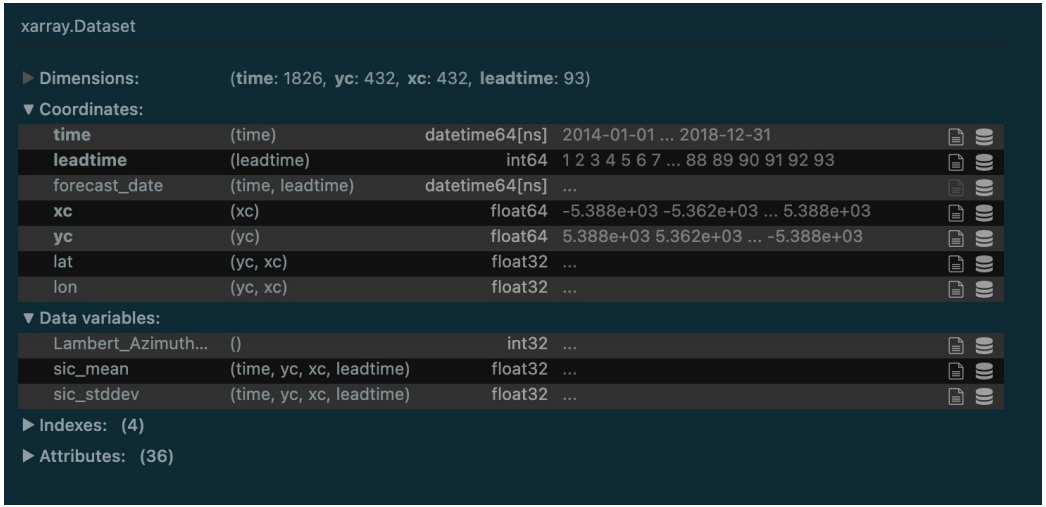
IceNet: A peek into validation commands

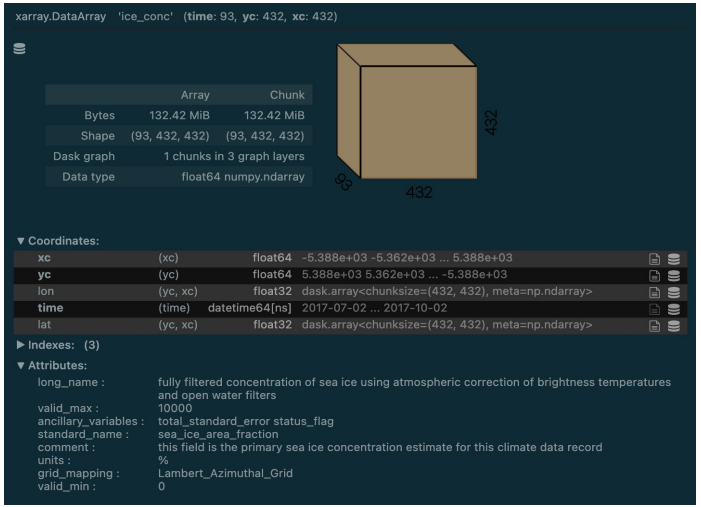
Ryan Chan

A peek into IceNet validation commands

After using the IceNet library to make forecasts, this is what our forecasts look like:



… and this is what our observed "ground truth" (OSISAF) data looks like:



