Developing a Deep Learning forecasting system for short term and high resolution prediction of Sea Ice Concentration

Masters' thesis in Computational Science: Geoscience, Spring 2023

Are Frode Kvanum[1,2]

[1]Development Centre for Weather Forecasting, Norwegian Meteorological Institute
[2]Department of Geosciences, University of Oslo

April 25, 2023

Contents

1 Introduction

The Arctic sea ice extent has continuously decreased since the first satellite observations of the Arctic was obtained in 1978 (Serreze and Meier, 2019), with an average decrease of 4% per decade (Cavalieri and Parkinson, 2012). The summer months are experiencing the greatest loss of sea ice extent (Comiso et al., 2017), with models from the Coupled Model Intercomparison Project Phase 6 (CMIP6) projecting the first sea ice-free Arctic summer before 2050 (Notz and Community, 2020). As a consequence of the sea ice retreat during the summer months, previously inaccessible oceanic areas have opened up causing an increase in maritime operations in the Arctic waters (Ho, 2010; Eguíluz et al., 2016). The expected influx of operators to the Arctic regions due to the prolonged open water season call for user-centric sea ice products on different spatial scales and resolutions to ensure maritime safety in the region (Wagner et al., 2020; Veland et al., 2021).

Current information on Arctic sea ice concentration can be discerned into several types of products with different spatial and temporal resolutions. Sea ice products designed for climate applications such as OSI-450, SICCI-25km and SICCI-50km provide daily sea ice concentration by merging observations from multiple sensors to create a historical dataset. The purpose of a climatology is to provide accurate reference data (Lavergne et al., 2019a) which can be used for e.g. forecast validation or anomaly detection. Satellite observations are also supplied as daily products, with a timeliness of a few hours on the same day and posing higher spatial resolutions than climatologies. For example, OSI-401-b (Tonboe et al., 2017) and OSI-408 (Lavelle et al., 2016) provide single sensor daily averaged sea ice concentration covering the northern and southern hemisphere, and can be used to force numerical weather prediction systems which only resolve the atmosphere (Müller et al., 2017).

Sea ice models are physically based models resolving the growth and movement of sea ice forward in time. Standalone models such as CiCE (Hunke and Dukowicz, 1997) and neXtSIM (Williams et al., 2021) can be used independently or coupled with ocean models (Röhrs et al., 2022) to create sea ice forecasting systems for short lead times. Finally, sea ice charts drawn analogously by a sea ice specialist merge recent sea ice observations from different sensors and satellites into a single daily product. The Ice Service of the Norwegian Meteorological Institute (NIS) provides regional ice charts covering the European Arctic. The product consists of polygons which are drawn to match the current resolution of the available observations, which range from 50m to several kilometers, and are assumed to have a low uncertainty due to the quality control exerted by the sea ice specialist (Dinessen et al., 2020).

The previously mentioned sea ice products serve different use cases, and it is possible to infer a correlation between the spatial and temporal resolution of a product and its application scenario for maritime end users. While lower resolution products at larger

temporal time scales can be used in long term planning, regional high resolution products delivered at a high frequency can assist strategic decision making and short term route planning (Wagner et al., 2020). However, it is currently reported by end users that available operational passive microwave satellite products are of a too low resolution, partly due to their insufficient ability to resolve leads and other high-resolution information necessary for maritime safety. Moreover, it is also reported that sea ice forecasting systems lack desired verification, are inadequate for operational use as well as being difficult to integrate with a vessel where computational resources and data-bandwidth are limited (Veland et al., 2021). Though sea ice charts provide maritime end users in the Arctic with information regarding where sea ice has been observed in the time after the previous ice chart has been published, the ice charts does not provide a description on the future outlook. Thus, the responsibility of interpreting the ice charts and other available sea ice information with a outlook on future development is delegated to the end-user and relies on their experience to ensure a continued safe navigation (Veland et al., 2021).

As such, a different approach to short-range sea ice forecasting may be necessary to deliver short-term sea ice information on a spatial scale that is relevant for end-users. Thus, this thesis proposes an alternative forecasting scheme that applies Convolutional deep learning in the form of a modified U-Net architecture (Ronneberger et al., 2015) to deliver a short lead time (1 - 3 days), 1km resolution forecasting product over a subsection of the European Arctic by utilizing the aforementioned Ice Charts as the ground truth. Moreover, the product is verified with regards to the position of the ice edge, which aims to demonstrate the operational relevance of the product (Veland et al., 2021; Melsom et al., 2019).

There have been made previous attempts to develop deep learning sea ice forecasting systems. Andersson et al. (2021) propose IceNet, a pan-arctic covering U-NET which predicts monthly averaged sea ice concentration (SIC) with 6 month lead time at a 25 km spatial resolution (Andersson et al., 2021). The model classifies sea ice concentration into one of the three classes open-water, marginal ice or full ice. IceNet showed an overall improvement over the numerical SEAS5 seasonal forecasting system (Johnson et al., 2019) for 2 months lead time and more, with the greatest improvement seen in the late summer months. The model is trained on SIC data provided by the European Organization for the Exploitation of Meteorological Satellites (EUMETSAT) Ocean and Sea Ice Satellite Application Facilities (OSI SAF) dataset (Lavergne et al., 2019a), as well as other climate variables obtained from the ERA5 reanalysis (Hersbach et al., 2020). Their model was validated against SEAS5, which is a seasonal forecasting system developed by the European Center for Medium-Range Weather Forecasts (ECMWF) (Johnson et al., 2019).

Similarly, Liu et al. (2021) propose a Convolutional long short-term memory network (ConvLSTM) which forecasts SIC with a lead time up to 6 weeks. The model uses climate variables and SIC from two reanalysis products ERA-Interim (Dee et al., 2011) and

ORAS4 (Balmaseda et al., 2013), covering the Barents Sea with a domain size of 24 (latitude) x 56 (longitude). Their results showed skill in beating numerical models as well as persistence.

Models such as those noted above consider input variables obtained from climatologies, and represent SIC on spatial scales far larger than what is needed for an operational short-term sea ice forecast. The possibility of using higher resolution input data was explored by Fritzner et al. (2020), which combined OSISAF SIC, sea surface temperature from the Multi-scale Ultra-high Resolution product, 2 meter air temperature from the ERA5 reanalysis as well as SIC from sea ice charts produced by the NIS. Fritzner et.al. developed a Fully Convolutional Network (FCN), which achieved similar performance to the Metroms coupled ocean and sea ice model version 0.3 (Kristensen et al., 2017). However, due to computational constraints of training the FCN, the subdomain was reduced to a resolution of 224 x 224 pixels which translates to 10 - 20km (Fritzner et al., 2020). Thus, the product has a limited accuracy for short term operational usage, similar to (Andersson et al., 2021) and (Liu et al., 2021).

Contrary to the authors above, Grigoryev et al. (2022) propose a 10 day lead time regional forecasting system with a 5km spatial resolution trained on a sequential (traditional) and recurrent U-Net architecture. The authors used 5km AMSR-2 sea ice concentration as the ground truth variable, and regrid atmospheric variables from the NCEP Global Forecast System (https://www.emc.ncep.noaa.gov/emc/pages/numerical_forecast_systems/gfs.php) to match the resolution of the ground truth. Their results showed that the recurrent setup slightly outperformed the sequential architecture for predictions with a lead time up to 3 days, with both architectures significantly outperforming persistence and the linear trend. However, the sequential architecture tended to outperform the recurrent architecture for 10 day forecasts, as the recurrent model was trained without weather data as it only had a lead time of 3 days.

As mentioned in (Andersson et al., 2021; Fritzner et al., 2020), the computational cost of producing a forecast using a pre-trained model is low, such that a laptop running consumer hardware is able to generate a forecast in seconds or minutes depending on the availability of a Graphics Processing Units (GPU). This is in stark contrast to numerical sea ice models, which could run for several hours on high-performance systems (Andersson et al., 2021). Training a model is a one time expense, and can be efficiently performed on a GPU. With the increased complexity, efficiency and availability of high end computing power, smart usage of the available memory allows for model training using high resolution fields. Current GPUs have seen a significant increase in the available video memory, which allows for higher resolution data to be utilized during training. This work will exploit the recent advances in GPU development, as well as incorporating techniques to reduce the floating point precision of the input meteorological variables, circumventing a reduction of the spatial resolution as seen in previous works.

Moreover, the U-Net architecture is part of the supervised learning paradigm of machine learning, which require labelled samples in order to train the network (Ronneberger et al., 2015). Furthermore, U-Nets perform pixel-level prediction where each pixel is classified according to a category. This work will utilize the image-to-image predictive capabilities of the U-Net to create a semantic segmentation based on its input variables simulating a forward in time propagation of the sea ice concentration akin to a physical model. This allows for the inspection of how changes to the architecture as well as input data configurations affect the behavior of the forecasting system.

In the present work, the development of a deep learning forecasting system will be explored. The choice and tuning of hyperparameters will be reasoned in light of the physical processes affecting sea ice and the surrounding variables. Furthermore, the quality of the machine learning forecasting system will be assessed against relevant benchmarks such as persistence, physical models and linear regression of the observed sea ice concentration. Due to the operational nature of the developed forecasting product, ice edge aware validation metrics such as the Integrated Ice Edge Error (Goessling et al., 2016) will be central to the performance analysis. Furthermore, this thesis aims at providing the framework for which a future operational sea ice prediction system can be built upon. As such, the choice and structure of data will be made with a potential operational transition in mind.

This thesis aims at exploring the following research questions:

- Can a deep learning system resolve regional sea ice concentration for high resolution, short lead time forecasts?
- How does a high resolution, short lead time U-Net forecasting system resolve the translation and accumulation of sea ice compared to a physical based model
- In what sense can a deep learning model be explainable / made transparent to explain the statistical reasoning behind the physical decision-making

The thesis is structured as follows. The first section will describe the datasets used, followed by the second section which will do a rundown of the methodological framework necessary to develop the U-Net as well as validation metrics used to assess forecast skill. The third section will detail the development process behind the U-Net, with the fourth section exploring the physical connections of the model. The fifth section will detail the performance assessment of the forecasts. In the sixth section, a discussion of the findings will be conducted, with the seventh and final section presenting conclusions and future outlook.

2 Datasets

To facilitate the development and verification of a high resolution short-term deep learning sea ice forecasting system, several datasets from observations and physical model forecasting systems have been chosen. When selecting appropriate datasets, their spatial resolution as well as release frequency has been considered. Even though several observational sea ice concentration products which cover the region of interest exists, a lot of the satellite products based on passive microwave retrievals are of a too coarse resolution (e.g. Lavergne et al. (2019a) or Kern et al. (2019)) to be able to aid in short term decision making (Wagner et al., 2020). Moreover Synthetic Aperture Radar (SAR) observations such as Sentinel 1A Interferometric Wide swath ($5\text{m} \times 20\text{m}$) or Extra-Wide swath ($20\text{m} \times 40\text{m}$) are on a sea ice structure resolving spatial resolution. However the daily SAR coverage is sparse in the Arctic (See Supporting Figure 55) and there are currently no sea ice concentration product based on retrieval algorithms of SAR observations which are known to the author.

Moreover, forecasts can be used as predictors for the deep learning system since they provide information regarding how the conditions should evolve in the period after the forecast has been initialized. Thus giving the deep learning system insight into the future state of the domain while still facilitating operational usage by not relying on e.g. future observations. Hence, atmospheric variables from a regional numerical weather prediction system will be included as input to the model. These variables (wind and temperature) have been chosen due to their physical impact on sea ice, and is assumed to encode information about the future state of sea ice concentration when seen in combination with past and present sea ice concentration by the deep learning system.

Finally, the highest resolution product with an appropriate temporal frequency available are the sea ice charts produced by the NIS (Dinessen et al., 2020). Moreover, the sea ice charts represents an interpretation of different sea ice observations delivered as a product directed towards operational users. Thus, the sea ice charts will serve as the ground truth for the model. Furthermore, as a deep learning system can increase its skill by combining correlated variables as input, this thesis will explore the impact caused by including several datasets covering both current observations, past trends as well as forecasted variables on different spatial resolutions as input predictors.

The following section will describe the domain covered for this thesis, followed by a rundown of the satellite products as well as physical models used. Table 1 presents the different products used for this thesis, and whether the product is used to train or verify the model.

Table 1: List of the products used, their applications as well as temporal regime. The dashed line separates observational products (above) from forecast products (below)

Product	Variables	Training	Verification	Time interval
Ice charts	SIC	Yes	Yes	Present / Future
OSI-SAF SSMIS	SIC trend	Yes	Yes	Past
OSI-SAF CDR	Ice edge length	No	Yes	Present
AMSR2	SIC	No	Yes	Future
AROME-Arctic	T2M, X-wind, Y-wind	Yes	No	Future
NeXtSIM	SIC	No	Yes	Future
Barents-2.5	SIC	No	Yes	Future

2.1 Region of interest

The domain covered by the deep learning system, covers part of the European Arctic, with Svalbard off-centered. The coast of Northern-Norway is located on the Southern Border, the archipelago of Novaya Zemlya on the eastern border with Franz Josef Land located to its north. The northern border reaches $(88°\text{N}, 79°\text{E})$. The region is an intersection between the domain covered by the Ice Charts (Dinessen et al., 2020) and AROME Arctic (Müller et al., 2017) as shown in figure (1). The domain has a 1km spatial resolution, and contains 1792×1792 equidistant grid points. Compared to the AROME Arctic grid, the model domain has a reduced southern and eastern extent. The study area is a commercially active region, with regards to fishing, tourism and shipping (Wagner et al., 2020). Moreover, the sea ice area and extent in the domain have a strong seasonal variability (Cavalieri and Parkinson, 2012).

The mean annual sea ice drift pattern governing the region of interest is in general positioned away from the arctic basin, with some local variations (Barry et al., 1993). In the area located towards the north of the domain, the sea ice drift is mainly driven by the Transpolar Drift Stream ocean current which transport the sea ice away from the Laptev Sea (76°N, 125°E) and towards the Fram Strait located between Greenland and the Svalbard archipelago (Colony and Thorndike, 1984). Moreover, the sea ice drift pattern in the Fram Strait is characterized by the strongest gradients for the entire Arctic basin, and is positioned parallel to the coast of greenland in a southwestward direction (Barry et al., 1993). Finally, a secondary drift pattern is observed originating in the Kara sea (77°N, 77°E) during the winter, where sea ice is drifting towards the western coast of Svalbard between Franz Josef Land and Novaya Zemlya (Kaur et al., 2018).

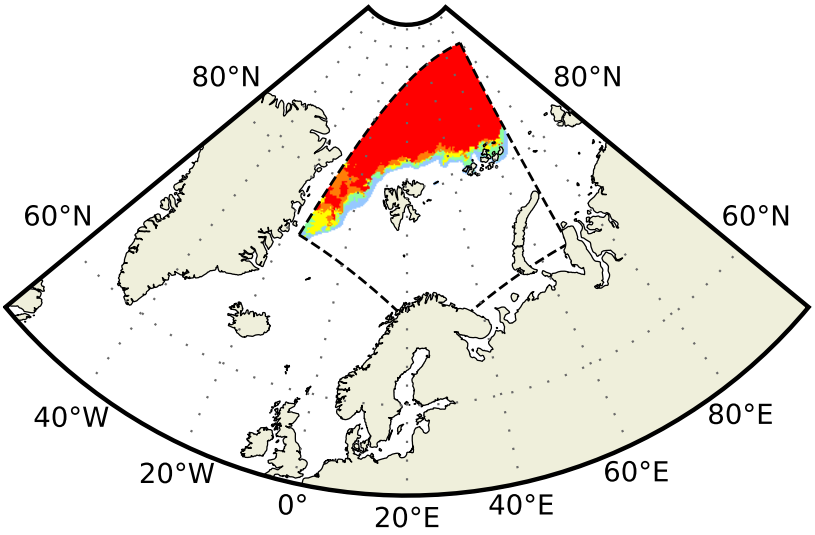


Figure 1: The model domain is shown by sea ice concentration contours retrieved from a sea ice chart (15 Sep 2022). No colorbar is shown, white is free open water and red is very close drift ice.

2.2 Observations

Observations are used to convey the current state of sea ice concentration. There is a lack of consistent in situ observations of sea ice concentration, due to the remoteness of the region. Thus, most independent observations are ship-based concentration estimates (Kern et al., 2019) or optical remote sensing, the latter is only available during summer. As a result, sea ice concentration is mainly observed automatically through passive microwave retrievals utilizing different sea ice retrieval algorithms (Lavergne et al., 2019b; Comiso et al., 1997; Spreen et al., 2008). Another source of sea ice observations are sea ice charts (https://usicecenter.gov/, Last Accessed 25 Jan 2023) (Dinessen et al., 2020), which are manually drawn interpretations combining available sea ice concentration observations such as SAR, passive microwave and optical imagery.

2.2.1 Sea Ice Charts

The sea ice charts utilized for this thesis are provided by the Norwegian Meteorological Institute through the Norwegian Ice Service. The product is manually drawn by a sea ice specialist, and is distributed every workday at 15:00 UTC. The Sea Ice specialist

assesses available SAR scenes from Sentinel 1 and Radarsat 2. However, due to the spatial variability in daily SAR coverage (See Supporting Figure (55)), visual, infrared and low resolution passive microwave observations are used in supplement to achieve a consistent spatial coverage (Dinessen et al., 2020). All observations used by the sea specialist are mainly gathered at the same date as the sea ice chart is drawn. The sea ice charts are not drawn onto any set resolution, although zoom-level as well as pixels per inch and screen size of the used monitor are factors which determine the drawing resolution at any given time. Hence, a gridded representation of the ice charts is only a representation of the mean value of the polygons contained inside each grid cell. The sea ice charts used in this work has been interpolated onto a 1-km grid with the same projection as AROME Arctic (Müller et al., 2017).

With regards to consistency, it is noted that the current sea ice chart product have no easily identifiable way of noting which observations were used by the sea ice analyst to draw each segment of the chart. As the different satellite products used have different spatial scales, from meters to kilometers (Dinessen et al., 2020), the underlying uncertainty and ability to resolve structures varies both spatially and temporally. The published sea ice charts as seen in Figure (2) shows the available SAR coverage as black contours, which is the preferred data source for the ice analysts (Dinessen et al., 2020).

Figure (3) shows the monthly distribution of sea ice concentration contours from the sea ice charts during the period of 2022. As can be seen from the figure, more than half of the region consists of ice free open water, with the other majority of an ice chart consisting of very close drift ice. Moreover, the figure shows the seasonal variability of the sea ice extent, with the ice free open water contributing between $\sim 38\%$ and $\sim 75\%$ of the entire domain depending on the month. The ice charts also resolve the intermediate sea ice concentration classes, which for the current region is mostly related to the marginal ice zone and the ice edge.

By inspecting Figure (4), it can be seen that the autocorrelation between two ice charts close in time is high. However, it can also be seen to steadily decline as the time lag increases. From the strong autocorrelation seen in Figure (4), it can be assumed that the persistence for short lead times (days) closely relate to the current sea ice concentration. Furthermore, the autocorrelation also renders previous sea ice concentration at short timescales as skillful at describing the current growth of the sea ice. The latter will be used as motivation to compute a sea ice concentration trend in a coming subsection.

Vurdere om dette avsnittet står seg bedre og selvstendig i performance assessment, eller omvendt?

The Sea Ice chart is an operational product mainly targeted for maritime operators. A single sea ice chart is usually drawn by one individual person from the NIS team. However, there are several sea ice specialists at the NIS whom draw ice charts. On the one hand, the operational nature of the product may influence the decision-making when creating the sea ice charts. Moreover, as a consequence of the human interaction with the production

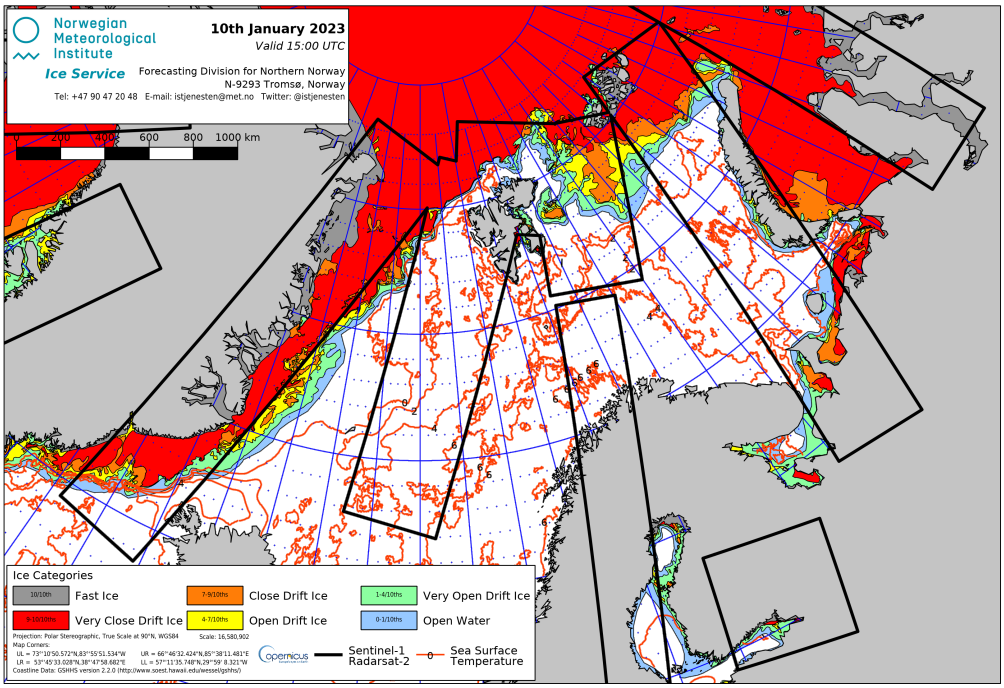


Figure 2: Sea Ice chart produced by the NIS covering 10 Jan 2023 at 15:00 UTC. Sea ice concentration categories are drawn as filled contours. The black lines indicate the available SAR data used to draw the sea ice chart.

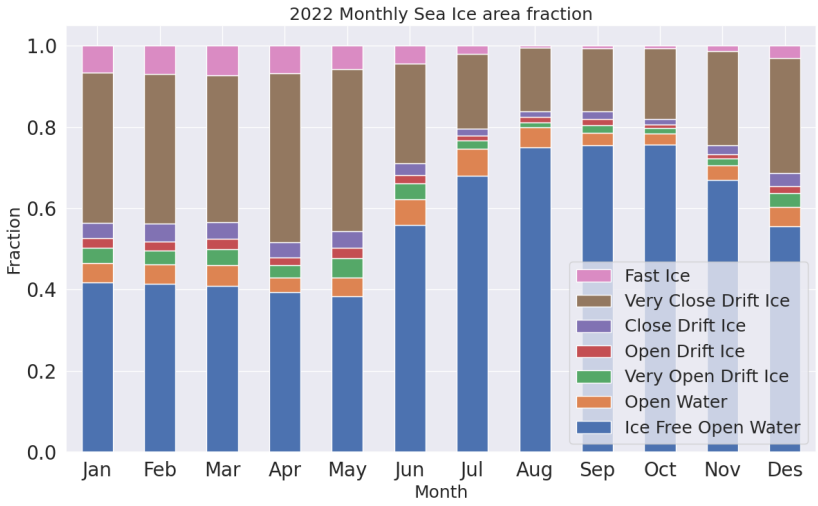


Figure 3: Monthly distribution of each concentration class as respective fraction of the total mean sea ice concentration for the sea ice charts covering 2022. **[Could extend to cover larger time period (e.g. from 2011), give a more climate perspective of the sea ice evolution], Add concentration ranges for each class**

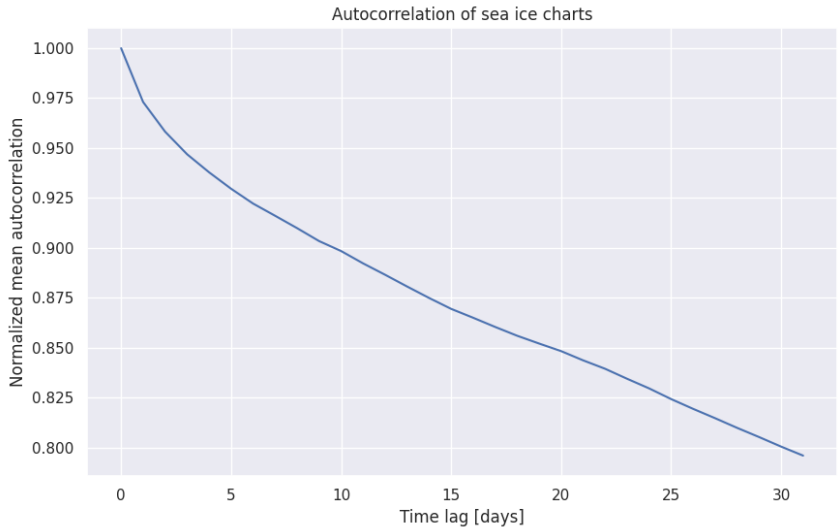


Figure 4: Autocorrelation of the sea ice charts from 2022. The x-axis is the time lag between to entries 2022 sea ice chart timeseries. The y-axis is the normalized autocorrelation, i.e. autocorrelation at a certain time lag divided by the autocorrelation at time lag 0. The autocorrelation is computed for a period covering 31 days.

of the sea ice charts it can be assumed that there is an unknown degree of personal bias added to the data. However, a recent exercise conducted by the Danish Meteorological Institute (DMI) compared the sea ice charts from five DMI sea ice specialists covering a SAR scene of the Greenland sea and showed that the inter specialist spread was ($\sim$ 9 - $\sim$ 34) pixels on the computer screen (Kreiner et al., 2023). On the other hand, the human involvement may also introduce a degree of quality control not seen in automatic sea ice concentration retrieval algorithms. Thus, the sea ice charts are assumed to have a low uncertainty, though there are no uncertainty estimates included (Dinessen et al., 2020).

In spite of the uncertainties outlined above, the sea ice charts are assumed to be the most accurate sea ice concentration product available for the purpose of high resolution data tailored towards operational end users. By utilizing the sea ice charts as the ground truth data when training the deep learning system, the developed model will fit towards the proposed high resolution operational use case.

2.2.2 OSI SAF

Two different sea-ice Concentration products are used from OSI SAF. OSI SAF Special Sensor Microwave Imager and Sounder (SSMIS) is an operational product delivering daily sea ice concentration on the northern (and southern) hemisphere. OSI SAF Climate Data Record (CDR) (Sørensen et al., 2021) delivers sea ice concentration beginning in 1979 (Lavergne et al., 2019a). The operational product will be used as a predictor for the model and for validation, whereas the CDR will be used only for validation purposes.

2.2.3 OSI SAF SSMIS

OSI SAF SSMIS is a passive microwave product derived from the (SSMIS) instrument. To convert brightness temperature to estimated sea ice concentration, a hybrid approach combining the Bootstrap algorithm (Comiso et al., 1997) and the Bristol algorithm (Smith, 1996) where the prior is used over open water and the latter used for ice concentrations above 40% (Tonboe et al., 2017). The algorithm uses data from the 19GHz frequency channel (Vertically polarized) and 37GHz channel (Vertically and Horizontally polarized), which are the two lowest spectral resolution channels for the SSMIS Tonboe et al. (2017). Finally, atmospheric corrections are made using analyses from the European Center for Medium Range Weather Forecasts (ECMWF). The end product is delivered every day on a 10km polar stereographic grid.

With regards to uncertainty, OSI SAF SSMIS is validated against pan-arctic sea ice charts from the U.S. National Ice Center as well as regional sea ice charts covering the Svalbard

region from the NIS. Moreover, the operational product is required to have a bias and standard deviation less then 10% ice concentration on an annual basis, when compared to the targets (https://osisaf-hl.met.no/sea-ice-conc-edge-validation, Last Accessed 24 Jan 2023) (Lavelle et al., 2017). This strengthens the assumption made at the end of Section (2.2.1) regarding the accuracy of the sea ice charts and their validity in terms of serving as an independent source for reference.

The operational OSI SAF SSMIS dataset is used to compute a coarse resolution (with respect to the ice charts) linear sea ice concentration trend in each grid cell, with a short term length covering a given amount of days backwards in time. The idea behind the computed trend is to encode multiple time-steps of sea ice concentration fields into a single 2d-array, in line with the lack of temporal awareness of the U-Net architecture. Moreover, the trend serve to limit the size of the training data, since the memory needed is equal to that of a single 2d-array regardless of the length of the trend. Furthermore, the ice concentration trend is computed from a separate sea ice product than the ice chart, with the intent to supply the model with correlated but not overlapping information, as the current day ice chart is already used as a predictor. However, it should also be noted that the lack of sea ice charts during the weekends (Dinessen et al., 2020) is also a contributing factor. As a sea ice concentration trend derived from Dinessen et al. (2020) would be limited to at most five days, which is not the case for OSI SAF SSMIS as there are no temporal gaps in the dataset. The coarser resolution also contributes to the OSI SAF trend serving as complementary information to the ice charts, as the coarse resolution makes the trend less resolvent of the local variability which is seen in the ice charts. As such, the trend serves as a indicator of where the sea ice growth / decline is occurring.

The temporal length used when deriving the trend will have an impact on how the trend reflects the current growth and decline zones, especially with regards to the volatile position of the ice edge on a daily timescale but also due to the seasonal variability of the ice area (Holland and Kimura, 2016). Hence, a too large lookbehind would cause a decorrelation between the current sea ice concentration and the computed trend. Nevertheless, Figure (4) shows that there is significant autocorrelation for sea ice concentration on a short time-range, as described previously. However, a trend computed from a sufficiently long temporal window could be assumed to better represent the spatial distribution of seasonal sea ice concentration growth and decline rather than representing the current growth and decline.

2.2.4 OSI SAF Climate Data Record

As briefly mentioned in Section (1), OSI SAF Climate Data record combines observations from different sensors (SMMR, SSM/I, SSMIS) as well as numerical weather prediction

fields from the ERA Interim reanalysis (Dee et al., 2011). The latter are utilized to correct for the atmospheric conditions. Two versions of the dataset has been used, version 2 (OSI-450) which covers (2011 - 2015), and the interim version (OSI-430-b) which cover (2016 - 2020) (https://osisaf-hl.met.no/osi-450-430-b-desc)(Last Accessed 18 Jan 2023). Both products are processed using the same algorithms, ensuring consistency (Lavergne et al., 2019b). The Interim version is serving as an extension of the original scope of OSI-450 (1979 - 2015), with a difference being its use of ECMWF analyses compared to the reanalysis and different SSMIS input data (https://osisaf-hl.met.no/osi-450-430-b-desc, Last Accessed 24 Jan 2023). Regardless, both products will hereby be referred to in tandem as OSI SAF CDR

The OSI SAF retrieval algorithm has been shown to have strong correlation against ship based measurements (Kern et al., 2019) as well as optical satellite observations during the summer (Kern et al., 2020). Hence, OSI SAF CDR is expected to serve as a low error representation of the Arctic sea ice concentration. However it is noted that no retrieval algorithm is able to match the true state of the sea ice concentration.

OSI SAF CDR is provided with a 25km spatial resolution on a Lambert Azimuthal Grid projection (Sørensen et al., 2021). The sea ice concentration data retrieved has been used to compute a climatological ice edge length for each day of the year, applying a daily mean across the time period (2011 - 2020). The ice edge length has been computed according to Melsom et al. (2019), which will be derived in Section (3.5.1). Note that though OSI SAF CDR provides a pan-arctic distribution of sea ice concentration, the data has been regridded onto the study region domain with the AROME Arctic projection and a 25km grid spacing before computing the ice edge length.

As can be seen in Figure (5), the Arctic sea ice edge experiences a strong seasonal variability. The computed climatological ice edge will be used as a normalization factor in order to use verification scores that are not seasonally dependent (Goessling et al., 2016; Zampieri et al., 2019; Palerme et al., 2019). Another benefit from utilizing a single ice edge length is to ensure that different sea ice products are normalized according to a common and independent factor. Furthermore, it will be shown in section 4.3.1 that the Integrated Ice Edge Error (Goessling et al., 2016) (3.5.2) normalized by the ice edge length is correlated with a changing resolution of the ice edge length, proving the validity of normalizing using a common, coarser resolution ice edge length.

2.2.5 AMSR2

The Advanced Microwave Scanning Radiometer 2 (AMSR2) data utilized for this thesis is the sea ice concentration product from the University of Bremen (https://seaice.uni-bremen.de/sea-ice-concentration/amsre-amsr2/)(Last Accessed 18 Jan 2023)

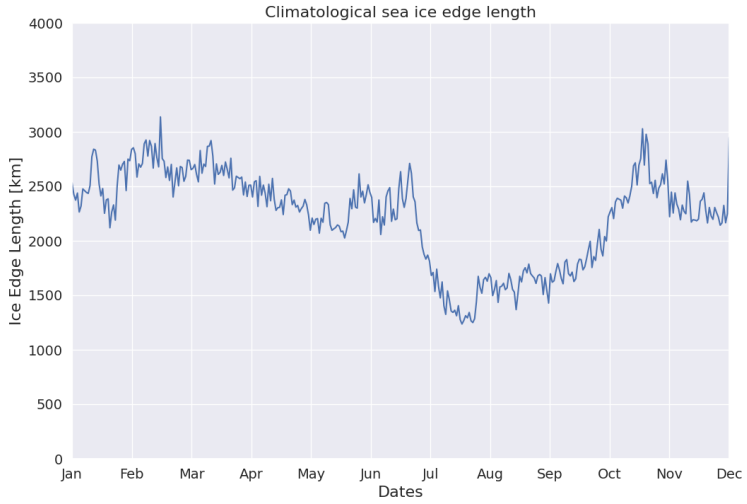


Figure 5: **FIX NONSENSICAL XTICKLABELS** Seasonal variability of the climatological ice edge length computed from satellite observations during the period 2011 - 2020. Only the part of the field projected onto the region of interest has been considered.

(Spreen et al., 2008). AMSR2 is a passive microwave sensor observing the microwaves emitted by the Earth, similar to **OSI SAF SSMIS**. AMSR2 is located on the JAXA GCOM-W1 satellite Melsheimer (2019), and the sea-ice concentration is retrieved using the ASI algorithm Spreen et al. (2008). The algorithm uses data from the 89GHz channel, which is the band with the highest spectral resolution, in both polarizations to determine the sea ice concentration. Bands at lower spectral resolutions are only used as weather filters, which can mask out false sea ice detected in the open ocean Spreen et al. (2008). The resulting data is a pan-arctic sea ice coverage with a spatial resolution of 6.25km.

The current AMSR2 product was chosen as the ASI retrieval algorithm (Spreen et al., 2008) results in a higher spatial resolution product compared to similar AMSR2 products such as the AMSR2 product from OSI SAF (Lavelle et al., 2016), which is delivered on a 10km spatial resolution.

Figure (6) shows the monthly distribution of sea ice contours for the AMSR2 dataset. Similarly to Figure (3), a majority of the scenes are covered by Ice Free Open Water. However, Figure (6) shows the AMSR2 dataset has less very close drift ice, which may stem from an increased fast ice contour. Furthermore, AMSR2 has a less resolved open water contour compared to the sea-ice charts. This may be a result of AMSR2 being a algorithmically derived product, whereas the sea-ice charts are drawn for operational use such that regions of potential sea ice encounters are exaggerated to ensure maritime safety. Lastly, Spreen et al. (2008) demonstrated that the ASI algorithm provide the

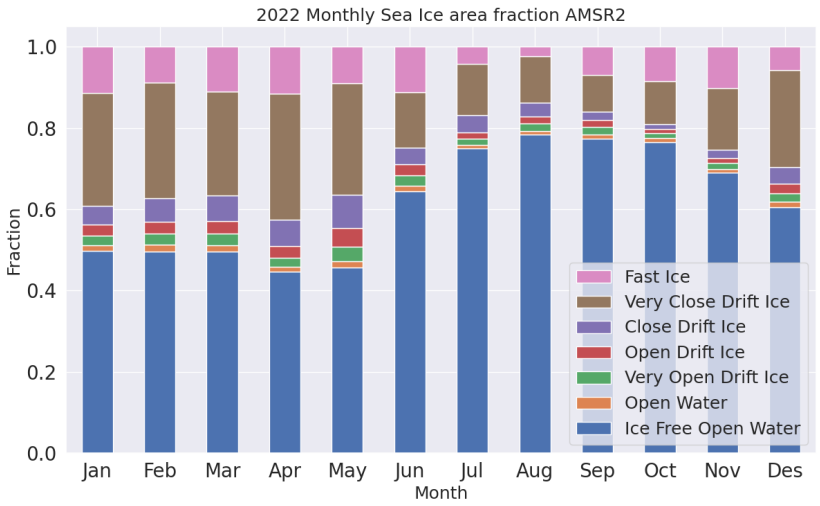


Figure 6: Monthly distribution of each sea ice concentration class fraction for the AMSR2 dataset covering 2022. The data has been projected onto the region of interest.

most certainty at concentrations above 65%, with lower concentrations having higher deviations, mainly due to the error contributed by the atmosphere.

As the sea ice charts are treated as the ground truth during training of the deep learning model, it can be assumed that the model is best at predicting sea ice concentration distributions similar to those found in the training data. As such, the AMSR2 data will serve as an independent dataset with high spatial resolution, and will be used for validation only. Thus, the performance of the deep learning system can be inspected with regards to another dataset that is less similar than the sea ice charts, which measures the generalizability of the model.

2.3 Forecasting systems

2.3.1 AROME Arctic

AROME Arctic is a non-hydrostatic, convection resolving high-resolution weather forecasting system which covers the European Arctic (Müller et al., 2017). The model covers the European Arctic similarly to Figure (1) which is the same domain though reduced, with a spatial resolution of 2.5km and 65 vertical levels. The first full year covered by AROME Arctic predictions was 2016, although the model have produced forecasts operationally since October 2015. AROME Arctic uses different data assimilation techniques for the atmosphere and surface variables. 3DVAR combines the atmospheric model back-

ground with observations and is forced by the deterministic ECMWF forecast, whereas optimal interpolation combines the surface model background with observations. Both parts of the data assimilation system is merged to produced the forecast analysis (Müller et al., 2017). As previously mentioned, variables influencing the sea ice concentration can aid in improving the predictive capabilities of a deep learning system. While observational products described above such as the ice charts (Dinessen et al., 2020) and OSI SAF SSMIS (Tonboe et al., 2017) describe the condition and dynamics of the sea ice concentration. Integrating weather forecast data as part of the model input can be used to describe the interaction between sea ice and atmospheric variables, thus providing relevant variables for predicting sea ice concentration. For the scope of this thesis, 2-meter temperature as well as 10-meter wind in the X and Y component have been selected.

Near surface winds influence the sea ice drift, with the sea ice in the European Arctic displaying a moderate to strong correlation between the sea ice drift speed and the wind speed during winter (Spreen et al., 2011). Moreover, sea ice drift speed is shown to be inversely proportional to the sea ice concentration (Yu et al., 2020). i.e. low concentration sea ice classes tend to have a higher drift speed than high concentration sea ice classes, though both classes display an increased drift speed given an increased near surface wind speed. Thus, including the X and Y component of the near surface wind from AROME Arctic provides the deep learning system with a high resolution proxy for the predicted sea ice drift.

