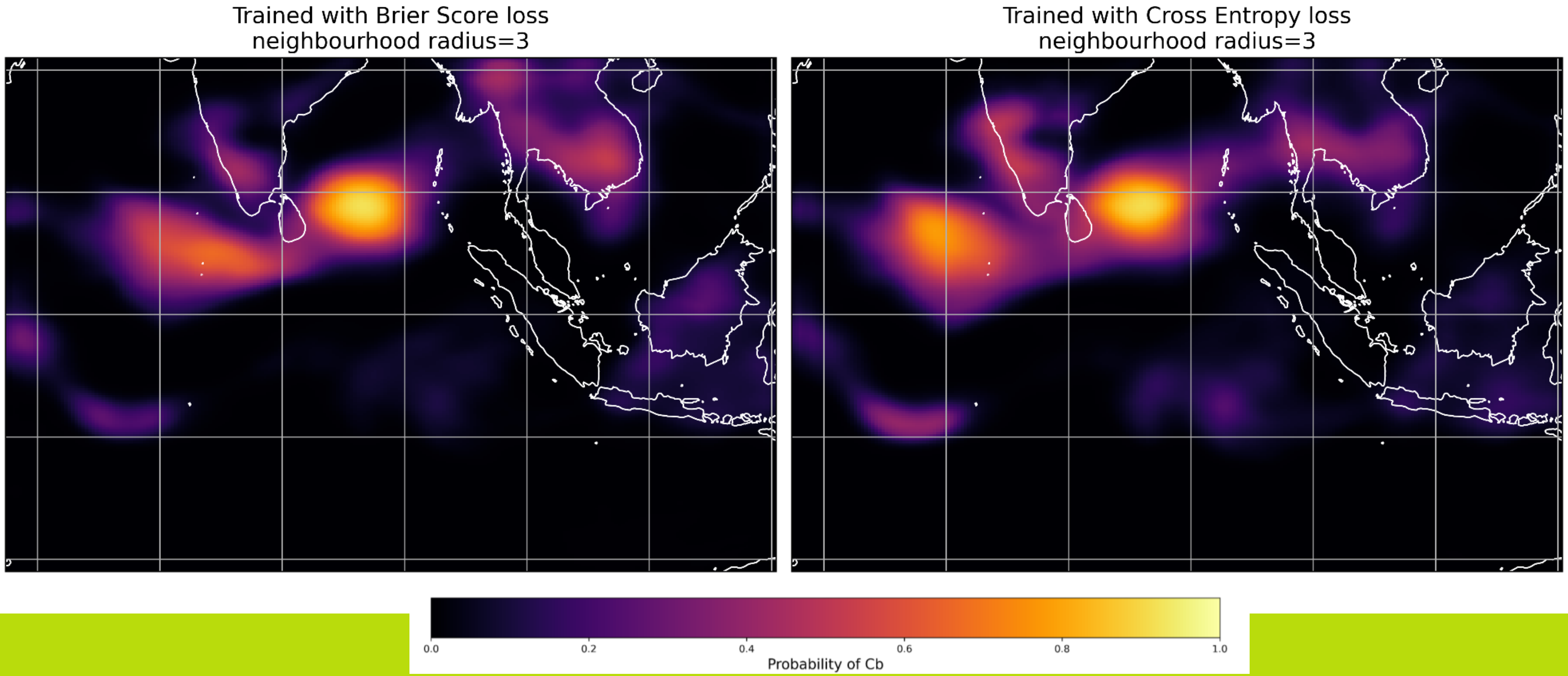
Met Office

What do the forecasts look like?