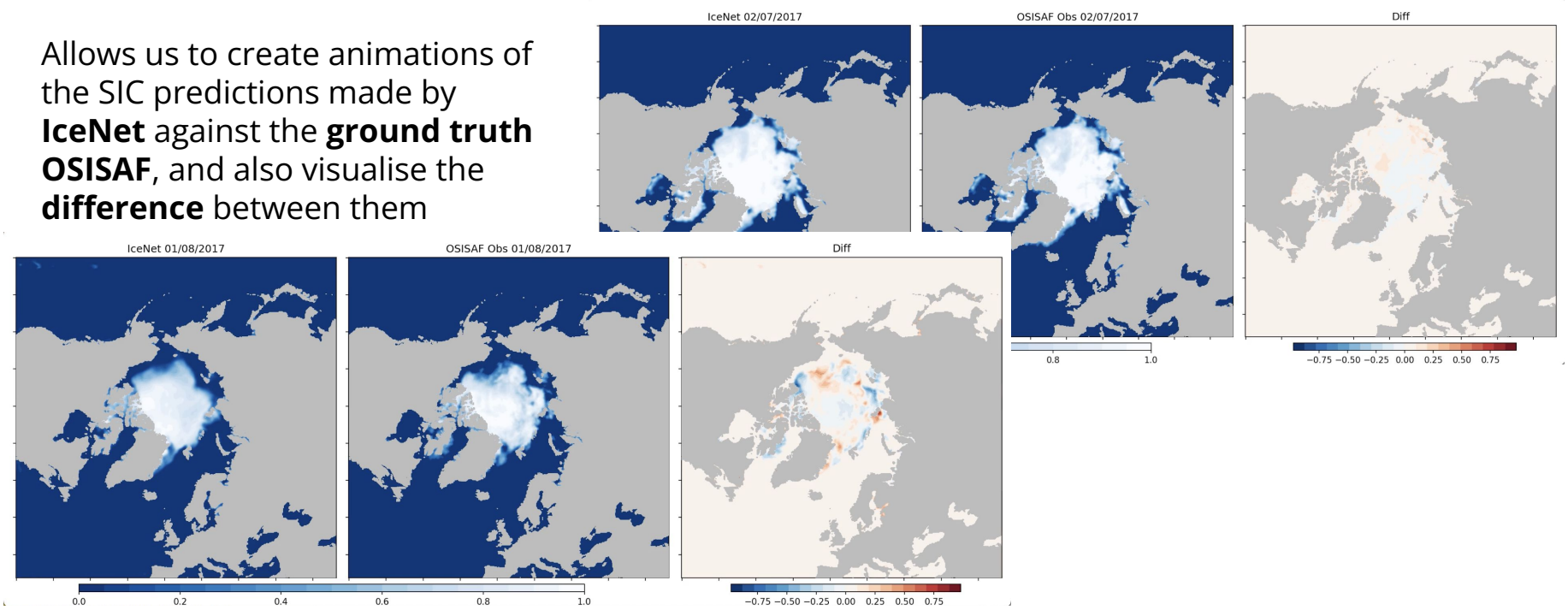
At each grid location, we have a **prediction** of SIC and a "ground truth" OSISAF measure of SIC

Question: How can we validate / evaluate our forecasts?

Visualising/animating sea ice concentration error

Using `sic_error_video` function (or `icenet_plot_sic_error` CLI)

Allows us to create animations of the SIC predictions made by **IceNet** against the **ground truth OSISAF**, and also visualise the **difference** between them



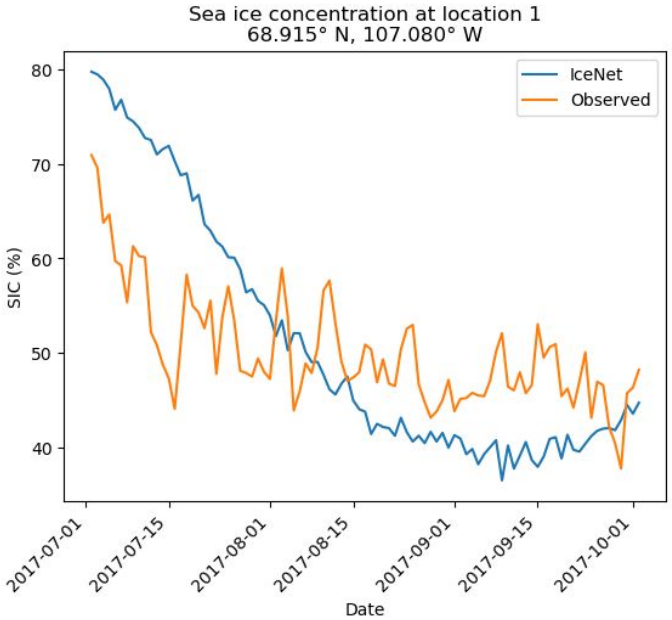
Visualising/animating sea ice concentration error