Similarly, surface temperature influences the sea ice mass balance by melting or facilitating sea ice growth (Hibler, 1979), for example through the formation of melt ponds on top of the sea ice. The 2-meter temperature from AROME Arctic is intended to serve as a proxy for the sea ice growth, by including a spatial distribution of temperature to the model. This may be correlated to areas in the model domain experiencing mean positive (melt) or negative (growth) temperatures during the forecast period.

AROME Arctic is shown to have lower RMSE in both 2-meter temperature and 10-meter zonal wind speed than both the deterministic (HRES) and ensemble (ENS) forecast as well as ERA-Interim from ECMWF, for all months when compared to measurements from 89 stations located in Finnmark, Svalbard as well as Jan Mayen and Bjørnøya (Müller et al., 2017). Hence, it is reasonable to assume that extracting the wind and temperature fields from AROME Arctic will provide the most precise information with regards to the strength and spatial location, compared to global medium range numerical weather prediction systems such as the ECMWF Integrated Forecasting System (IFS) Cycle 47r3 (Haiden et al., 2022). However, it is noted that operational numerical weather prediction systems such as those described by Müller et al. (2017) and Haiden et al. (2022) are in constant development, with new improvements added without any retroactive effect for previous data. Firstly, the comparison made in Müller et al. (2017) was with HRES and ENS as of Cycle 38r2 Bauer et al. (2013) is not necessarily represen-

tative of the current state of both products. Secondly, significant advances in model development may cause data before and after the implementation date to be inconsistent, e.g. by introducing a permanent shift in bias for a variable. Problems regarding model updates could be avoided by using variables from a re-forecast or reanalysis product such as CARRA (Køltzow et al., 2022). However, CARRA similarly to other reanalysis products is produced in delayed-mode (see https://climate.copernicus.eu/copernicus-arctic-regional-reanalysis-service, Last Accessed 21 Jan 2023), which would inhibit the operational aspect of the developed deep learning system. It is also noted that CARRA specifically only have a 30 hour lead time, which limits the desired "up to 3 day" lead time desired for the developed deep learning system.

With regards to model development, a major development in AROME Arctic in terms of temperature representation over sea ice ocurred 10 Oct 2018 (AROME Arctic Changelog, Last Access 21 Jan 2023), in the form of a *snow on ice* variable. As this change is expected to have changed the distribution of 2-meter temperature significantly, especially over sea ice covered grid cells (Batrak and Müller, 2019), it has been opted to only consider near surface temperature data from AROME Arctic from 2019 and onwards. This decision is made to avoid having a shift in temperature distribution present in the data, which would exert a negative impact on training the deep learning model.

Though the different datasets in Table (1) have been chosen with the intention to serve as independent products without any intra coupling, it is noted that the sea ice observations used to compute the sea ice concentration trend (Tonboe et al., 2017) is also used to force AROME Arctic with sea ice concentration at the initial timestep (Müller et al., 2017). it is suboptimal to provide input parameters derived from other input parameters, as their correlation may render one of the input parameters obsolete in terms of additional information the deep learning system will infer from the "redundant" predictor. Nonetheless, it is assumed that the impact of the sea ice concentration forcing is low when combined with other surface forcings during the assimilation process. Furthermore, as the sea ice concentration is kept constant at all timesteps (Müller et al., 2017), the correlation between sea ice concentration and atmospheric variables can be assumed to be decaying with time. Thus, both products will be used as input variables, and their overlap is assumed to tend toward zero.

2.3.2 NeXtSIM

The neXt generation Sea Ice Model (neXtSIM) is developed by the Nansen Environmental and Remote Sensing Center and performs the physical simulations for the neXtSIM-F deterministic forecasting platform (Williams et al., 2021). NeXtSIM-F assimilates sea ice concentration from operational OSI SAF sea ice concentration products (Tonboe et al., 2017; Lavelle et al., 2016) and forces the model with oceanic and atmospheric forecasts.

Furthermore, the neXtSIM-F platform is not a coupled system, i.e. the neXtSIM sea ice model is not coupled to either and atmospheric or oceanic model. The version of neXtSIM-F data used for this thesis is supplied on a 3km polar stereographic grid on a pan-arctic domain.

NeXtSIM differentiates itself from comparative physical sea ice models as it does not apply a rheology based on the Viscous-Plastic scheme. The rheology of a sea ice model refers to how the model relates ice deformation and ice thickness with the internal stresses in the ice (Hibler, 1979). Instead, NeXtSIM applies a brittle sea ice rheology, specifically the Brittle Bingham-Maxwell (BBM) rheology which treats the sea ice as a brittle material rather than a viscous fluid (Ólason et al., 2022). With the implementation of a brittle rheology scheme, neXtSIM-F is the first sea ice forecasting system not to use a rheology from the viscous-plastic branch of rheologies (Williams et al., 2021).

With a forecast range of 7 days, data from neXtSIM-F will be used to validate the deep learning system against current high resolution operational sea ice forecasts by serving as a comparable product. NeXtSIM-F is the highest resolution model distributed as part of the Copernicus Marine Environmental Monitoring Service (European Union-Copernicus Marine Service, 2020).

2.3.3 Barents-2.5

Barents-2.5, (hereby Barents) is an operational coupled ocean and sea ice forecasting model under development at MET Norway (Röhrs et al., 2022). The model has been in operation since September 2021. Barents poses the same resolution and projection as AA, i.e. Lambert Conformal Conic with a 2.5km resolution (Röhrs et al., 2022; Müller et al., 2017). Furthermore, Barents also forecasts with a lead time of up to 66 hours, which is the same as AROME Arctic. Since Barents covers the same spatial domain as the deep learning system and forecast with a lead time close to three days, its predicted sea ice concentration will be used for validation purposes.

The sea ice model used in Barents is the Los Alamos sea ice model (CICE) version 5.1, which uses an Elastic Viscous Plastic sea ice Rheology (Hunke et al., 2015). Thus, the CICE model represents sea ice as a viscous fluid which creeps slowly given small stresses and deforms plastically under large stress. It is also noted that the elastic behavior was introduced to benefit the numerical aspects of the model, and can be considered unrealistic from a physical point of view (Hunke and Dukowicz, 1997).

Barents includes an Ensemble Prediction System with 6 members executed for each of the four model runs situated at (00, 06, 12 and 18) (Röhrs et al., 2022). As part of its forcing routine, Barents performs non-homogenous atmospheric forcing of its ensemble members, with one member of each ensemble being forced with AA while the rest of

the members are forced using atmospheric data from ECMWF. As such, the members forced with AA seem to perform best with regards to ocean currents, but the atmospheric forcing's impact on SIC performance is unknown at the time of writing (Johannes Röhrs, 2022, pers. commun.). However, there is generally little spread within one ensemble with regards to sea ice (Röhrs et al., 2022).

The data assimilation scheme applied for Barents is a Deterministic Ensemble Kalman filter, which solves for the analysis with a background error covariance matrix estimated as the variance of the ensemble of background members (Röhrs et al., 2022). Furthermore, it has been expressed by the developers of Barents that the model performance was unsatisfactory up until May / June 2022 due to spin up time of the data assimilation system (Johannes Röhrs, 2022, pers. commun.). As such, forecasts initiated prior to May 2022 will not be assessed for validation purposes due to the expected shift in performance as expressed by the model developers.

Similarly to the neXtSIM-F data in Section (2.3.2), Barents will also be used to validate the deep learning system. However, the forecast range of Barents is only 66 hours, which cuts it short of producing three full daily means. Furthermore, due to the ensemble setup of Barents, it is possible to present a forecast both through the ensemble mean as well as a pseudo deterministic run (single member). However, a forecast from a single Barents member would still be influenced by the other ensemble members during the assimilation stage.

ECMWF IFS is used to force both neXtSIM and Barents with atmospheric variables, whereas the ocean and sea ice model TOPAZ (Sakov et al., 2012) is used to force neXtSIM (Williams et al., 2021) while only nudging the boundaries of Barents (Röhrs et al., 2022). However, their differences in terms of ensemble setup, model coupling, sea ice rheology as well as domain coverage has led to both products being included for validation of the deep learning system. Moreover, both physical products are of a spatial and temporal scale for operational relevancy (Wagner et al., 2020), similar to the deep learning system.

3 Methodological framework

This section will first outline the theoretical background of convolutions from a deep learning point of view, as well as provide a brief overview of image segmentation as a computer vision task. Second, the methodological framework of the U-Net architecture, the deep neural network which is used in the present work, is outlined with a detailed description of its training loop and central algorithms. Thirdly, validation metrics which are used to asses the performance of the developed deep learning system will be described.

Dette avsnittet ble kjelkete, men ønsker å si noe om at det er mer relevant å sammenlikne mot disse to modellene, og ikke

Finally, aspects of explainable ai will be explored in terms of understanding how a deep learning system make a single decision.

3.1 Convolutional layers

Convolutional layers incorporated into a deep neural network which are utilizing the backpropagation algorithm (Rumelhart et al., 1986) was initially proposed by LeCun et al. (1989) to classify handwritten numbers. The layer LeCun et al. (1989) presented consists of an arbitrary amount of filters, which are small two dimensional matrices (e.g. (3×3) pixels) designed to capture a certain structure in the image such as lines or edges . Each filter contains trainable weights, which are learned from the data during backpropagation (LeCun et al., 1989) and gradient descent. When a filter is convolved with all possible local neighborhoods from the input, it outputs a feature map which represent where the input image triggered a response from the filter (Zeiler et al., 2010). Moreover, inputting feature maps to a convolutional layer allows for the filters to respond to combinations of lower level structures, which trains the layer to detect more complicated patterns (Fukushima, 1980). Additionally, stacking convolutional layers in a network-architecture structure increases the field of view for each subsequent layer, which makes each layer observe an increasingly complex pattern of higher order feature maps at increasingly larger spatial scales (Fukushima, 1980). As a result, convolutional layers are able to discern between object and background as they perceive only a limited view of the scene. The convolutional layer is also invariant to the translation of the object, since the filter is constant when creating the feature map, i.e. the filter is detecting the same feature at all locations in the image, known as weight sharing (LeCun et al., 1989).

Sitere boka til Goodfellow2016?

The number of trainable parameters for a convolutional layer is equal to the size of a filter times the number of filters. As a result, the number of trainable parameters is invariant to the spatial extent of the input images. Contrarily, fitting a fully connected layer to spatial gridded data consists of associating a separate trainable parameter to each pixel. As such, the size of a fully connected layer scales with the size of the image, which increases the risk of overfitting the network. In the case of the convolutional layer, LeCun et al. (1989) notes that reducing the number of trainable parameters through weight sharing constrains the solution space such that overfitting is avoided while still having enough trainable parameters to fit the layer to the data. Furthermore, the fully connected layer is not invariant to translation as each trainable parameter is exclusive to their respective pixel, hence no weight sharing. As such, the layer is unable to detect a similar object at a different position, reducing their usefulness for image-based prediction tasks.

Finally, Ciresan et al. (2012b) showed that the processing time of a convolutional layer is significantly shortened by utilizing a graphics processing unit (GPU), due to their large amount of compute cores compared to traditional Central Processing Units (CPUs).

Furthermore, the authors of Krizhevsky et al. (2012) provided the first publicly available implementation of a CNN running on a GPU by utilizing the Nvidia Compute Unified Device Architecture (CUDA) api. Krizhevsky et al. (2012) also demonstrated that their results are tied to the performance of the GPU in terms of available memory as well as the rate of floating point operations per second, with the implication that a better GPU as well as larger datasets would improve their results. As such, both larger convolutional layers as well as deeper convolutional architectures written in a machine learning library which interacts with the GPU through CUDA (Jia et al., 2014; Abadi et al., 2015; Paszke et al., 2019) can be initiated. This is due to the speed up induced by the GPU, which allows for larger datasets to be processed, thus satisfying the increased parameter-count of the architecture.

since they can process greater datasets consisting of larger samples due to their GPU implementation.

The convolutional layer can be described mathematically by utilizing the previously described principle of allowing the filter to only perceive a local neighborhood of the input. Consider the value of a single point $y_{i,j} \subset Y \in \mathbb{R}^2$ where i, j denote the position in the x and y direction as a single output from a convolution. Let $X \in \mathbb{R}^3$ be an input image of size $(A \times B \times D)$ consisting of a single channel, and $W \in \mathbb{R}^3$ be a symmetric filter of size $(r \times r \times D)$. Then, the value at a single point $y_{i,j}$ is given as follows,

$$y_{i,j} = \sum_{a=1-\frac{r}{2}}^{\frac{r}{2}} \sum_{b=1-\frac{r}{2}}^{\frac{r}{2}} \sum_{d=1}^{D} W_{a+\frac{r}{2},b+\frac{r}{2},d} X_{i+a,j+b,d} \tag{1}$$

Where the subscript notation is used in W and X to denote indexes similar to Y. Equation (1) is described graphically in figure (7)

Repeating equation (1) across all points $x \subset X$ by applying a sliding window technique returns the convolution of X with filter W, which results in an output Y with size $(A-r+1) \times (B-r+1)$. Note that the above definition only applies for $X_{1 \leq i+a \leq A, 1 \leq j+b \leq B \cdot}$ The size of the output can be adjusted by padding the input X by a size P in each direction or increasing the stride S of the sliding window, which reformulates the output size of Y in a single dimension as a function

$$Y_{\text{dim}} = \lfloor \frac{A - r + 2P}{S} + 1 \rfloor \tag{2}$$

The convolutional layer adds the convolution described in equation (1) with a bias term $B \in \mathbb{R}^2$ of the same spatial shape as Y, as well as applying an activation function g to

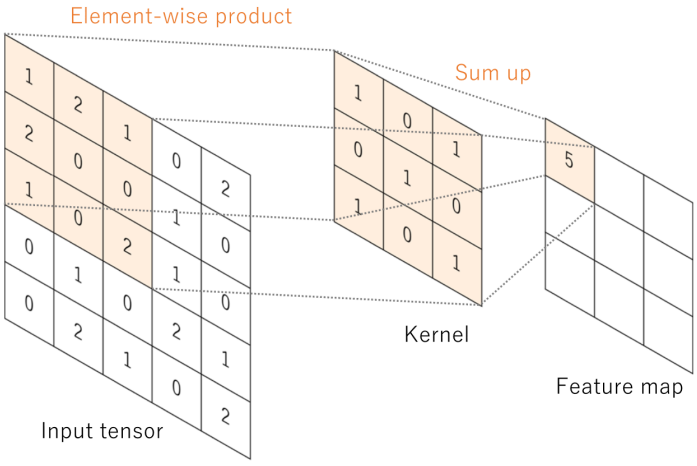


Figure 7: Convolution applied at a single point given a two dimensional input. Figure adapted from (Yamashita et al., 2018).

each $y_{i,j}$ which introduces nonlinearity. In summary, the output of a convolutional layer can be described as

$$Y' = g(Y + B) = g(W^T X + B) \tag{3}$$

If the number of filters increases from 1 to N, equation (3) is repeated for all filters, resulting in an output $Y \in \mathbb{R}^3$ of size $(Y_{\text{dim1}}, Y_{\text{dim2}}, N)$.

Finally, the receptive field is refers to the number of neurons seen by a neuron deeper in the network, and can be seen in figure (7) where the feature map pixel sees nine input tensor pixels. Following the derivations described in Araujo et al. (2019), the receptive field at a layer is mathematically defined as

$$r_0 = \sum_{l=1}^{L} \left((k_l - 1) \prod_{i=1}^{l-1} s_i \right) + 1 \tag{4}$$

where r_0 is the receptive field at layer L, k_l is the kernel size at layer l and s_i is the stride at layer i. Note that the stride in the last convolutional layer does not influence on the receptive field. The receptive field defined in equation 4 may be regarded as the theoretical upper bound, with recent results such as Luo et al. (2017) showing that the effective receptive field attains a Gaussian shape with a peak at the center of the receptive field. Hence the effective receptive field is smaller than the theoretical maximum, and assumes that pixels closer to the center of the receptive field are more important.

3.2 Image segmentation

Image segmentation is a computer vision task where pixels are assigned labels according to some predetermined rules. It is common to define an image segmentation task either as a study of countable *things* (Instance segmentation), or recognizing similarly textured *stuff* (Semantic segmentation) (Kirillov et al., 2018). The task for this thesis, which is labeling sea ice concentration according to its predicted concentration class, falls into the latter category following the definition of *stuff* in Adelson (2001). I.e. the current task is to assign each pixel in a predicted scene a single class label.

Network architectures based on the Convolutional Neural Network (CNN) (LeCun et al., 1989; Ciresan et al., 2012b; Krizhevsky et al., 2012; Simonyan and Zisserman, 2014; Szegedy et al., 2014; He et al., 2015b; Huang et al., 2016) can be used to perform pixelwise semantic segmentation, however the CNN architectures listed have been developed for image classification i.e. predicting a single label for the entire image. Ciresan et al. (2012a) presented an approach where a CNN (see the architecture of Ciresan et al. (2012b)) was used to predict a label for all pixels in an image. Instead of processing the entire image at once, Ciresan et al. (2012a) applied a sliding window technique which predicted each pixel by using their surrounding neighborhood as input. However, due to only processing parts of the image at once, the segmentation algorithm in Ciresan et al. (2012a) is computationally expensive as the CNN must be run for all possible neighborhoods. Additionally, the context for each CNN is limited to the local neighborhood surrounding the pixel (Ronneberger et al., 2015).

To capture the global context of a scene, network architectures such as Long et al. (2015); Noh et al. (2015); Ronneberger et al. (2015); Badrinarayanan et al. (2017); Chen et al. (2018) implement the Encoder-Decoder architecture, where the entire input scene is first processed by a CNN-like architecture referred to as the Encoder to produce a signal. The signal is then used as input to a subsequent network which reconstructs the encoded signal to match the resolution of the original image through upsampling. Long et al. (2015); Ronneberger et al. (2015); Badrinarayanan et al. (2017) all apply the Deconvolution architecture proposed by Zeiler et al. (2010) to upsample the encoded signal through the use of a trainable deconvolutional layer, which will be described in greater detail in Subsection (3.3.3). However, other upsampling techniques exists, such as unpooling used in Noh et al. (2015) which performs a upsampling by performing the opposite operation of a maxpool layer (maxpooling is described in Subsection (3.3.2)).

This thesis will utilize the U-Net architecture proposed by Ronneberger et al. (2015). The U-Net was initially developed for medical image segmentation, however the architecture has shown promising results for both pan-arctic seasonal (Andersson et al., 2021) and regional short term (Grigoryev et al., 2022) sea ice concentration forecasting amongst other applications. Another aspect which makes the U-Net more suitable to the current

task, compared to other previously mentioned image-to-image architectures is that the network converges quickly, which is ideal when working with a dataset consisting of few samples (Ronneberger et al., 2015).

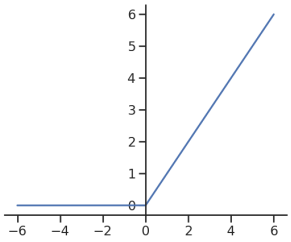
3.3 Describing the U-Net architecture

Figure (8) shows the U-Net architecture. This section intends to describe the different components constituting the architecture from a technical point of view.

3.3.1 The convolutional block

A single convolutional block consists of two repeat convolutional layers, each followed by the Rectified Linear Unit (ReLU) (Nair and Hinton, 2010) nonlinear activation function. The ReLU activation function is defined as follows

$$f(x) = \max(0, x) \tag{5}$$



The ReLU function, similar to other activation functions used in deep neural networks, introduce non-linearities to the connections in the network. Thus the network is able to learn non-linear connections in the data.

Each convolution is performed using a 3×3 window. The original formulation of the U-Net also does not apply padding to the input, resulting the convolutional filter only being applied to the entries of the input where the filter is never out of bounds. With a stride $S = 1$, this results in each convolutional layer reducing the spatial extent by two pixels in each direction following equation (2). It is also noted that the number of feature maps is doubled after each downsampling step, which is performed by the pooling layers.

Either in this section or next section write about equation for calculating do-

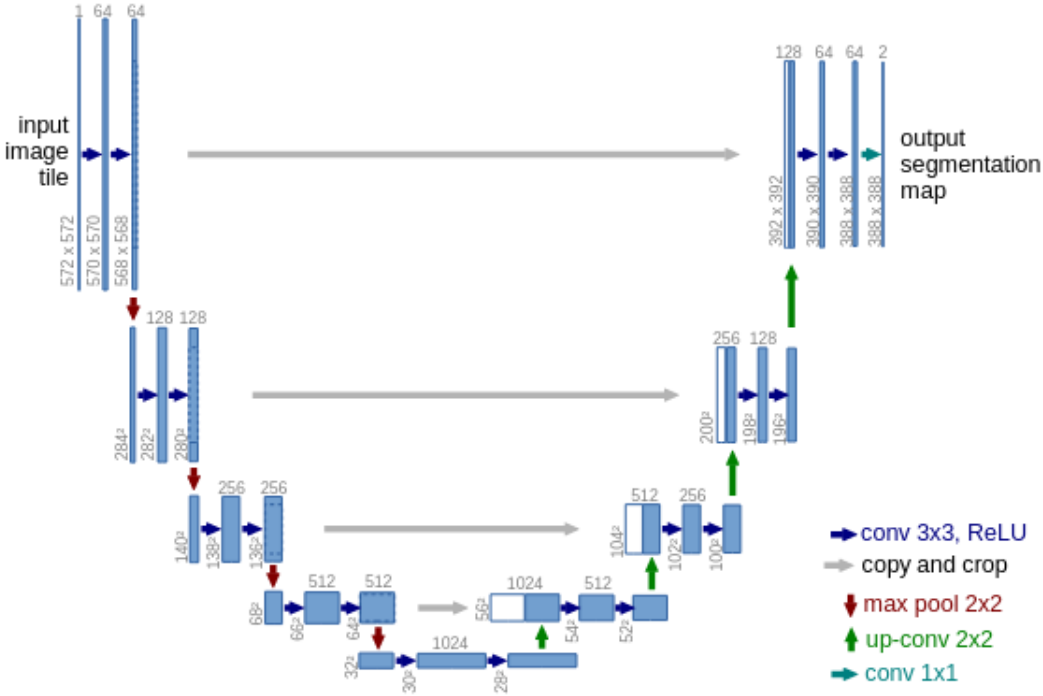


Figure 8: The U-Net architecture. The blue boxes represent feature maps, with the lower left numbers determining the spatial resolution and the top number the amount of feature maps. White boxes in the expansive path (right side / decoder) are the copied feature maps from the contractive path (left side / encoder). Arrows denote the different operations. Note that the original U-Net only performs *valid* convolutions, i.e. convolution without padding to match the input. This causes a convolutional layer to slightly decrease the spatial extent. As a result, the copied features from the contracting path are also cropped to match the dimensionality in the expansive path. Figure extracted from Ronneberger et al. (2015).

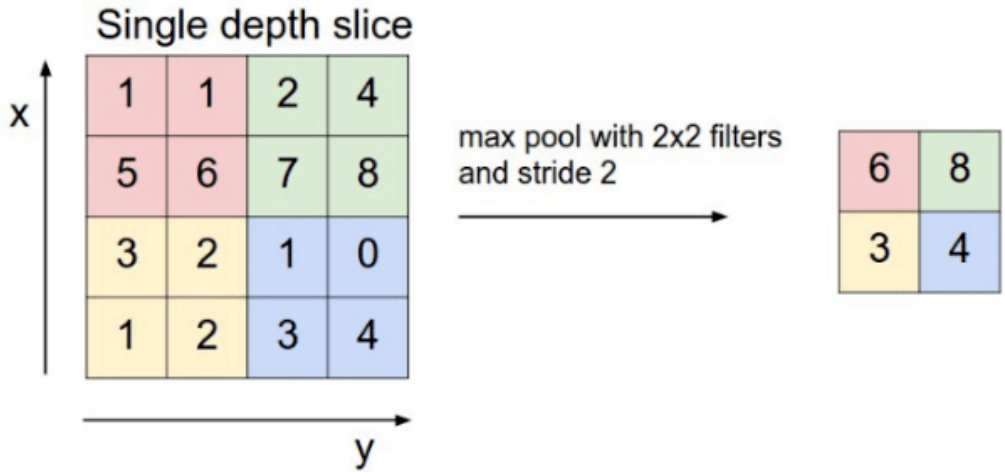


Figure 9: The max-pool operation for a 2×2 filter with a stride of 2. Figure taken from (RADU et al., 2020)

3.3.2 Maxpooling

Pooling operations are used to reduce the spatial extent of the current feature maps, by downsampling the data in the spatial dimensions. As seen in Figure (8), the U-Net downsamples the data in the contracting path through 2×2 maximum pool layers with a stride of 2. This specific configuration causes the spatial resolution to be halved. In the max-pool layer, a filter runs through each input channel and and chooses the maximum value inside the neighborhood of the filter. As such, the extreme values in each feature map is retained at the expense of rejecting the rest of the data. Since the maxpooling operation is rejecting some parts of the data, it may be regarded as a regularizer for the network which aid in keeping the model generalized, which is a topic further explored in the final paragraph of section 3.4. See Figure (9) for a graphical description.

3.3.3 Transposed convolutions

Transposed convolution was proposed by Zeiler et al. (2010) (note the incorrect use of deconvolution, this is not the mathematical inverse of a convolution) to increase the resolution of a feature map. The method was first utilized by Long et al. (2015) to connect the coarse output of an encoder with the image resolution of the target (it is referred to as both *backwards convolution* and *deconvolution* in the proceedings paper). Similar to the convolutional layer, the transposed convolutional layer involves striding a convolutional filter with trainable parameters across a feature map. However, the transposed convolutional layer projects a singular entry from the input through the convolutional kernel to

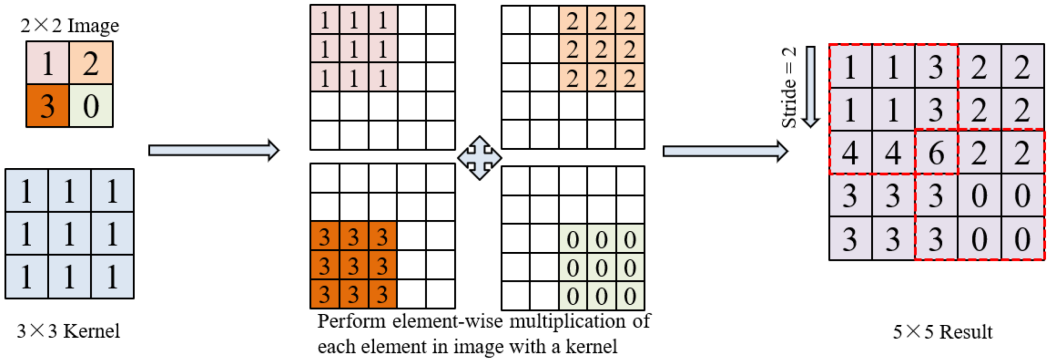


Figure 10: Figure demonstrating the computations performed by a transposed convolutional layer. Figure adapted from (Wu et al., 2022)

produce an output that is larger than the input. Figure (10) shows a graphical description of transposed convolutions.

In the Encoder architecture, lower level feature maps provide spatial information regarding where stuff is located in a scene, whereas higher level feature maps contain information regarding what is in the scene at the expense of losing spatial information (Long et al., 2015). To circumvent this, Ronneberger et al. (2015) concatenate the features from the contracting path with the output from the transposed convolution at the same level of depth, i.e. where the number of feature maps are equal at the end of the convolutional block. The concatenation operation is possible in Ronneberger et al. (2015) since they crop the feature maps in the encoder in their spatial dimensions to match the spatial dimensionality of the feature maps in the decoder. The operation can be seen in Figure(8) denoted by the gray arrow. The resulting convolutional layer is then trained to make a more precise prediction due to the concatenated input (Ronneberger et al., 2015).

3.3.4 Outputs

The output layer of the U-Net is denoted by the turquoise arrow at the right side of Figure (8). The arrow denote that the input is processed by a convolutional layer which have as many filters as there are output classes. Each filter is of size (1×1) with stride $S = 1$ and maps each layer in the input feature map to their respective class probability map of equal spatial shape (Ronneberger et al., 2015). By inspecting Figure (8), the U-Net outputs two feature maps, and from each feature map the pixelwise probability of belonging to the associated class can be computed.

3.4 Training procedure for the U-Net

This subsection aims to demonstrate how Ronneberger et al. (2015) trained the U-Net, and will consequently highlight some different hyperparameters and exemplify some functions and operations which are used in the training. Hyperparameters refer to model parameters which are not updated during training (Yu and Zhu, 2020), and may directly influence the model architecture or the training procedure. This section will not describe how samples were preprocessed and loaded, and modifications made which reflects concerns regarding medical images may be noted but not explained.

Training the U-Net starts by assigning random values to the weights of the network. Since the U-Net utilizes the ReLU activation function after each convolutional layer in the convolutional blocks (Ronneberger et al., 2015), it is standard for each layer to draw the weights from a normal distribution with $\mu = 0$ and standard deviation $\sigma = \sqrt{\frac{2}{n_l}}$, where n_l is the number of inputs to the layer He et al. (2015a). This weight initialization scheme ensures that variance of the feature maps are approximately equal, i.e. avoids varying the activation of input signals between layers He et al. (2015a); Ronneberger et al. (2015).

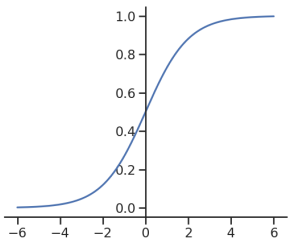
The process of training the U-Net involves making predictions on all training data. For each sample, the prediction is compared against a ground truth label. For the U-Net, a pixelwise prediction map is created by computing the pixelwise softmax which is an extension of the softmax function (Bridle, 1990) defined as

$$p_{k,i,j} = \frac{e^{a_{k,i,j}}}{\sum_{k'=1}^{K} e^{a_{k',i,j}}} \tag{6}$$

where a_k is the feature map for feature channel k of input x and K is the number of output classes and $p \in [0, 1]$. i, j are the spatial coordinates. Similarly to the standard softmax function (Bridle, 1990), equation (6) is ≈ 1 for the class that has maximum $a_{k,i,j}$ and ≈ 0 for all other classes, albeit depthwise in the channel dimension for all pixels (Ronneberger et al., 2015). The sum of the depthwise output from the pixelwise softmax is 1, hence the function maps each pixel with the probability of that pixel belonging to each class.

For the case of binary classification, i.e. when the number of classes $k = 2$, the softmax function in equation (6) is reduced to the Sigmoid function which is defined as,

$$p_{i,j} = \frac{e^{a_{i,j}}}{e^{a_{i,j}} + 1} \tag{7}$$



To quantify the prediction error, a loss function is defined. The overall goal of training a neural network is to minimize the loss function with respect to the trainable weights. For the U-Net, a weighted variation of the cross entropy loss function is proposed (Ronneberger et al., 2015).

$$L(p) = \sum_{i,j \subset \mathbb{Z}} w_{i,j} log(p_{l,i,j}) \tag{8}$$

where w is a predefined weight map and $p_{l,i,j}$ (equation 6) is the prediction made at pixel i, j at the true label l.

The error computed by the loss function is then sent backwards throughout the network according to the backpropagation algorithm (Rumelhart et al., 1986), which effectively computes the gradient of the loss function with regards to the trainable parameters

$$\frac{\partial L}{\partial w_l} = \frac{\partial L}{\partial p} \frac{\partial p}{\partial w_l} \tag{9}$$

where w_l is the trainable parameters associated with the l-th layer. The gradient of the loss for a weight at a given layer shown in equation (9) is used by an optimizer to adjust the weights such that the loss is minimized with respect to the weights (gradient descent).

Ronneberger et al. (2015) uses the stochastic gradient descent with momentum optimizer implemented in the Machine Learning library Caffe (Jia et al., 2014), where the optimizer is defined as follows,

$$w_l^{t+1} = \gamma(w_l^t - w_l^{t-1}) - \mu \frac{\partial L}{\partial w_l^t} \tag{10}$$

In equation (10), the superscript t was added to w and refers to training step, which is defined as a prediction and subsequent backpropagation of a batch of samples, where the size of a batch is a pre-determined hyperparameter. γ and μ are the momentum

and learning rate hyperparameters respectively. Note that γ is introduced by momentum stochastic gradient descent, whereas the learning rate μ is a hyperparameter common for all deep learning models and determines the rate of weight adjustment as seen in equation (10).

When all training samples have been inspected once by the U-Net, the training data is shuffled and the above outlined training procedure is repeated. The process of going through all the training data once is defined as an epoch. The number of epochs is a hyperparameter which can be adjusted, and is tied to the bias-variance tradeoff dilemma (Geman et al., 1992). Moreover, the number of epochs determines the duration of training time, and is influenced by the available computing resources.

Geman et al. (1992) states that the cost of low bias in a model is high variance. A model with high bias and low variance is assumed to not have underwent much, if any training, and is thus underfitted to the data. Consequently, a model with low bias but high variance has been trained for a high number of epochs, and is overfitted towards the training-data. An overfitted model is, due to its high variance, ideal at explaining the training data, but lacks the ability to generalize to external datasets. For the training procedure described above, the optimum model has been trained for a sufficient amount of epochs, where it is neither underfitted nor overfitted.

3.5 Forecast verification metrics

Verification schemes provide insight into how a forecasting system performs. For this thesis, verification metrics serve a dual purpose. From a model development point of view, verification metrics will be used to increase the skill of the model. However, the same metrics will also be utilized to assess the quality of a prediction as well as explain the physical interpretation of the model (Casati et al., 2008). The model developed for this thesis predicts a scene consisting of labelled pixels, as described in section (3.2). It was mentioned in section (1) that the developed model is aimed towards operational end users, which is partly achieved by validating the model against metrics of end user relevance. Furthermore, it can be assumed that the model and target observations will not differ much outside of the marginal ice zone (Fritzner et al., 2020). Thus, this section will introduce metrics which are relevant for evaluating the sea-ice edge position, as the sea ice edge is important information for maritime operators in the Arctic (Melsom et al., 2019). The following subsections will describe how to determine the position of the sea ice edge, as well as its length according to Melsom et al. (2019), and derive the Integrated Ice Edge Error (Goessling et al., 2016), with regards to a spatially gridded dataset of deterministic sea ice concentration values.

The Integrated Ice Edge Error is chosen among similar sea ice edge metrics (Melsom et al.,

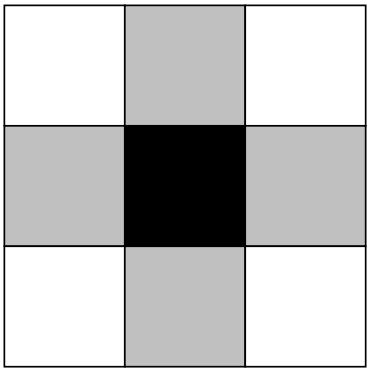


Figure 11: The gray pixels forms the 4-connected neighborhood of adjacent grid cells for the center pixel.

2019; Dukhovskoy et al., 2015) as it has been shown to be less sensitive to isolated ice patches (Palerme et al., 2019). Furthermore, the work of Melsom et al. (2019) recommends the Integrated Ice Edge Error amongst other metrics for its intuitive interpretation as well as for the possibility to provide the spatial distribution of IIEE areas.

3.5.1 Defining the Sea Ice Edge

The sea ice edge for a given spatial distribution of sea ice concentration values is derived on a per pixel basis. Let $C \in \mathbb{R}^2$ be gridded sea ice concentration values. Then, the sea ice edge is defined as the entries in C which meets the following condition,

$$c_{i,j} \geq c_e \wedge \min(c_{i-1,j}, c_{i+1,j}, c_{i,j-1}, c_{i,j+1}) < c_e \tag{11}$$

In Condition (11), $c \subset C$ are sea ice concentration values, with i, j denoting indexes. c_e is a given concentration threshold.

Next, let $E \in \mathbb{R}^2$ be the set containing sea ice concentration pixels constituting the sea ice edge. It can be seen that the entries $c_{i,j}$ which adhere to condition (11) form the set E (Melsom et al., 2019).

Moreover, all the entries in E each contribute to the total length of the sea ice edge, with each entries' length contribution determined based on that entries' 4-connected adjacent grid points, (see figure 11). Using this formulation, the different combination of neighborhoods in E can result in three different length contributions. For the following contributions, s is the spatial resolution of the grid.

- A neighborless pixel is assumed to yield a contribution equal to the length of the

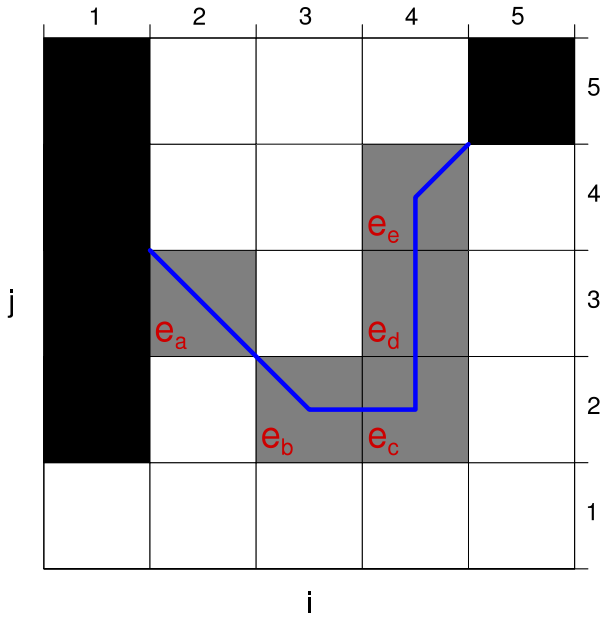


Figure 12: Sketch of an example gridded ice edge. The gray cells denote ice edge cells, which are labelled and illustrates the ice edge contained in the cell. The black cells denote land. Figure fetched from (Melsom et al., 2019)

diagonal of a grid cell ($l = \sqrt{2}s$). Here it is assumed that the grid cell only have diagonal neighbors (e_a in figure 12).

- A pixel with one of the four possible adjacent grid points contributes with the mean value between the length of the grid cell and length og the diagonal of the grid cell $l = \frac{s+\sqrt{2}s}{2}$. It is assumed that the grid cell also has a diagonal neighbor (e_b and e_e in figure 12).
- A pixel with two or more of the four adjacent grid points contributes with its spatial resolution (length of the grid cell) $l = s$ (e_c and e_d in figure 12).

The final length of the sea ice edge length then becomes

$$L = \sum_{\text{e in E}} l^e \tag{12}$$

where the superscript l^e denotes the length associated with the entry e according to the algorithm listed above, I.e. the sum of all contributions.

3.5.2 Integrated Ice Edge Error

The IIEE is an error metric which compares a forecast f to a predefined ground truth target t Goessling et al. (2016). The metric is defined as

$$\text{IIEE} = \text{O} + \text{U} \tag{13}$$

where

$$\text{O} = \int_A \max(C_f - C_t, 0) dA \tag{14}$$

and

$$\text{U} = \int_A \max(C_t - C_f, 0) dA \tag{15}$$

with $A \in \mathbb{R}^2$ being the area of interest, and is of similar size as C. Subscript f, t denotes whether C contains forecasted or target sea ice concentration values. In Equations 14 and 15, C is binary and is equal to 1 if its concentration value is above a predefined threshold, and 0 elsewhere (Goessling et al., 2016). From the definition of the metric, it can be seen that the IIEE is a sum of the forecast overshoot and undershoot compared to the ground truth target. For a graphical description, see figure (13)

Additionally, the IIEE can also be represented as a spatial metric by removing the integral with respect to A in equation (14 and 15). In this way, the metric is used to define the set of pixels which constitutes its area. To clearly distinguish between the area O (overestimation) and the set of pixels used to compute O, A^+ will be used to note the latter. Similarly, A^- will represent the set of pixels constituting U (underestimation). Finally, it can be seen that A^+ and A^- represent the spatial distribution of False Positive and False Negatives of the forecast respectively.