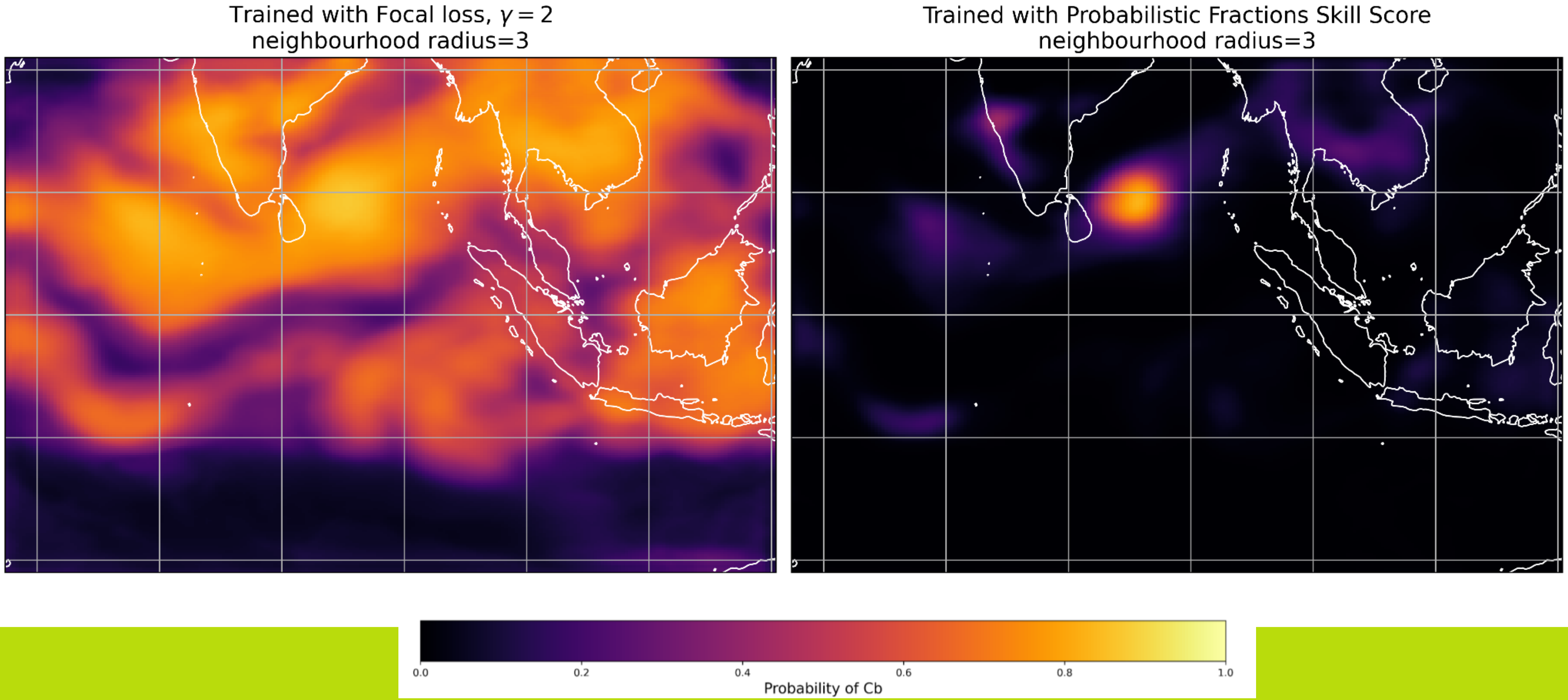


Met Office

What do the forecasts look like?



What do the forecasts look like?



Met Office

How do the forecasts perform?

- Pick out meteorological features.
- Smoother and more confident than baseline predictions.
- Improve on a baseline when compared with ROC/Reliability curves.

- Different loss functions gives different characteristics

Next Steps

More tuning of loss functions – neighbourhood size, weighting parameters.

Use whole ensemble in model training.

Detailed model evaluation and verification.

Predictions at multiple timesteps.

Summary

Image Segmentation models applied to produce forecasts of Cbs.

Spatially aware loss functions used during model training.

Choice of loss function important for characteristics and performance of models.

Please get in touch!

