

# MiSpace 2025

Technical Overview

Teja Koduru

Cindy Yang

December 4, 2025

# 1 Frontend Description

The frontend we developed was meant to be smooth, easy to understand for users, and clean. It's built with React and bundled using Vite for fast development and optimized production builds. The ice forecast visualization is rendered on an interactive HTML5 Canvas, enabling real-time mapping of ice concentration grids. The UI is styled using Tailwind CSS, giving the dashboard a clean, responsive layout with consistent spacing, color, and typography.

A sidebar on the user interface also adds qualitative information about the ice patterns on the lakes, such as expected shipping delays, information on percent increase in ice coverage, and more.

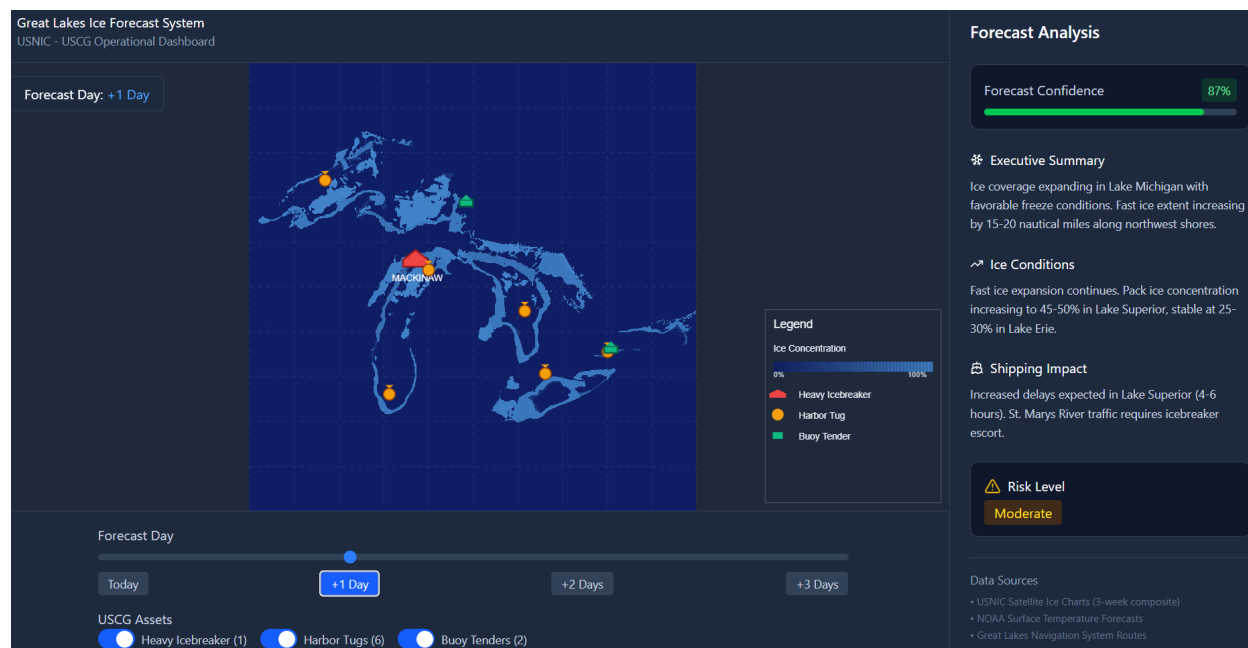


Figure 1: Figure 1. Our frontend

## 2 Backend Image Generation Pipeline

The backend was the most challenging and most experimental part of our project. Much of the development process was driven by iterative trial-and-error and exploration of different model types. What follows is a detailed narrative of everything we tried: things that worked, things that didn't, and what we finally settled on.

### 2.1 First attempt: Classical Deep Learning (U-Net)

Our first idea was to treat the problem like an image-to-image task, which led us to a UNet architecture. We started with a simple encoder-decoder model, using HRRR atmospheric fields as

the inputs to the model, and sea-ice concentration as the target output.

Though the model trained well initially and produced relatively decent outputs, it quickly became clear that the model was overfitting and the outputs didn't seem natural. The data simply didn't span enough of a time period to teach the model about the physics underlying ice formation, which led us to pursue other options.