The length of the ice edge has a strong influence on the IIEE (Goessling and Jung, 2018; Palerme et al., 2019). Hence, to ensure that forecast errors are comparable across seasons, IIEE is normalized with the length of the ice edge, as mentioned in section (2.2.2). Furthermore, the normalized IIEE provides an estimate of the displacement error between the forecasted and target sea ice edge (Melsom et al., 2019).

Figure 13: 15% sea ice concentration contours for a forecast (blue) and target (red) sea ice concentration product. The IIEE is the sum of the overestimated (O, blue) and underestimated (U, red). White denotes the union between the products. Figure fetched from (Goessling et al., 2016).

3.6 AI explainability

Explainable Artificial Intelligence (XAI) is a field which has seen a recent relevance growth in conjunction with the renewed interest in deep learning methods (especially for image analysis) launched by the network proposed by Krizhevsky et al. (2012). XAI covers methods which aim to provide insight into the "black-box problem" of machine learning (Adadi and Berrada, 2018), which for deep learning models arise partly due to the complex structures and many nonlinear connections found in the models (Lopes et al., 2022). For the purpose of this thesis, XAI methods which aim explaining a single decision will be applied both to increase the transparency of the developed deep learning system, as well as attempting to connect a single prediction to the underlying physics present in the input variables.

3.6.1 Gradient-weighted Class Activation Mapping for semantic segmentation

Several methods of XAI for visual explanations for decisions made by CNN-based models exists, such as (Simonyan et al., 2013; Zhou et al., 2016; Selvaraju et al., 2016; Sundararajan et al., 2017; Lundberg and Lee, 2017). The aforementioned methods are designed for image classification tasks, which covers problems where the entire image is associated with a single label. However, there has been limited work related to understanding the

decisions made by semantic segmentation models (Linardatos et al., 2020).

A method for activation based explanation of semantic segmentation predictions was proposed by Vinogradova et al. (2020), which is a modification of the Gradient-weighted Class Activation Map (Grad-CAM) that was first introduced by Selvaraju et al. (2016). Grad-CAM is an activation based explanation method, hence, Grad-CAM constructs a class activated map that highlights a weighted combination of all feature maps at a given layer for a given class. The assigned weights determines whether each feature map was considered important when predicting the considered class, and are computed from the gradient of the predicted logit with respect to each feature map at the considered layer (Selvaraju et al., 2016). Grad-CAM is mathematically defined as

$$L^c = \text{ReLU}\left(\sum_k \alpha_k^c A^k\right) \tag{16}$$

with the weights computed as

$$\alpha_k^c = \frac{1}{Z}\sum_{i,j}\frac{\partial y^c}{\partial A_{ij}^k} \tag{17}$$

where c denote the chosen class of interest, k the number of feature maps and i, j the spatial dimensions. $A \in \mathbb{R}^3$ are k feature maps of size i, j and y^c is the predicted logit for class c. Finally Z denote the number of nodes in feature map A.

Figure (14) provides an overview which summarizes equations (16 and 17). The figure also demonstrates the use of the deepest layer as the chosen feature maps, which Selvaraju et al. (2016) argues for as they contain object specific information rather than positional information, although the choice of feature map is a free parameter. Moreover, the outputted class activation map to the lower left corner of figure (14) represent where the model had to look to make the prediction (Selvaraju et al., 2016).

Furthermore, the modification presented by Vinogradova et al. (2020), hereafter referred to as seg-Grad-CAM, replaces y^c in equation (17) with a new term

$$\sum_{(i,j)\in M} y_{ij}^c \tag{18}$$

where M is a set of pixel indices (Vinogradova et al., 2020). Thus, the pixel indices chosen becomes a free parameter. Similarly to (Selvaraju et al., 2016), Vinogradova et al. (2020) also argues that computing the gradient w.r.t. the lowest resolution feature maps returns the most informative class activation maps. Hence, contrary to Grad-CAM which consider

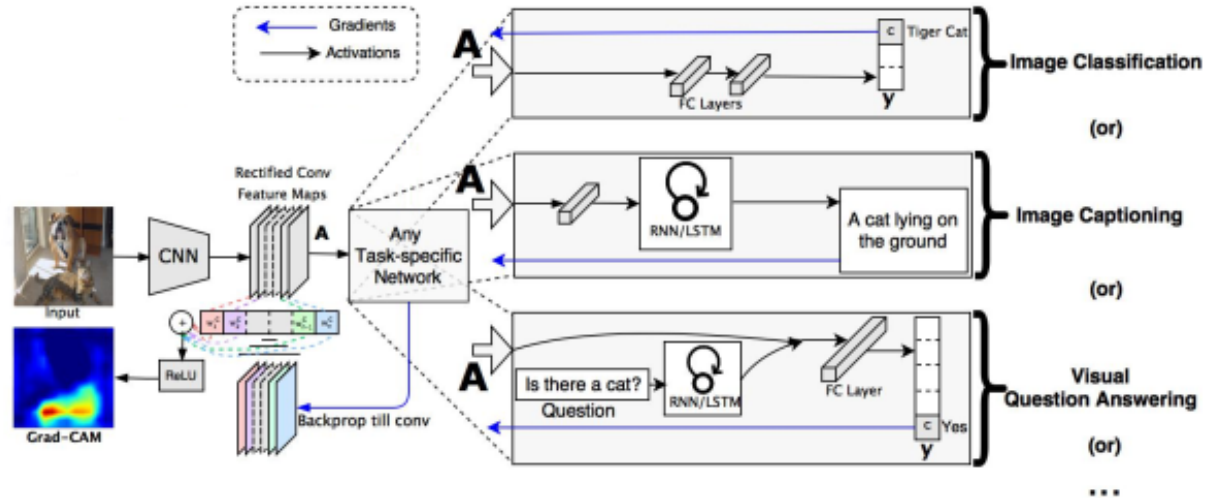


Figure 14: Overview showing how a class activation map is constructed form a single image. The bottom left image (captioned "Grad-CAM") provides an example class activation map for the class "tiger cat". The figure is heavily adapted from (Selvaraju et al., 2016), where references in the figure to Guided Grad-CAM and Guided backpropagation has been removed.

encoder only networks where the final convolutional layer contain the lowest resolution feature maps, seg-Grad-CAM compute the gradient w.r.t. the second convolutional layer located in the bottleneck of the U-Net, which in Figure (8) is represented by the lower right blue rectangle (followed by the first up-conv arrow).

4 Model development

This section will cover the implementation of the U-Net architecture, as well as related processes such as data preparation and writing a custom dataloader. Furthermore, this section will present intermediate results obtained during development to highlight technical decisions made as well as their consequence for model performance. Decisions made will be highlighted from a statistical point of view, and when relevant they will also be explored in a context of the underlying physics.

4.1 Data preprocessing

The deep learning system can be disassembled into two parts, a dataloader followed by the deep learning model itself. The dataloader structures already preprocessed data that

are then provided to the deep learning model during training. However, the predictors from a sample are from different datasets provided on different spatial grids and temporal frequency. Thus, preliminary computations are required in order to extract the desired spatiotemporal information from each predictor onto a common grid. Since the preparatory work on the predictors only need to be done once, the preprocessing is performed in advance of the training process. The following sections describes how the preprocessing is performed on the different datasets.

An overview of the data pipeline and workflow is described in figure (15). As can be seen from figure (15), a sample is constructed from a recent sea ice chart (Dinessen et al., 2020), the computed sea ice concentration trend over a set amount of previous days from OSI SAF SSMIS observations (Tonboe et al., 2017), recent AROME Arctic (Müller et al., 2017) 2-meter temperature and wind forecasts as well as a land-sea mask, which is the same land-sea mask used in AROME Arctic.

4.1.1 Regridding data

All input data loaded into the U-Net are on the AROME Arctic projection with a 1km spatial resolution and cover the same domain as required by the input layer of the U-Net architecture (Ronneberger et al., 2015). For geographic data, this requirement implies that all data used for training, validation or testing of the deep learning system has to be on a common grid. As such, following the region of interest outlined in section (2.1), the sea ice charts described in section (4.1.2) are used as the reference for the domain as they are already supplied on the desired projection and spatial resolution. Other products which are not on the AROME Arctic grid (Müller et al., 2017) or on a 1km spatial resolution (Dinessen et al., 2020) have to be reprojected and resampled to match the target grid.

For this work, the process of re-projecting and interpolation is performed on a per-product basis as part of the preprocessing routine. Re-projection of the datasets are performed with the Python library **Pyproj** (Snow et al., 2022), while interpolation onto the new coordinate system is done using nearest neighbor interpolation. In cases where the data are already present on the desired projection, but on a different resolution than the target, only nearest neighbor resampling is performed.

4.1.2 Sea Ice Charts

The sea ice charts used for this thesis have been made available by Nick Hughes of the Norwegian Ice Service. Hughes' work has involved readying the sea ice charts by gridding

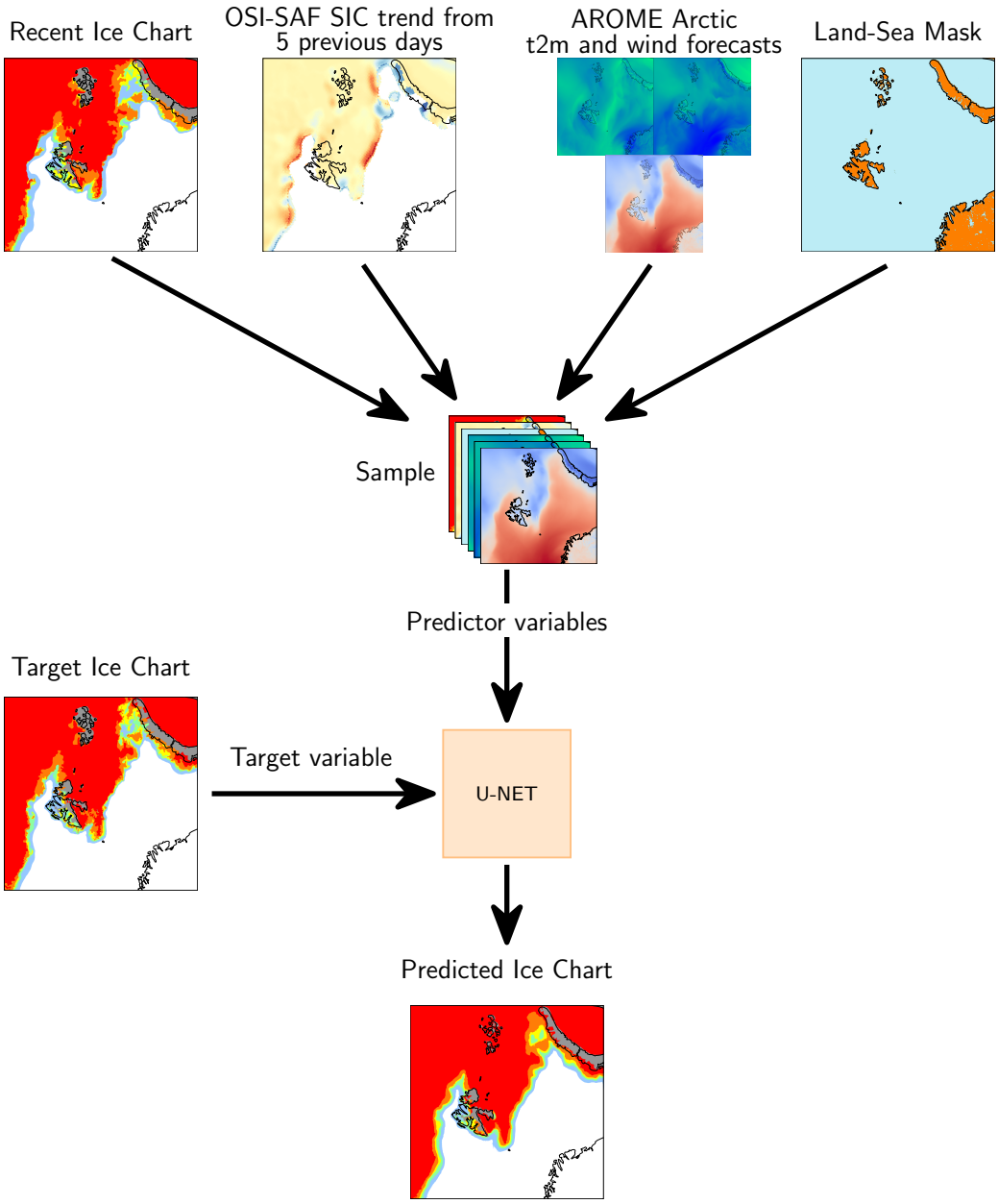


Figure 15: Workflow figure providing an overview of the data pipeline. Data are sampled from four sources (Sea Ice Charts, OSI SAF SSMIS, AROME Arctic and a Land Sea Mask), preprocessed and merged into a single sample. The sample is fed into the network together with an associated sea ice chart which is the target variable. The predicted sea ice chart is compared against the ground truth sea ice chart, and their binary cross entropy error is propagated backwards throughout the network, which constitutes a step in the training loop.

the dataset from a GIS production environment (Dinessen et al., 2020) where concentration contours are drawn onto a 1km spatial resolution grid with the AROME Arctic projection and domain (Müller et al., 2017). Nearest neighbor interpolation is used when projecting the polygons onto the AROME Arctic grid (Nick Hughes, 2022, pers. commun.). Moreover, the drawn vector polygons run under land, such that all the sea ice charts are fully consistent fields with no missing values. However, there is no systematicity to how the land pixels have been treated, and it has been advised by Hughes to mask out the land by the use of the land-sea mask from AROME Arctic (Nick Hughes, 2022, pers. commun.).

Since the U-Net architecture imposes the restriction that all predictors must consist of valid numerical values at all pixels (Ronneberger et al., 2015), two different methods for filling the masked land pixels have been attempted. The first method involves setting all land pixels to 0, thus labelling the land pixels to ice-free open water. However, it is noted that this approach adds additional ice-free open water to the region of interest (see figure 3), and thus may further skew the sea ice concentration distribution. The second approach is inspired by the work of Wang et al. (2017), which replaced land pixels by their mirrored counterpart. However, instead of mirroring as in Wang et al. (2017), a nearest neighbor interpolation of the surrounding pixels was used to fill the land pixels. Since the convolutional kernel only inspects a local neighborhood of pixel values (Yamashita et al., 2018), it is assumed that the nearest neighbor approach diminishes the amount of abrupt category change occurring within a filter compared to the initially proposed method. E.g. filling land pixels with open water would create a strong gradient if present next to fast ice, which the filter could detect as a notable feature.

Finally, as can be seen from the sea ice chart sample in figure (15), sea ice in the Baltic as well as any polygon drawn under land of the Norwegian and Russian mainland is filtered out. This is deliberate, since the task for the developed model is to predict Arctic sea ice only. However, filtering out Baltic sea ice is also important from a validational point of view, since if left unattended the Baltic sea ice would influence forecast verification.

4.1.3 OSI SAF linear SIC trend

As noted in section (2.2.2), the OSI SAF passive microwave product is pan-Arctic and delivered on a 10km polar stereographic grid. The processing of OSI SAF SSMIS can be described in two steps. Firstly, pixelwise linear trends are computed over the previous days preceding the forecast start date. Secondly, the computed trends are regridded to match the projection and resolution of the region of interest following the process described in section (4.1.1).

Kommer det tydelig frem at OSI SAF starter dgane før?

4.1.4 Atmospheric predictors from AROME Arctic

The forecast production scheme for AROME Arctic initiates four 66-hour forecasts each day at evenly spaced 6-hour intervals starting at 00:00 UTC. The goal of the deep learning prediction system is to provide a forecast at a given lead time on the same date as the predictor sea ice chart was published. Since the purpose of using atmospheric data from AROME Arctic is to provide the deep learning system with information regarding the future state of the atmosphere, it follows that the forecast should cover the time region between predictor publication time (15:00 UTC) and forecast target time (1-3 day lead time). When considering the publication scheme and lead time of AROME Arctic, the forecast initiated at 18:00 UTC was chosen. Regarding operationaility of the deep learning system, the timeliness of AROME Arctic forecasts are approximately 2.5 hours.

Burde dette nevnes et tidligere sted?

Furthermore, the sea ice charts represents the mean sea ice condition from the available observations up until the time of publication. Thus, when taking into account the publication time of observations which have been accounted for by the sea ice specialist, and to potentially fully utilize the 66-hour lead time of AROME Arctic, a target time of 12:00 UTC at the day of publication was considered for AROME Arctic forecasts. Thus, the above selection scheme provides atmospheric data that are contained within the observation time window of a forecast with respect to the chosen lead times.

All AROME Arctic data are stored as double precision floats (8 bytes), which when considering the target domain of (1792×1792) results in each field requiring $\approx 25.6Mb$ of memory. Because predictors and model variables are stored in memory at the same time, predictor formulations which reduce the amount of memory reserved for each input variable has been explored. In the case of AROME Arctic data, instead of inputting each relevant timestep for each field as a stack of separate predictors, a cumulative mean along the temporal dimension between forecast initialization and 12:00 UTC at target valid-date is computed. Thus, the memory requirements for each AROME Arctic field is reduced while temporal changes is encoded into the variable as it supplies the network information about the mean state of the 10-meter winds and 2-meter temperature during the forecasted period.

Studies such as Obite et al. (2020) have shown that artificial neural networks model multicollinear data better compared to traditional ordinary least squares regression, which indicates that feature engineering is less necessary for machine learning models. However the negative impacts of multicollinear predictors such as interdependance between variables and difficulties measuring the impact of a single variable still persist for deep learning methods (Chan et al., 2022). As such, providing AROME Arctic predictors as cumulative temporal mean fields intends to reduce the memory requirements of each predictor which enables greater batch sizes. Concurrently, the cumulative mean formulation may also aid to mitigate problems related to predictor multicollinearity.

Another motivation for supplying AROME Arctic data (a similar reasoning applies to the OSI SAF trend as well) as fields with temporal information embedded is to more easily assess the impact of each physical predictor on model performance. A temporal sequence of data both restricts the possibility of altering predictors for explainability purposes as well as introducing complexity to the nonlinear relationship between the model parameters and input data. Hence the current predictor formulation is intended to ease the inspection of predictor importance (see Section (**NOT YET IMPLEMENTED**)), strengthening the statistical relationship between a prediction and the underlying physics in the predictors.

The winds extracted from AROME Arctic are the u and v wind component at 10-meter height, i.e. the zonal and meridional wind components. However, the zonal and meridional wind do not retain the same orientation throughout the region covered by the study area, and as such the x and y components of the 10-meter winds with regards to the projection coordinate system are utilized. The x and y wind fields are normal components with respect to each other at all locations, and ensures that local neighborhoods considered in a convolutional layer always contain winds pointing towards the same direction.

4.1.5 Targets

The U-Net architecture adopted for this work performs classification, similarly to the U-Net outlined by Ronneberger et al. (2015). Thus the sea ice concentration categories present in the sea ice charts (Dinessen et al., 2020) will be used as ground truth labels when training the deep learning system. Furthermore, as the purpose of the model is to forecast the future state of sea ice concentration categories, following the sample creation date t_0 the following target sea ice chart will be $t_0 + (1 - 3)$ days depending on the choice of forecast lead time. Otherwise, as the sea ice concentration targets are drawn from the same pool as predictor sea ice charts, the same preparation considerations are applied.

Motivated by the sea ice concentration distribution presented in figure (3) along with limited fraction of the spatial domain covered by the Marginal Ice Zone ($0.15 \leq \text{sic} \leq 0.80$) as seen in the sea ice charts (e.g. figure 2) and figure 2 in Strong (2012), each sea ice concentration category in the sea ice chart is defined as cumulative contours and predicted independently. In terms of the U-Net architecture, instead of having a singular output layer (turqoise arrow in figure 8), the U-Net has individual output layers for each target sea ice concentration. Since each sea ice category represent a range of sea ice concentrations (Dinessen et al., 2020), each cumulative contour contains sea ice concentration categories equal to or greater than the lowermost category used to define the contour. This way, each cumulative contour represent a greater fraction of the domain than the individual classes, and predicting each contour separately is thought to be a more balanced prediction

task than predicting all classes simultaneously following the original U-Net architecture (Ronneberger et al., 2015).

The mathematical definition of cumulative contours are given as follows. let $C \in \mathbb{R}^3$ be a set representing ($N > 2$) contour elements with spatial indexes i, j and elements $c^n_{i,j}$. Moreover, let $S \in \mathbb{R}^2$ represent a sea ice chart, with $s_{i,j}$ being the sea ice concentration values between 0 and 1. Then, let $k_n \in [0, 1]$ be thresholds

$$0 \leq k_1 < k_2 < ... < k_n \leq 1 \tag{19}$$

Hence, each cumulative contour is defined as

$$c^n_{i,j} = \begin{cases} 1 & \text{if } s_{i,j} \geq k_n \\ 0 & \text{if } s_{i,j} < k_n \end{cases} \tag{20}$$

By adopting the scheme presented in equation (20), a more balanced representation of each contour is intended. Contrary to predicting each contour simultaneously, where the target dataset is skewed in disfavor of the MIZ contours, the cumulative contours reduce the classification task into multiple binary predictions where each contour is given a larger spatial distribution. Further details regarding technical implementation and how each predicted contour is reconstructed into a sea ice chart can be found in section (4.2)

4.1.6 Preparing and loading data

Sections (4.1.2, 4.1.3, 4.1.4 and 4.1.5) described how the data are preprocessed with respect to a common grid in addition to explaining how temporality is accounted for in the different datasets. The next step after the data are preprocessed is to store the data in predefined samples, as visualized in figure (15). For this work, a single sample is stored as an Hierarchial Data Format version 5 (.hdf5) file that make up the predictors and target for a given date at a given lead time. Hence, each considered lead time can be considered a separate dataset, as the composition of a sample differ based on the time offset between target and predictor enforced by the chosen lead time. The choice of storing samples as individual files is made to limit the amount of memory needed to store all samples simultaneously in memory, although at the expected cost of increased I/O overhead. A single .hdf5 sample occupies 467 Megabytes of memory.

The main dataset used in this thesis covers the period between 2019 and 2022 following the update to AROME Arctic in 2018 were an added snow on ice parameterization removed a significant temperature bias (Batrak and Müller, 2019). Regardless, datasets constructed from additional years will be inspected although specifically noted for clarity. Following

Table 2: Table showing the subset affiliation of each year, as well as the number of samples belonging to each year for all lead times. The dashed line is used to separate the core form the extended dataset, and represents the change in AROME Arctic following the implementation of (Batrak and Müller, 2019).

year	subset	1 day lead time	2 day lead time	3 day lead time
2022	test	196	147	142
2021	validation	198	147	142
2020	train	198	146	142
2019	train	192	143	144
2018	train	194	146	144
2017	train	187	139	140
2016	train	200	151	150

common machine learning practices, the dataset is split into separate train, validation and test subsets. The data used for training and validating the deep learning model can be categorized in two distinct groups. The first group is the data known by the system, which is used during training to increase or validate model performance. Additional to the data used during training is external data, which is needed to validate the generalizability of the model. I.e., how well does the model perform with unknown data, which is assumed to be drawn from the same distribution as the data used during training. It is standard practice to arbitrarily split by a given fraction into the three datasets (training, validation, testing). However, due to the seasonal dependency seen in the current dataset, a naive split of the data could result in seasonally unbalanced datasets. As such, the data split is done manually such that each data subset covers at least a full year. Thus, no dataset is assumed to be seasonally skewed.

Currently the data is split such that 2022 is the test dataset, 2021 is the validation dataset and ≤ 2020 is the training data. See table (2) for a summary. Letting each subset consist of at least a full year ensures that each subset covers the full seasonal cycle of sea ice concentration (Cavalieri and Parkinson, 2012), see also figure (3). However, it is noted that the above deterministic approach for splitting data deviates from common machine learning practices, where the data split is stochastic. A stochastic approach ensures especially that the test data is unknown to the developer as well as unseen by the model. Hence the latter approach associates less bias towards the data upon the model. Nevertheless an uneven seasonality is considered a greater detriment to model generalization than bias associated with a priori knowledge of the test dataset.

To increase the concurrency during training, a custom dataloader has been developed

which reads data and prepares samples simultaneously as the model trains on previously loaded samples. The preparations performed by the dataloader is twofold. Firstly, the predictors are normalized according to the min-max normalization equation

$$x' = \frac{x - min_s(x)}{max_s(x) - min_s(x)} \tag{21}$$

where x is the predictor with $'$ denoting the normalized variant. Subscript s refers to the subset minimum and maximum, implying the minimum and maximum of an entire data subset (train / validation / test). Each predictor is normalized according to a separate global minimum and maximum. Restricting the normalized data between $[0, 1]$ through min-max normalization ensures that the numerical predictors such as atmospheric data from AROME Arctic and the OSI SAF trend fall within the same value range as the categorical sea ice chart and land sea mask predictors. Min-max normalization is preferred for the current problem as it is invariant to the predictors being drawn from different distributions, e.g. the sea ice charts in figure (3) resembles a heavy tailed distribution whereas 2-meter temperature from AROME Arctic would more so resemble a Gussian, since all predictors are scaled to the same range.

Secondly, the dataloader separates the target sea ice chart into individual binary contours constructed cumulatively as described previously in section (4.1.5). During training, the dataloader also ensures that the samples are shuffled at the start of each epoch, which generalizes the optimizer (see e.g. equation 10) as the computed gradient of the loss is not biased towards the sample sequence.

4.2 Model implementation

The purpose of this subsection is to describe the implemented deep learning architecture. Considerations taken with respect to the format of the input data will be explored, such as the decomposition of the target ice chart into cumulative contours explained in section (4.1.5). Initial choices made, as well as integration of recent developments in deep learning based image processing will also be described.

4.2.1 Overall structure of the network

The developed model adopts the overall encoder-decoder structure of the U-Net architecture as described previously in section (3.3). The model architecture can be decomposed into four main components, the input layer, encoder, decoder and output layers. Moreover, both the encoder and decoder consists of multiple convolutional blocks, which chain

together a string of computations. The following sections will describe the technical details of each model component. The U-Net was developed using the machine learning framework Tensorflow v2.11 (Abadi et al., 2015) together with the Keras library (Chollet et al., 2015) using the Python v3.8.10 programming language.

4.2.2 Input layer

The future state of sea ice concentration is predicted on lead times 1 - 3 days using the current ice chart, atmospheric predictors from AROME Arctic as well as the linear sea ice concentration trend derived from OSI SAF passive microwave (section (4.1.6)). These predictors creates a sample consisting of 7 channels. The spatial shape of each predictor has been set to 1792×1792 and covers parts of the European Arctic as shown in section (2.1). The spatial size was set to be the even number 1792, as it is four times divisible by 4, thus allowing for four consecutive max pool operations in the encoder (Ronneberger et al., 2015) each reducing the domain by a factor of 4.

Another reason for the reduced domain extent was to limit the amount of memory needed when loading data during training. The AROME Arctic domain has a spatial shape of 2370×1845 after being resampled onto a 1km grid. A reduced spatial shape also reduces the number of computations performed in the network at each layer, which speeds up training. The southern and eastern extent of the domain was trimmed due to considerations of the expected sea ice dynamics with the goal to avoid targeting likely sea ice concentration containing grid cells. Firstly, there is no sea ice expected to occur at the southernmost grid cells. Secondly, the eastern part of the domain only experiences freezing during winter (Serreze and Meier, 2019).

4.2.3 Encoder

The encoder captures spatial features including temporal structures for the predictors with embedded temporality, which are used to capture the context of the scene (Ronneberger et al., 2015). The encoder is constructed by stacking a sequence of convolutional blocks with average pooling layers. Average pooling is similar to max pooling as it downsamples what is given as input, however instead of choosing the maximum value from a sliding filter the layer reduces what is seen by the filter to a mean value. Average pooling is described in figure (16).

Due to the large spatial extent of the input predictors, the average pooling layers are defined to have a 4×4 filter with a stride of 4. Hence, each pooling layer reduces the spatial size of the feature maps by a factor of 4, increasing the domain of influence captured by the receptive field of each pixel in the final feature map of the bottleneck.

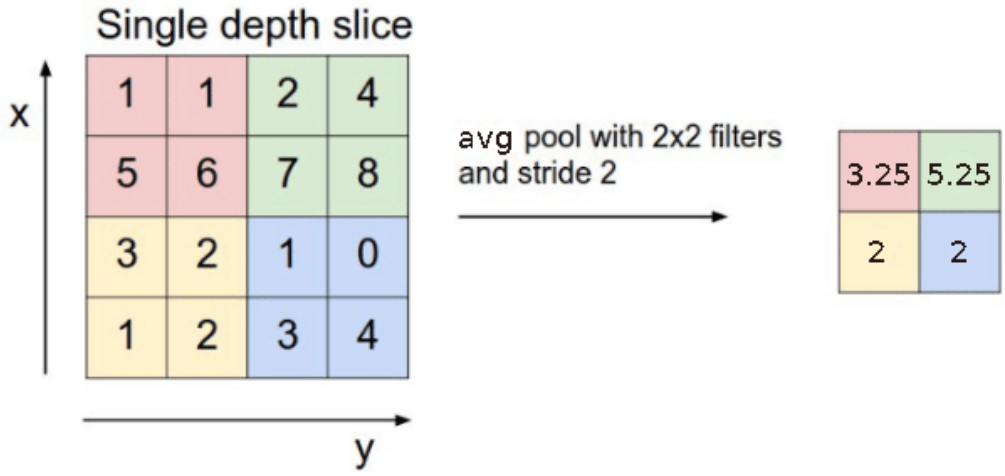


Figure 16: The average pool operation for a 2×2 filter with a stride of 2. Figure modified from (RADU et al., 2020)

4.2.4 The convolutional block (find new title, same as in methodology)

The convolutional block from Ronneberger et al. (2015) is adopted for this work, with both the (3×3) filter size and ReLU (Nair and Hinton, 2010) activation functions. To retain the spatial size of the of the input scene, each convolution is performed with zero-padding, which was omitted in Ronneberger et al. (2015) as it has the potential to create visual artefacts in the border region. Each convolutional block contain two convolutional layers with 2^{6+n} number of output feature maps, where n is the stage of the encoder starting from $n = 0$. Figure 8 visualizes how the number of feature maps follows the depth of the U-Net. The objective of the convolutional block is to detect features and construct feature maps from the incoming tensors, with each filter in the convolutional layers being sensitive only to a single pattern (Fukushima, 1980).

The current implementation of the convolutional block deviates from the original U-Net architecture as a normalization layer is added after the each activation function to speed up and stabilize training (Ioffe and Szegedy, 2015). Although Batch Normalization has been commonly implemented in deep networks, works such as Wu and He (2018) demonstrate that the technique exerts drawbacks for small batch size training. In Wu and He (2018), the drawbacks in batch normalization are attributed to the normalization statistics computed along the batch dimension of a feature map. Furthermore, Wu and He (2018) presents an analogous technique for computing normalization statistics, albeit computed along the channel dimension which is divided into connected groups (Wu and He, 2018). The number of groups is set with a hyperparameter G. Figure (17) visualize the different normalization techniques in Ioffe and Szegedy (2015) and Wu and He (2018).

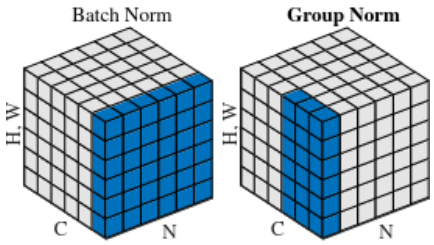


Figure 17: Schematic showing a feature map tensor of spatial shape (H,W) with N batches and C channels. The blue pixels are normalized by the same mean and variance. The group normalization hyperparameter $G = 2$. Figure modified from (Wu and He, 2018)

As increasing the batch size quickly saturates the available memory due to the high resolution predictors, group normalization is adapted for normalizing the feature maps. This follows the results in Wu and He (2018) where group normalization was shown to reduce network error for small (< 8) batch sizes. Following the recommendations by Wu and He (2018), the hyperparameter $G = 32$.

4.2.5 Decoder

The decoder restores the feature maps outputted by the encoder to image-resolution through the use of transposed convolutions (Zeiler et al., 2010). There are a similar amount of transposed convolutional layers as there are pooling layers, yet the number of convolutional blocks is reduced by one when compared to the encoder (see figure (8)). The convolutional blocks in the decoder have the same structure as those used in the encoder. Each transposed convolution has a filter size and stride equal to the pooling factor, which has been set to 4. Moreover, the output space for each transposed convolution is $2^{5-m+n_{\max}}$ starting with $m = 0$ for the first transposed convolutional layer.

4.2.6 Output layers

The feature maps at the final stage of the encoder is fed to the output component of the network. The output component is comprised of C individual output layers as found in Ronneberger et al. (2015) (convolutional layer with (1×1) filter size). However, the number of output channels of each output layer has been reduced to 1, following the definition of cumulative target contours described in section (4.1.5). Hence, each output layer facilitates a binary classification task in which each pixel is predicted to belong in the cumulative contour associated with the layer. Finally, each prediction is activated pixel-wise with the sigmoid function (equation (7)) which outputs a probability score

$[0, 1]$ for belonging to the predicted contour. The output from the network is on the shape $(C, 1792, 1792, 1)$.

Given that each prediction is of a spatially decreasing cumulative contour (see section 4.1.5), the predicted sea ice chart is constructed as a sum of the individual predictions along the first dimension of the model output. Here, it is assumed that the individual predictions adopt the cumulative property of the target cumulative contours.

Initially, a more conventional architecture with a single output layer and multiple output categories (Ronneberger et al., 2015) was attempted. Conversely, the model applies the softmax activation function (equation (6)) to the outputs, which ensures that a single class is selected as most likely. However, due to unsatisfying results during the initial stages of development, further pursue of the architecture was dropped in favour of the above described model. An example prediction visualizing the problematic aspects of the model can be seen in section (4.3.2).

4.2.7 Training environment

The model is trained on a GPU workstation with an Nvidia A100 80-Gb GPU available. The largest achievable batch-size in the environment was four. To both speed up training and reduce the memory footprint of the predictors, mixed precision training was utilized (Micikevicius et al., 2017). Mixed precision refers to storing the predictors as half-precision floats, whereas parameters in the model are stored as single-precision floats. Similarly to Ronneberger et al. (2015), the model weights are HE-initialized (He et al., 2015a) since the ReLU activation function (Nair and Hinton, 2010) is used in the convolutional blocks.

The loss function implemented is the pixelwise Binary Cross Entropy loss, which is an unweighted variation of the loss implemented in (Ronneberger et al., 2015) (equation (8)) for binary classification tasks.

$$L = -\frac{1}{N}\sum_{n=1}^{N}\sum_{p\in\mathbb{Z}^2}\left(y_p^n \log\left(\hat{y}_p^n\right) + \left(1 - y_p^n\right)\log\left(1 - \hat{y}_p^n\right)\right) \tag{22}$$

where N is the batch size, y is the true label and $\hat{y}$ is the predicted probability by the model. Subscript p refers to the pixels in y and $\hat{y}$. As a consequence of the architecture described in section (4.2.6), the computed losses at each output layer exerts individual contributions to the convolutional layers after the split at the end of the decoder. Furthermore, at the end of the decoder, each loss contribution is reduced to a sum before further propagated through the network during backpropagation. The optimizer used is the ADAM optimizer (Kingma and Ba, 2014).

Motivated by the bias-variance tradeoff described at the end of section (3.4), a validation dataset is used to determine at which epoch the model achieves highest generalizability during training. Further motivated by the regularization effects offered by early stopping (Graves et al., 2013), while still allowing the network to converge further, a model checkpoint technique is deployed where the state of the model weights is stored every time the validational loss achieves a new minimum. As noted, the validation loss is monitored, which will be explored further in coming sections.

4.3 Hyperparameter tuning and model selection

Throughout this section, the ice edge displacement metric is referred to frequently. If not otherwise stated, what is referred to is the IIEE of the ($> 10\%$) contour normalized with respect to the climatological sea ice edge (section 2.2.4). Notations such as NIIEE refer to this metric. The details are described in the following section 4.3.1.

4.3.1 Computing a climatological sea ice edge

The IIEE (Goessling et al., 2016) was derived in section 3.5.2 as an ice edge aware metric, which reports the average sea ice edge displacement error between two products when divided by the length of the sea ice edge (Melsom et al., 2019). Moreover, section 2.2.4 presented the OSI SAF CDR as an independent observational product which will be utilized to derive a climatological sea ice edge which will serve as a normalization factor for the computed IIEE. What follows are the considerations made to construct the climatological sea ice edge.

Following the definition of the MIZ from Strong (2012), the lower boundary of the MIZ is used to define the sea ice edge concentration threshold. Using the 15% sea ice concentration as a threshold for the sea ice edge is customary in similar works on sea ice edge verification metrics (Dukhovskoy et al., 2015; Goessling et al., 2016; Goessling and Jung, 2018; Melsom et al., 2019). With a threshold for the sea ice edge defined, computing the sea ice edge is done following the methodology of Melsom et al. (2019) described in section 3.5.1. As previously mentioned in section 2.2.4, the climatology sea ice edge is computed from a ten year mean (2011 - 2020) and is delivered on a daily frequency.

The relationship between IIEE and a varying sea ice edge length is shown in figure 18. The IIEE used in this figure was computed at 1km spatial resolution from the contour starting at 10% sea ice concentration, with the resolution of the sea ice edge being varied. When computing the sea ice edge at 10km, both sea ice concentration fields were interpolated onto a 10km resolution grid using nearest neighbor interpolation. The correlation between the two Normalized IIEE curves in figure 18 is 0.98.

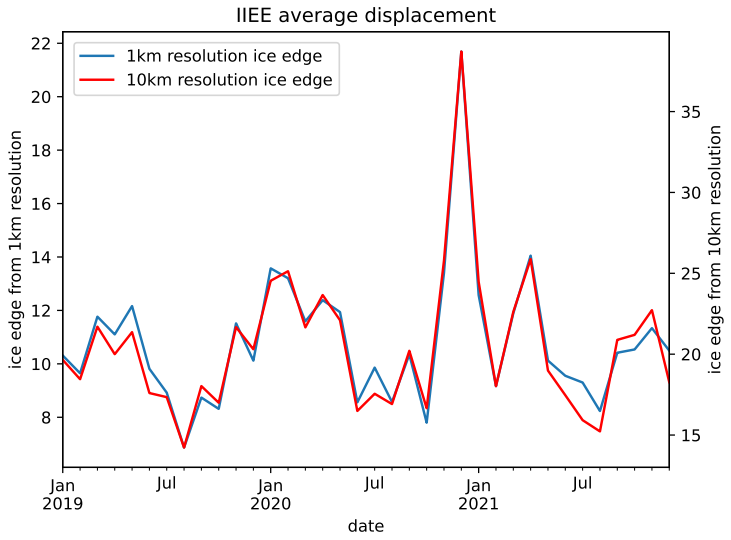


Figure 18: Integrated Ice Edge Error computed across three years (2019, 2020 and 2021) where the target is the sea ice charts and the forecast is persistence with a two day lead time. The IIEE was computed using a 1km spatial resolution, with the resolution of the sea ice edge varied. The ice edge length was computed as the mean sea ice edge length between the two products.