andrew.creswick@metoffice.gov.uk

https://doi.org/10.5194/egusphere-egu25-9783

Met Office

Segmentation Model - UNet

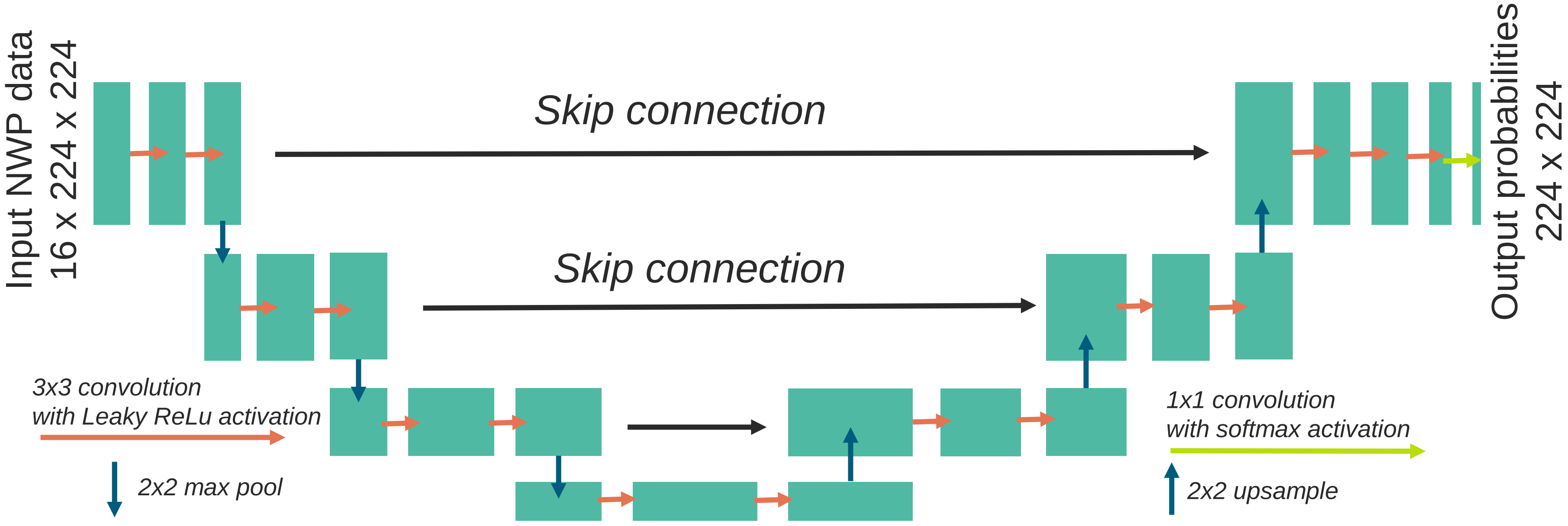
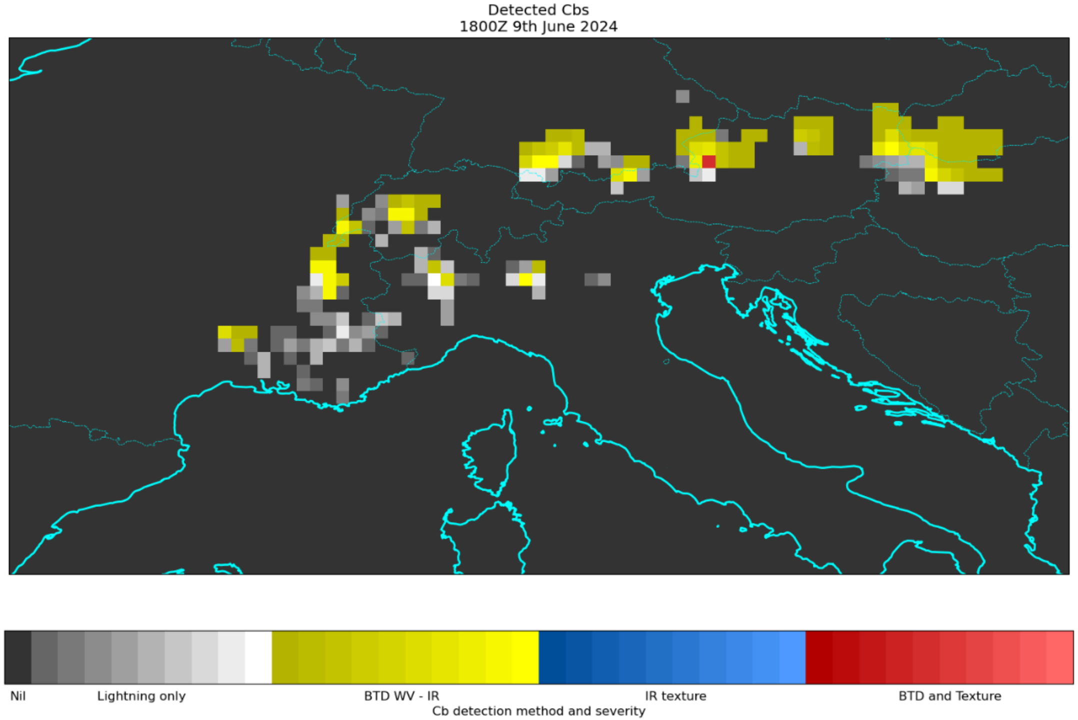


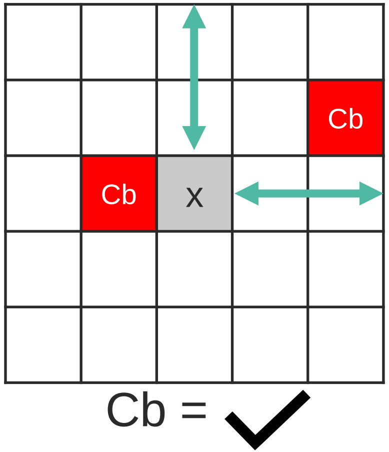
Image adapted from [Ronnenberger et al, 2015]; [Lagerquist et al, 2021]

Observation Dataset



- Satellite based Cb detection algorithm.
- Uses characteristics of overshooting tops in IR and WV channels.
- Input from lightning detectors.

“Spatially Aware” Loss Functions



Process truth/observations by looking in neighbourhood around each point.

Use neighbourhood truth in the traditional loss function definition.