## 2.2 Option 2: CNN Regression Models

After the U-Net plateaued, we pivoted to smaller CNNs. We thought that a lighter model would resist overfitting and be better at discovering a relationship between our HRRR inputs and ice concentration outputs.

Unfortunately, these smaller models performed even worse. They captured broad climatological patterns, but the outputs tended to blur together. The models tended to predict the 'safe' answer (A giant, semi-ice field), than anything physically meaningful.

## 2.3 Exploring Temporal Models

Sea-ice is inherently dynamic, so we attempted to incorporate time. We tried stacking multiple timesteps as input, using 3D convolutions and sequence-to-sequence CNN architectures.

Though these models performed slightly better, they still could not understand the true underlying physics behind ice formation. The models essentially learned that "yesterday's ice is similar to today's ice," but not why it changes, how it drifts, or how atmospheric forcing interacts with ocean and ice dynamics.

This led us to our final idea, which was to try a physics-based simulation.

## 2.4 Physics Based Modelling

Our team began by exploring publicly available ice formation prediction models, especially the CICE Consortium's Icepack framework.

This direction had enormous potential. Icepack explicitly models:

- sea-ice thermodynamics,
- mechanical deformation,
- rheology,
- melt/freeze processes,
- and the response of ice to atmospheric forcing.

However, in reality, we ran into tons of issues with installing and running Icepack, and also getting our data into a format that Icepack could run with. Though we never achieved a fully

operational Icepack simulation pipeline, the attempt highlighted exactly which physical processes were missing from our earlier approaches.

## 2.5 A custom physics simulation

Given the challenges with pure ML and the full Icepack solver, our final approach was a mix of data and physics. We implemented a simplified, 0-layer, 5-parameter thermodynamic sea-ice model tailored specifically for short-term forecasting on our existing grid.

This model treats the sea-ice pack as a single slab (0-layer) and calculates the change in ice thickness ( $\Delta H$ ) and concentration ( $\Delta A$ ) based on the net energy balance. The governing equation for our simulation is driven by heat flux to the region:

$$\Delta H = \frac{F_{net}}{\rho_i L_f} \Delta t$$

where  $F_{net}$  is the sum of all surface and ocean heat fluxes,  $\rho_i$  is the density of ice,  $L_f$  is the latent heat of fusion, and  $\Delta t$  is the time step.

The model explicitly relies on five key parameters, which represent the major physical uncertainties or simplifications required for robust short-term forecasting:

1.  $T_{frz\ offset}$ : Sea-ice freezing point correction (in °C).
2.  $C_h/C_d$ : Turbulent exchange coefficient ratio.
3.  $F_{bo}$ : Constant ocean heat flux (W/m<sup>2</sup>).
4.  $C_{rain}$ : Rain heat flux scaling factor.
5.  $\alpha_{ice}$ : Ice albedo (reflectivity) value.

## 2.6 Hyperparameter Optimization

The challenge then shifted from deep learning training to **hyperparameter optimization** (HPO). Since training a complex ML model was out of the picture, we decided to build a pipeline to do the following:

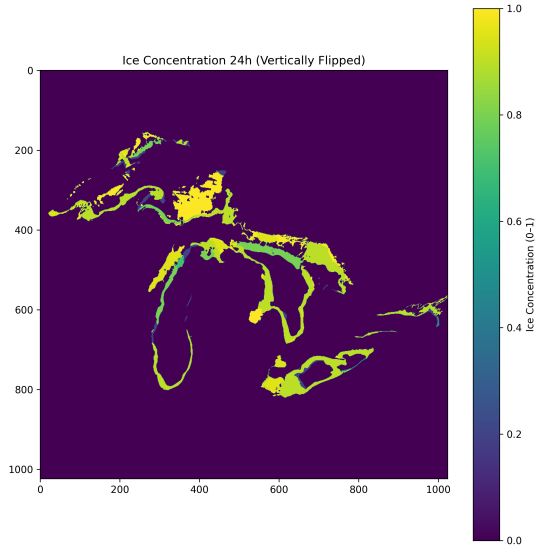
1. **Defines a Search Space:** Established realistic bounds for each of the five physical parameters.
2. **Runs the Simulation:** Executes the 96-hour thermodynamic model using a candidate parameter set on historical forcing data.
3. **Calculates Loss:** Evaluates the model’s output (forecasted ice concentration) against the ground-truth historical ice concentration using a standard loss metric (Mean Squared Error).