Moreover, the correlation between NIIEE computed from a 1km sea ice concentration field and a 1 km sea ice edge with the NIIEE computed from a 10km sea ice concentration field and 10 km sea ice edge is 0.98 as well. Finally, the mean difference between NIIEE from a 1km sea ice concentration field divided by a 10km sea ice edge and NIIEE from a 10km sea ice concentration field and a 10km sea ice edge is 0.1km.

4.3.2 Single output, multiple label model

During early stages of model development, an iteration of the deep learning architecture with a single output layer and multiple target labels was developed. Figure (19) is an example prediction made with the described model. The categories very open drift ice ($< 40\%$) and open drift ice ($< 70\%$) are not resolved by the model, and persists for all samples (not shown).

4.3.3 General training performance

Training the deep learning system takes $\sim$ 3h30min on the GPU workstation, although the training time have varied positively and negatively following driver updates and other non-

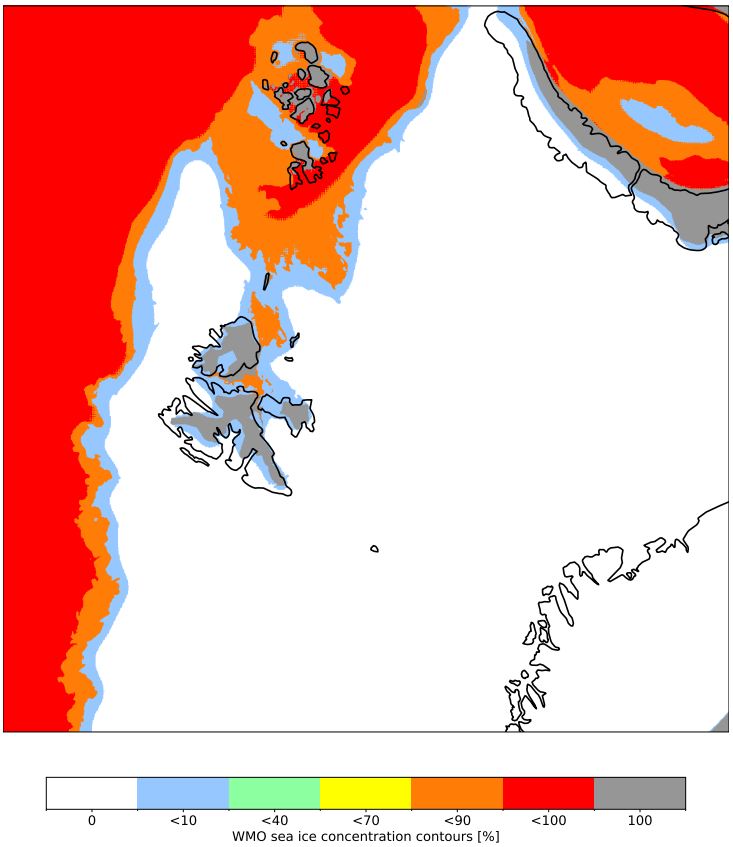


Figure 19: Prediction with a two day lead time, single output multiple labels U-Net 06 Jan 2021.

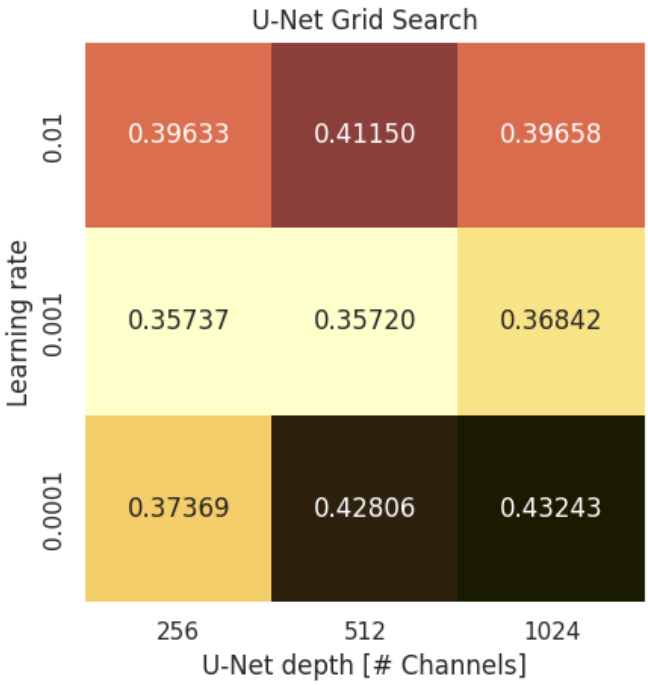


Figure 20: Grid search performed over variations of the learning rate as well as an increasing U-Net depth (represented by the number of feature maps at the at the final convolutional block). Each cell contains the minimum obtained validation loss for its respective combination.

transparent backend operations. Iterating through the training data takes $\sim$ 6 minutes, and the validation data $\sim$ 8 minutes for a single epoch. Memory usage varies between $\sim$ 19.4 and $\sim$ 55 gb, and scales with the depth of the unet. With a pre-trained model, performing a single prediction on a workstation CPU (AMD EPYC 7282 16-Core) takes $\sim$ 6 seconds, while on a laptop CPU (Intel(R) Core(TM) i7-8565U 8-Core) takes $\sim$ 30 seconds.

To determine the optimal learning rate and U-Net depth, a grid search was conducted across variations of the aforementioned variables. The result is shown in figure 20. It can be seen from the figure that the validation loss increases with U-Net depth. At the same time, the validation loss also increases when the learning rate deviates from 0.001.

Training curves for the model in figure 20 which achieved a loss of 0.35737 (lr = 0.001, U-Net depth = 256) are shown in figure 21. The figure plots both the training and validation loss, as well as a third curve which keeps track of the current minimum validation loss. The lowest validational loss is achieved at epoch 17.

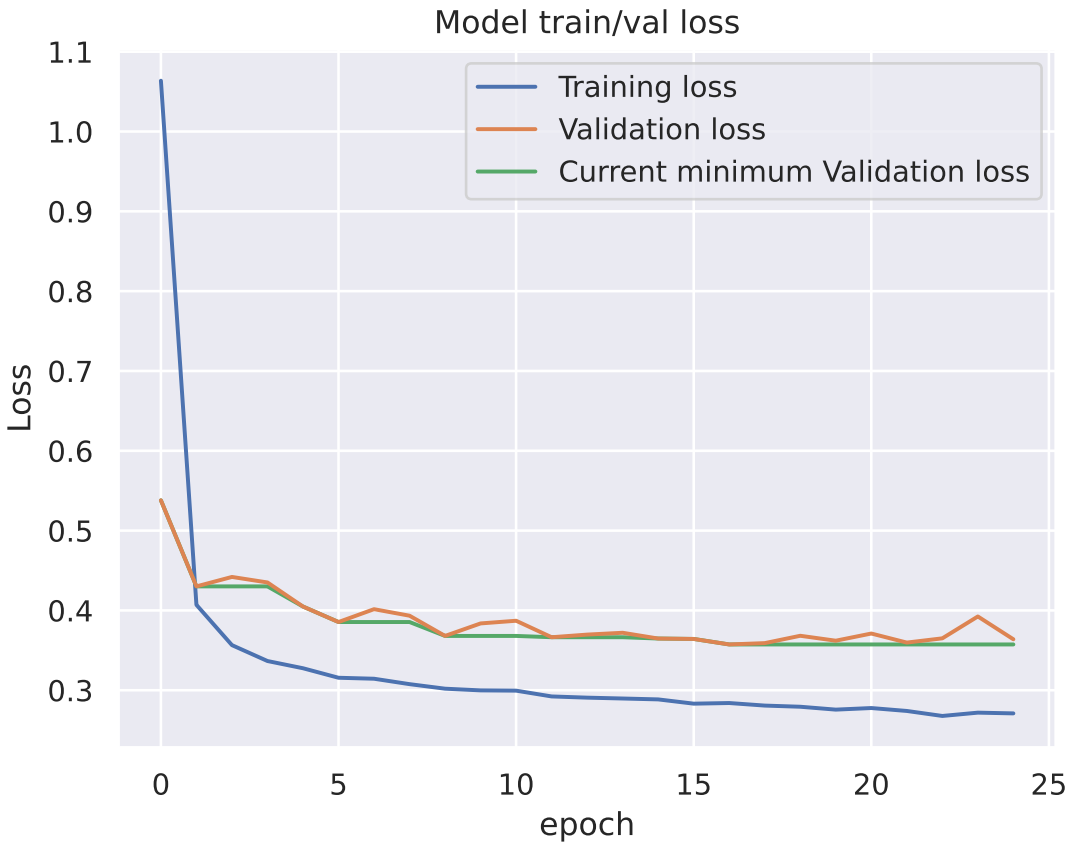


Figure 21: Training and validation loss from the middle leftmost model in Figure (20) (lr = 0.001, depth = 256). The current minimum validation loss is also displayed.

Figure 22 shows the impact that model depth have on the predictions. Both models are trained on the same data, and the best model for both training procedures is selected according to section 4.2.7. Both models resolve the scene comparatively, with mean annual statistics of ice edge displacement error being 28.2 km for the model in figure 22a and 30.7 km for the model in figure 22b. The total number of trainable parameters in the 256 architecture is $\sim$ 2.4 million, where $\sim$ 1.15 million are located in the encoder and $\sim$ 1.25 million in the decoder. The rightmost model in figure 22, which has a depth of 1024 filters, contain $\sim$ 16 times more parameters than the model with a depth of 256 filters, with a total of $\sim$ 39 million parameters. Decomposing the total number of parameters into encoder and decoder results in $\sim$ 19 million parameters in the encoder and $\sim$ 20 million parameters in the decoder. For reference, the 512 depth model contain $\sim$ 9.8 million parameters. The receptive field of the bottleneck (final feature map in the encoder) for both models is calculated using equation 4. The model with a depth of 256 has an encoder with a theoretical receptive field of 145 pixels in each spatial dimension, whereas the model with a depth of 1024 has an encoder with a theoretical receptive field of 2385 pixels in each direction. For the second model, a receptive field of 2385 results in the entire input scene being used as context for each pixel in the final encoder feature map. Note that the theoretical receptive field is invariant to the input shape.

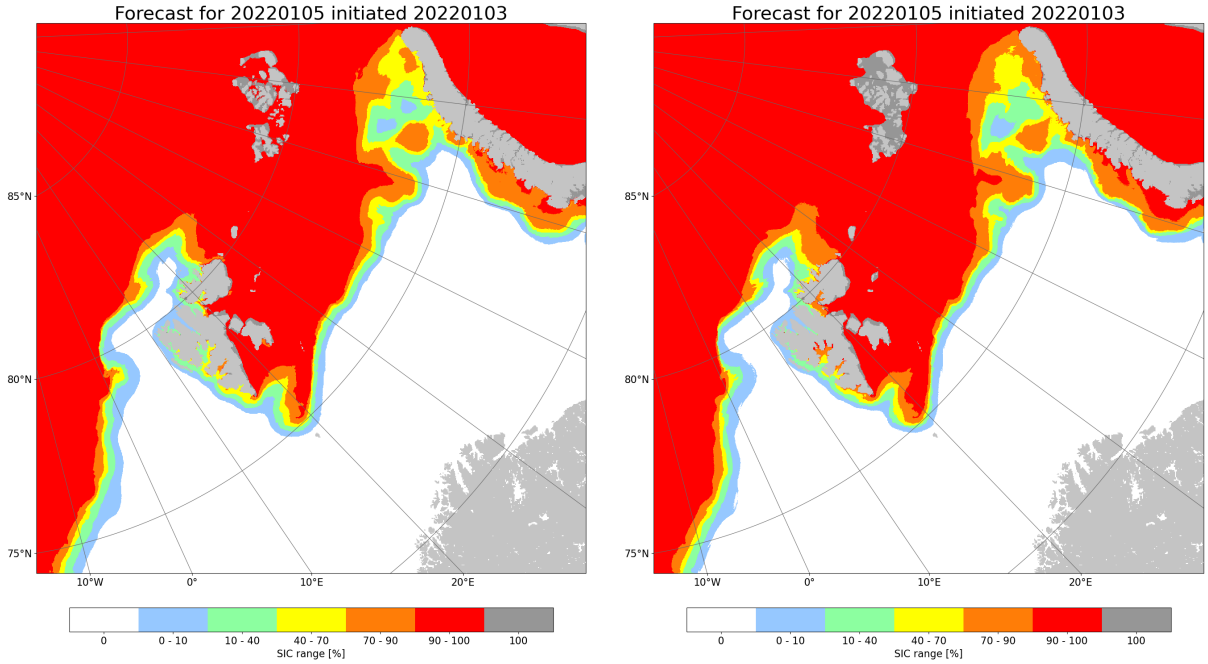
A full year of forecasts, where each month is represented by the first prediction made that month using a 2 day lead time model and 256 architecture is shown in figure 23. Figure 23 also visualizes the sea ice edge computed from OSI SAF ssmis observations at 10km spatial resolution. The figure is intended as an example of the predictive capabilities of the model across the entire test dataset.

Include sea ice edge from osisaf as in the next figure

The effect of training across varying lead time is shown in figure 24. From figure 24 a), it can be seen that both persistence and the deep learning forecast increase their mean annual NIIEE with increasing lead times. It can also be seen that persistence achieves a higher mean annual NIIEE than the deep learning model for all lead times. Figure 24 b) show that the forecast improvement increase with the lead time, which follows the divergence between the deep learning and persistence NIIEE curves in a) which show a diverging pattern. Note that it is expected for persistence to lose skill with increasing lead time.

Include WMO color-bar for this figure

An inspection on the effect of appending additional years to the core (2019 and 2020) training data is summarized in figure 25. Both validation loss and NIIEE is shown in figure 25. The model trained with a training dataset starting in 2016 and including all years up until including 2020 has higher validation loss (2021) and NIIEE (2022) than the other models in figure 25. The model trained with 2017 up until including 2020 achieves



(a) Prediction with a 2 day lead time for 05 Jan 2022 using a deep learning model with lr = 0.001 and depth 256.

(b) Prediction with a 2 day lead time for 05 Jan 2022 using a deep learning model with lr = 0.001 and depth 1024.

Figure 22: Prediction of the same date with two different models with parameters as in figure 20. The model in figure (a) contains $\sim$ 2.4 million parameters, and achieves a mean annual ice edge displacement ($>$ 10% contour) of 28.2 km. The model in figure (b) contains $\sim$ 39 million parameters, and achieves a mean annual ice edge displacement of 30.7 km

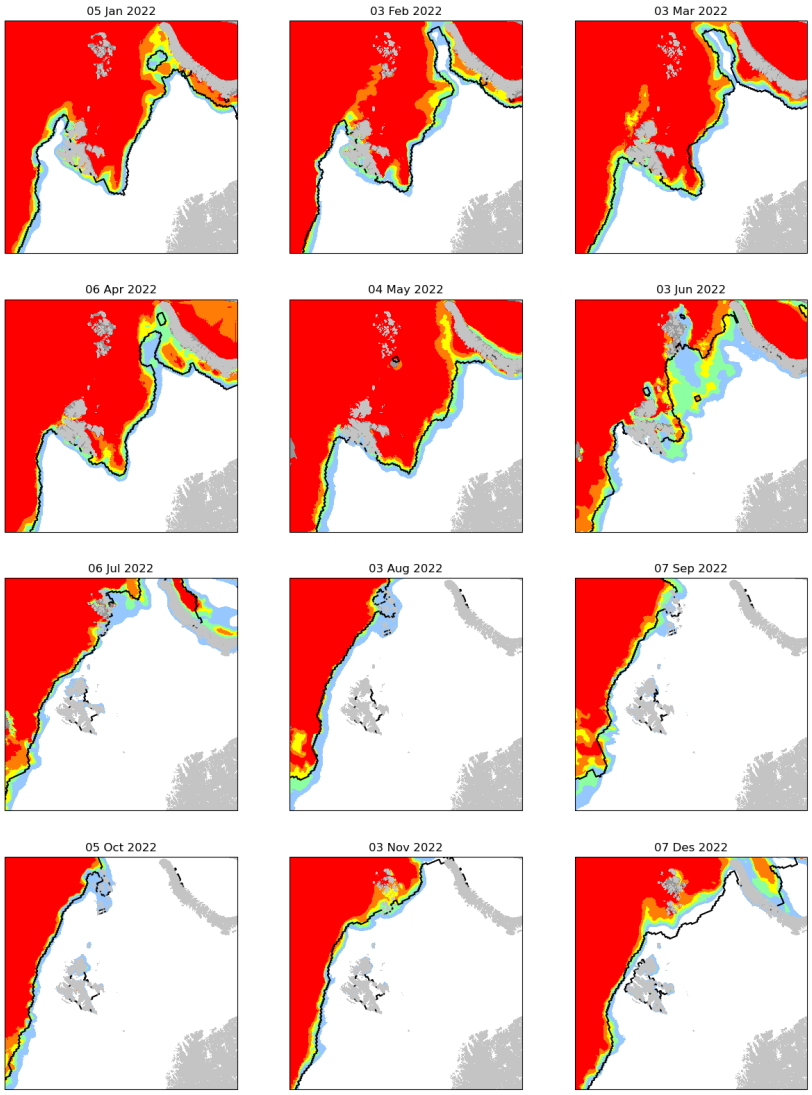


Figure 23: Prediction with a 2 day lead time given the first available ice chart each month of 2022. The black line is the 15% ice edge computed from OSI SAF ssmis. The figure exemplifies the predicitve capabilities of the model. Subplots for each month exemplifies seasonal variability of the model. The WMO color code is preserved, no colorbar is shown.

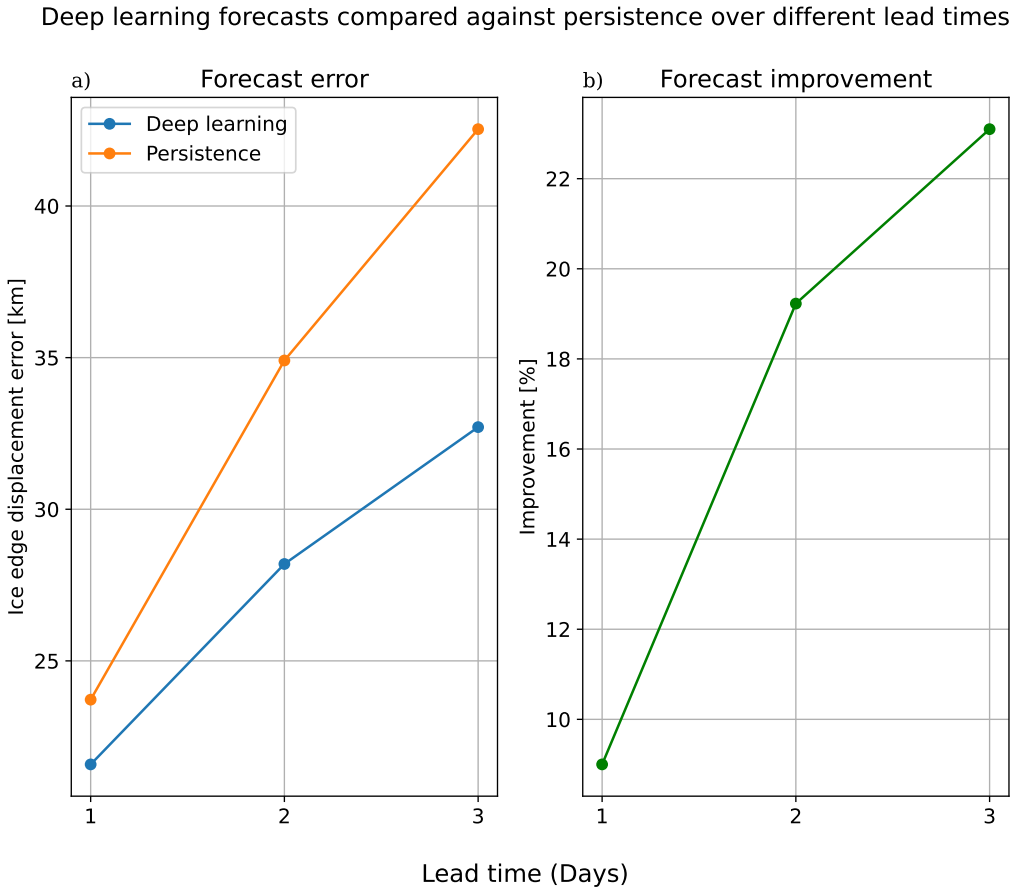


Figure 24: Comparing the effect of training the deep learning system against varying target lead time. The ice edge displacement reported in subfigure a) is the normalized IIEE with regards to the ($> 10\%$) contour. The improvement compared to persistence in subfigure b) is computed in favour of the deep learning forecast.

the lowest score on both monitored metrics.

An inspection of the interannual variability of the NIIEE and Sea Ice Extent of persistence with regards to a two day lead time for the years covered by the full extent of the training data (2016 to 2022) is covered in figure 26. The sea ice extent shown in the topmost figure (a) in figure 26 is computed as the sum of all grid cell areas with category equal to or higher than the targeted contour. No discernable trend across the years can be seen in either (a) or (b) in figure 26

The effect of the non-linear activation function was assessed by training a model were the activation functions were replaced with a linear mapping. The mean annual NIIEE on the test set was 41.35 km, which is 13.15km more than the benchmark model with a mean annual NIIEE of 28.20 km. A qualitative prediction made with the model can be seen in figure 27. Inspecting figure 27 reveals visual artefacts produced by the linear model, not seen in comparable predictions from non linear models. E.g. the sea ice contours east and west of Novaya Semlya exerts a prominent checkerboard-like pattern, which is repeated throughout the scene.

4.3.4 Modifying predictor related hyperparameters

The effect of setting all pixels in the predictor ice chart covered by the land mask as ice-free open water (category 0) and reducing the number of ice chart classes was inspected. When replacing all land covered predictor pixels to ice-free open water, a mean annual NIIEE of 29.72 km was achieved compared to 28.20 km when nearest neighbor interpolation is used.

When reducing the number of possible classes in the ice chart, two contours were modified. The ($< 10\%$) sea ice concentration contour was set to ice-free open water, whereas the (100%, land fast ice) contour was set to the class below ($< 100\%$, very close drift ice). The model was still trained similarly to other models, and achieved a mean annual NIIEE of 28.61 km compared to 28.20 km when all contours are predicted.

4.3.5 Connecting validation loss with NIIEE

Section 4.2.7 described how model selection is performed, where the model that performs best on the validation dataset (2021) with regards to minimizing the loss is selected. This subsection presents a result where the IIEE displacement error (Goessling et al., 2016; Melsom et al., 2019) with respect to the ($> 10\%$) contour is performed.

The IIEE was normalized with respect to a climatological ice edge length derived from ten years of OSI SAF data as described in section 2.2.4. When iterating through the

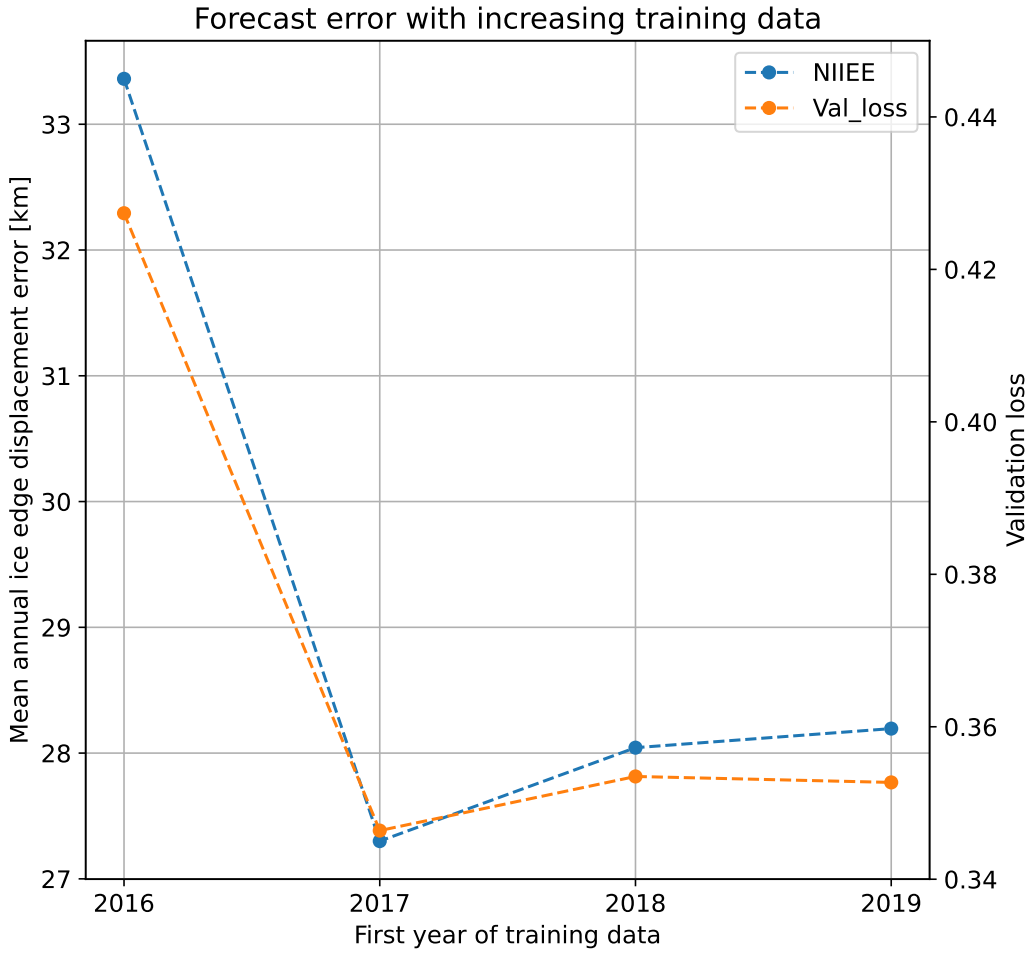


Figure 25: The lines visualize the relationship between start year for the training data (upper bound is always 2020) with the ice edge displacement error for the ($> 10\%$) contour and validation loss.

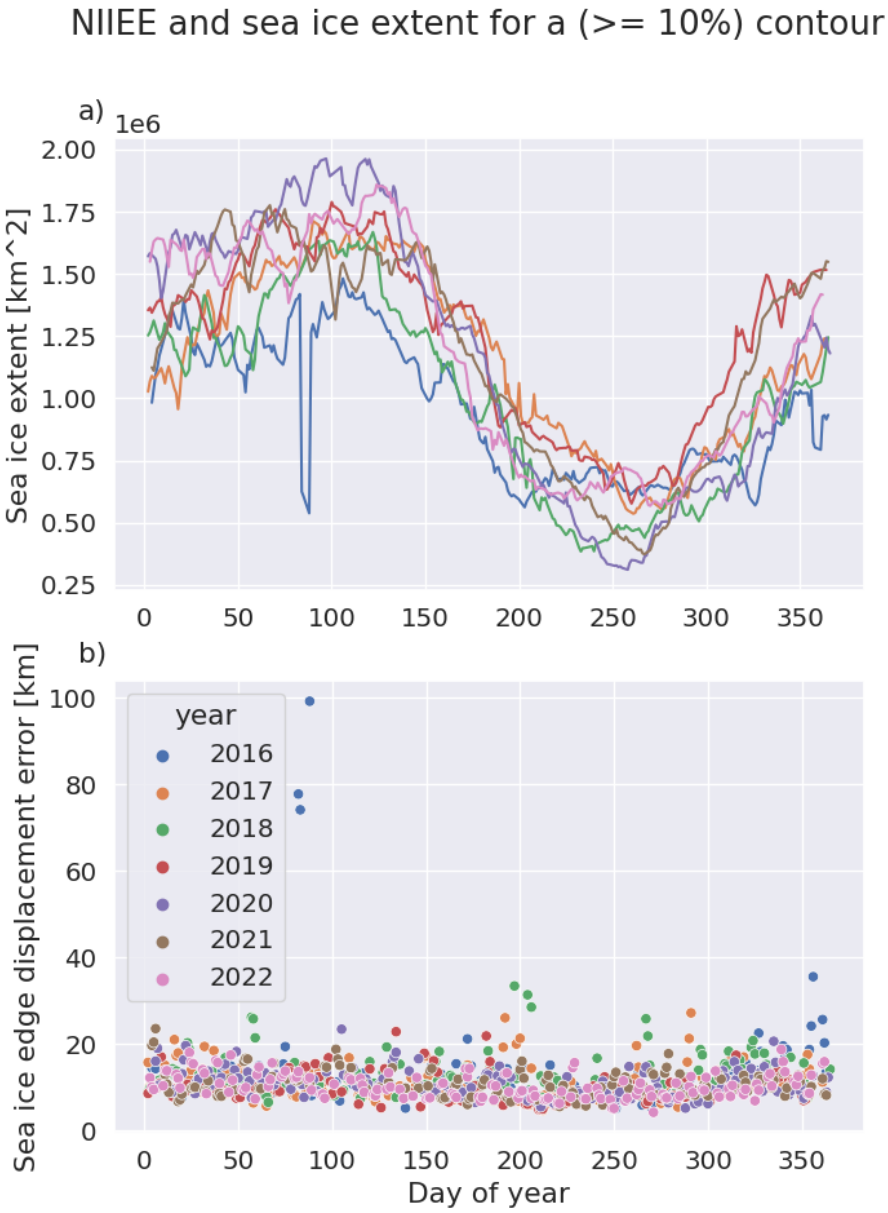


Figure 26: Interannual variability of sea ice extent (a) and NIIEE (b) for persistence with a two day lead time.

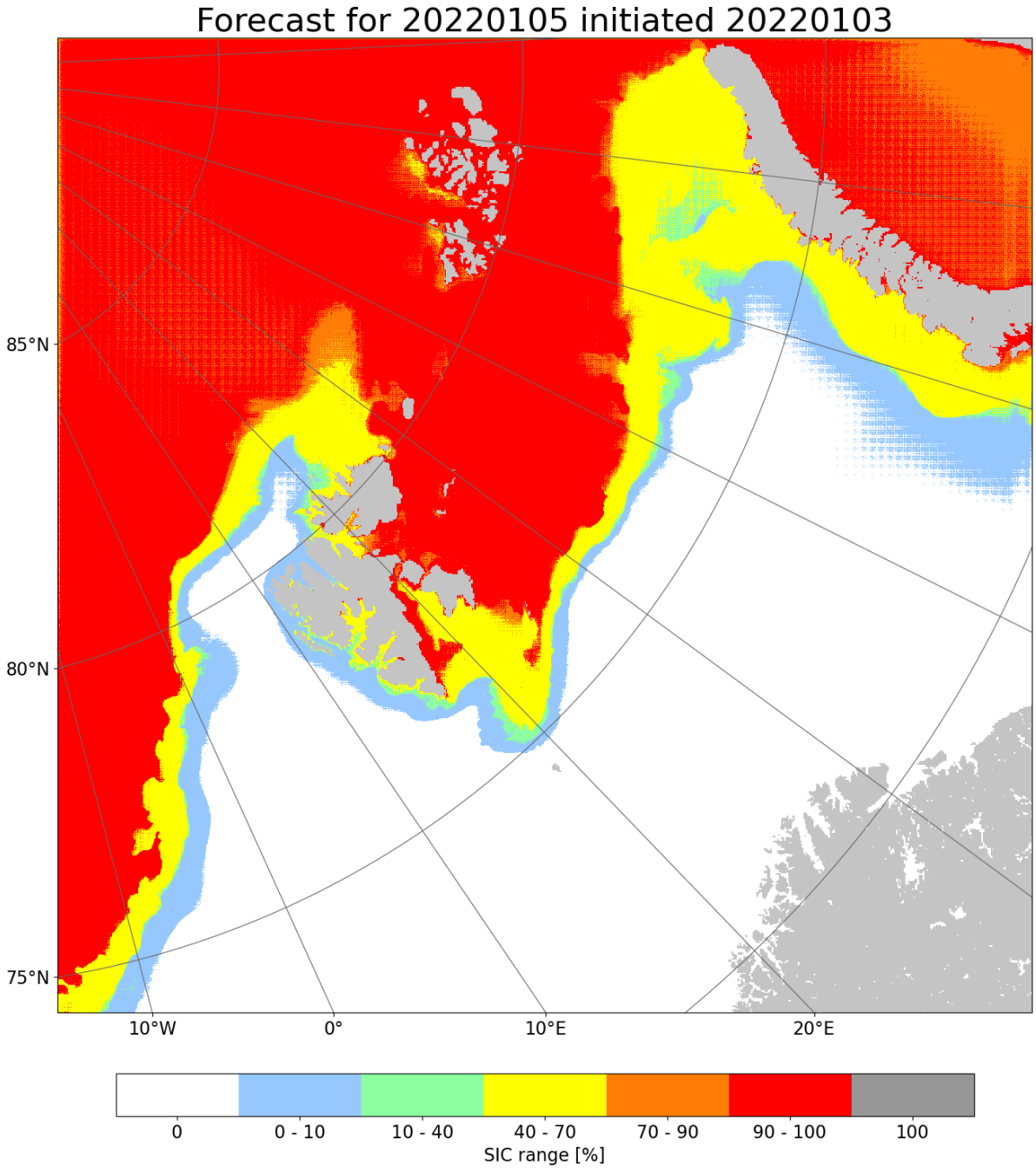


Figure 27: Prediction with a two day lead time model, where all non-linear activation functions were replaced with linear mappings. The figure aim to qualitatively demonstrate a prediction made with a linear model. FIX CATEGORY LABELS

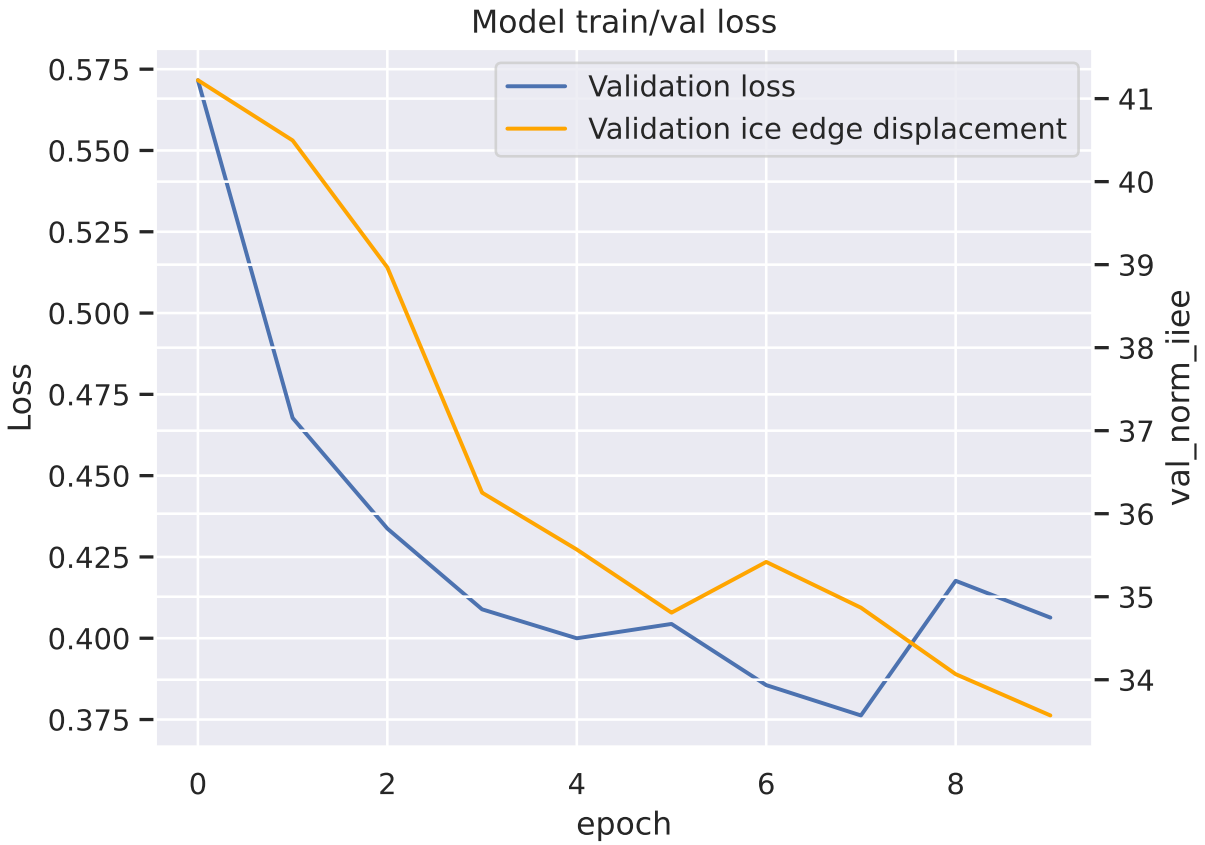


Figure 28: Validation loss and validation normalized ice edge displacement as a function of epoch. A training environment with a 2 day lead time was used.

validational dataset after an epoch is completed, all validational predictions are used to compute the IIEE with respect to their associated ground truth label. The results are summarized in figure 28

The correlation between the validation loss and validation normalized ice edge displacement reported in figure 28 is 0.82. Moreover, training the model for 10 epochs with the previously described IIEE validation scheme took 21 hours. A single IIEE validation iteration took 2 hours. Therefore, we decided to use the binary cross-entropy as the validation metric for model selection during training.

5 Model performance

The following section intends to explore the performance and capabilities of the deep learning system. Where the previous section 4.3 assessed the intra-training model performance, the current section will compare a benchmark deep learning model against baselines and physical models. The physical models have been previously described in section 2, and the baselines (although previously mentioned and to some extent utilized) will be derived in the following subsection. This section will first assess model performance against persistence. Afterwards, the deep learning system will be compared against other physical models. Setup and considerations will be described as they become relevant.

5.1 Baselines

Two types of baselines are considered, persistence and a linear trend. A persistence forecast is constant in time. Regardless of the forecast lead time the initial values for all grid cells are kept constant. Moreover, the autocorrelation of sea ice concentration from the sea ice charts was shown in section 2.2.1 to be high for short lead times. To determine if a forecast has predictive skill, the forecast has to achieve a lower NIIEE than persistence, which is a similar approach as employed in Zampieri et al. (2019). We believe that using this threshold as the definition of a skillful forecast preserves the intent of validating the sea ice forecast in a manner relevant for maritime end users (Melsom et al., 2019; Veland et al., 2021).

The second baseline uses the linear trend, as described in section 2.2.3 and used as predictor for the deep learning system 4.1.3. However, the computed linear trend will be applied pixelwise to advance the initial state forward in time to a given lead time. As the linear trend is computed from OSI SAF ssmis observations, it will consequently be applied to the same dataset. For clarity, the linear trend forecast is computed on the 1km AROME Arctic grid, and the computed values are clipped to match the valid value range, i.e. values $< 0 \rightarrow$ values $= 0 \wedge$ values $> 100 \rightarrow$ values $= 100$.