Met Office

Loss Functions Chosen

Brier Score
- Probabilistic version of Mean Square Error

Binary Cross Entropy
- Standard classification loss function

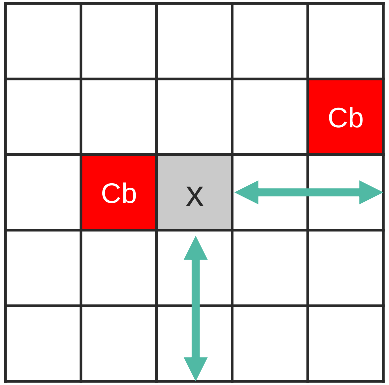
(probabilistic) Fractions Skill Score
- Compare fractions of events (forecast v obs) in a neighbourhood.

Binary Focal Loss
- Extension of Cross Entropy
- *Focussing* parameter gives higher weight to correct classification of rarer events.
- Aims to deal with data imbalance.

Neighbourhood sizes: 0 (pixelwise), 3 (~ 60km), 5 (~100km)

Neighbourhood Brier Score example

- Brier Score = Mean Square Error for probabilities
- $BS = \frac{1}{N}\sum_{i=1}^{N}\left(p_i - y_{r,i}\right)^2$
- p_i is probability of Cb at i^th gridpoint
- $y_{r,i}$ is 1 if Cb in neighbourhood of radius r around i^th gridpoint, else 0.
- N is the number of gridpoints.



$y_{2,i} = 1$

Met Office

Probabilistic Fractions Skill Score

$$pFSS = 1 - \frac{\sum_{i=1}^{N}\left(\overline{p_i}(r) - \overline{y_i}(r)\right)^2}{\sum_{i=1}^{N}\left(\overline{p_i}(r) + \overline{y_i}(r)\right)^2}$$

$\overline{y_i}(r)$ is the mean observed value (i.e. 0 or 1) within radius *r* of point *i.*
i.e. the fraction of points in the neighbourhood that have Cb

$\overline{p_i}(r)$ is the mean probability within radius *r* of point *i.*
i.e. the mean probability of Cb in the neighbourhood.

Met Office

Binary Cross Entropy Loss

$$BCE = -\frac{1}{N}\sum_{i=1}^{N}\log(p_{t,i}(r))$$

p_i is probability of Cb at i^th gridpoint
$y_{r,i}$ is 1 if Cb in neighbourhood of radius r around i^th gridpoint, else 0.
N is the number of gridpoints.

$$p_{t,i}(r) = \begin{cases} p_i, & if\ y_i^{max}(r) = 1 \\ 1 - p_i, & if\ y_i^{max}(r) = 0 \end{cases}$$

Focal Loss

$$FL = -\frac{1}{N}\sum_{i=1}^{N} \alpha_{t,i}(r)(1-p_{t,i}(r))^{\gamma}\log(p_{t,i}(r))$$

$$p_{t,i}(r) = \begin{cases} p_i, & if\ y_i^{max}(r) = 1 \\ 1-p_i, & if\ y_i^{max}(r) = 0 \end{cases}$$

$$\alpha_{t,i}(r) = \begin{cases} \alpha, & if\ y_i^{max}(r) = 1 \\ 1-\alpha, & if\ y_i^{max}(r) = 0 \end{cases}$$

γ is a tunable “focussing parameter” adjusting weight of more easily classified samples

Weighting factor, α adjusts for class imbalance

Focal Loss explained – weight α

- Weight α is given to the rare *cb (within neighbourhood)* event
- Weight $1 - \alpha$ to the common *no cb (within neighbourhood)* event.

- Value of α chosen by calculating the relative frequency of Cb/No Cb.

Focal Loss explained – tuning parameter $\gamma \geq 0$

- $(1 - p_{t,i}(r))^{\gamma}$

- Low prob of Cb, No Cb observed => $(1 - p_{t,i}(r))$ approx. 1
- Low prob of Cb, Cb observed => $(1 - p_{t,i}(r))$ small
- High prob of Cb, No Cb observed => $(1 - p_{t,i}(r))$ small
- High prob of Cb, Cb observed => $(1 - p_{t,i}(r))$ approx 1

- $(Approx\ 1)^{\gamma} \rightarrow 1$ as $\gamma \rightarrow \infty$
- $(small)^{\gamma} \rightarrow 0$ as $\gamma \rightarrow \infty$

Focal Loss explained – tuning parameter $\gamma \geq 0$

- So with larger values of γ the correct classification of the minority case (Cb) is given a lesser penalty.
- Incorrect classification of the majority case (no Cb) given greater penalty.

- So use higher values of γ to encourage better classification of the minority case.

Experiments

24 models trained and tested.

2 model architectures (UNet + FCN),

4 loss functions,

3 neighbourhood sizes.

3 months of training data – MJJ 2021

T+12 forecasts as input

Satellite Cb obs as truth.