4. **\*\*Iterates:\*\*** Used an automated search algorithm (such as a random search or a basic grid search) to explore the parameter space and identify the combination that yielded the lowest error, thus "training" the simplified physics model to best match the historical observations.

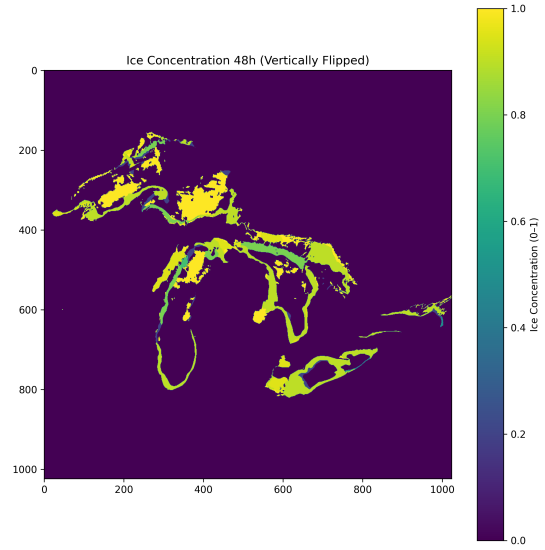
This HPO approach was the key to success. It allowed us to leverage the deterministic, stable nature of the physics simulation while using the historical data to empirically tune the most sensitive variables (like ocean heat flux and albedo), resulting in our final submitted forecast pipeline.

See the final generated images on the next page.

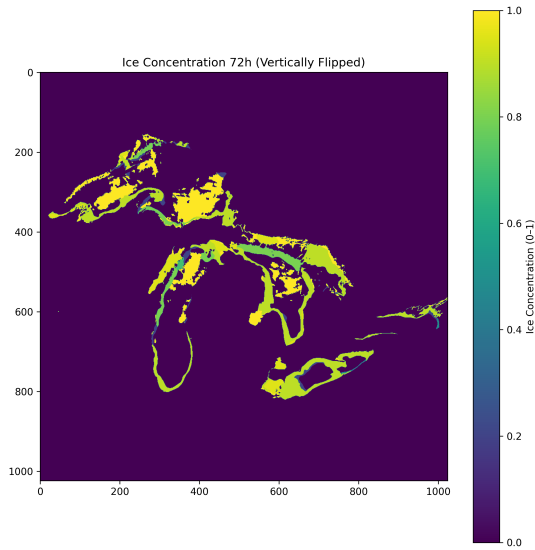
### 3 Final images



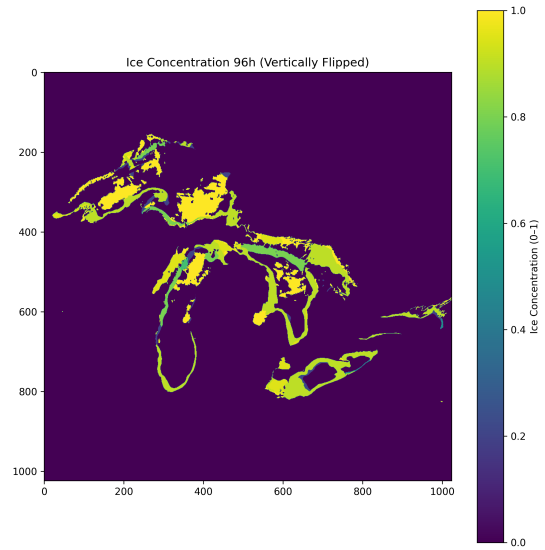
(a) 24-hour Forecast



(b) 48-hour Forecast



(c) 72-hour Forecast



(d) 96-hour Forecast

Figure 2: Ice concentration forecasts at 24h, 48h, 72h, and 96h.