5.2 Verifying performance against persistence

For this section, a model representing a benchmark with a depth of 256 channels in the final feature map, learning rate = 0.001 and all predictor variables have been used. Only the core training dataset was used for training(2019 and 2020).

The seasonal distribution of average ice edge displacement for all sea ice categories found in the sea ice charts are shown for the deep learning system and persistence are displayed in

figure 29. Figure 29 demonstrates the predictive performance for the deep learning system measured at each resolved contour. In figure 29 b), c), d) and e), the deep learning system achieves a lower median 25-th and 75-th percentile than persistence.

Figures 30 and 31 shows the model confidence as an annual mean for all output contours (figure 30) and the ($> 10\%$) contour distributed seasonally (figure 31). The confidence values shown are output pixel values after the sigmoid (equation 7), such that values closer than 1 are pixels that the model is more confident to belong in the outputted contour. Likewise, values closer to 0 are confident not to belong to the targeted contour.

Figure 30 show that all cumulative contours, except $= 100\%$, have a similar confidence pattern similar to the seasonal cycle of sea ice concentration. Which is expected since for all contours at all dates the leftmost sea ice concentration is always present, whereas the less confident areas east of Svalbard exert spatial variability through mobility. However, it is noted that the $= 100\%$ contour (figure 30 (f)) show a lower overall confidence level, as well as only being restricted to the land structures present in the scene.

The seasonal confidence cycle for the $= 100\%$ contour is shown in figure 31. It is seen that the spatial distribution of confidence tend cover the land-covered pixels for all seasons, although with varying levels of confidence.

The monthly mean sea ice edge length is shown in figure 32. From figure 32, the predicted ice edge follow a similar seasonal pattern to the ice edge length from the target ice charts. Each monthly mean predicted sea ice edge length is biased towards shorter lengths, with the annual mean bias for 2022 being -2146km.

Moreover, the monthly distribution of the different sea ice categories is shown in figure 33. The figure show that the deep learning resolve the area of each contour with a similar scale and variability as the target sea ice charts.

5.3 Inter-product comparison

This section covers results regarding the multi-product comparison. First, the preparation of samples as well as setup of the comparison environment is described. The physical models considered for this comparison are neXtSIM (Williams et al., 2021) presented in section 2.3.2 and Barents-2.5 (Röhrs et al., 2022) presented in section 2.3.3, whereas the considered baselines are persistence and the linear sea ice concentration trend described in section 5.1. Two different products are used as targets. The first product are the sea ice charts, which will be utilized similarly as when comparing only against persistence in section 5.2. Secondly, the independent AMSR2 observations produced by Spreen et al. (2008) are also utilized as ground truth targets.

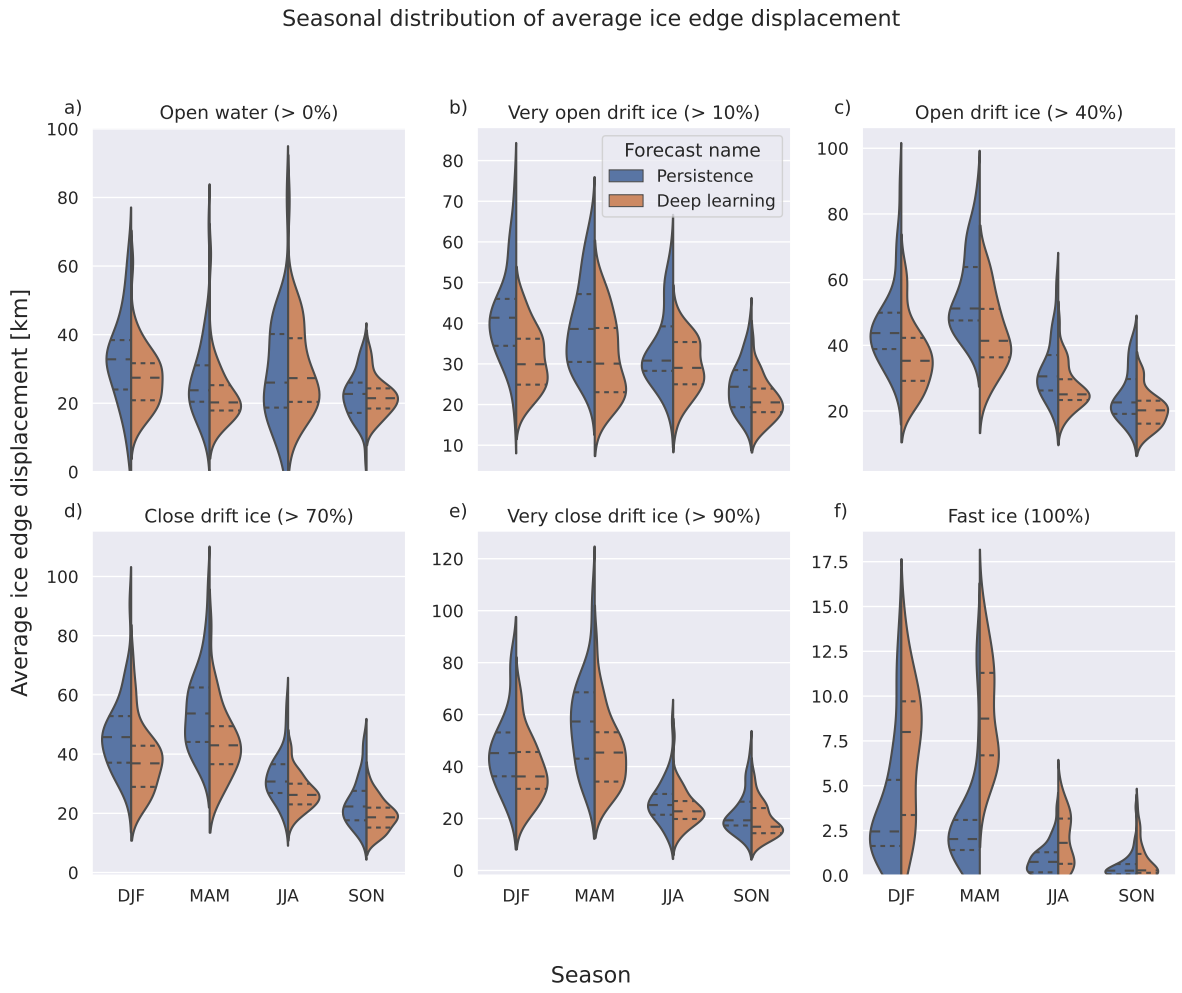


Figure 29: Seasonal distribution of the mean ice edge displacement (Normalized IIEE) for the different sea ice chart categories in the form of cumulative contours. The related sea ice concentration range for each contour is also included. The lower and upper dashed line denote the interquartile range, with the middlemost dashed line showing the distribution median.

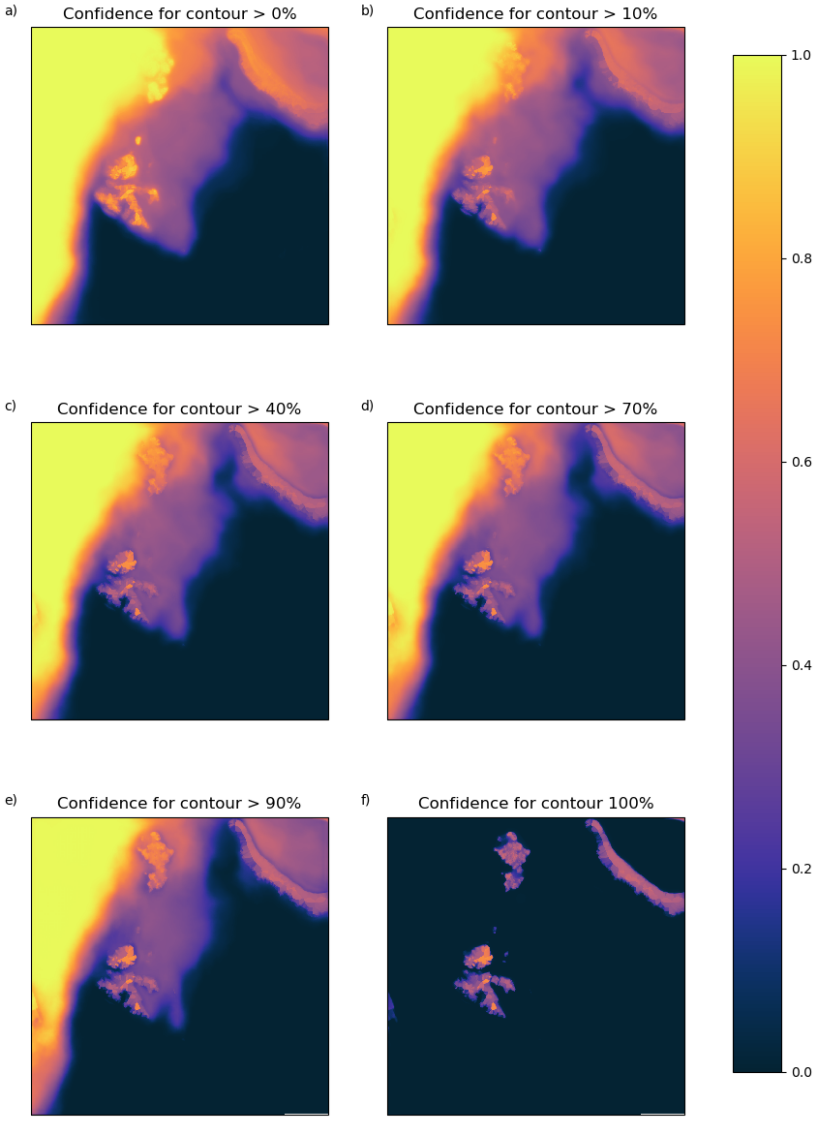


Figure 30: Mean annual probabilities for the different cumulative contours outputted by the model (the class ice free open water is not shown).

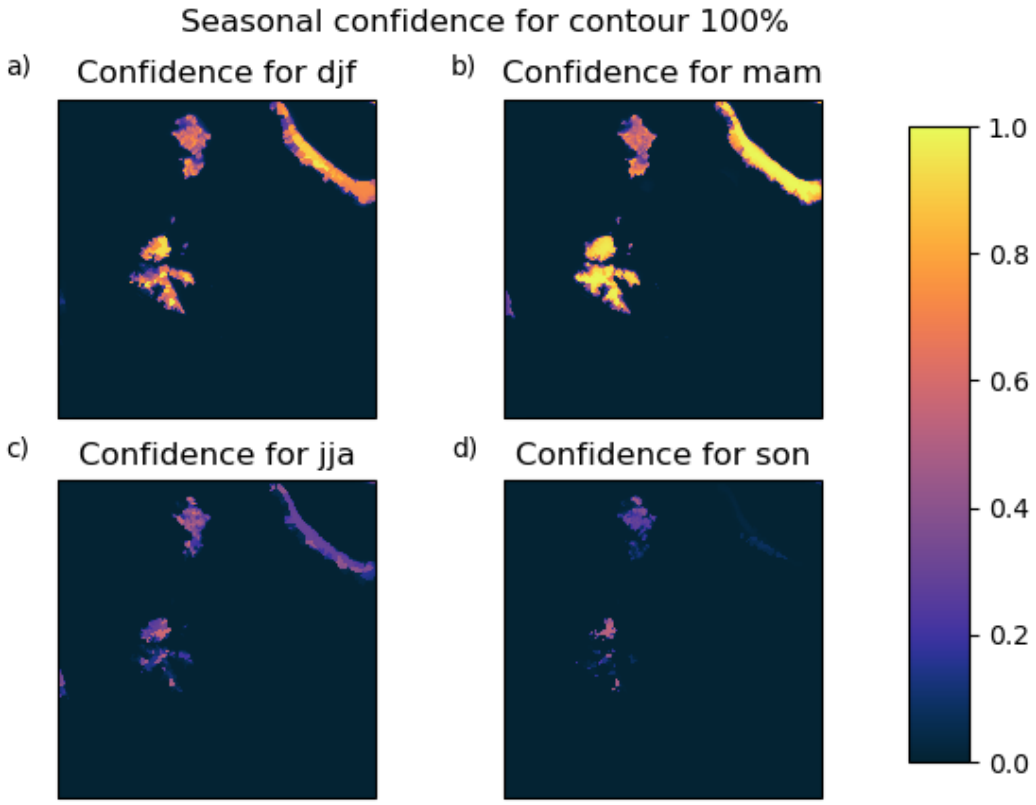


Figure 31: Mean seasonal confidence for the (= 100%) cumulative contour.

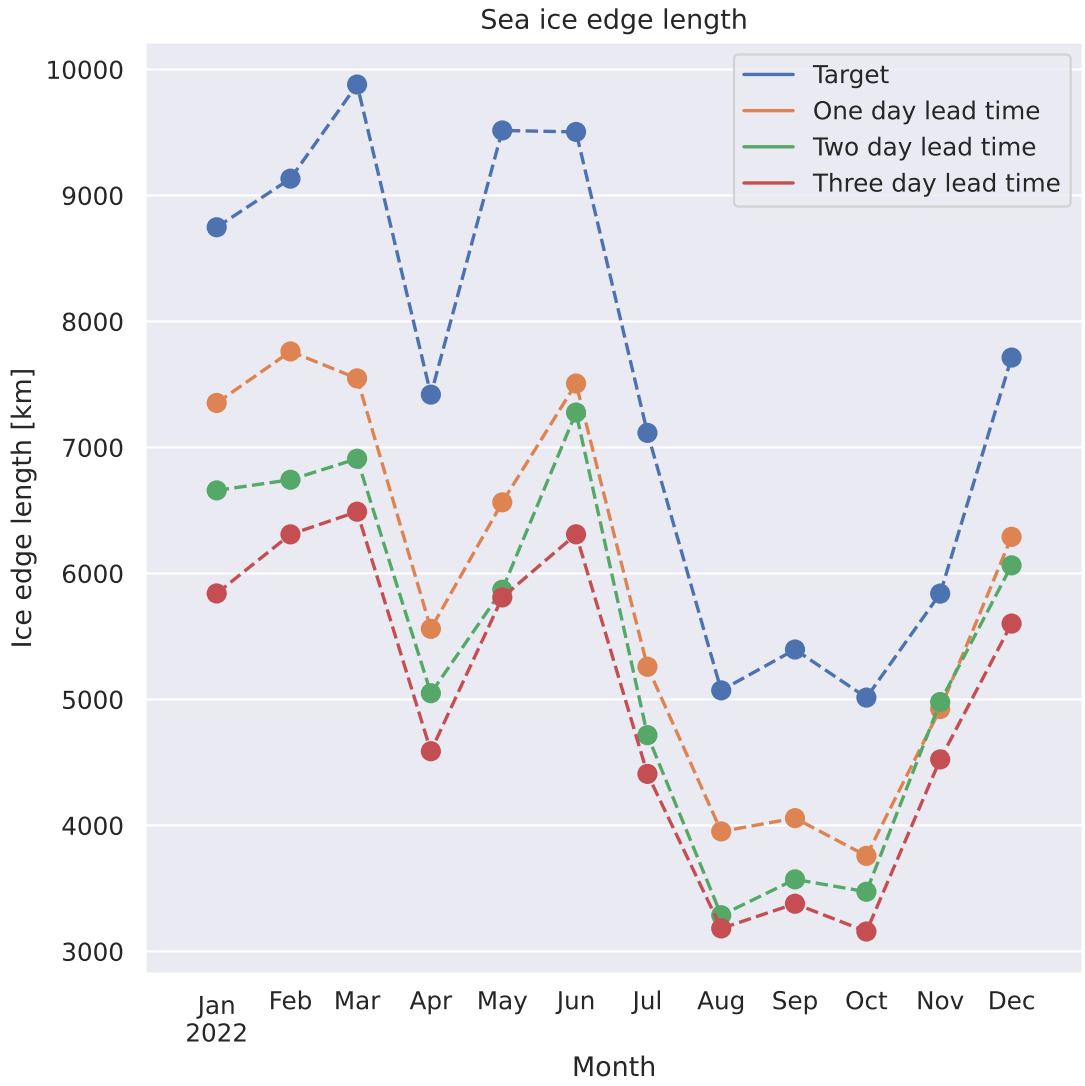


Figure 32: Mean monthly sea ice edge length for the entire 2022 test dataset. The ice edge is defined from a 10% threshold, which results in the ($> 10\%$) contour being used to define the ice edge. Each entry in the defined sea ice edge are on a 1km resolution. Each deep learning marker is annotated with the mean monthly bias with respect to the target sea ice edge length.

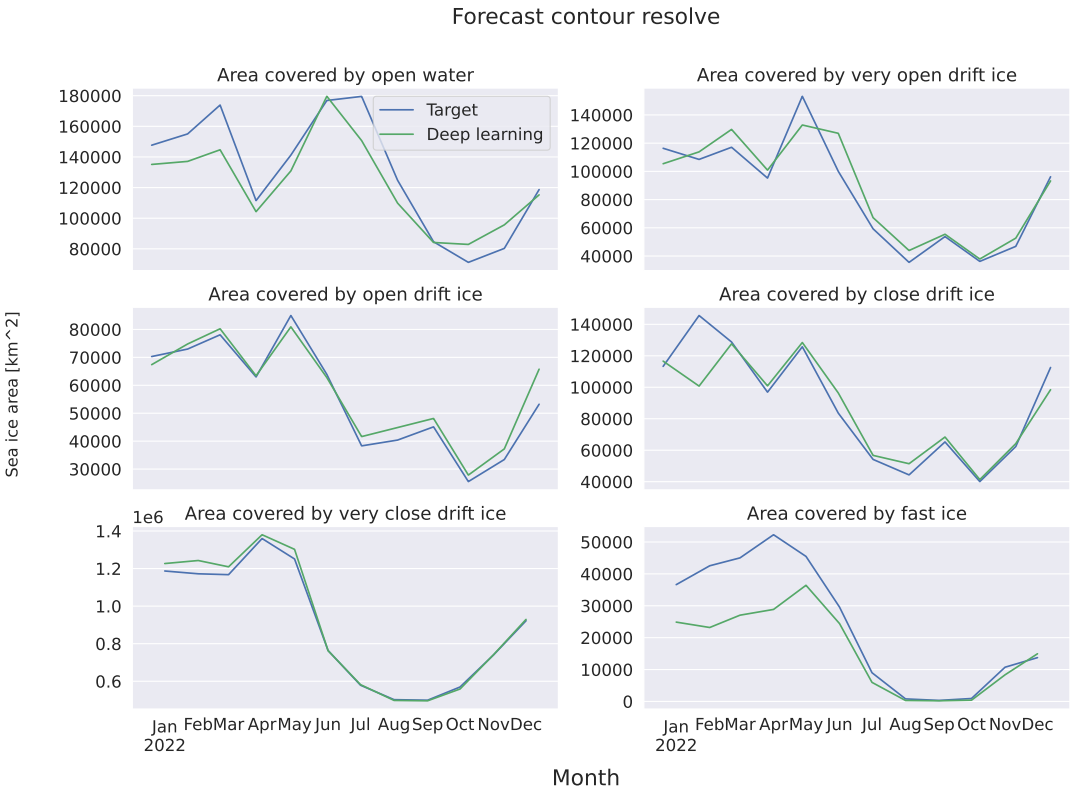


Figure 33: Mean monthly sea ice category distribution for the model and the target sea ice charts for the 2022 test dataset. Each contour is represented by the sea ice area, which is computed from the sum of pixels in each contour times their spatial extent.

When comparing against multiple products, the coarsest resolution model is used as a common spatial resolution. Also, the projection of the coarsest resolution is used for all products, such that other products have to be interpolated onto the grid of the coarsest resolution model, which is done using nearest neighbor interpolation. As both baselines have a daily forecast frequency, comparing either with a deep learning prediction involves identifying the forecast with similar bulletin- and valid date, i.e. initialized at the same day and targeting the same lead time. When utilizing the sea ice charts as the ground truth, the spatial resolution of neXtSIM (3km) is the coarsest, and thus all products are interpolated onto the same resolution.

Comparing against the two physical models requires a consideration of the hourly forecast frequency (Williams et al., 2021; Röhrs et al., 2022) of both model. First, given a published sea ice chart, the comparable physical model is initialized the following day at 00:00 UTC. Furthermore, a daily mean is computed from the 24 steps forward in time taken by the physical model when it covers the valid date of the deep learning forecast. Even though the sea ice charts only convey information about the sea ice concentration up until their publication time, the operational product is considered a reference for the entirety of the publication date. Moreover, to reduce introducing a bias towards the time of day to the physical forecasts as well as limiting the spatial variability induced by the lack of a temporal mean, reducing the physical forecasts to daily averages is considered a more comparable approach than e.g. selecting a single hour (15:00 UTC) from the forecasts.

Since the AMSR2 observations are supplied on a 6.25 km spatial resolution (Spreen et al., 2008), when AMSR2 is used as the ground truth all data is interpolated to match the resolution of AMRS2. Although the AMSR2 data have a substantially coarser spatial resolution compared to the sea ice charts or the deep learning system, the data makes it possible to assess the generalizability of the deep learning performance when targeting an unseen ground truth.

From this setup, the mean of the first 24 hours of a forecast from a physical model is compared against a deep learning prediction with one day lead time, the mean between 24 and 48 hours are compared against a deep learning prediction with two day lead and the mean of the third predicted day is compared against a deep learning prediction with three day lead time. Figure 34 summarizes the process. Note that Barents-2.5 only have a 66 hour lead time (Röhrs et al., 2022), thus the mean between $t = 48$ and $t = 66$ is computed when comparing against a three day lead time prediction.

It is noted that when comparing against multiple forecast products as described in figure 34, only the common dates shared between all products are used. With the current setup, where neXtSIM, Persistence, Deep learning, OSI SAF trend and Barents-2.5 are considered, the test dataset is reduced from 196 to 171 samples, 147 to 130 samples and

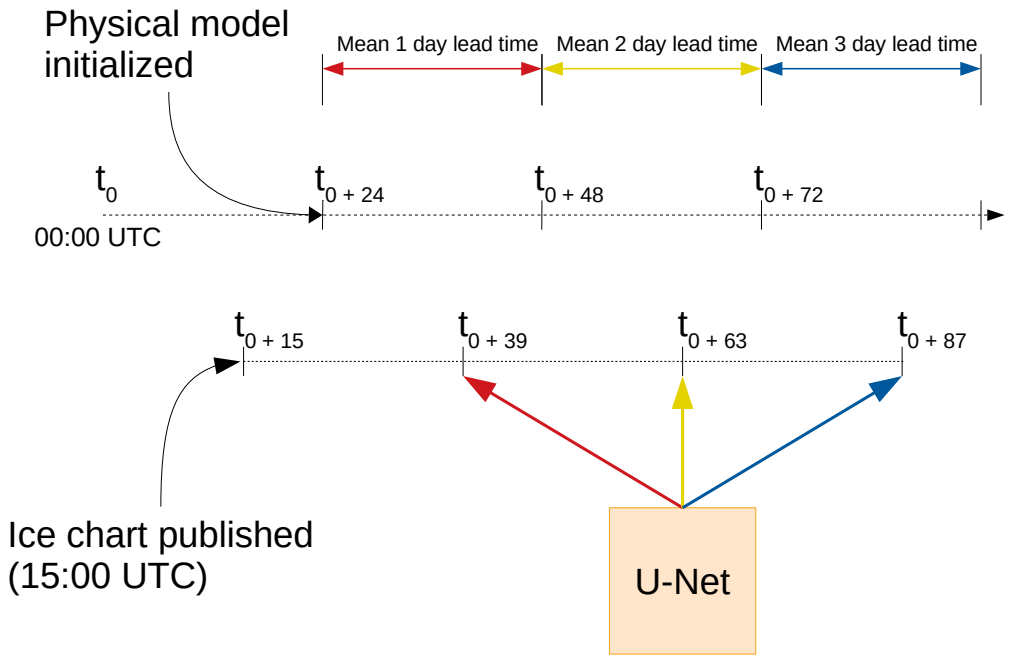


Figure 34: Overview describing how a physical model with an hourly frequency is compared against a deep learning forecast. Timestamps are hourly, and relative to 00:00 UTC the day a sea ice chart is published. The physical model is initialized the following day. Colors are used to denote lead time comparability, with red = 1, yellow = 2 and green = 3 day lead time.

142 to 125 samples for 1, 2, and 3 day lead time respectively. Moreover, Barents-2.5 is only considered starting with the month of June, to comply with the spin up time of its data assimilation system (Röhrs et al., 2022).

Figure 35 shows the seasonal distribution of NIIEE for the different forecast systems and benchmarks, following the setup described in figure 34. By inspecting figure 35, it can be seen that only the products based on the sea ice chart are able to achieve consistently low NIIEE for the $> 0\%$ contour. Furthermore, for the $\geq (10, 40, 70, 90)\%$ contours, the deep learning system achieves the lowest median and mean values compared to all the other products. It can also be seen that neXtSIM tend to increase its mean and median as well as spread for increasing contours, with a similar although not as consistent pattern for Barents-2.5. Moreover, the OSI SAF trend typically have the highest (visible) outliers. Finally, no product is able to achieve a lower mean or median NIIEE compared to persistence when inspecting the 100% (fast ice) contour.

The fraction of days where Deep learning achieves lower NIIEE compared to each considered product is shown in Figure 36. The figure shows that the deep learning system consistently achieves a $\geq 50\%$ success rate compared to all products, except for Persistence 1 day lead time in July, August and September as well as Barents-2.5 2 day lead time in November and December. When compared to neXtSIM at 1 day lead time (figure 36 (a)), the Deep learning system achieves a lower NIIEE at all considered dates in the test data. However, it can also be seen that a lower amount of days with lower NIIEE than neXtSIM are achieved as the lead time increases. The same pattern may also be seen in the Barents data as the mean fraction of days with lower NIIEE for the Deep learning system also decrease with lead time, although Barents is only able to achieve lower NIIEE more than 50% of the dates for a 2 day lead time as previously noted. With respect to persistence, the Deep learning forecasts seem to achieve a higher fraction of days with lower NIIEE as lead time increases, although there is no trend for the individual months. At the ($\geq 10\%$) contour, the OSI SAF trend is consistently beat by the deep learning system during Winter and Spring, with less consistency observed during the Summer and Autumn seasons.

The spatial distribution of product error is shown in Figure 37. From the figure, it can be seen that both sea ice chart products have lower bias than the three other products, as well as only exerting biases in the MIZ. Moreover, it can be seen from the top row in figure 37 the neXtSIM data have a negative bias along the sea ice edge, which is prominent during Winter and Spring. Moreover, the OSI SAF trend seem to have a strong negative bias along a wide sea ice edge. Finally, Barents seem to have a positive bias around Svalbard in the Summer, with a less prominent overall bias during the Fall.

The seasonal NIIEE distributions shown in figure 38 is created similarly as figure 35, but with AMSR2 as the ground truth data, which also implies that all data have been in-

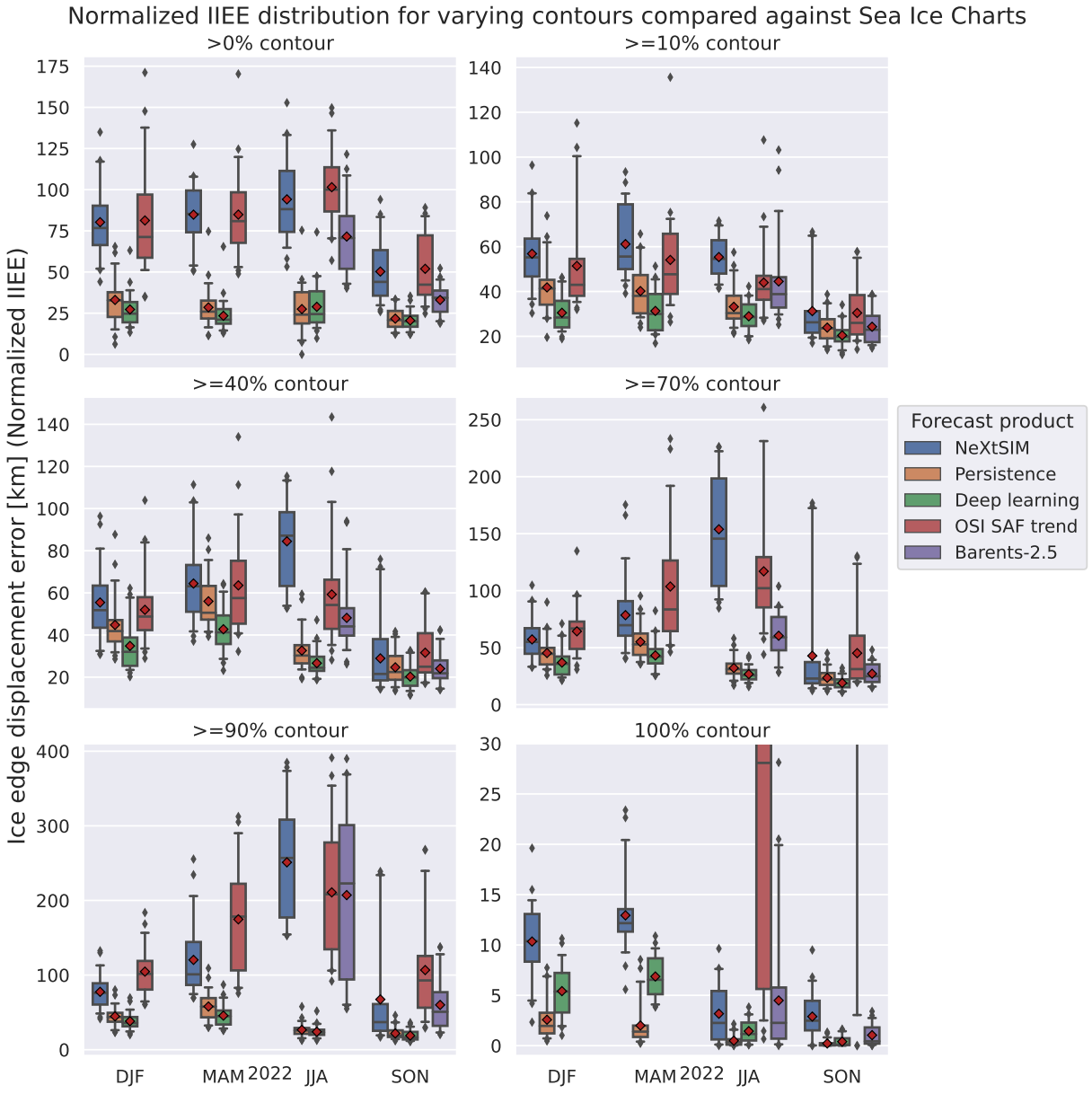


Figure 35: Model intercomparisson with a two day lead time. The boxes are constructed from seasonally distributed NIIEE values computed from the test dataset (2022). The sea ice charts are considered as targets. Each box cover the interquartile range (25th - 75th percentile), with whiskers covering the 5th and 95th percentile. The line in each box is the median, and the red diamond is the mean. The IIEE is normalized according to the climatological sea ice edge at the forecast valid date. The extent of the y axis is limited in such a way that the distributions are easily readable, at the expense of some outliers not being visible. The OSI SAF trend is computed from the past 7 days.

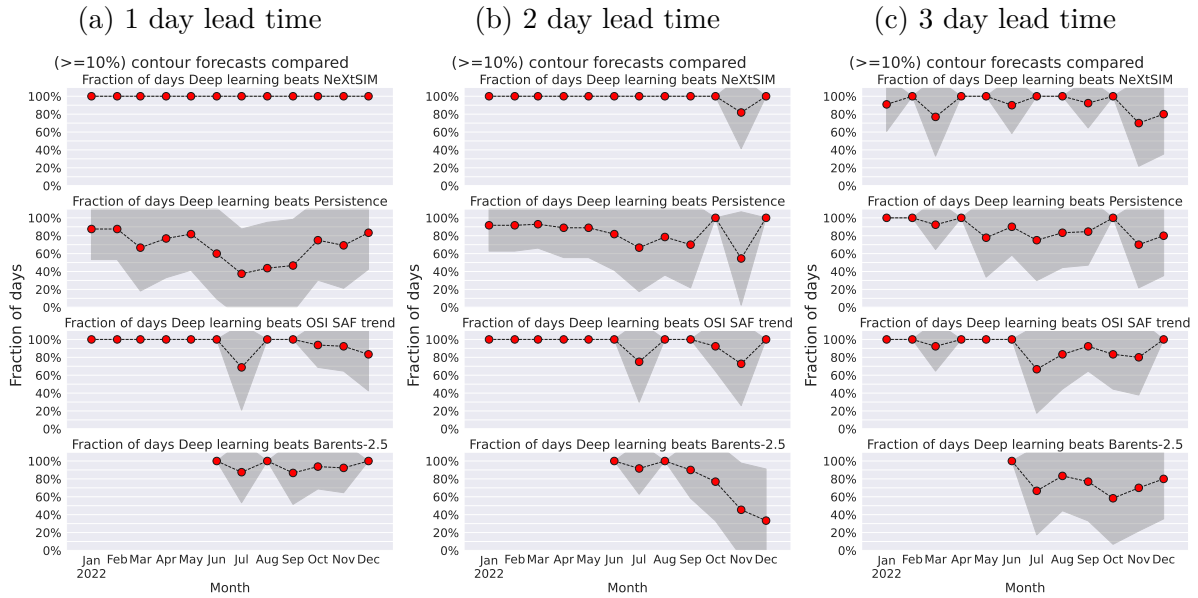


Figure 36: Fraction of days where the Deep learning forecast achieves a lower NIIEE than the compared product, distributed monthly for all lead times. Only the ($\geq 10\%$) contour has been considered, due to the relevance of the contour with respect to the definition of the sea ice edge and its application to operational end users. Gray contours denote the uncertainty (standard deviation) for each month. The sea ice chart has been used as ground truth target when computing the IIEE, and the score has been normalized according to the climatological sea ice edge.

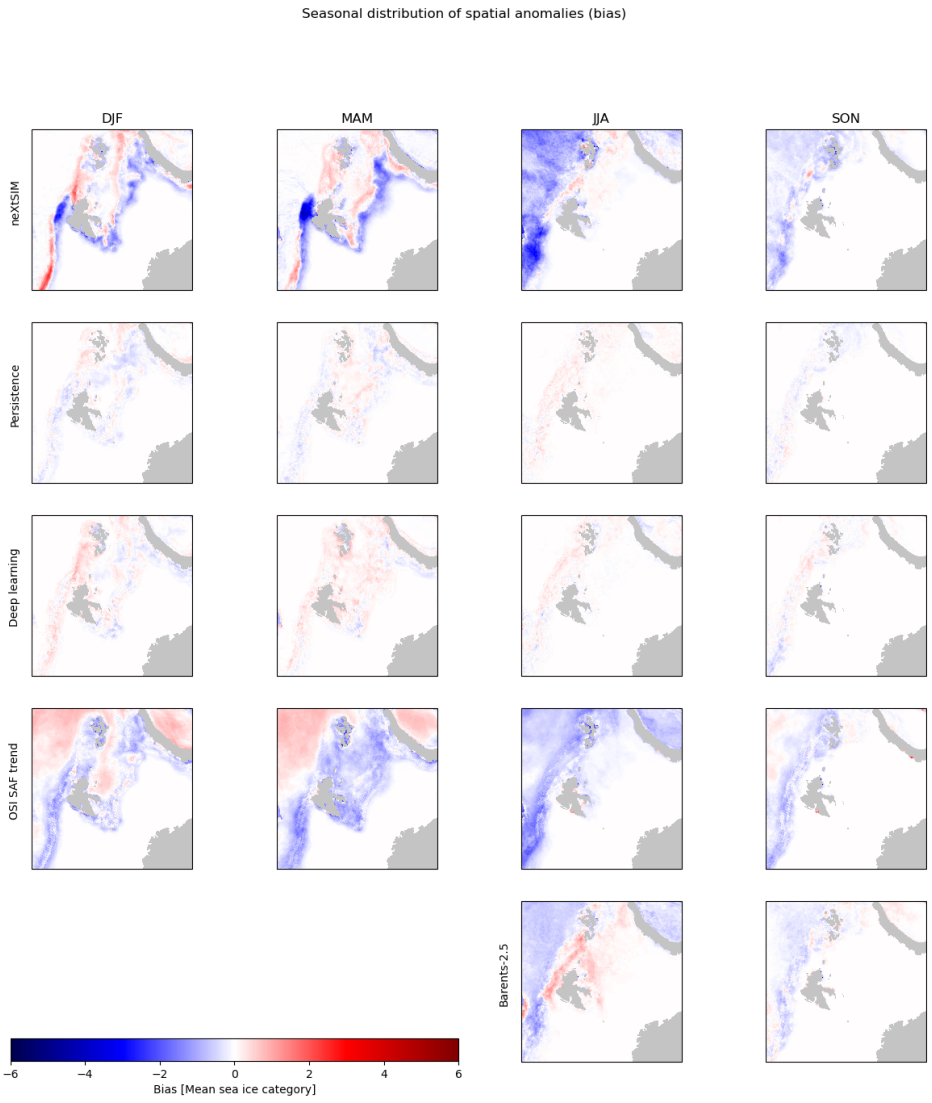


Figure 37: Spatial distribution of the mean seasonal error in predicted sea ice categories between the compared products. The data is interpolated onto the neXtSIM grid, and the test dataset is considered.

terpolated onto the 6.25km AMSR2 grid (Spreen et al., 2008). Contrary to what was observed in figure 35, the sea ice chart based products in figure 38 exert significantly higher NIIEE at the $> 0\%$ contour. However, Barents-2.5 also exert a similar increased NIIEE as the sea ice chart based products at the same contour. Moreover, the sea ice chart based models are within the interquartile range of neXtSIM and OSI SAF trend starting at the ($\geq 10\%$) contour. At the 0 and 10% contours, the OSI SAF trend exerts the lowest mean and median NIIEE for all months except SON where neXtSIM achieves the lowest median and mean. However, starting at the ($\geq 40\%$) the deep learning system has the lowest median and mean NIIEE, which lasts until the 100% contour where performance is comparable between all products except for the OSI SAF trend during DJF and MAM.

Following the result seen in the upper leftmost distribution in Figure 38, Figure 39 shows a comparable figure but with a deep learning model which does not predict the 10% and 100% contours as described in section 4.3.4. By inspecting the >0% contour, it can be seen that the deep learning system achieves significantly lower NIIEE than persistence, as well as the deep learning system in figure 38. Otherwise for the other contours, the performance of the deep learning system is comparable to the deep learning system in figure 38.

The boxplots in Figure 40 computes the >0% contour NIIEE against AMSR2 with the model used in Figure 38 but with the predicted >0% contour removed. The distribution seen in the figure resembles that in Figure 39, with the Deep learning forecasts performing significantly better than persistence.

Kanskje dette også skal i appendix?

6 Model explainability and physical connections

This section aims at presenting results intended to explain aspects of the deep learning system, such as predictor importance, how the model responds to the predictors as well as understanding how decisions where made by the deep learning system. This will be divided into three subsections. Firstly, model response to modified predictors will be measured. Secondly, individual predictions will be used to highlight how the model interprets the input data for decision-making. Thirdly, a case study will be conducted with the intent to relate the previous explainability results to a real world and in-distribution scenario.