Only used control member

Validation on MJJ 2022, Testing on MJJ 2023

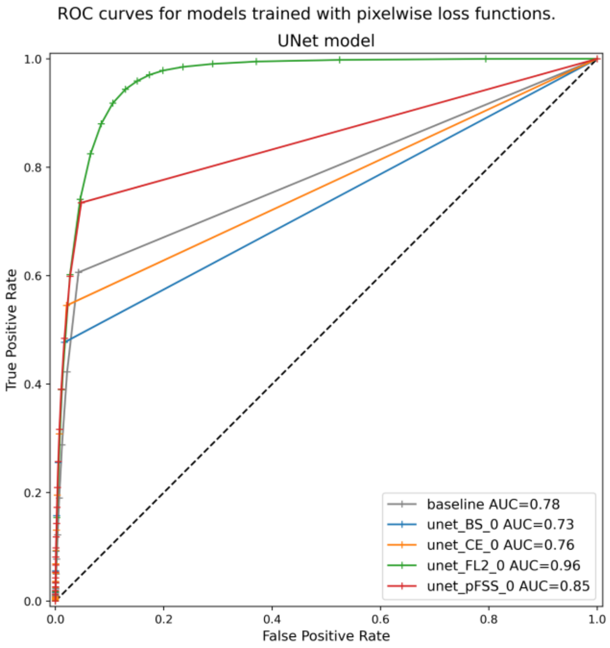
Experiments – more details

- 24 models trained and tested.
- Training data – May-July 21
- Validation data – May-July 22
- Test data – May – July 23
- Generate patches from original inputs, i.e. smaller sets of size 224 x 224 from the 1280 x 960 grid.

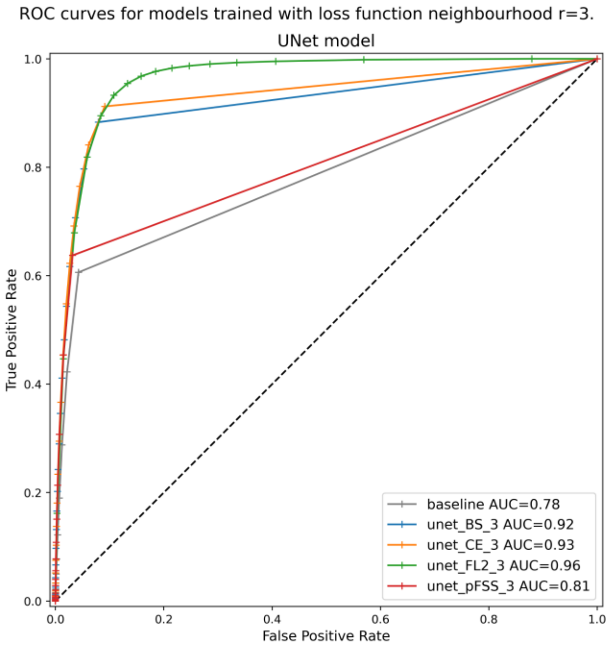
- 100 epochs
  - Early stopping of 30 epochs with no improvement.
  - Adam optimizer – reduce learning rate after 10 epochs with no improvement
  - 3.5 – 10 hours training time on 2 Nvidia A100 GPUs from orchid cluster on JASMIN.