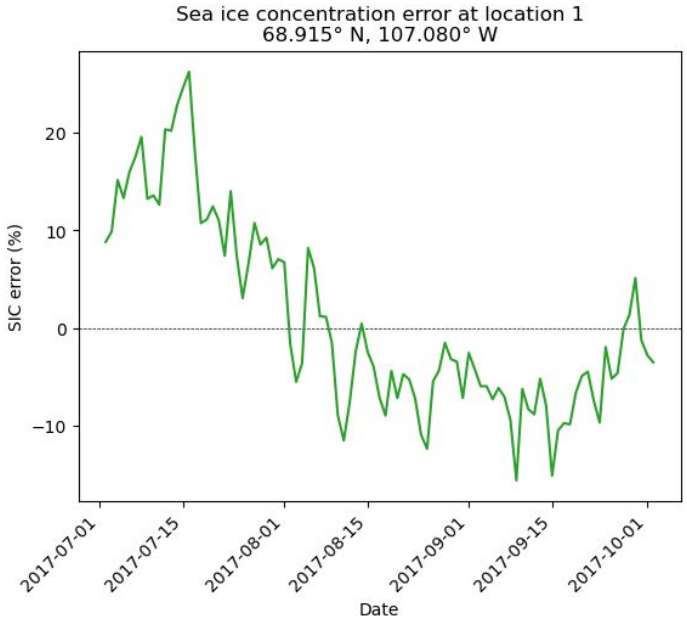
Using `sic_error_video` function (or `icenet_plot_sic_error` CLI)

Zooming into a particular grid location

Using `sic_error_local_plots` function (or `icenet_plot_sic_error_local` CLI)

Allows us to zoom in and evaluate the SIC error of the forecast at a **particular grid location / coordinate**

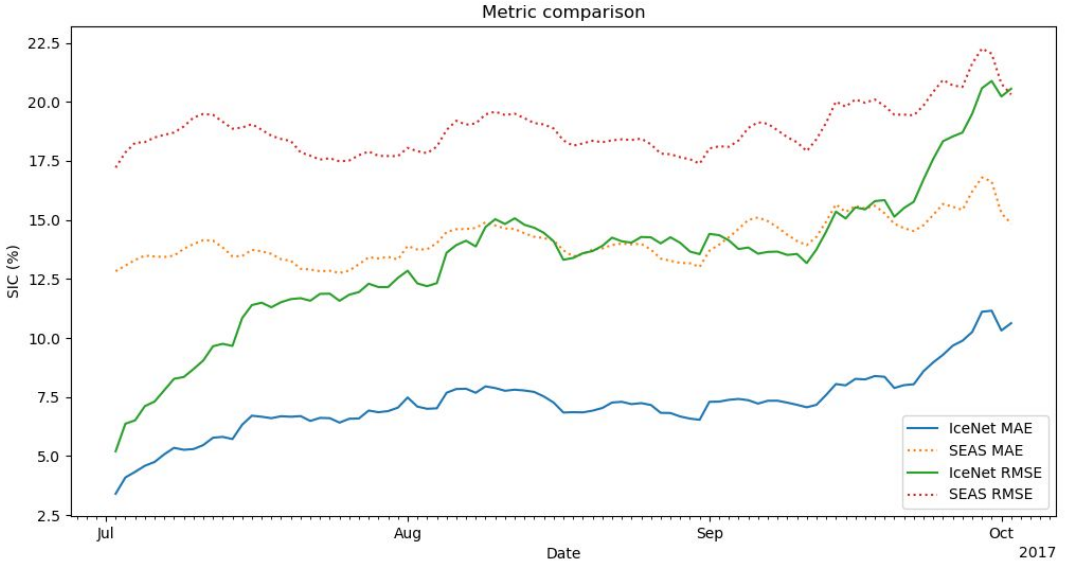




Summary metrics based on pure sea ice concentration

Using `plot_metrics` function (or `icenet_plot_metrics` CLI)