6.1 Predictor importance

To measure the impact of each predictor, an experiment was conducted where the deep learning system was fitted to subset of the predictors, and the Normalized IIEE with

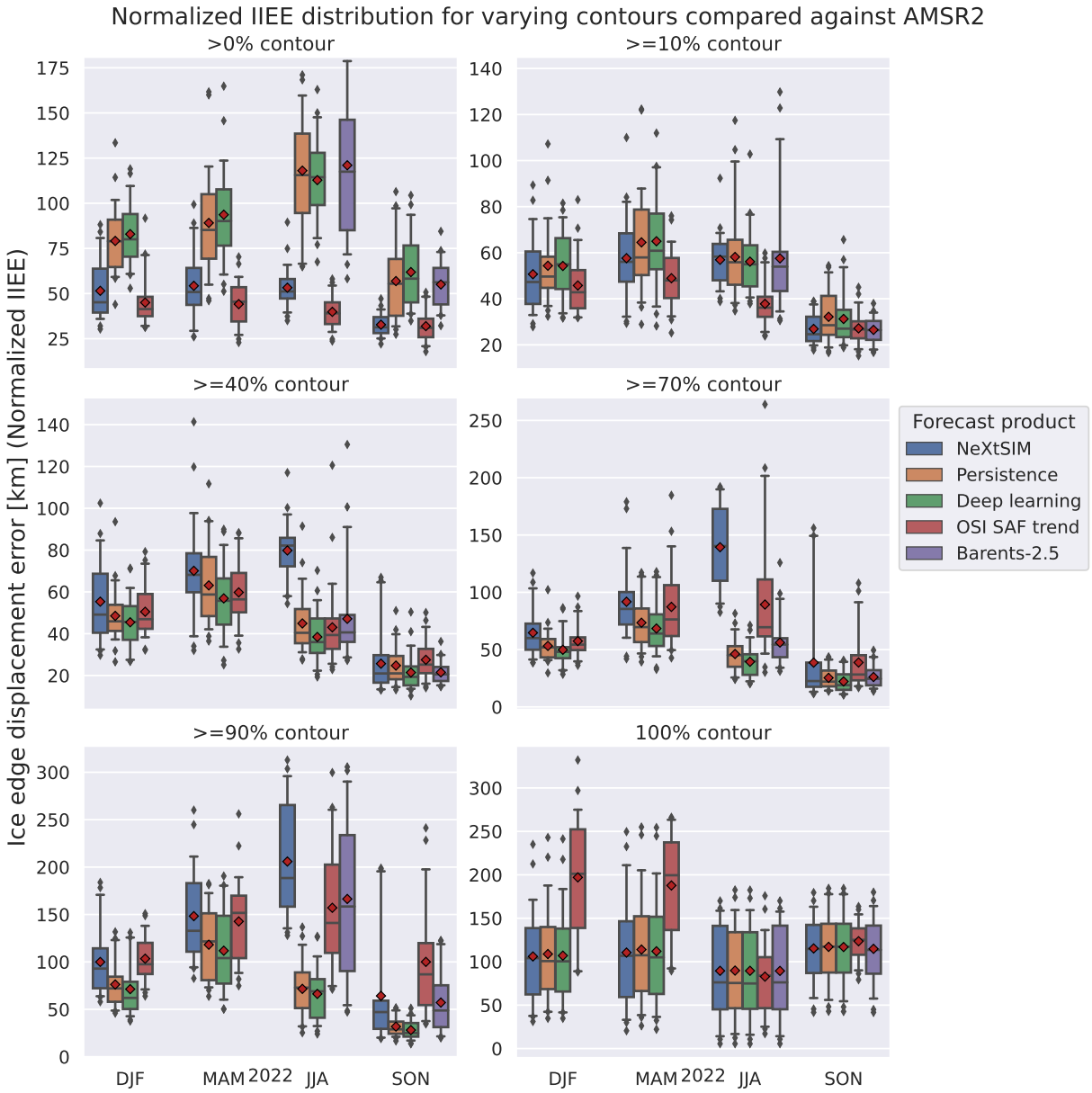


Figure 38: Same as figure 35, but with AMSR2 sea ice concentration as the target ground truth data.

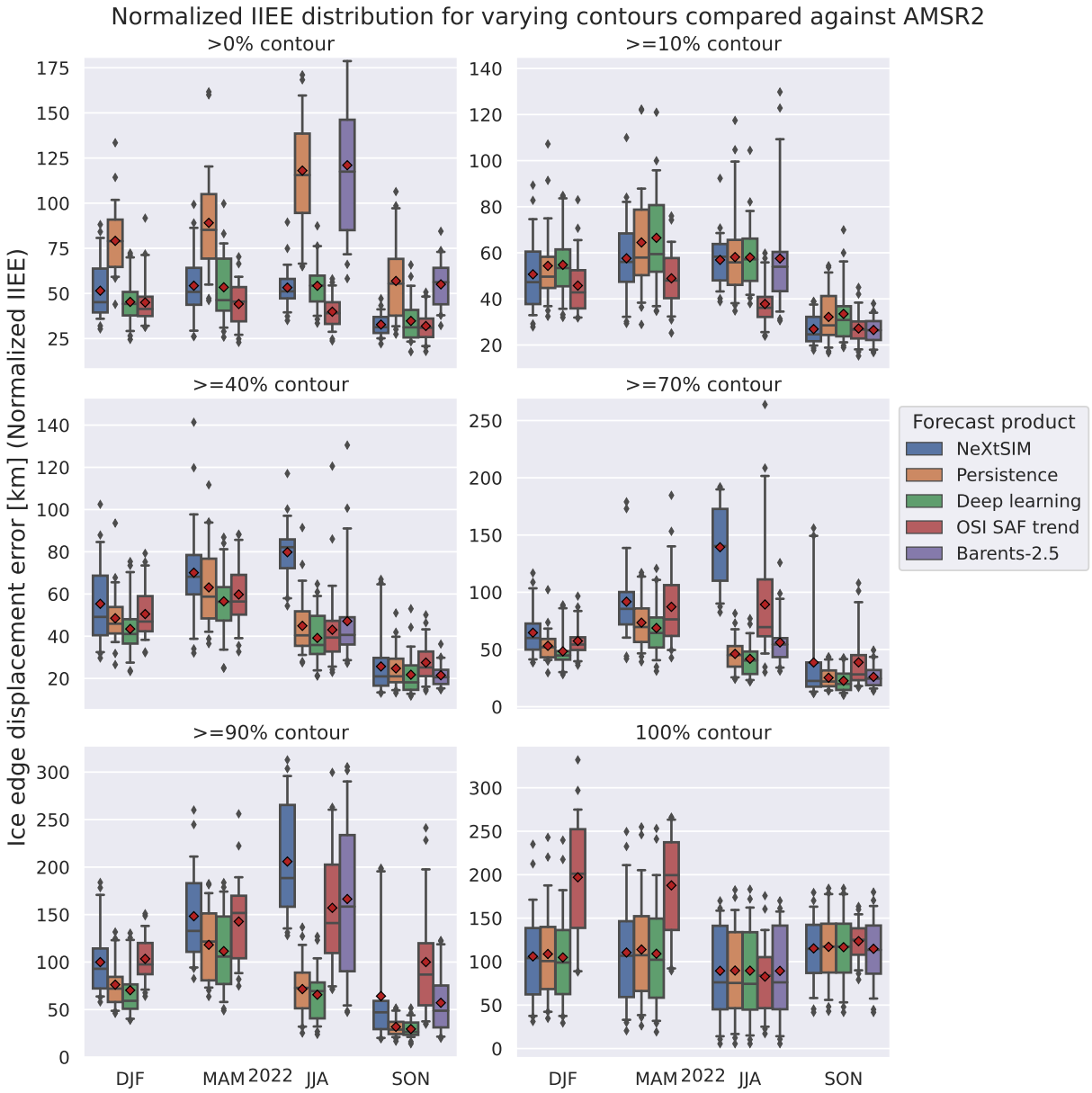


Figure 39: Same as figure 38, but the deep learning system used has reduced output classes.

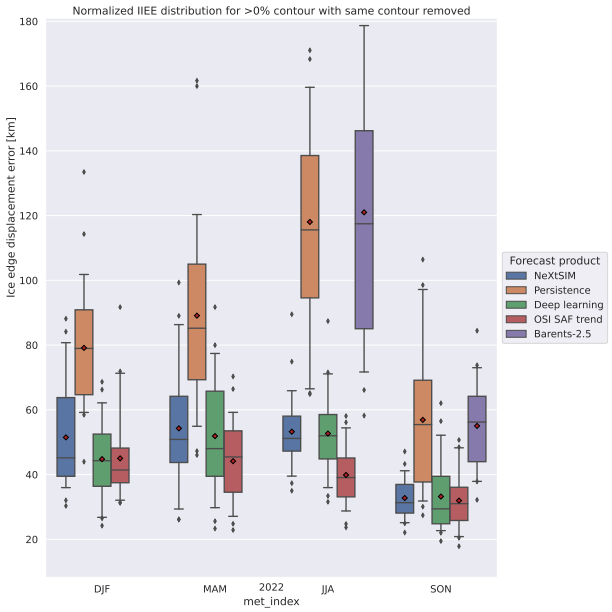


Figure 40: NIIEE for the >0% contour with the model from Figure 38, but with the values in the >0% contour set to category 0 (ice free open water)

respect to the ($\geq 10\%$) contour was measured as a yearly mean value. The result of the experiment can be seen in Figure 41, where also persistence as well as an all predictor baseline was included for reference. From figure 41, it can be seen that removing the recent sea ice chart as a predictor causes the deep learning system to achieve the highest NIIEE. Moreover, only when removing all AROME Arctic atmospheric predictors does the deep learning system perform worse than persistence for the 2022 test data period. Finally, removing t2m seem to stochastically improve the forecast beyond the baseline with all predictors, as further indicated by training a new model without t2m which achieves similar performance (orange line).

Moreover, two more experiments where conducted to explore how sensitive a model fitted to all parameters are to perturbations in the input data. Firstly, it is noted that the model is trained on unaltered training and validation data, only the test dataset is permuted. The first experiment involves swapping all predictors with uniform noise between 0 and 1, which is the value range for the predictors after normalization (see section 4.1.6). The purpose of replacing a predictor with a uniform noise field is to study how much it alters the prediction of the model, i.e. if the model is strongly fitted to the predictor. The result of the first experiment is seen in figure 42, and although not shown permuting SIC results in the values [383, 439, 870, 851] following the same seasonal sequence as figure 42.

Inspecting figure 42 reveals that permuting SIC results in the highest NIIEE seasonal means. Following SIC, when swapping temperature out with random uniform noise, the

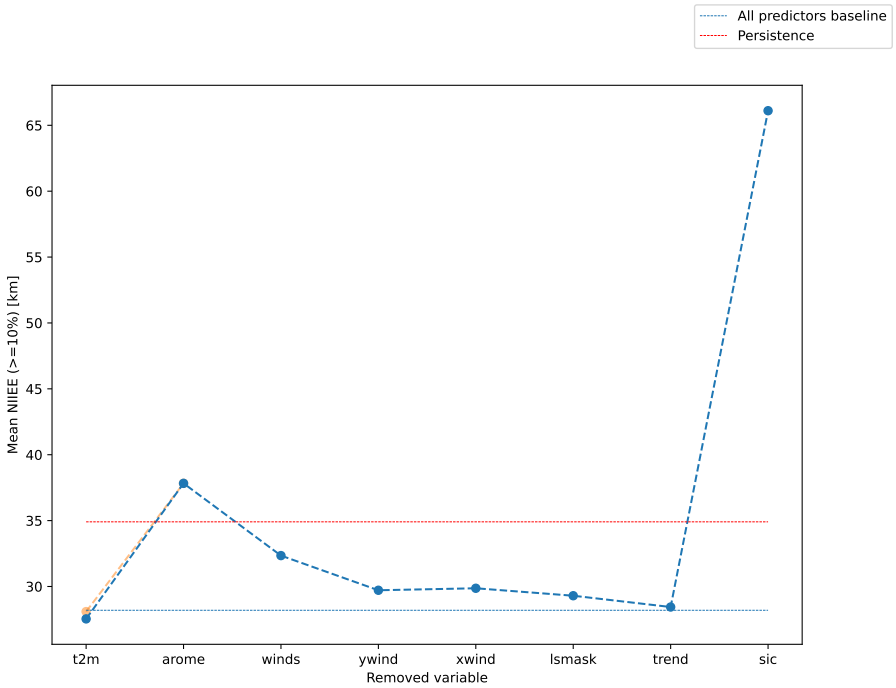


Figure 41: Yearly mean NIIEE for the ($\geq$ 10%) contour computed for deep learning systems where one (or a group) of the predictors have been removed. The red dashed line denotes the yearly mean NIIEE computed from persistence, whereas the blue dashed line denotes the yearly mean NIIEE for a benchmark deep learning model with all predictors. A two day lead time was considered. The orange mark and line seen for t2m is from an independent rerun. Trend refers to the OSI SAF linear sea ice trend. Winds refer to removing both the x and y component of the wind, whereas arome means removing all AROME Arctic atmospheric predictors.

NIIEE seasonal means are higher then persistence for both JJA and SON.

The second experiment conducted is similarly constructed to the previously described experiment of swapping out predictors with random noise, however instead of replacing each predictor with random noise each predictor have their 2022 test dataset sequence swapped. Thus, the distribution in which samples are drawn from is preserved. To account for seasonality, each prediction on the test data was repeated ten times for each predictor. The results of swapping predictor fields with fields from a different date is seen in figure 43. As with figure 42, SIC is not shown but achieves the values [186 ± 13, 225 ± 29, 266 ± 36, 230 ± 29] distributed in the same seasonal sequence as in figure 43.

From figure 43, it can be seen that swapping t2m is the only predictor which achieves a significantly higher NIIEE than persistence for the months JJA and SON. Moreover, for all seasons, all AROME Arctic predictors cause the model to have significantly higher NIIEE than the non-AROME Arctic predictors for all seasons. Figure 43 also shows that the swapped OSI SAF trend exerts no discernable standard deviation at all seasons.

6.2 Synthetic AROME Artcic fields

Following the predictor importance experiments described previously in section 6.1, this section will describe an experiment which only targets the atmospheric predictors provided by AROME Arctic by constructing synthetic atmospheric fields and measuring how the deep learning system reacts. Firstly, a brief rundown of the synthetically created AROME Arctic fields and the experiment environment will be provided.

The deep learning model chosen is, similar to previous results, a model trained on all predictors with a two day lead time. Four prediction dates covering months from different seasons have been chosen when measuring how the model responds to artificial AROME Arctic data, with the intent to also measure any seasonal variability in the responses. The chosen dates were 03rd March, 03rd June, 07th September and 07th December, with the dates referring to the valid date of the forecasts. Moreover, different synthetic fields were created.

With regards to temperature, four fields were created where temperature increases linearly from one end of the domain to the other, starting at the lowest possible t2m value in the test dataset and ending at the maximum t2m value in the test dataset. Moreover, two homogenous fields containing only minimum or maximum values were created. For the winds, seven different fields were set up. The first field contains no wind in either x or y direction. Additionally, four fields where the wind blew in one direction (x or y, as well as positive or negative) where created. Finally, two fields were the winds were set to their maximum and minimum value in x and y at the same time was initiated.

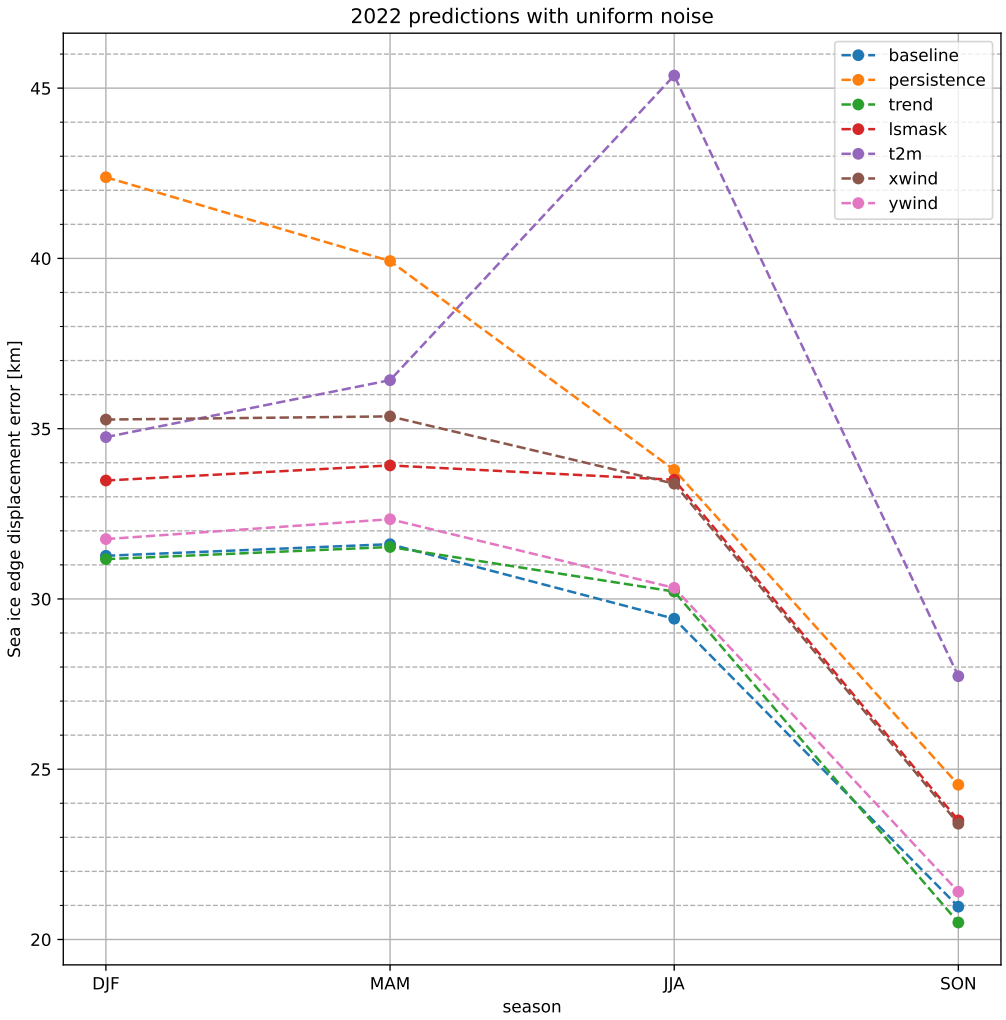


Figure 42: Seasonally distributed mean NIIEE for the ($\geq 10\%$) contour where each of the predictors have been swapped out with uniform noise between 0 and 1. Each colored line represent a variable that has been replaced, however the blue and orange line represent a no permuted baseline and persistence respectively. SIC is not shown, as it achieves the values [383, 439, 870, 851] distributed in the same seasonal sequence as the shown lines.

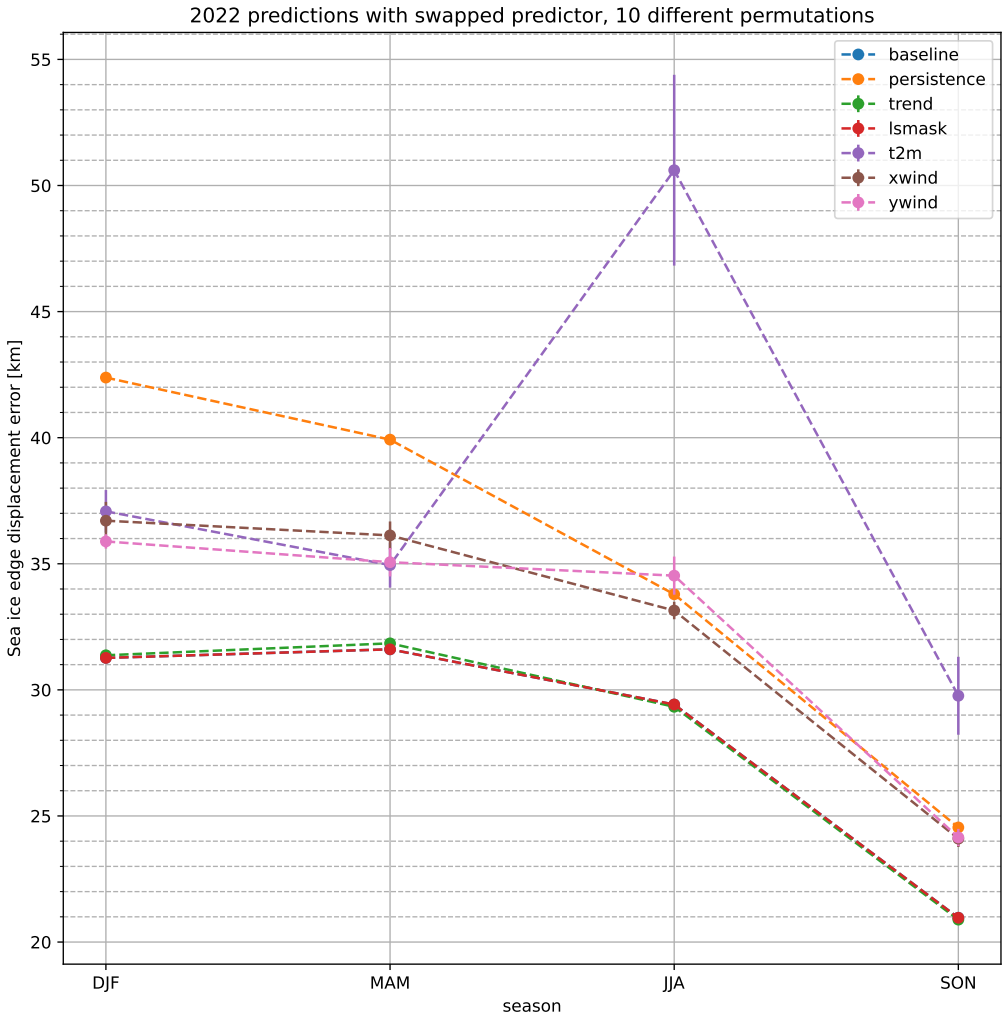


Figure 43: Seasonally distributed mean NIIEE for the ($\geq 10\%$) contour across 10 runs for each predictor, where each predictor have had their fields swapped internally. Each line represents one predictor. SIC is not shown, but it achieves the values [186 ± 13, 225 ± 29, 266 ± 36, 230 ± 29].

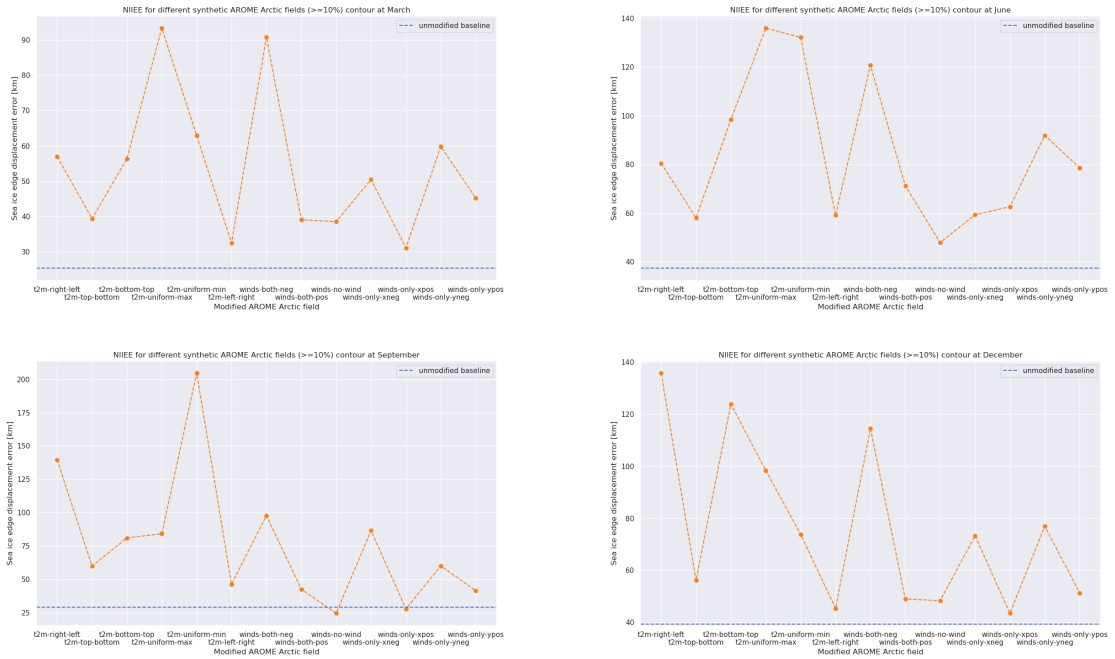


Figure 44: NIIEE across different synthetic AROME Arctic fields measured at the ($\geq$ 10%) contour for different seasons. The deep learning system has a two day lead time. The blue dashed line show the NIIEE value for a baseline model with no synthetic AROME Arctic field. The x axis describe how the different field were created.

Figure 44 show how the model respond to synthetic AROME Arctic fields in terms of the NIIEE computed at the ($\geq$ 10%) contour. From the figure, it can be seen that the model respond differently to the different fields for the inspected seasons. The synthetic fields tend to give the deep learning system higher NIIEE, except for two wind related fields for the september prediction. For all seasons, the highest NIIEE value is achieved with a synthetic temperature field, although having both the x and y winds at a maximum negative direction causes the highest NIIEE for all seasons compared to the other synthetic wind fields.

Spatial errors are shown in figure 45, where some of the predictions made with the synthetic fields have been chosen. The top row of figure 45 show two examples where synthetic wind fields have been constructed, and it is noted that a negative y-wind direction is towards the top of the domain. When both wind fields are pointed in a negative direction with maximum velocity from the test set, the sea ice concentration rend towards lower categories as seen by the negative difference along the sea ice edge. With only x-wind in the positive direction with a maximum test set magnitude, a varied distribution of differences occur along the sea ice edge, with an overall weak signature.

The lowermost row contain two prediction differences made with synthetic t2m. The lower leftmost figure show a consistent increase in sea ice concentration when the entire domain is covered by the lowest t2m value found in the test dataset. Sea ice is forming in areas where it is usually not found during September, and sea ice is also forming along the coast of Norway. A similar response can be seen in the December prediction to the lower right, where temperature is linearly increasing from the minimum to maximum value of the test dataset form the bottom of the domain to the top of the domain. Sea ice is forming along the lower border of the domain, as well as above Russia. Moreover, following the temperature gradient, the sea ice tend to decrease as the t2m increases. With the topmost past of the domain experiencing consistent melting.

6.3 Understanding predictions

Sections 6.1 and 6.2 have presented results which attempt to explain the relationship between model performance and predictors. In the current section, predictions will be used to explain model decision-making through the use of segmentation gradient class activation maps (seg-GradCAM) (Vinogradova et al., 2020) which was presented and described in section 3.6.1. Due to the formulation of the targets as cumulative contours (section 4.1.5), the gradient has to be computed from a single target output contour. Thus, each seg-GradCAM will only contain information related to wether a pixel was important for predicting the chosen cumulative contour, as a network used to create a segmentation gradient class activation map will be limited to predict only one cumulative contour.

When creating seg-GradCAMs, only the final layer of the encoder will be considered following the recommendations made by Vinogradova et al. (2020). This approach is also compliant with the originally proposed gradient class activation mapping (GradCAM) for classification tasks (Selvaraju et al., 2016), where it is recommended to compute the gradient with respect to the final convolutional layer. Since neither GradCAM or seg-GradCAM is intended for quantitative analysis, nor are there any studies known to the author where the methods have been applied to compute statistics, a case study approach is adopted where seg-GradCAMs from varying dates and contours will be inspected and compared. To experiments are conducted, one where the baseline model is compared against the model with reduced classes at different contours, and one where the baseline model is compared against the model trained without 2-meter temperature. For the first experiment where target contours are varied, 5th of January 2022 is chosen as the target date. For the second experiment where dates are varied. the same dates as those used in the synthetic AROME Arctic fields experiments are chosen, namely 03rd March, 03rd June, 7th September and 7th December.

For all the following figures, a shared colorscale is used as values have been normalized between 0 and 1 following Vinogradova et al. (2020). However, the actual values are not

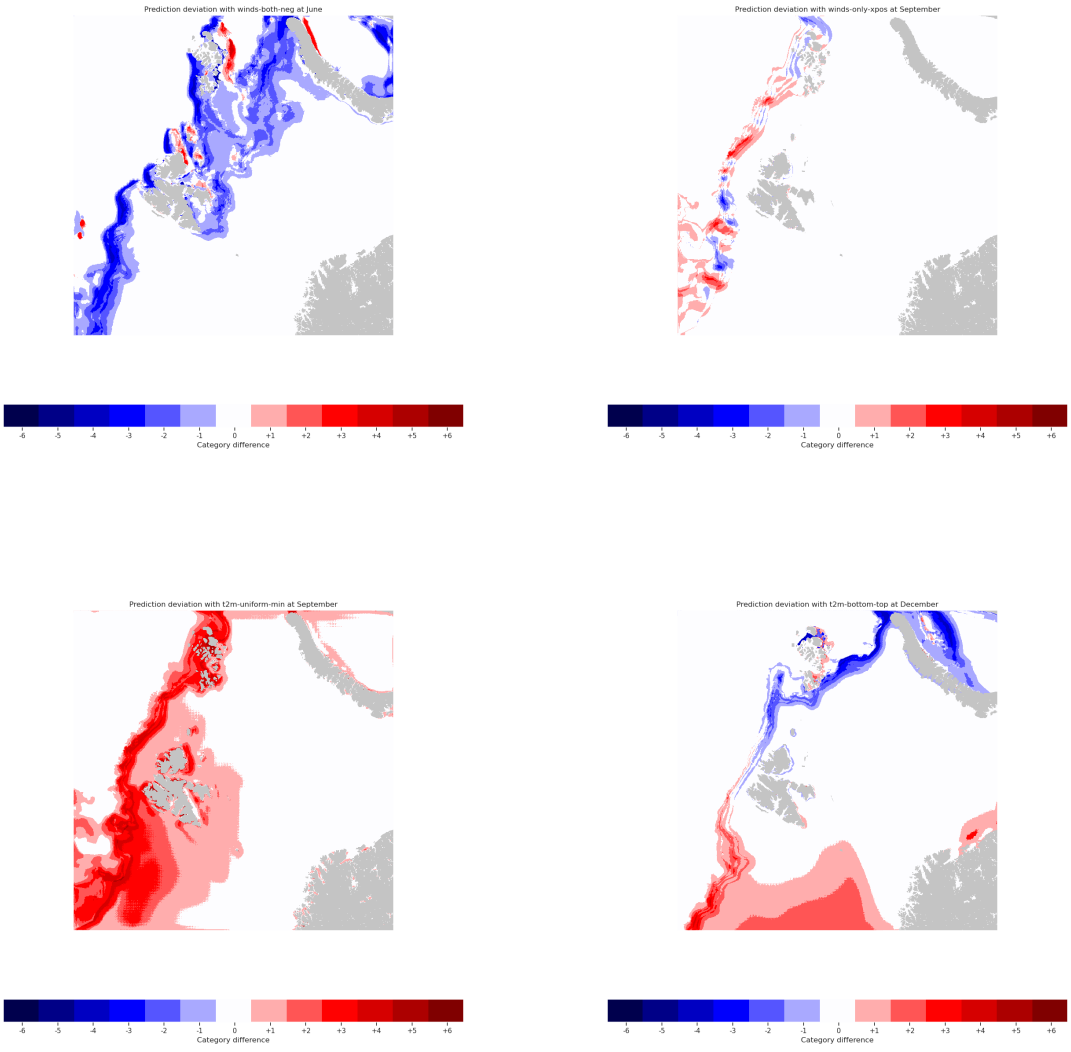


Figure 45: Sea ice contour error with respective to the baseline prediction with no synthetic AROME Arctic field. The figure show a selection of the synthetic fields, with the purpose of visualizing spatially how the deep learning system responds.

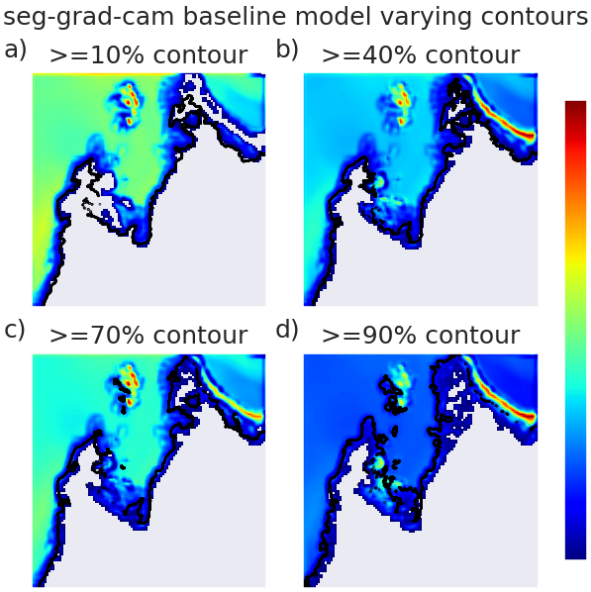


Figure 46: Segmentation class activation maps for varying target contours for the same date. The model used is the baseline model predicting with a two day lead time. The black line is the sea ice edge for the relevant contour from the input sea ice chart.

comparable, hence each seg-GradCAM visualizes the relative importance of each pixel for each contour. Furthermore, although Vinogradova et al. (2020) presents seg-GradCAM as a GradCAM computed from an arbitrary region of interest, for the following experiments the region of interest was defined as all pixels classified as part of the cumulative contour (both true and false positives). Furthermore, only non-zero values are shown.

A seg-GradCAM for varying target contours for the baseline model with a two day lead time is shown in figure 46. The maps are spatially similar, although the intensity of the activated features tend to vary without a clear trend. The activated features seem to closely resemble the position of the sea ice edge. For figure 46 (a, b, c and d) the features with highest activation seem to be located in Frans Josef Land and Novaya Semlya (the latter not for (a)).

The seg-GradCAMs for the model without the $< 10\%$ and $= 100\%$ / fast-ice contour is shown in figure 47. Each seg-GradCAM in figure 47 has significantly higher relative values compared to figure 46. Moreover, whereas the highest activation values occurred over Frans Josef Land and Novaya Semlya for figure 46, the opposite seems to be the effect for figure 47. Furthermore, some spurious activations occur in figure 47 (a, b, c) located towards the lower right where mainland Norway is located.

Få inn hvilken date dette er

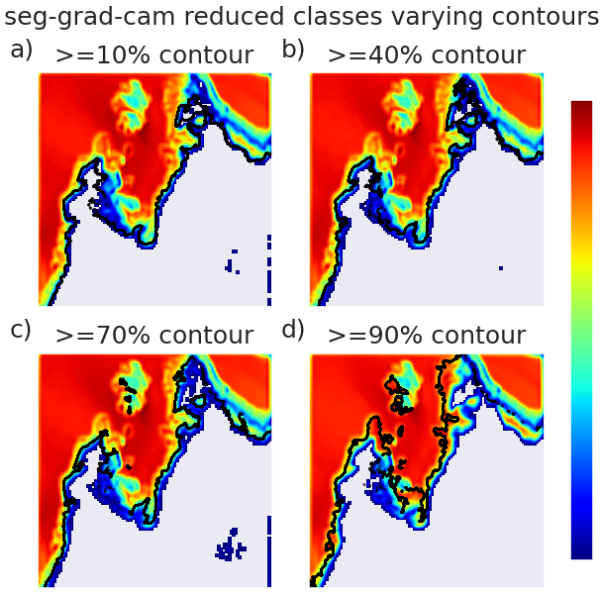


Figure 47: Segmentation class activation maps for varying target contours for the same date. The model used is the reduced classes model predicting with a two day lead time. The black line is the sea ice edge for the relevant contour from the input sea ice chart.

The varying dates experiment for the baseline model is shown in figure 48. As with figure 46, the activated features tend to follow the sea ice edge, with no activated pixels located significantly outside of the sea ice edge. The relative value tend to be towards the upper limit, especially for figure 48 (b, c and d).

Finally, seg-GradCAM from the model trained without 2-meter temperature is shown in figure 49. Compared to figure 48, the relative values tend towards lower values. Contrary to figure 48, figure 49 contain several large patches of activated features with low values located outside the border of the sea ice edge. For figure 49 (a and b), features located under Norway have been consistently activated. Moreover, for figure 49 (d) there appears to be activated features located in the inlet fjords of Svalbard, which is not seen in figure 48 (d).

Mean 2-meter temperature fields from AROME Arctic, used as predictors for the baseline model used in figure 48. The sea ice chart is included to highlight the temperature-gradient in the scenes.

Kanskje i appendix

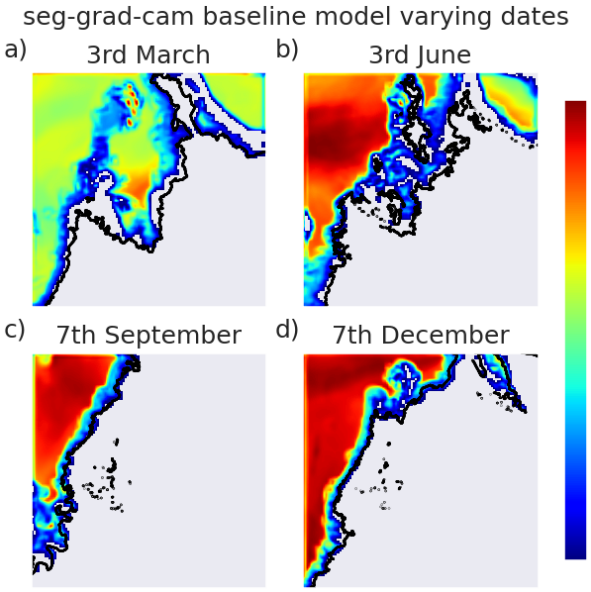


Figure 48: Segmentation class activation maps for varying dates for the same target contour. The model used is the baseline model predicting with a two day lead time. The black line is the sea ice edge for the relevant date from the input sea ice chart.

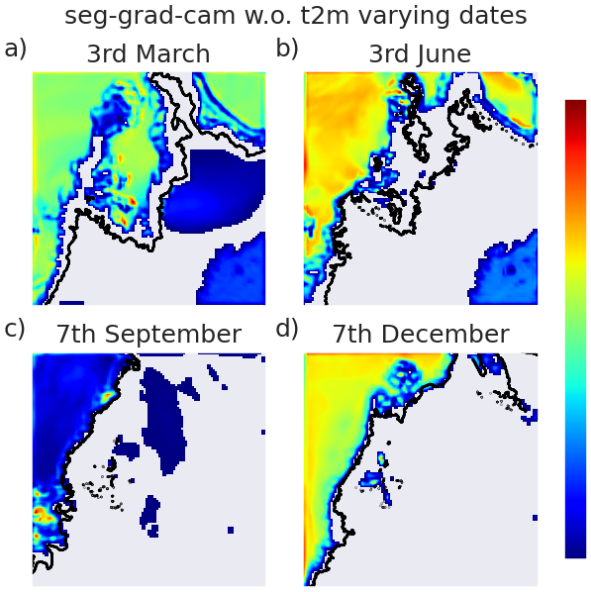


Figure 49: Segmentation class activation maps for varying dates for the same target contour. The model used is a model without 2-meter temperature as a predictor predicting with a two day lead time. The black line is the sea ice edge for the relevant date from the input sea ice chart.