Some verification results – On test set

ROC Curves

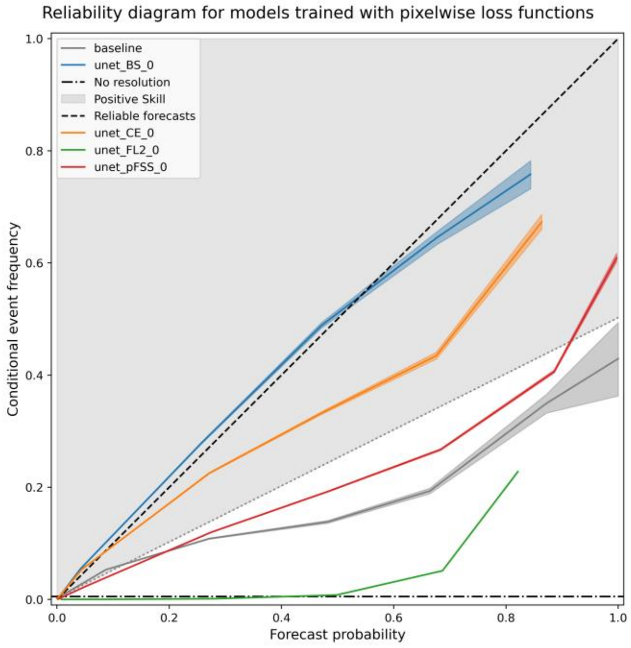


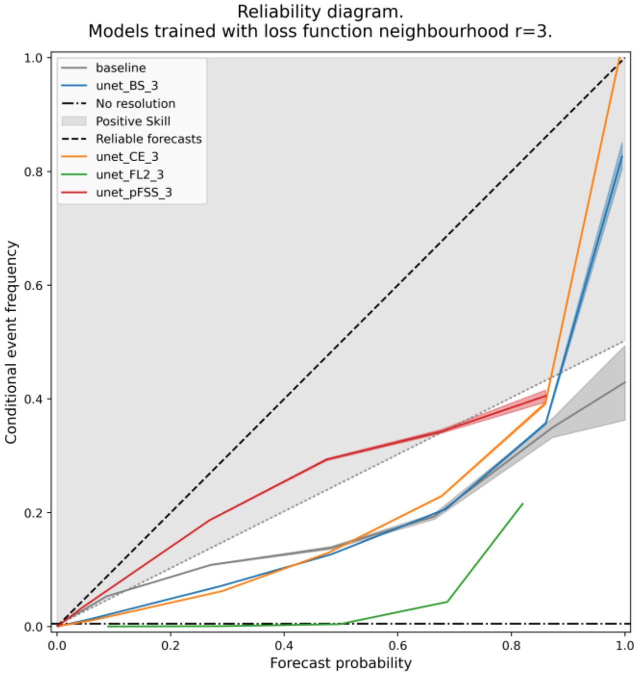
ROC curves for models trained with pixelwise loss functions.
UNet model
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
baseline AUC=0.78
unet_BS_0 AUC=0.73
unet_CE_0 AUC=0.76
unet_FL2_0 AUC=0.96
unet_pFSS_0 AUC=0.85



ROC curves for models trained with loss function neighbourhood r=3.
UNet model
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
baseline AUC=0.78
unet_BS_3 AUC=0.92
unet_CE_3 AUC=0.93
unet_FL2_3 AUC=0.96
unet_pFSS_3 AUC=0.81