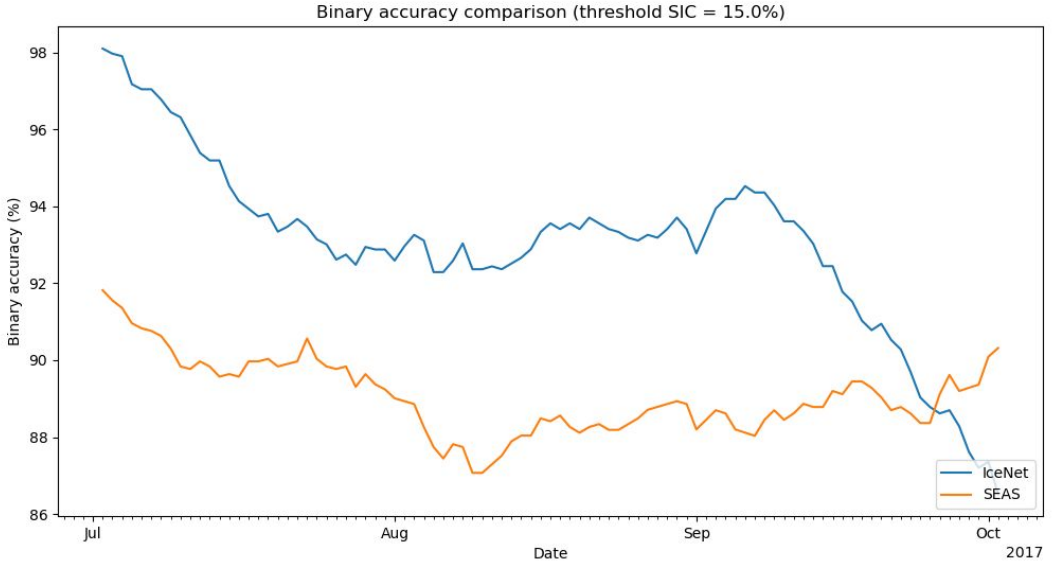
Allows us to compute various metrics (***MAE, MSE, RMSE***) to summarise the overall error of a forecast (i.e. aggregate the metric at each grid location)



Binary accuracy metric

Using `plot_binary_accuracy` function (or `icenet_plot_bin_accuracy` CLI)

Assess the performance of the model on the **binary task of predicting ice** (if *SIC* >= *15%*) and no-ice (if *SIC* < *15%*) for each grid cell