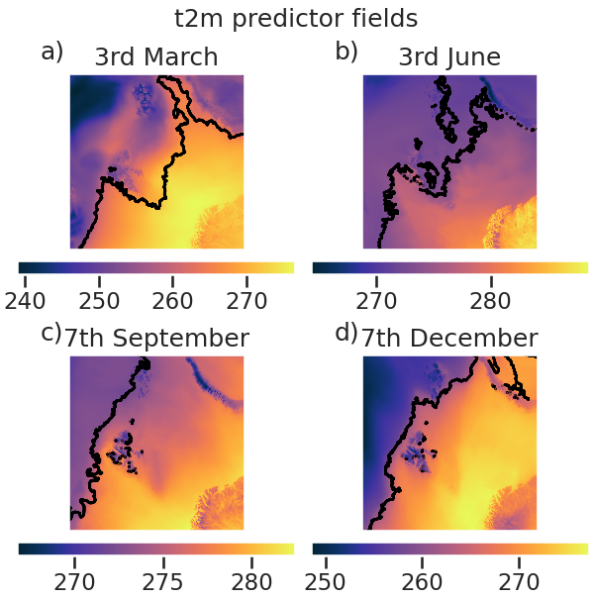


Figure 50: 2-meter temperature mean fields used as predictor for the baseline model for varying dates. The black line is the sea ice edge for the relevant date from the input sea ice chart.

6.4 Case study

A case study is conducted for the date with the highest reported IIEE value from the baseline machine learning model, with a two day lead time. The motivation behind presenting a case study is to relate the previous model explainability results in sections 6.1 and 6.2 to an example where the predictors are in-distribution, thus representing a possible model prediction. The day chosen was 18th of March 2022 (valid date), which implies that the deep learning forecast was initialized 16th of March 2022 (bulletin date). For the current date, the deep learning forecast achieved an NIIEE of 50.8km, with persistence achieving an NIIEE of 52.5km for the same date. The chosen date is 14th highest persistence NIIEE value. The purpose of including a case study is understand if connections can be drawn between the input variables and output forecast, with the date of highest NIIEE for the test dataset chosen as it may be characterized by outlier like conditions compared to the mean annual NIIEE for the baseline model (28.2km). We believe that discussing an outlier prediction is instructive in terms of model explainability, as it may provide further insight into how the model responds to the input predictors. Furthermore, exploring the lower limits of model performance may reveal sea ice conditions in which the model is not well suited to predict.

The sea ice conditions for the bulletin and valid date, as well as the sea ice forecast for the

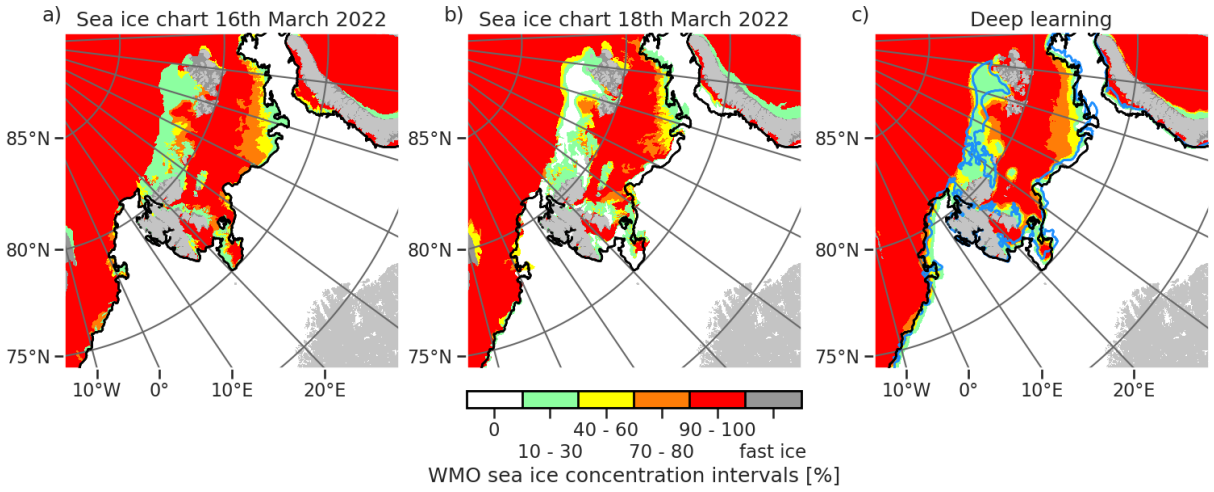


Figure 51: Sea ice charts for the 16th (a) and 18th (b) of March 2022, with a deep learning prediction for 18th of March 2022 initialized 16th of March 2022 in (c). The black line is the sea ice edge for the sea ice chart in (a) and blue line is the sea ice edge for the sea ice chart in (b), both are thresholded by 10% sea ice concentration. The $< 10\%$ sea ice concentration contour is not shown.

valid date is visualized in figure 51. To aid in visualizing the differences between the sea ice concentration fields, the 10% sea ice edge from the bulletin (figure 51 (a)) is visualized in figure 51 (a, b and c). The sea ice edge for the valid sea ice field is only shown in figure 51 (c) as the blue line. The two sea ice edges closely match along MIZ, however a major deviation between the two dates is the occurrence of a major loss of $10 - 30\%$ sea ice north east of Svalbard for the bulletin date sea ice chart (figure 51 (b)). It is seen in figure 51 (c) that the deep learning forecast has not predicted this particular loss of sea ice. For clarity, the sea ice chart in figure 51 (a) was used as input for the deep learning system to make the forecast seen in figure 51 (c).

The atmospheric predictors used to make the forecast in figure 51 (c) are given in figure 52. The fields shown are 2-meter temperature, as well as the x and y wind, which have been computed as two day mean fields between 16th of March 2022 18:00UTC and 18th of March 2022 12:00UTC following the approach for atmospheric predictor construction described in section 4.1.4. The 2-meter temperature field contains values ~ 273K, although significantly lower values are seen towards the top left of the scene (figure 52 (a)). The x-wind field mostly contain values centered about 0 m/s, although some outlier values are located along the north of Novaya Semlya blowing strongly towards the left of the scene (figure 52 (b)). The y-wind component show that the winds vary between blowing towards the top and bottom of the domain, although the y-wind along the sea ice edge (MIZ) tend to be directed towards the bottom of the domain (figure 52 (c)).

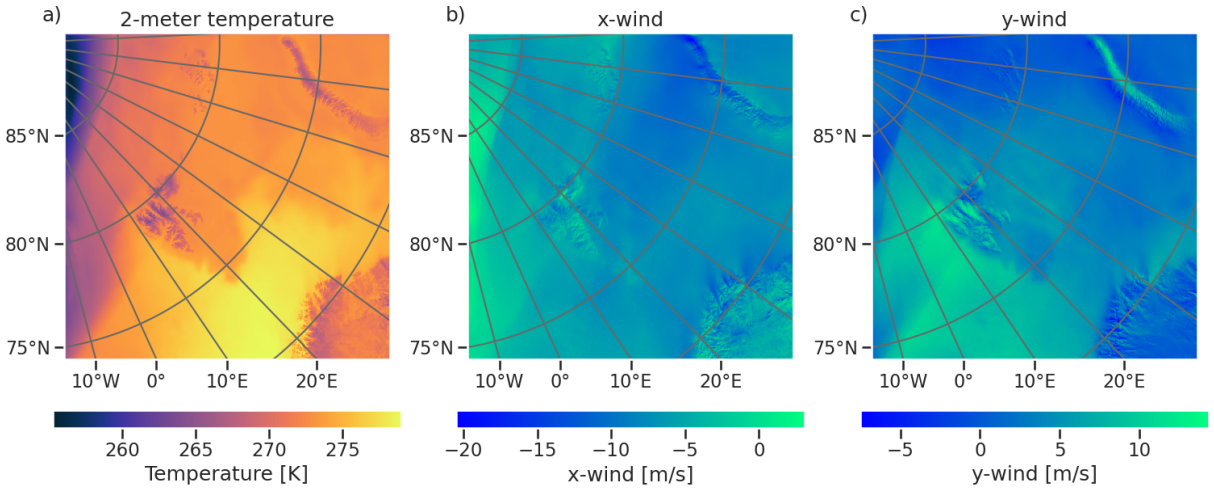


Figure 52: Atmospheric predictors for a sample initialized 16th of March 2022 for a two day lead time model. (a) is the two day mean AROME Arctic 2-meter temperature field. (b) and (c) are the two day mean x and y wind fields from AROME Arctic respectively.

The spatial distribution of deep learning forecast sea ice concentration overestimation and underestimation (Goessling et al., 2016) with respect to the valid date sea ice chart is shown in figure 53. Furthermore, areas where the two products agree are denoted as sea ice (for positive agreement) and Ocean (negative agreement). The land sea mask is included in figure 53, as the IIEE is computed sans the land pixels. The deep learning forecast unresolved lack of sea ice north east of Svalbard (figure 51 (c)) is clearly shown as overestimated sea ice concentration in figure 53. Furthermore, figure 53 show that the sea ice edge along the MIZ is overestimated towards the bottom of the domain, with some spurious overestimation located south of Svalbard. A consistent overestimation occurs towards the top of the domain, at the sea ice edge located opposite the northern edge of Novaya Semlya. Contrary, the sea ice concentration is overestimated along the coast of Novaya Semlya.

7 Discussion

The following section will discuss the results presented in sections 4, 5 and 6. Results will be discussed separately and in conjunction where appropriate.

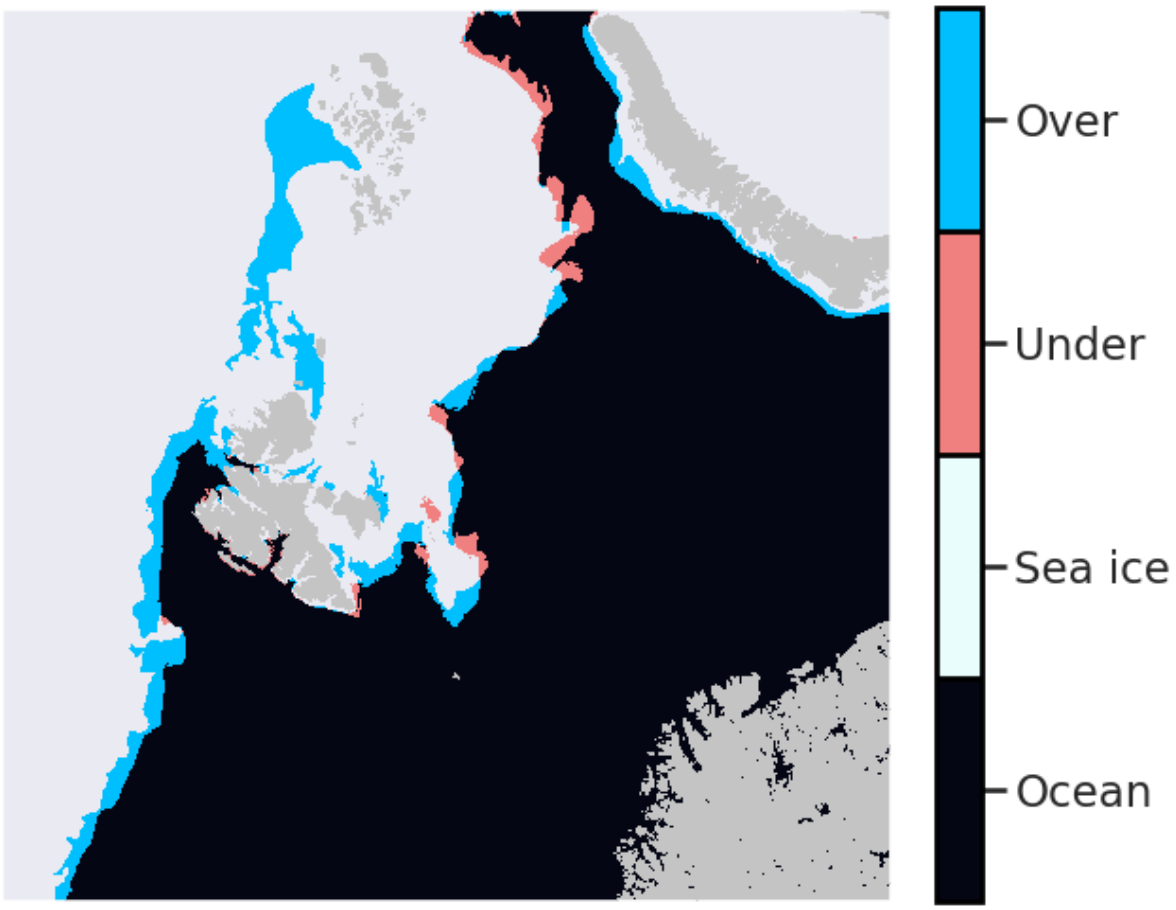


Figure 53: IIEE computed between the deep learning forecast valid 18th of March, initialized 16th of March 2022 and the sea ice chart for 18th of March 2022. Sea ice and Ocean denote true positives and true negative for the deep learning forecast.

7.1 Development

7.2 Initial attempt

Section 4.3.2 presents the initial attempt at designing a deep learning architecture for sea ice category classification. The initial architecture closely resembled the U-Net architecture as described by Ronneberger et al. (2015), with a single output layer with multiple channels such that multiple classes is predicted at the same time. Following the U-Net architecture and training procedure descriptions given in sections 3.3 and 3.4, each channel in the output layer represents a target class with the channels containing unactivated probabilities (logits) for each pixel to belong to the class. To determine which category each pixel is most likely to belong to, the softmax function (equation 6) converts the logits to probabilities where for each pixel is assigned the category belonging to the channel with highest probability. As seen in figure 19, neither the $(40-60\%)$ nor $(70-80\%)$ categories are resolved. This occurs for all 2022 test data (not shown). It appeared from the monthly distribution of sea ice categories presented in figure 3 that the intermediate sea ice categories (very open drift ice, open drift ice, close drift ice) constituted a significantly low fraction of the different sea ice categories. Due to the skewness in the sea ice category distribution towards ice free open water and very close drift seen in the sea ice charts, the current formulation of the segmentation task is highly imbalanced in favour of the aforementioned classes. Furthermore, due to the presence of class imbalance, it is expected that the computed loss (which is the unweighted variant of 8) is dominated by the more likely classes. Assuming that each contribution to the loss is treated equally due to the absence of a class- or pixel level weight term, the more likely classes supply a larger fraction of the learning signal which has the potential to dwarf or counteract the portion of the learning signal computed from the less likely classes (Lin et al., 2017).

With regards to the intended operationality and usefulness of the developed deep learning system, resolving the MIZ is a crucial aspect to achieve skillful forecasts that ensure maritime safety (Wagner et al., 2020). One way to approach the problem of class imbalance would be to modify the weight parameter w in equation 8. In the work of Ronneberger et al. (2015), w was defined as a precomputed weight map which assigned some pixels more importance than others to compensate for the different frequency of pixels between the classes. However, it is noted that the weight map computed by (Ronneberger et al., 2015) is specifically designed for biomedical image segmentation, and not directly transferable to other domains. Another approach to which indirectly defines the weight w is through rewriting the loss function, such as the focal loss proposed by Lin et al. (2017). The focal loss introduces a focusing parameter and a modulating factor which down-weights the loss contribution from predictions with a probability, i.e. easy to predict samples (Lin et al., 2017). Another way to approach the problem of class imbalance is to reformulate the prediction task to a more balanced problem as described in section 4.1.5 with the intro-

duction of cumulative contours. Due to the immediate improvement in contour resolve seen when implementing the cumulative contours, the approach was further pursued in favour of the single output layer model.

7.2.1 Determining the depth of the model

Finding the optimal learning rate and depth of the desired deep learning architecture (multiple outputs, single label) is summarized in figures 20 and 21. Based on the conducted grid search in figure 20, we see that the validation loss tend to increase when the model becomes deeper, as well as when deviating from $lr = 0.001$. Although the lowest validation loss is achieved with $lr = 0.001$ and a depth of 512 channels in the bottleneck, the difference is marginal compared to the 256 depth counterpart. However, the model with a depth of 256 channels has 4 times less parameters than the 512 depth model, which indicates that the 256 model is satisfactory fit to the data without needing the additional parameters found in the deeper network. This is further indicated by figure 21, which show that the 256-depth model is marginally improved beyond the 10th epoch.

Moreover, when comparing a prediction from a 1024 depth unet with a 256 depth unet (figure 22), the predictions are generally visually similar. The similarity can be further seen when comparing the difference in NIIEE, which is only 2.5km between the two models. As such, the increased complexity gained from increasing the parameter-count of the model does not seem necessary to increase the predictive skill of the model. Viewed in conjunction with figure 21, a possible explanation may be that the 256 model already rapidly fits the training data. Hence, the limited number of training samples does not necessitate a more complex architecture in terms of encoder depth.

Using equation 4 from Araujo et al. (2019), it was calculated that the bottleneck of the 256 encoder have a theoretical receptive field of 145 pixels in each direction, whereas the 1024 encoder have a receptive field covering the entirety of the input fields. As such, each high level feature in the 256 bottleneck have only been influenced by lower level features in a 145km radius, whereas each high level feature in the bottleneck of the 1024 model have been influenced by features from the entirety of the domain. Hence, although the encoded signals in the 1024 bottleneck contain influences from the entire input field, figure 20 and 22 indicate that the performance is still reduced, as the model has a higher NIIEE compared to the 256 model with a limited influential range for the bottleneck.

It is shown in Luo et al. (2017) that the effective receptive field is limited compared to the theoretical receptive field, with the effective receptive field attaining an asymptotic gaussian shape. Moreover, Luo et al. (2017) also show an example where although the theoretical receptive field is bigger than the input size, the effective receptive field is not able to fit the whole image, which is a consequence of the relative shrink of the effective

receptive field shown to follow the relationship $\propto \frac{1}{\sqrt{N}}$ where N is the number of layers (Luo et al., 2017). Thus, based on the results of Luo et al. (2017), all features of the bottlenck in the 1024 model are unlikely to be influenced by the entirety of the domain despite the coverage implied by the theoretical receptive field. Moreover, the full domain theoretical receptive field coverage was attained by increasing model complexity by a factor of ~ 16. Thus, the current results seem to discourage the need of fitting deep and complex deep neural networks to high resolution and spatially dependent data, as their theoretical ability to encode high level features influenced by all grid cells from the input data is opposed by a combination of the increased model complexity and reduced effective receptive field fraction.

7.2.2 Demonstrating seasonality

The deep learning system is able to preserve seasonality, as indicated by figure 23 which show that the predicted sea ice concentration for a two day lead time follows the observed sea ice edge for all months. Based on the demonstrated predictive capabilities of the model seen in figure 23, it can be seen that the deep learning system is able recreate the seasonal variability of the sea ice edge. Furthermore, the model is able to follow the seasonal cycle of sea ice concentration from the physical predictors only, which indicates that the model is able to infer the day of year based on the combined state of the sea ice concentration and atmosphere. This contrasts the work of Grigoryev et al. (2022), where the date was explicitly given to the model as a predictor. Thus, the model is able to capture the seasonal differences of sea ice dynamics from physical input fields only.

7.2.3 Using NIIEE as a metric

Throughout this thesis, model performance have been measured according to the normalized sea ice edge displacement error (Goessling et al., 2016; Melsom et al., 2019; Palerme et al., 2019; Zampieri et al., 2019) for different contours. The works of Goessling et al. (2016); Palerme et al. (2019); Zampieri et al. (2019) all applies the IIEE to seasonal prediction systems with coarse spatial resolution ($\gg$ 1km), hence it is explored how the IIEE responds to high spatial resolution sea ice concentration. Moreover, as the NIIEE compared in the inter-product comparisson (section 5.3) a high resolution IIEE with a coarser resolution sea ice edge length is also inspected in section 4.3.1. Firstly, based on the computed correlations between the NIIEE computed from both sic and sea ice edge at 1km and NIIEE where both fields were at a 10km resolution, the NIIEE is conserved at increasing resolutions.

Secondly, when dividing the IIEE from a 1km spatial resolution grid by a sea ice edge length computed from a 10km sea ice concentration field, the variability of the NIIEE

is similar as when the 1km IIEE is divided by a 1km sea ice edge length (figure 18). The results indicate that the IIEE is a relevant metric also when applied to high spatial resolutions. Moreover, the variability of the NIIEE is preserved with increasing sea ice edge resolutions. Note that the sea ice edges used in section 4.3.1 were not independent of the sea ice concentration, as is the case with the sea ice edge length derived from OSI SAF CDR.

With regards to model selection, it was shown in section 4.3.5 and figure 28 that the correlation between the NIIEE and validation loss is strong with a value of 0.82. Moreover, figure 28 presents an example training where after each epoch, the model was measured in terms of the NIIEE on the validation set. As validating the deep learning system against the NIIEE was considerably slower (2 hours) compared to validating against the validation loss (8 minutes) for a single epoch, it was decided to select models in terms of the validation loss since the correlation between the metrics is strong.

7.2.4 Increasing the size of the training data

The size of the core training data (2019 and 2020) of 390, 289 or 286 samples for one, two and three day lead time respectively is considerably small when training an encoder-decoder deep learning model. Furthermore, the strong autocorrelation seen in the sea ice charts (figure 4) may act to further reduce the total training dataset, as the sea ice concentration for consecutive days have a low variability (Fritzner et al., 2020). However, by increasing the predictor pool through introducing atmospheric variables from AROME Arctic the deep learning model, the deep learning model is also learning correlations between the sea ice concentration development and atmospheric (temperature and wind fields) development. Hence, the strong autocorrelation between sea ice concentration may not cause the unwanted effect of rendering certain samples redundant, as the atmospheric predictors allows the model to learn a more complex pattern from the input data.

It was mentioned in section 4.1.6 that the core training data only consists of the year 2019 and 2020 due to a major update regarding the representation of near surface temperature. It is common to assume that all data used to train, test and validate a machine learning algorithm is independently and identically distributed (IID), but this assumption does not hold when using predictors from AROME Arctic which has been continuously updated after it was released (Müller et al., 2017). The choice of limiting the training dataset to the aforementioned two year period was made based on an assessment of which updates would have a strong influence on the IID assumption for the chosen variables, where the snow on ice parameterization was deemed significant (Batrak and Müller, 2019). An example of how the updated changed the near surface temperature distribution can be seen in Figure 54, with the figure showing a clear bias reduction compared to observations.

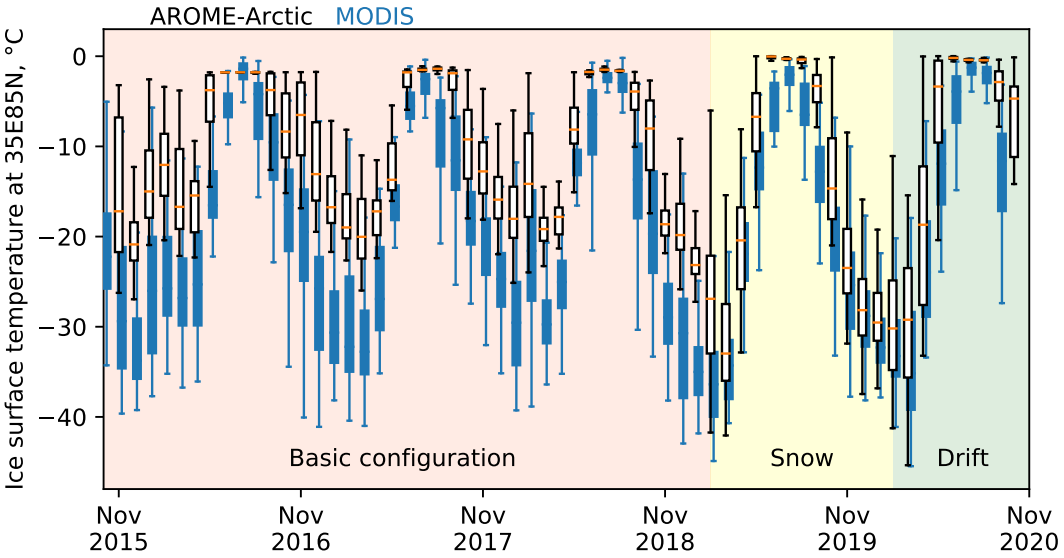


Figure 54: Ice surface temperature from AROME-Arctic compared with observations. Arome Arctic is distributed as white boxes, whereas observations are blue boxes. The snow on ice parameterization was added starting at the yellow background. Figure courtesy of Yurii Batrak.

However, with regards to the problems occurring as a consequence of limited data, the effect of training on an increasing amount of years was shown in figure 25. From figure 25, it can be seen that both the NIIEE and validation loss is reduced when 2018 and 2017 is prepended to the training dataset, although 2016 causes the model to lose skill. Figure 26 show that the sea ice charts for the years covered by the training, validation and test dataset follow a similar distribution. Furthermore, there are no trends or clear outlier between the years. As such, it can be assumed that the sea ice charts are IID also when the training data has additional years prepended. Thus, the loss of skill caused by 2016 is likely to be caused by predictors from AROME Arctic which, due to continuous model development, are not IID. That the loss of skill only happens with 2016 included could be a reflection of 2016 being the first full year of operation for AROME Arctic, where significant updates are expected to have occurred frequently.

It should also be noted that the inclusion of 2018 and 2017 seems to improve model performance, despite the years not including the snow on ice parameterization deemed significant with regards to the representation of t2m. This could be explained by the model being more impacted by the winds rather than the temperature from AROME Arctic, thus causing the shift in temperature variance to be negligible. A further discussion of predictor importance and model explainability will be done in section **NOT YET**.

Figure 25 highlights the difficulties of fitting a deep learning model to data from an operational product, as the data is updated frequently in order to improvement the product performance through intentional shifts in model bias or variance (Batrak and Müller, 2019) (see also figure 54). It was mentioned in section 2.3.1 that re-forecasts or reanalysis products such as CARRA (Køltzow et al., 2022) or ERA5 (Hersbach et al., 2020) would alleviate the problem of temporal inconsistencies in the data, as the model used to create a reanalysis is based on a single model cycle. However, reanalysis products generally have a publishing delay from days to months, which is not compatible with the operational timeliness of providing forecasts on the same day as the predictors are valid envisioned for the developed deep learning system. Furthermore, a reanalysis differs from a forecast in that data is complete and consistent, which in terms of the temporal mean approach for the atmospheric predictors would limit the data diversity as the reanalysis provide the same values for a given date regardless of the initialization date of the deep learning forecast. This is contrasted by a forecast, which output different values at shared timestamps depending on the forecast initialization date. Hence, despite the expected increase in skill of a reanalysis product, the above may serve to limit the overall usefulness for reanalysis products as deep learning predictors in the case where there is a temporal overlap between predictors. On the other hand, re-forecasts as a static model cycle providing full range forecasts would not have the same restraints since predictors with overlapping timestamps would be computed from different values.

Although this project have attempted to increase the dataset size directly by adding more data, similar studies utilized other techniques such as Grigoryev et al. (2022) which trained their deep learning model on multiple regions and performed geometric transformations on the data during training. Their results show that geometric augmentations have a small positive influence for some areas, and negligible for one area. Though adding multiple regions to increase dataset size is outside the boundaries of this work, applying geometric augmentations to the data is a known generalization technique for image based deep learning which Grigoryev et al. (2022) could alleviate the limited dataset if implemented correctly (Shorten and Khoshgoftaar, 2019). However, given that atmospheric variables have a strong spatial dependency inferred by their lat/lon coordinates, it is questioned to what extent geometric transformations applied to the entirety of a scene can teach the network new relations between the variables.

7.2.5 Tuning model related parameters

The effect of interpolating sea ice concentration over land covered pixels following the approach of Wang et al. (2017) was recorded in section 4.3.4. The approach was compared with assigning all land-covered pixels as the category ice free open water (open-ocean mask), where the latter approached increased the mean annual NIIEE with 1.5km. This

shows that how the sea ice is represented in areas which are ignored during validation due to the land sea mask impacts how surrounding grid cells are classified. Moreover, the result show that supplying the land-sea mask as a predictor is not by itself sufficient to teach the network to discern between land pixels and water pixels, which may be explained by how the convolutional kernel computes features from a local neighborhood separately for each channel (Fukushima, 1980).

Another interpretation of the open-ocean mask is that it relates the affected predictor to be treated as an incomplete partial convolution (Liu et al., 2018). Partial convolution is a variation of the convolutional layer which stem from image inpainting, which is the task of filling holes in an image (Liu et al., 2018). The technique involves multiplying the convolutional filter with a masked variation of the local neighborhood determined by the holes in the image, then the computed value is scaled to adjust for the number of unmasked pixels in the neighborhood (Liu et al., 2018). Since the integration of the land-sea mask into a predictor through the open-ocean mask was without a consideration of scaling the output with regards to the varying number of masked pixels in the local neighborhood, the sharp gradient computed between land-fast ice and ice-free open water is left unscaled thus detected as a notable feature. Thus, the work of Liu et al. (2018) may provide a possible explanation as to why the open ocean mask degrades the performance of the model.

However, the result also highlights a different approach to integrate the land-sea mask into the predictors, as partial convolutions would make it possible to mask a subset or all predictors with the land sea mask thus removing it from the pool of predictors. This may aid to increase the relevance of the land-sea mask, as although the intended behavior of the predictor is help the model treat land pixels differently than ocean pixels it was previously discussed that this behavior was not directly inferred from the land-sea mask. By utilizing partial convolutions, and using the land-sea mask as a mask, the convolutional result would only depend on the non-hole regions (Liu et al., 2018).

Reducing the number of classes to output does not seem to improve the model beyond the baseline. The model trained in section 4.3.4 where the $> 0\%$ and $= 100\%$ contours was removed performs similarly to the baseline model which outputs all target classes, with an increase of the NIIEE of 0.4km. This result may suggest that although removing contours have a theoretical impact on training, the effect is not reflected by the predictions. From the description of how the individual losses computed with equation 22 are treated as a sum starting at the decoder during backpropagation (section 4.2.7), it can be seen that removing contours decreases the magnitude of the loss function. Moreover, due to the cumulative contours formulation, removing outputs also has the effect of lowering the amount of redundant information as the number of overlapping pixels is reduced. Hence, the result of removing contours, especially the 100% contour, is that the computed loss is less weighted by the land-fast ice pixels , causing other contours to have a greater impact

The first sentence below is also seen in amsr2 model compare fig

This is what is seen by the saliency grad-cam experi-

on the training.

The mean seasonal confidence fields from figure 31 can be used in conjunction with the loss function defined in equation 22 to determine the imapct of the = 100% contour have on the overall loss starting from the decoder. Equation 22 is defined such that a greater deviation from the true value (1) results in a greater loss. Moreover, the = 100% contour is usually present beneath the land areas as the fast-ice contour drawn close to land by the sea ice specialist is interpolated onto the land-pixels by the nearest neighbor interpolation (Wang et al., 2017) performed on the sea ice charts during data preprocessing (section 4.1.2). Figure 31 reveal that the = 100% contour has a seasonal contribution to the loss, where the winter and spring contribution is lower than the summer and autumn contribution. Since the confidence levels are high during winter and spring, the impact from the = 100% contour expected to be small. However, the overall low confidence levels during summer and autumn will increase the loss hence samples from the summer and autumn seasons trained on networks where the = 100% contour is used as an output may decrease model performance if the loss from the other contours are dominated by the loss of the = 100% contour. As it is ideal for the model to perform well at resolving the MIZ related contours, reducing the number of contours such that unrelated contours with a high loss (such as just described for the = 100% contour) does not negatively impact model performance through high influence on the loss is advisable.

Another possible way to mitigate the loss from one cumulative contour to dominate the loss contributions from the other cumulative contours is to reformulate the network architecture as a multitask learning problem. Multitask learning is a branch of machine learning where a single network is used to perform mutliple tasks simultaneously, with the goal of sharing as much information as possible while at the same time reducing negative interactions between the tasks (Crawshaw, 2020). For the current task, a negative interaction could be the aforementioned high loss from the (= 100%) contour which decrease the ability for the other contours to influence the loss when training for certain seasons. Multitask networks for computer vision tasks generally employ an approach originally proposed by Zhang et al. (2014) where a common feature extractor is shared by individual output branches for each task (Crawshaw, 2020). This general approach can be translated to the U-Net architecture, where the encoder is used as a common feature extractor and each task is assigned a individual decoder (Jha et al., 2020). Given the cumulative contour formulation, employing a multitask learning U-Net architecture similar to Jha et al. (2020) where each contour is predicted individually with its own decoder could help differentiating the predicted contours as they do not share the same pathway, as well as reduce negative interactions as described above. However, this approach could cause the network to disassociate the cumulative contours from one another, causing unwanted effects on the output forecast such as a higher amount of sporadic category change which is currently not the case for the shared decoder models. E.g. both figure 22 and

23 show the contours tend to gradually change starting from the lowermost contour, with few occurrences of sharp category changes.

The model response to replacing all ReLU non-linear activation functions (Nair and Hinton, 2010) with a linear mapping was visually exemplified in figure 27. The linear model increased the mean annual NIIEE from the baseline by 13.15km, which 6km higher than persistence mean annual NIIEE for the test data as shown in figure 41. The purpose of assessing the forecast skill with a linear deep learning system is to understand the effect of the non-linear activation function. We see that predictions made with a linear model perform significantly worse than persistence (for a two day lead time), which renders the forecasts skilless. The example prediction seen in figure 27 may provide insight as to why the linear model performs worse than the non-linear counterpart. Firstly, there are several instances of checkerboard artefacts commonly caused by deconvolutional layers. The U-Net architecure contains skip connections from the encoder to the corresponding layer of the decoder (see figure 8, gray arrow) where spatial information in the encoder is concatenated with the upsampled signal from the deconvolutional layer and may help to supress the checkerboard artefact (Ronneberger et al., 2015). However, the two concatenated feature maps are not merged until after the first convolutional layer and subsequent activation function. Hence, when the non-linear activation is replaced by a linear mapping, the convolutional layers are not able to supress the checkerboard artefact by themselves, which shows that the non-linear connections introduced with the ReLU activation function are crucial for model performance.

Burde jeg liste opp NIIEE for relevant modeller som er diskutert i en tabell?

Secondly, the linear model seem to underestimate the $\geq 10\%$ contour in favour of the $\geq 40\%$ contour when compared to the same prediction with the baseline model in figure 22 (a). The linear model is also unable to resolve the isolated patch $\geq 70\%$ sea ice concentration North of Novaya Semlya seen in figure 22. Moreover, detailed structures such as the what is seen for the $\geq 90\%$ contour towards the lower left of figure 27 are likely remnants of the sea ice chart used as predictor. Thus, the model is repurposing parts of the input when making a prediction, which is probably done due to the high autocorrelation seen between the sea ice charts (figure 4). However, this means that the linear model have learned limited connections between the predictors, indicating that the non-linear activation function is necessary for the model to correlate sea ice dynamics with the state of the atmosphere as discussed in section 7.2.2.

7.3 Performance

As described in section 5.1, a forecast is skillful if it achieves a lower NIIEE than persistence. The performance of the developed deep learning system was measured in ways. Firstly, the performance of the deep learning system was measured against persistence, utilizing the sea ice charts as the ground truth target. Secondly, the performance of the

deep learning system was compared against four other sea ice concentration products, with two different ground truth datasets utilized where one of them (AMSR2) is completely independent of the deep learning system to measure the generalizability of the product. We begin by discussing the performance of the deep learning system in isolation, before broadening the discussion by including other products.

7.3.1 Model performance with a two day lead time

The 256 architecture forecasting with a two day lead time was compared against persistence, computing the NIIEE for all contours. The result was shown in figure 29, where it was shown that the deep learning forecast achieves a lower median and interquartile range boundaries for the $\geq (10, 40, 70, 90)\%$. The model tend to perform significantly worse than persistence for the fast-ice contour (figure 29 (f)), which can be explained by the limited fraction of the scene which contains fast-ice (figure 3) causing the scene to be skewed in disfavour of the category even with the cumulative contour definition. This is further elaborated through figures 30 (f) and 31, which shows that the model predicts a small fraction of the domain as the contour, and that the confidence of the prediction is tied to seasonality. Furthermore, inspecting the other confidence fields in figure 30 reveal that the model predict the other categories with higher probabilities and with a strong resemblance to each other. Thus, when each contribution to the total loss is summed at the end of the decoder during backpropagation, the similar signals from the intermediate contours increases the magnitude of the computed loss in the shared decoder.

For the $> 0\%$ contour, the model is performing worse than persistence in the summer, with the summer distribution for the deep learning system closely resembling the persistence distribution although it appears somewhat denser around the 25th percentile. Moreover, it is seen that the deep learning forecasts have a smaller shift in interquartile range during the summer and autumn compared to winter and spring for all contours (except $(= 100\%)$). This may in part be explained through the IIEE, which is proportional to the sea ice edge length Goessling et al. (2016) as the potential area of error is reduced. However, it was previously discussed in section 7.2.5 that as a result of not utilizing a multitask approach to the U-Net architecture, the $= 100\%$ contour could negatively impact training due to potential high errors over the land-fast ice contour dominating the computed loss. The apparent seasonality in model performance seen in figure 29 could be explained by the negative interactions between the $= 100\%$ as it has low confidence which causes high errors during summer and autumn. Following the discussion in section 7.2.5, that the $= 100\%$ has a different confidence spatial extent compared to the other cumulative contours (figure 31) and especially during winter and autumn where the fast-ice contour is generally disconnected from the other contours. Thus, the negative impact caused when the different contours interact in the shared encoder decreases model performance during

summer and autumn as seen in figure 29. This may not be a problem during winter and spring, as the = 100% contour is not disconnected from the other cumulative contours, hence not contributing with loss values irrelevant to the extent of the sea ice edge.

Finally, figure 29 demonstrates the potential for machine learning forecasts of sea ice concentration, as the deep learning system generally outperforms persistence. The relevance towards maritime end users is encouraged by the performance at the lower concentration contours (such as the $\geq 10\%$) (Wagner et al., 2020; Veland et al., 2021). Moreover, the results prove the effectiveness of formulating the targets as cumulative contours rather than multiclass classification which was intially discussed in section 7.2. The cumulative contours tend to sufficiently resolve the intermediate sea ice categories (MIZ). It appears that the intermediate classes are well resolved at the expense of the fast-ice contour with the cumulative contour formulation, as seen in figure 33. With respect to end user relevance, this is an apt tradeoff.

7.3.2 Model performance for varying lead times

The model performance decreases in terms of absolute NIIEE as a function of lead time. However, the relative improvement over persistence increases with lead time, as shown in figure 24. Based on the results of Zampieri et al. (2019), it is expected that the NIIEE for persistence increase fast with lead time. However, the results from Zampieri et al. (2019) also show that no seasonal sea ice prediction system has predictive skill for the first few (~ 2) lead times, indicating that persistence forecasts are relatively skillful for short lead times. Hence, achieving lower NIIEE than persistence for all considered lead times further demonstrates the predictive capabilities of the deep learning system. Moreover, that the relative improvement over persistence seen in figure 24 increases with lead time can be attributed to the rapid loss of skill for persistence forecasts. However, the deep learning system is able to fit the target lead time as well as increased AROME Arctic cumulative mean, as the forecast error in figure 24 is not linearly growing.