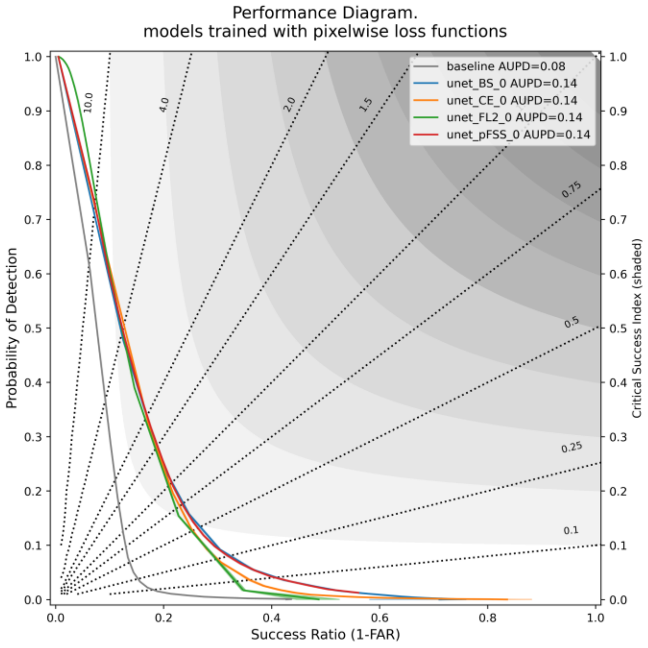
Reliability Diagrams

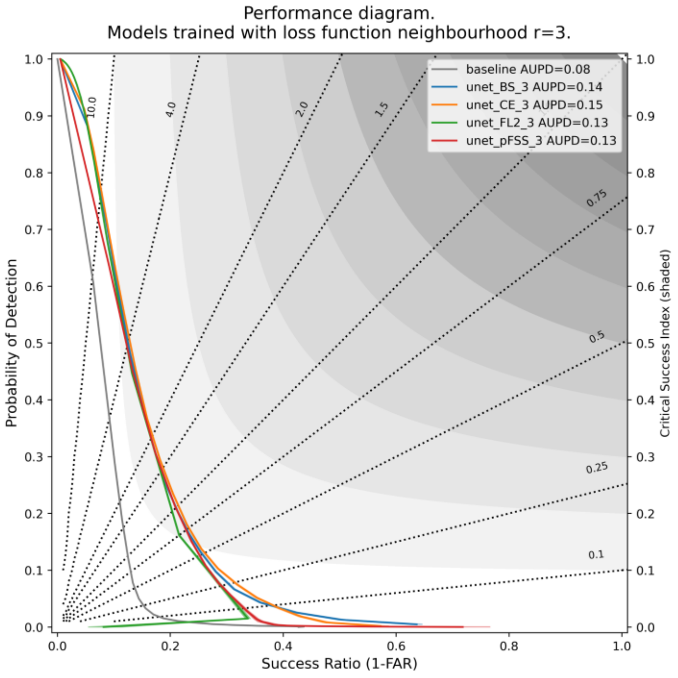




Met Office

Performance Diagrams





Next Steps: Using the whole ensemble

Can all the ensemble be used in training and prediction, …

… without having an unmanageably large training dataset, …

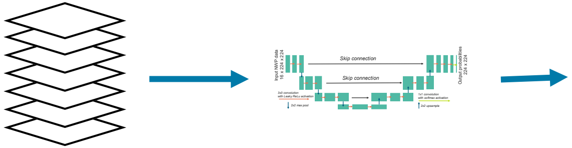
… and still take advantage of how the ML networks work.

The following slides suggest some approaches

Met Office

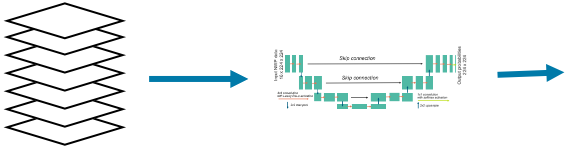
Generating forecasts from whole ensemble