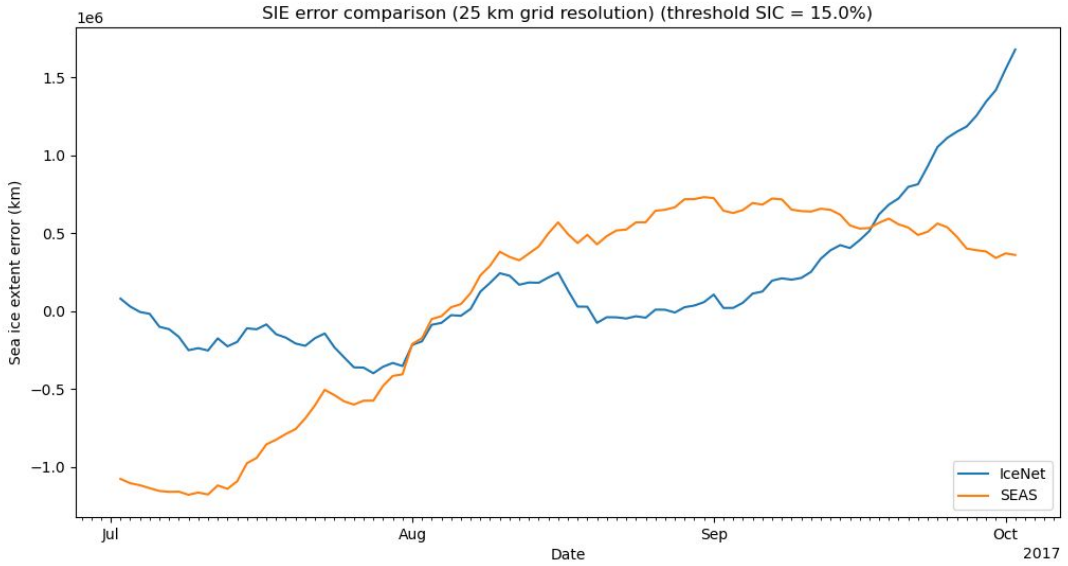


Sea ice extent (SIE) error

Using `plot_sea_ice_extent_error` function (or `icenet_plot_sie_error` CLI)

Sea ice extent (SIE) is the total area of grid cells that has ice (*SIC > 15%*)

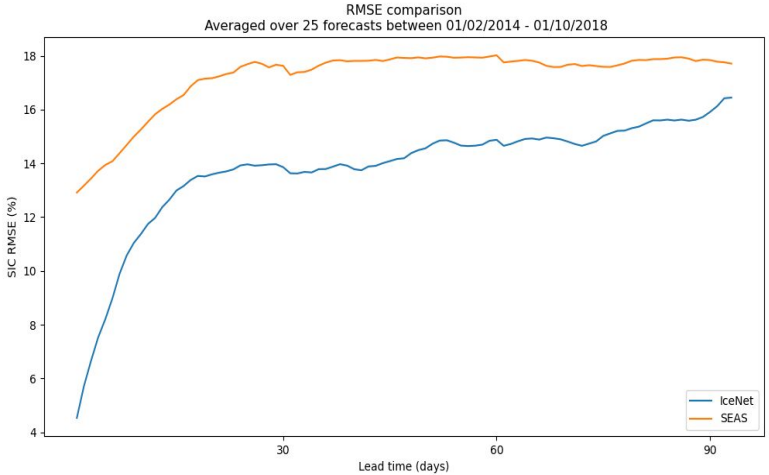
We assess the model by looking at the difference in total area of ice between the forecast and OSISAF

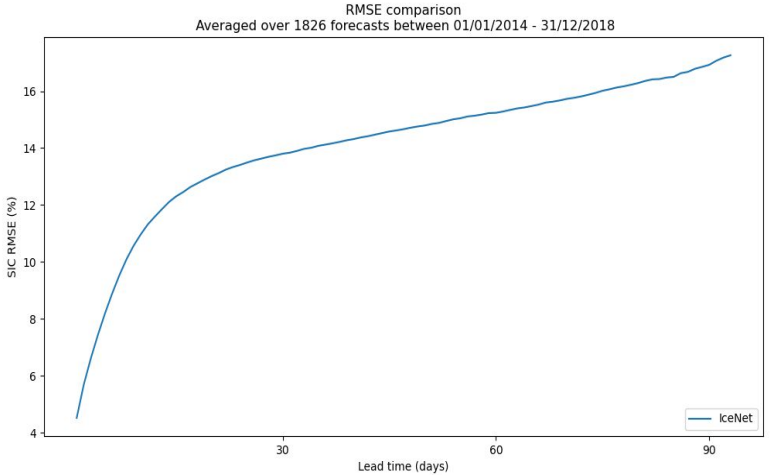


Leadtime averaged plots (averaging over all forecasts)

Using `plot_metrics_leadtime_avg` function (or `icenet_plot_leadtime_avg` CLI)

Rather than looking at the error of a forecast at each leadtime, we might be interested in **average performance of several forecasts** made between some time period





Leadtime averaged plots (averaged over month or day)

Using `plot_metrics_leadtime_avg` function (or `icenet_plot_leadtime_avg` CLI)

Rather than averaging over *all* forecasts, we can also average over month and day to get an idea of seasonal IceNet performance