Mem 0



0.1	0.2	0.1
0.2	0.6	0.5
0.1	0.4	0.3

Mem 1

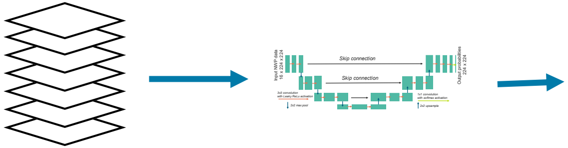


0.1	0.2	0.1
0.2	0.6	0.5
0.1	0.4	0.3

…

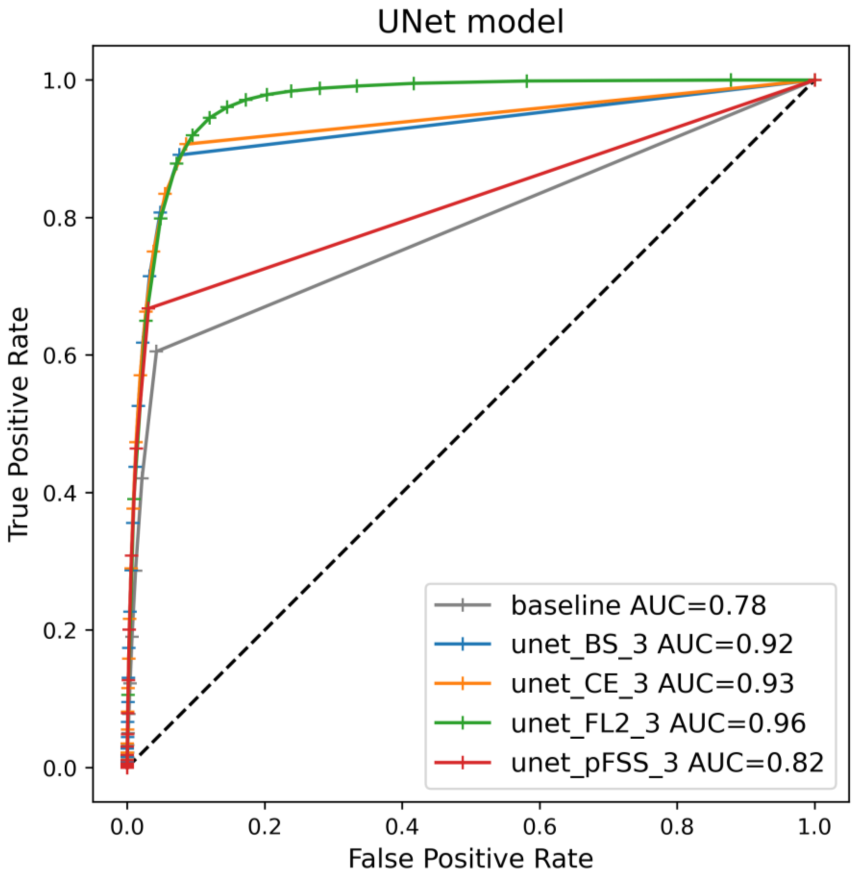
…

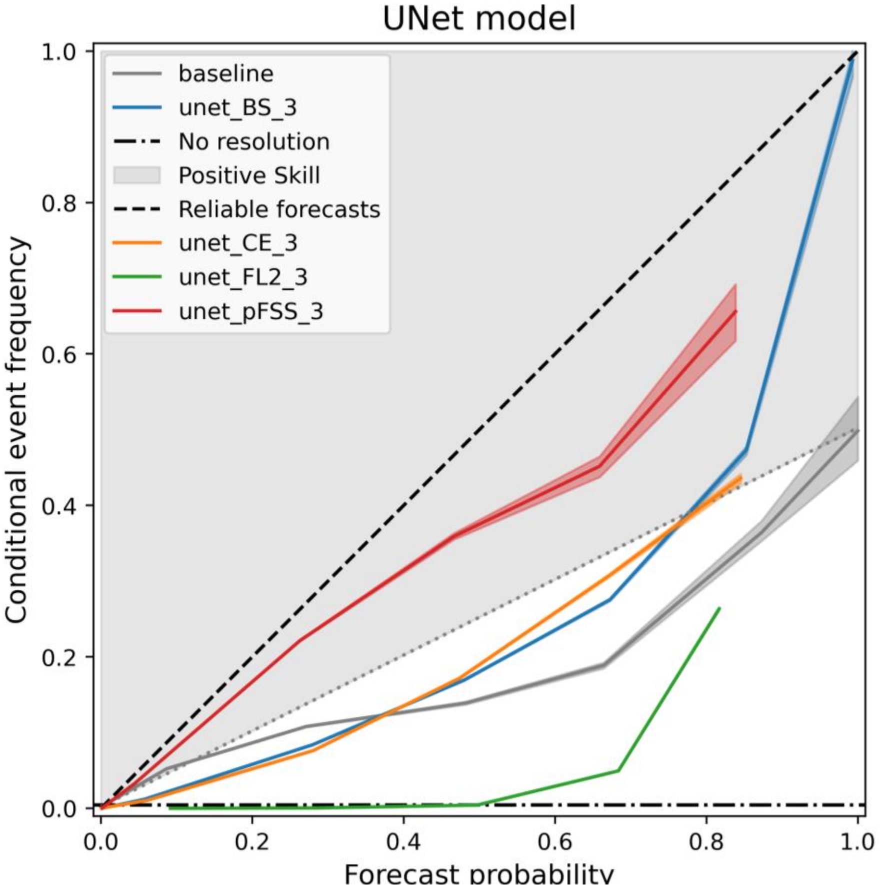
Mem 17



0.1	0.2	0.1
0.2	0.6	0.5
0.1	0.4	0.3

Take ensemble mean of these probabilities



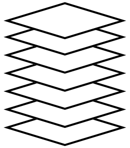


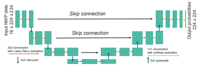
Loss function neighbourhood r=3, ensemble mean of probabilities, 1 month only

Met Office

Generating forecasts from whole ensemble – alternative approaches

Mem 0

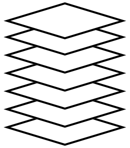


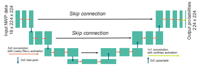


0.1	0.2	0.1
0.2	0.6	0.5
0.1	0.4	0.3

Cb if Prob > threshold

Mem 1





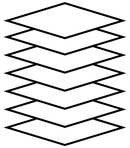
0.1	0.2	0.1
0.2	0.6	0.5
0.1	0.4	0.3

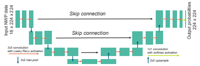
Cb if Prob > threshold

…

…

Mem 17





0.1	0.2	0.1
0.2	0.6	0.5
0.1	0.4	0.3

Cb if Prob > threshold

Count proportion of members with Cb present

Met Office

Generating forecasts from whole ensemble – alternative approaches

- Use percentiles for each parameter as inputs to an LSTM model [1]
  - E.g. CAPE 0,10, 25, 50, 75, 90, 100 percentile
  - But make predictions for each pixel in isolation – no spatial info

- Tree-based feature selection to choose model ensemble statistics [2]
  - Then these fed into a CNN

[1] Pre-tactical convection prediction for air traffic flow management using LSTM neural network - Jardines - 2024 – MetApps
[2] Thunderstorm prediction using convolutional neural networks to support air traffic management in Switzerland, EMS Annual Meeting 2024
[3] A machine-learning approach to thunderstorm forecasting through post-processing of simulation data - Vahid Yousefnia - 2024 - QJRMS

Software/Hardware used

- **Pytorch** – to write model architecture
- **Pytorch Lightning** – to handle model training, save checkpoints, use of multiple GPUs.
- **MLFlow** – tracking/logging of training runs.
- **TorchMetrics** – integrates more complex metrics with Lightning. Also implemented own metrics.
- **xarray** – to read in netcdf data
- **xbatcher** – to create patches from data in xarray. Minimises size of batch given at training time.

- Orchid GPU cluster on JASMIN
- Tried on standard Met Office scientific research compute resource (CPU based) but took way too long.
- Python packages (and almost all online support) based on using GPUs.

Please get in touch!

andrew.creswick@metoffice.gov.uk

https://doi.org/10.5194/egusphere-egu25-9783