Figure 32 provide some insight into how the different deep learning systems resolve sea ice concentration contours by showing the mean monthly sea ice edge length. From figure 32, it can be seen that the sea ice edge length decreases as a function of lead time. Moreover, the figure already shows that there is a negative bias in terms of sea ice edge length when comparing the forecasts with the target sea ice chart. The negative bias in sea ice edge length can be explained when comparing figures 22 or 23 with an example sea ice chart as presented in figure 2, where it can be seen that the predicted sea ice charts have a smoother appearance rather than the sea ice chart which resolve individual structures with a high level of detail. As the sea ice edge length is computed as a sum of grid cells (see equation 12) (Melsom et al., 2019), it follows that a less detailed sea ice edge results in a shorter sea ice edge length. Hence, since the ice edge length becomes shorter

for increasing lead times, it is expected that the deep learning forecasts produce higher variance forecasts which are less precise, i.e. smoother contours.

7.3.3 Comparing against multiple products

A difference between the two comparisons in sections 5.2 and 5.3 is the choice of grid, where the AROME Arctic grid native to the sea ice charts and deep learning system used in section 5.2 is exchanged with the grid of the product with largest spatial resolution in section 5.3. Hence, when utilizing the sea ice charts as ground truth, the 3km neXtSIM grid is utilized (Williams et al., 2021). When AMSR2 is used as ground truth, the 6.25km AMRS2 grid is utlized (Spreen et al., 2008). Since nearest neighbor interpolation is used to downsample the products onto the target grid, it is expected that the higher resolution products such as the deep learning system or persistence receive a larger amount of interpolation artefacts compared to the products with a grid which is closer to the target. Although the interpolation artefacts remain unmeasured, it is assumed that they have a negative effect proportional to the grid size difference on the products.

The inter-product comparison made between NeXtSIM, persistence, the baseline deep learning model, the linear OSI SAF trend and Barents-2.5 for a two day lead time was shown in figure 35. It is initially noted in section 5.3 that only the sea ice chart based products are able to resolve the $> 0\%$ contour. When comparing data against the $> 0\%$ contour, it is important to note that the contour itself is not resolving the sea ice edge, nor is it directly tied to any observation of sea ice. However, the contour is drawn by the sea ice analyst to denote a buffer tracing the actually observed sea ice edge denoted by the $\geq 10\%$ contour. Thus, the $> 0\%$ contour is a result of the operational nature of the sea ice charts, rather than a actual occurrence. Thus, this result is expected, since the other sea ice concentration products used in the inter-comparison are independent of the sea ice charts. With the OSI SAF linear trend being computed directly from satellite observations (Tonboe et al., 2017) and both physical models being forced by the ocean and sea ice model TOPAZ (Sakov et al., 2012).

Similarly to when the deep learning system was compared only against persistence in section 5.2 figure 29, the deep learning system exceeds all other products in terms of distribution mean and median for the $\geq (10, 40, 70, 90)\%$ contours, further demonstrating the skill of the deep learning forecasts. Figure 35 also provide insight into the performance of the different physical sea ice prediction system, where Barents-2.5 tend to provide lower median and mean values compared to NeXtSIM. Moreover, neXtSIM tend to increase the distribution spread for increasing contours, especially during the summer season. The OSI SAF linear trend is performing worse than persistence as a baseline product, and the OSI SAF linear trend NIIEE distribution tend to increase with increased target contour. The lack of performance for the OSI SAF linear trend can be connected to the subpar

performance of the linear deep learning model discussed in section 7.2.5, and indicates that the sea ice concentration dynamics are of a non-linear nature (Grigoryev et al., 2022).

The spatial anomalies for all products used in the inter-product comparison with respect to the sea ice chart targets was shown in figure 37. The overall good performance for both the sea ice charts as well as persistence is reflected in figure 37 since the biases are overall low. Moreover, compared to all the other products, the sea ice chart based forecasts have no bias in the upper left corner for all seasons. This corresponds with where the deep learning forecast always showed high confidence for predictions, further indicating the aforementioned feedback in that the model is apt at predicting where the cumulative targets overlap due to the shared decoder. Furthermore, figure 37 provide some insight into the performance of neXtSIM, as there is a consistent negative bias along the sea ice edge which is especially prominent during winter and spring. Hence, figure 37 show that neXtSIM underestimates the sea ice edge during winter and spring, and the entire scene for summer and autumn.

The anomalies for Barents-2.5 are generally low, but covers large parts of the available scenes. The consistent underestimation of sea ice concentration seen in Barents, neXtSIM and to some degree the OSI SAF linear trend in figure 37 during summer and autumn may occur due to the physical models assigning the grid-cells concentration values less than 90%, whereas the $\geq 90\%$ sea ice category is frequently used by the sea ice specialist to denote all sea ice that is not part of an ice edge. This highlights potential weakness in comparing the physical models against sea ice charts, and may also explain why the physical models are less comparable than the sea ice chart products in figure 35 as it seems the non-sea ice chart products forecast lower concentration values, especially during summer and autumn.

With respect to operational concerns of the developed deep learning system, as well as with regards to providing user-relevant validation metrics which are easily interpretable (Veland et al., 2021), the fraction of days where the deep learning forecast offers an improvement compared to the other products was shown in figure 36. Firstly, it should be noted that the high uncertainty for months where the fraction of days is less than 100% arise due to limited monthly sample sizes (table 2). However, figure 36 is consistent with the performance for varying lead times shown in figure 24 when comparing the deep learning system against persistence. It can be seen in the top row of figures in figure 36 that the improvement against neXtSIM is decreasing with increasing lead time, which may indicate that neXtSIM gains performance with lead time.

A similar trend occurs when comparing against Barents-2.5, however the Barents forecast is able achieve more days with lower NIIEE for November and December with a two day lead time. The consistent high performance of Barents seen in figure 35 and 37 motivate

for an inclusion of the model into the pool of predictors. E.g. could Barents-2.5 provide forecasted sea ice concentration as a predictor similar to the atmospheric fields from AROME Arctic. Although Barents-2.5 is still in development, providing re-forecasts of the model as described in section 7.2.4 would allow for coverage of the training dataset with the current model performance as of after the spin up time of the data assimilation system (Röhrs et al., 2022). Furthermore, the problem regarding predictor multicollinearity as described in section 4.1.4 is avoided as Barents-2.5 provide sea ice concentration information independent of the sea ice charts, as well as providing data with a future perspective which could greatly enhance the skill of the deep learning system.

To assess the generalized performance of the deep learning system, the inter-product comparison was extended to also compare against independent AMSR-2 sea ice concentration observations (Spreen et al., 2008). Firstly, the sea ice chart based products are not skillful for the $\geq 0\%$ contour, confirming the unphysical nature of the contour for the sea ice charts. However, for increasing contours starting from the $\geq 40\%$ contour, the deep learning system consistently achieves the lowest median and mean, which means that the deep learning system is not biased towards only having skillful predictions for sea ice chart like sea ice concentration conditions. Since figure 38 show that the deep learning system outperforms the other products when targeted with data which has not been seen during training. Thus, the generalizability of the deep learning system is satisfactory. That the deep learning system is outperformed for the $\geq 10\%$ may be a consequence of the uncertainty related to the AMSR-2 observations. Where it was described in section 2.2.5 that the ASI sea ice algorithm exerts higher uncertainties for lower concentration values, mainly due to errors introduced through atmospheric interactions (Spreen et al., 2008). It is further noted the AMSR-2 observations are most certain at 65%, thus the deep learning system achieves the lowest NIIEE distribution for the cumulative contours which correspond to where the target sea ice concentration values are most certain.

Finally, the model trained with reduced classes was not shown in figure 39 to improve the contours which it targeted in common with the baseline model used in figure 38. This is consistent with the results discussed in section 7.2.5 with the mean annual NIIEE of the reduced class model being similar to the baseline model. However, whe not targeting the unphysical $\geq 10\%$ contour, the model is able to achieve comparable performance akin to the other products independent from the sea ice charts, as seen in the top left of figure 39. Thus, to achieve model generalizability in terms of other sea ice concentration products than the sea ice charts, the $\geq 10\%$ contour is advisable to remove as it hinders model performance at the contour.

7.4 Explainability

Two general approaches were used to determine the overall effect each predictor had on the deep learning system. The first approach was to leave one predictor out, and train the model with the remaining predictors. Although deep learning models fitted towards different pools of predictors are not directly comparable since a model is uniquely fitted to the available predictors, the overall skill of the differently fitted models can still be compared. The second approach involved modifying predictors in the test data set, such that model response to differently constructed out-of-distribution predictors could be measured. Out-of-distribution predictors signify that a input-channel or input combination is different than what was observed during training (DeVries and Taylor, 2018), which for the current work is reflected through replacing a predictor with uniform noise 42 or shuffling the sequence for only a predictor-variable while the other remain untouched 43. It is common for deep neural networks to provide incorrect predictions with a high level of confidence when fed meaningless predictors (DeVries and Taylor, 2018). However, given that the predictors for the current work are physically bound, although deemed nonsensical by the deep learning system the modification performed which causes the predictor to become out-of-sample can still be explained and in some cases related to physical processes. Moreover, given the limited size of the training data, it can be assumed that there exists a number of valid atmospheric configurations not seen by the deep learning system which to some extent can be replicated by the following experiments. Thus, a systematic model response to a consistent and explainable predictor modification may provide insight into model response which is not just artificial hallucination.

7.4.1 Model response to predictors

It was shown in figure 41 that removing either the recent sea ice chart or all AROME Arctic variables caused the deep learning system to achieve worse performance than persistence. In terms of removing the recent sea ice chart, this result indicates that the recent sea ice chart is the most important predictor for the deep learning system. This is followed up by figures 42 and 43, where the model response to both modifications to the recent sea ice chart caused the model performance to significantly worsen. The high values when replacing the recent sea ice chart with uniform noise in figure 42 indicates that when the model is strongly fitted to the recent sea ice chart since the response is very much higher than for any other predictor. Comparably, the seasonal NIIEE is lower when the recent sea ice chart is swapped within the distribution, indicating that the model is able to recreate a sea ice edge when fed a actual sea ice chart. Hence, given that the model is strongly fitted to recent sea ice chart, it can be assumed that the predictions made by the deep learning system are in general modifications of the recent ice chart. This may be an effect of the strong autocorrelation inherent to the sea ice charts (figure 4). Such

that when given a sea ice chart from a potential other season in figure 43, the forecast retains the seasonality inherent of predictor sea ice chart, which is not possible when the recent sea ice chart is replaced by uniform noise in figure 43. Thus explaining the large difference in seasonal NIIEE values for sea ice concentration in figures 42 and 43.

Figure 41 also shows that the deep learning forecast is not able to outperform persistence without atmospheric predictors from AROME Arctic. Thus, it seems to be the case that the deep learning system is not able to fit the presumed non-linear nature of sea ice dynamics (Grigoryev et al., 2022) without correlations from additional predictors such as the atmospheric conditions. Moreover, given the model responds well to relevant physical predictors, it can be assumed that further increasing the predictor pool with physical processes related to the MIZ may increase the skill of the forecast. Currently, the pool of predictors cover sea ice and atmospheric interactions, however the ice-ocean interactions present in the MIZ are missing. A driver of sea ice breakage in the MIZ is the interaction between the sea ice edge and surface waves, causing ice breakage which resonates with large scale dynamics and thermodynamical properties of the sea ice (Williams et al., 2013). Given that the model responds positively to physical predictors from the atmosphere, including relevant fields from a high resolution wave forecasting system such as Carrasco et al. (2022) where significant wave height and wave direction to be included as model predictors.

Figures 41, 42 and 43 also show that the model gains no skill from the linear sea ice trend computed from the OSI SAF observations. In figure 41, this is shown as model performance is similar with and without the predictor. In figure 43 it is seen as the trend follows the baseline and exerts no variability when swapped with trends from other seasons. It is however seen in figure 42 that replacing the trend with uniform noise improves some seasons, although stochastically. This may be show that the predictor have a very slight degrading effect on the deep learning forecast, however it is more likely to be stochastic hallucination as the other figures 41 or 43 does not recreate a similar trend.

7.4.2 Model inferred physics

The same precautions regarding out-of-distribution samples given in section 7.4 also apply to the synthetic AROME Arctic samples explored in section 6.2, since the synthetic fields are likely to contain combinations of valid values which have not been covered by the training dataset. With respect to the forecast errors obtained when comparing a non-synthetic forecast with a forecast where one or several AROME Arctic fields have been replaced with a synthetic variation in figure 44, no clear seasonal cycle to the NIIEE can be seen. Such a random change in NIIEE for the same synthetic field at different seasons may show that the model is able to infer some relationship between the state of a physical field and the other predictors. This is further demonstrated by the two synthetic fields

in the lower leftmost plot in figure 44 where the model improved the NIIEE when being fed no-winds or only positive winds in the x direction. Although there is possibility of these results occurring stochastically, it is noteworthy that there may be a possibility to engineer more ideal conditions if the state of other predictors are known.

These results also indicate that the model does interpret each atmospheric field separately at the beginning of the encoder before the first convolutional layer merges the different input channels into a single feature map, since the model is able to evolve the sea ice concentration field differently given different states of the atmosphere. Additionally, when figure 44 is seen in relation to swapping the order of a single AROME Arctic field in figure 43, the general model response to a modified state of the atmosphere is to degrade the forecast skill. Which is expected given that the 2-meter temperature field and the wind fields are assumed to have a strong correlation, such that when a uncorrelated atmospheric field is introduced to the predictor pool the encoded signal is weaker resulting in less skillful forecasts.

The response to some of the synthetic AROME Arctic predictions in figure 44 was shown in figure 45 as spatial anomalies. The top row of figure 45 show model response to synthetic wind fields, whereas the bottom row show model response to synthetic 2-meter temperature fields. A major difference between the top and bottom row of figure 45 is where sea ice growth and decline occurs. For the top row, where only the winds are synthetic, it appears that the spatial distribution of sea ice is located where sea ice already exists. However, for the 2-meter temperature synthetic fields in the bottom row, it is seen that the sea ice growth and decline occurs throughout the entirety of the scene, without relation to where the sea ice concentration is located in the recent sea ice chart predictor. This response is compliant with the expected physical interactions between the sea ice and atmosphere.

Firstly, the x and y-component of the wind is only able to affect sea ice dynamics for already existing sea ice (Spreen et al., 2011; Yu et al., 2020). From the upper leftmost plot in figure 45, it can be seen that then both wind components are directed in a negative direction, the extent of the sea ice edge is lowered. This is compliant with how the sea ice edge tend to be sharply defined when experiencing incoming winds, since the negative direction of the wind restricts the outward transportation of low concentration sea ice (Yu et al., 2020). The contrary seem to occur in the top right plot in figure 45, since the sea ice edge is relatively normal to the direction of the winds (only x wind in positive direction) causing the winds to transport sea ice away from the sea ice edge. This is reflected by some areas experiencing a wider MIZ, although with some inconsistencies which may be due to the positive x-wind not affecting the scene in isolation. Secondly, with synthetic 2-meter temperature fields the deep learning system is able to infer growth in areas where there are no sea ice in the recent sea ice chart similar to the occurrence of sea ice formation as a response to freezing temperatures (Hibler, 1979). Furthermore, the

lower rightmost plot in figure 45 show a 2-meter temperature field which linearly increases from the lowest possible to the highest possible values in the test data from bottom to top of the scene. The sea ice concentration response to the synthetic forcing is similar, with growth in the bottom half of the domain and growth in the top half, as expected with regards to the physical response where cold temperatures facilitate sea ice growth and vice versa (Hibler, 1979). These results indicate that the deep learning system is able to infer physical relationships and responses from the predictors without having learnt them explicitly. Furthermore, the deep learning system is able to recreate physical responses between predictors to some degree, without having the framework to simulate or resolve the underlying physics.

7.5 Explainable predictions

This work opted to implemented the seg-GradCAM technique (Vinogradova et al., 2020) to increase the transparency and explainability of the developed deep learning system. An initial attempt implementing the technique for the baseline model with a two day lead time at different target contours was shown in figure 46. The visually highlighted region seem to follow the marked sea ice edge, although the regions in figure 46 (b, c and d) also seem to include parts of Svalbard to a greater extent than (a) showing that the model look at different regions to determine the different contours. The only class each output layer is attempting to predict is whether each pixel belongs to the cumulative contour. Figure 46 show that the regions of the predictors which were important for predicting pixels as part of the cumulative contour were the pixels which constituted the predictor sea ice chart. Although for some contours the model is utilizing pixels outside the sea ice edge for the given contour, the region of the predictors which overlap with the non-highlighted portion of figure 46 were not important for predicting cumulative contours.

In section 7.2.1, is was shown that the shallow model without an scene-encompassing theoretical receptive field at the bottleneck outperformed the deeper model where each feature in the bottleneck had a receptive field which covered the bottleneck. Although it was discussed that the effective receptive field covered a fraction of the theoretical receptive field (Luo et al., 2017), based on figure 46 it can also be seen that by increasing the theoretical receptive field each encoded feature in the bottlenck are computed from a lot of features from unimportant spatial locations are included. On the contrary, a model such as the 256 U-Net architecture which only had a theoretical receptive field of 145 in the bottleneck would have a some features in the bottleneck only computed from important pixels, and some encoded features only computed from unimportant pixels. Thus, it may be the case that due to the clear dichotomy of low level features, the signal is easier to decode compared to when all features are a mix of important and unimportant pixels as for the 1028 U-Net architecture.

8 Conclusion and future outlook

A consequence of the operational aspect is the possibility to force decoupled NWP-systems with updated Sea Ice Concentration.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, URL https://www.tensorflow.org/, software available from tensorflow.org, 2015.

Adadi, A. and Berrada, M.: Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI), IEEE Access, 6, 52 138–52 160, https://doi.org/10.1109/access.2018.2870052, 2018.

Adelson, E. H.: On seeing stuff: the perception of materials by humans and machines, in: SPIE Proceedings, edited by Rogowitz, B. E. and Pappas, T. N., SPIE, https://doi.org/10.1117/12.429489, 2001.

Andersson, T. R., Hosking, J. S., Pérez-Ortiz, M., Paige, B., Elliott, A., Russell, C., Law, S., Jones, D. C., Wilkinson, J., Phillips, T., Byrne, J., Tietsche, S., Sarojini, B. B., Blanchard-Wrigglesworth, E., Aksenov, Y., Downie, R., and Shuckburgh, E.: Seasonal Arctic sea ice forecasting with probabilistic deep learning, Nature Communications, 12, https://doi.org/10.1038/s41467-021-25257-4, 2021.

Araujo, A., Norris, W., and Sim, J.: Computing Receptive Fields of Convolutional Neural Networks, Distill, 4, https://doi.org/10.23915/distill.00021, 2019.

Badrinarayanan, V., Kendall, A., and Cipolla, R.: SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 39, 2481–2495, https://doi.org/10.1109/tpami.2016.2644615, 2017.

Relatere seg grad cam til shared en-coder

Relatere seg-GradCAM til fig-urene over, mod-ellen er sterkt fittet til t2m, men den gir nega-tive effek-ter. Her tar vi inn sea ice edge og t2m kontur sam-men-heng

Forklare case study, fork-lare hva som skjer.

Balmaseda, M. A., Mogensen, K., and Weaver, A. T.: Evaluation of the ECMWF ocean reanalysis system ORAS4, Quarterly Journal of the Royal Meteorological Society, 139, 1132–1161, https://doi.org/https://doi.org/10.1002/qj.2063, URL https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.2063, 2013.

Barry, R. G., Serreze, M. C., Maslanik, J. A., and Preller, R. H.: The Arctic Sea Ice-Climate System: Observations and modeling, Reviews of Geophysics, 31, 397, https://doi.org/10.1029/93rg01998, 1993.

Batrak, Y. and Müller, M.: On the warm bias in atmospheric reanalyses induced by the missing snow over Arctic sea-ice, Nature Communications, 10, https://doi.org/10.1038/s41467-019-11975-3, 2019.

Bauer, P., Beljaars, A., Ahlgrimm, M., Bechtold, P., Bidlot, J.-R., Bonavita, M., Bozzo, A., Forbes, R., Hólm, E., Leutbecher, M., Lopez, P., Magnusson, L., Prates, F., Rodwell, M., Sandu, I., Untch, A., and Vitart, F.: Model Cycle 38r2: Components and Performance, https://doi.org/10.21957/XC1R0LJ6L, 2013.

Bridle, J. S.: Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition, in: Neurocomputing, pp. 227–236, Springer Berlin Heidelberg, https://doi.org/10.1007/978-3-642-76153-9_28, 1990.

Carrasco, A., Øyvind Saetra, Burud, A., Müller, M., and Melsom, A.: PRODUCT USER MANUAL For Arctic Ocean Wave Analysis and Forecasting Products ARCTIC_ANALYSIS_FORECAST_WAV_002_014, Tech. rep., 2022.

Casati, B., Wilson, L. J., Stephenson, D. B., Nurmi, P., Ghelli, A., Pocernich, M., Damrath, U., Ebert, E. E., Brown, B. G., and Mason, S.: Forecast verification: current status and future directions, Meteorological Applications, 15, 3–18, https://doi.org/10.1002/met.52, 2008.

Cavalieri, D. J. and Parkinson, C. L.: Arctic sea ice variability and trends, 1979–2010, The Cryosphere, 6, 881–889, https://doi.org/10.5194/tc-6-881-2012, 2012.

Chan, J. Y.-L., Leow, S. M. H., Bea, K. T., Cheng, W. K., Phoong, S. W., Hong, Z.-W., and Chen, Y.-L.: Mitigating the Multicollinearity Problem and Its Machine Learning Approach: A Review, Mathematics, 10, 1283, https://doi.org/10.3390/math10081283, 2022.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L.: DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs, IEEE Transactions on Pattern Analysis and Machine Intelligence, 40, 834–848, https://doi.org/10.1109/tpami.2017.2699184, 2018.

Chollet, F. et al.: Keras, https://keras.io, 2015.

Ciresan, D., Giusti, A., Gambardella, L., and Schmidhuber, J.: Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images, in: Advances in Neural Information Processing Systems, edited by Pereira, F., Burges, C., Bottou, L., and Weinberger, K., vol. 25, Curran Associates, Inc., URL https://proceedings.neurips.cc/paper/2012/file/459a4ddcb586f24efd9395aa7662bc7c-Paper.pdf, 2012a.

Ciresan, D., Meier, U., and Schmidhuber, J.: Multi-column deep neural networks for image classification, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, https://doi.org/10.1109/cvpr.2012.6248110, 2012b.

Colony, R. and Thorndike, A. S.: An estimate of the mean field of Arctic sea ice motion, Journal of Geophysical Research, 89, 10 623, https://doi.org/10.1029/jc089ic06p10623, 1984.

Comiso, J. C., Cavalieri, D. J., Parkinson, C. L., and Gloersen, P.: Passive microwave algorithms for sea ice concentration: A comparison of two techniques, Remote Sensing of Environment, 60, 357–384, https://doi.org/10.1016/s0034-4257(96)00220-9, 1997.

Comiso, J. C., Meier, W. N., and Gersten, R.: Variability and trends in the Arctic Sea ice cover: Results from different techniques, Journal of Geophysical Research: Oceans, 122, 6883–6900, https://doi.org/10.1002/2017jc012768, 2017.

Crawshaw, M.: Multi-Task Learning with Deep Neural Networks: A Survey, https://doi.org/10.48550/ARXIV.2009.09796, 2020.

Dee, D. P., Uppala, S. M., Simmons, A. J., Berrisford, P., Poli, P., Kobayashi, S., Andrae, U., Balmaseda, M. A., Balsamo, G., Bauer, P., Bechtold, P., Beljaars, A. C. M., van de Berg, L., Bidlot, J., Bormann, N., Delsol, C., Dragani, R., Fuentes, M., Geer, A. J., Haimberger, L., Healy, S. B., Hersbach, H., Hólm, E. V., Isaksen, L., Kållberg, P., Köhler, M., Matricardi, M., McNally, A. P., Monge-Sanz, B. M., Morcrette, J.-J., Park, B.-K., Peubey, C., de Rosnay, P., Tavolato, C., Thépaut, J.-N., and Vitart, F.: The ERA-Interim reanalysis: configuration and performance of the data assimilation system, Quarterly Journal of the Royal Meteorological Society, 137, 553–597, https://doi.org/https://doi.org/10.1002/qj.828, URL https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.828, 2011.

DeVries, T. and Taylor, G. W.: Learning Confidence for Out-of-Distribution Detection in Neural Networks, https://doi.org/10.48550/ARXIV.1802.04865, 2018.

Dinessen, F., Hackett, B., and Kreiner, M. B.: Product User Manual For Regional High Resolution Sea Ice Charts Svalbard and Greenland Region, Tech. rep., Norwegian Meteorological Institute, 2020.

Dukhovskoy, D. S., Ubnoske, J., Blanchard-Wrigglesworth, E., Hiester, H. R., and Proshutinsky, A.: Skill metrics for evaluation and comparison of sea ice models, Journal of Geophysical Research: Oceans, 120, 5910–5931, https://doi.org/10.1002/2015jc010989, 2015.

Eguíluz, V. M., Fernández-Gracia, J., Irigoien, X., and Duarte, C. M.: A quantitative assessment of Arctic shipping in 2010–2014, Scientific Reports, 6, https://doi.org/10.1038/srep30682, 2016.

European Union-Copernicus Marine Service: Arctic Ocean Sea Ice Analysis and Forecast, https://doi.org/10.48670/MOI-00004, 2020.

Fritzner, S., Graversen, R., and Christensen, K. H.: Assessment of High-Resolution Dynamical and Machine Learning Models for Prediction of Sea Ice Concentration in a

Regional Application, 125, https://doi.org/10.1029/2020jc016277, neural Networks for predicting Sea-Ice concentration are only slightly more accurate than persistence forecasting for short-term predictions., 2020.

Fukushima, K.: Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, Biological Cybernetics, 36, 193–202, https://doi.org/10.1007/bf00344251, 1980.

Geman, S., Bienenstock, E., and Doursat, R.: Neural Networks and the Bias/Variance Dilemma, Neural Computation, 4, 1–58, https://doi.org/10.1162/neco.1992.4.1.1, 1992.

Goessling, H. F. and Jung, T.: A probabilistic verification score for contours: Methodology and application to Arctic ice-edge forecasts, Quarterly Journal of the Royal Meteorological Society, 144, 735–743, https://doi.org/10.1002/qj.3242, 2018.

Goessling, H. F., Tietsche, S., Day, J. J., Hawkins, E., and Jung, T.: Predictability of the Arctic sea ice edge, Geophysical Research Letters, 43, 1642–1650, https://doi.org/10.1002/2015gl067232, 2016.

Graves, A., rahman Mohamed, A., and Hinton, G.: Speech recognition with deep recurrent neural networks, in: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, https://doi.org/10.1109/icassp.2013.6638947, 2013.

Grigoryev, T., Verezemskaya, P., Krinitskiy, M., Anikin, N., Gavrikov, A., Trofimov, I., Balabin, N., Shpilman, A., Eremchenko, A., Gulev, S., Burnaev, E., and Vanovskiy, V.: Data-Driven Short-Term Daily Operational Sea Ice Regional Forecasting, Remote Sensing, 14, https://doi.org/10.3390/rs14225837, URL https://www.mdpi.com/2072-4292/14/22/5837, 2022.

Haiden, T., Janousek, M., Vitart, F., Ben-Bouallegue, Z., Ferranti, L., Prates, F., and Richardson, D.: Evaluation of ECMWF forecasts, including the 2021 upgrade, https://doi.org/10.21957/XQNU5O3P, 2022.

He, K., Zhang, X., Ren, S., and Sun, J.: Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, https://doi.org/10.48550/ARXIV.1502.01852, 2015a.

He, K., Zhang, X., Ren, S., and Sun, J.: Deep Residual Learning for Image Recognition, 2015b.

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu, R., Schepers, D., Simmons, A., Soci, C., Abdalla, S., Abellan, X., Balsamo, G., Bechtold, P., Biavati, G., Bidlot, J., Bonavita, M., De Chiara, G., Dahlgren, P., Dee, D., Diamantakis, M., Dragani, R., Flemming, J., Forbes, R., Fuentes, M., Geer, A., Haimberger, L., Healy, S., Hogan, R. J., Hólm, E., Janisková, M., Keeley, S., Laloyaux, P., Lopez, P., Lupu, C., Radnoti, G., de Rosnay, P., Rozum, I., Vamborg, F., Villaume, S., and Thépaut, J.-N.: The ERA5 global reanalysis, Quarterly Journal of the Royal Meteorological Society, 146, 1999–2049, https://doi.org/https://doi.org/10.1002/qj.3803, URL https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.3803, 2020.

Hibler, W. D.: A Dynamic Thermodynamic Sea Ice Model, Journal of Physical Oceanography, 9, 815–846, https://doi.org/10.1175/1520-0485(1979)009<0815:adtsim>2.0.co;2, 1979.

Ho, J.: The implications of Arctic sea ice decline on shipping, Marine Policy, 34, 713–715, https://doi.org/10.1016/j.marpol.2009.10.009, 2010.

Holland, P. R. and Kimura, N.: Observed Concentration Budgets of Arctic and Antarctic Sea Ice, Journal of Climate, 29, 5241–5249, https://doi.org/10.1175/jcli-d-16-0121.1, 2016.

Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. Q.: Densely Connected Convolutional Networks, https://doi.org/10.48550/ARXIV.1608.06993, 2016.

Hunke, E. C. and Dukowicz, J. K.: An Elastic–Viscous–Plastic Model for Sea Ice Dynamics, Journal of Physical Oceanography, 27, 1849–1867, https://doi.org/10.1175/1520-0485(1997)027<1849:aevpmf>2.0.co;2, 1997.

Hunke, E. C., Lipscomb, W. H., Turner, A. K., Jeffery, N., and Elliott, S.: CICE: the Los Alamos Sea Ice Model Documentation and Software User's Manual Version 5.1 LA-CC-06-012, techreport, Los Alamos National Laboratory, Los Alamos NM 87545, 2015.

Ioffe, S. and Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, 2015.

Jha, A., Kumar, A., Pande, S., Banerjee, B., and Chaudhuri, S.: MT-UNET: A Novel U-Net Based Multi-Task Architecture For Visual Scene Understanding, in: 2020 IEEE International Conference on Image Processing (ICIP), IEEE, https://doi.org/10.1109/icip40778.2020.9190695, 2020.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T.: Caffe: Convolutional Architecture for Fast Feature Embedding, https://doi.org/10.48550/ARXIV.1408.5093, 2014.

Johnson, S. J., Stockdale, T. N., Ferranti, L., Balmaseda, M. A., Molteni, F., Magnusson, L., Tietsche, S., Decremer, D., Weisheimer, A., Balsamo, G., Keeley, S. P. E., Mogensen, K., Zuo, H., and Monge-Sanz, B. M.: SEAS5: the new ECMWF seasonal forecast system, Geoscientific Model Development, 12, 1087–1117, https://doi.org/10.5194/gmd-12-1087-2019, URL https://gmd.copernicus.org/articles/12/1087/2019/, 2019.

Kaur, S., Ehn, J. K., and Barber, D. G.: Pan-arctic winter drift speeds and changing patterns of sea ice motion: 1979–2015, Polar Record, 54, 303–311, https://doi.org/10.1017/s0032247418000566, 2018.

Kern, S., Lavergne, T., Notz, D., Pedersen, L. T., Tonboe, R. T., Saldo, R., and Sørensen, A. M.: Satellite passive microwave sea-ice concentration data set intercomparison: closed ice and ship-based observations, The Cryosphere, 13, 3261–3307, https://doi.org/10.5194/tc-13-3261-2019, 2019.

Kern, S., Lavergne, T., Notz, D., Pedersen, L. T., and Tonboe, R.: Satellite passive

microwave sea-ice concentration data set inter-comparison for Arctic summer conditions, The Cryosphere, 14, 2469–2493, https://doi.org/10.5194/tc-14-2469-2020, 2020.
Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization, https://doi.org/10.48550/ARXIV.1412.6980, 2014.
Kirillov, A., He, K., Girshick, R., Rother, C., and Dollár, P.: Panoptic Segmentation, https://doi.org/10.48550/ARXIV.1801.00868, 2018.
Køltzow, M., Schyberg, H., Støylen, E., and Yang, X.: Value of the Copernicus Arctic Regional Reanalysis (CARRA) in representing near-surface temperature and wind speed in the north-east European Arctic, Polar Research, 41, https://doi.org/10.33265/polar.v41.8002, 2022.
Kreiner, M. B., Wulf, T., Jakobsen, J., Nielsen, A. A., and Pedersen, L. T.: Poster: Inter- and intra-analyst ice edge assessment, https://doi.org/10.6084/M9.FIGSHARE.22312648.V1, 2023.
Kristensen, N. M., JensBDebernard, SebastianMaartensson, Keguang Wang, and Hedstrom, K.: Metno/Metroms: Version 0.3 - Before Merge, https://doi.org/10.5281/ZENODO.1046114, 2017.
Krizhevsky, A., Sutskever, I., and Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks, in: Advances in Neural Information Processing Systems, edited by Pereira, F., Burges, C., Bottou, L., and Weinberger, K., vol. 25, Curran Associates, Inc., URL https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf, 2012.
Lavelle, J., Tonboe, R., Tian, T., Pfeiffer, R.-H., and Howe, E.: Product User Manual for the OSI SAF AMSR-2 Global Sea Ice Concentration Product OSI-408, Tech. Rep. 1.1, Danish Meteorological Institute, 2016.
Lavelle, J., Tonboe, R., Jensen, M. B., and Howe, E.: Validation Report for OSI SAF Global Sea Ice Concentration Product OSI-401-b, Tech. Rep. 1.2, Danish Meteorological Institute, 2017.
Lavergne, T., Sørensen, A. M., Kern, S., Tonboe, R., Notz, D., Aaboe, S., Bell, L., Dybkjær, G., Eastwood, S., Gabarro, C., Heygster, G., Killie, M. A., Brandt Kreiner, M., Lavelle, J., Saldo, R., Sandven, S., and Pedersen, L. T.: Version 2 of the EUMETSAT OSI SAF and ESA CCI sea-ice concentration climate data records, The Cryosphere, 13, 49–78, https://doi.org/10.5194/tc-13-49-2019, URL https://tc.copernicus.org/articles/13/49/2019/, 2019a.
Lavergne, T., Tonboe, R., Lavelle, J., and Eastwood, S.: Algorithm Theoretical Basis Document for the OSI SAF Global Sea Ice Concentration Climate Data Record OSI-450, OSI-430-b, techreport 1.2, 2019b.
LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D.: Backpropagation Applied to Handwritten Zip Code Recognition, Neural Computation, 1, 541–551, https://doi.org/10.1162/neco.1989.1.4.541, 1989.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P.: Focal Loss for Dense Object Detection, 2017.

Linardatos, P., Papastefanopoulos, V., and Kotsiantis, S.: Explainable AI: A Review of Machine Learning Interpretability Methods, Entropy, 23, 18, https://doi.org/10.3390/e23010018, 2020.

Liu, G., Reda, F. A., Shih, K. J., Wang, T.-C., Tao, A., and Catanzaro, B.: Image Inpainting for Irregular Holes Using Partial Convolutions, https://doi.org/10.48550/ARXIV.1804.07723, 2018.

Liu, Y., Bogaardt, L., Attema, J., and Hazeleger, W.: Extended Range Arctic Sea Ice Forecast with Convolutional Long-Short Term Memory Networks, Monthly Weather Review, https://doi.org/10.1175/mwr-d-20-0113.1, 2021.

Long, J., Shelhamer, E., and Darrell, T.: Fully convolutional networks for semantic segmentation, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, https://doi.org/10.1109/cvpr.2015.7298965, 2015.

Lopes, P., Silva, E., Braga, C., Oliveira, T., and Rosado, L.: XAI Systems Evaluation: A Review of Human and Computer-Centred Methods, Applied Sciences, 12, 9423, https://doi.org/10.3390/app12199423, 2022.

Lundberg, S. M. and Lee, S.-I.: A Unified Approach to Interpreting Model Predictions, in: Advances in Neural Information Processing Systems, edited by Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., vol. 30, Curran Associates, Inc., URL https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf, 2017.

Luo, W., Li, Y., Urtasun, R., and Zemel, R.: Understanding the Effective Receptive Field in Deep Convolutional Neural Networks, https://doi.org/10.48550/ARXIV.1701.04128, 2017.

Melsheimer, C.: ASI Version 5 Sea Ice Concentration User Guide, Tech. rep., Institute of Environmental Physics, University of Bremen, 2019.

Melsom, A., Palerme, C., and Müller, M.: Validation metrics for ice edge position forecasts, Ocean Science, 15, 615–630, https://doi.org/10.5194/os-15-615-2019, 2019.

Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., and Wu, H.: Mixed Precision Training, 2017.

Müller, M., Batrak, Y., Kristiansen, J., Køltzow, M. A. Ø., Noer, G., and Korosov, A.: Characteristics of a Convective-Scale Weather Forecasting System for the European Arctic, Monthly Weather Review, 145, 4771–4787, https://doi.org/10.1175/mwr-d-17-0194.1, 2017.

Nair, V. and Hinton, G.: Rectified Linear Units Improve Restricted Boltzmann Machines Vinod Nair, vol. 27, pp. 807–814, 2010.

Noh, H., Hong, S., and Han, B.: Learning Deconvolution Network for Semantic Segmen-

tation, in: 2015 IEEE International Conference on Computer Vision (ICCV), IEEE, https://doi.org/10.1109/iccv.2015.178, 2015.

Notz, D. and Community, S.: Arctic Sea Ice in CMIP6, Geophysical Research Letters, 47, https://doi.org/10.1029/2019gl086749, 2020.

Obite, C. P., Olewuezi, N. P., Ugwuanyim, G. U., and Bartholomew, D. C.: Multicollinearity Effect in Regression Analysis: A Feed Forward Artificial Neural Network Approach, Asian Journal of Probability and Statistics, pp. 22–33, https://doi.org/10.9734/ajpas/2020/v6i130151, 2020.

Ólason, E., Boutin, G., Korosov, A., Rampal, P., Williams, T., Kimmritz, M., Dansereau, V., and Samaké, A.: A New Brittle Rheology and Numerical Framework for Large-Scale Sea-Ice Models, Journal of Advances in Modeling Earth Systems, 14, https://doi.org/10.1029/2021ms002685, 2022.

Palerme, C., Müller, M., and Melsom, A.: An Intercomparison of Verification Scores for Evaluating the Sea Ice Edge Position in Seasonal Forecasts, Geophysical Research Letters, 46, 4757–4763, https://doi.org/10.1029/2019gl082482, 2019.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S.: PyTorch: An Imperative Style, High-Performance Deep Learning Library, https://doi.org/10.48550/ARXIV.1912.01703, 2019.

RADU, M. D., COSTEA, I. M., and STAN, V. A.: Automatic Traffic Sign Recognition Artificial Inteligence - Deep Learning Algorithm, in: 2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), IEEE, https://doi.org/10.1109/ecai50035.2020.9223186, 2020.

Ronneberger, O., Fischer, P., and Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation, in: Lecture Notes in Computer Science, pp. 234–241, Springer International Publishing, https://doi.org/10.1007/978-3-319-24574-4_28, 2015.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J.: Learning representations by back-propagating errors, Nature, 323, 533–536, https://doi.org/10.1038/323533a0, 1986.

Röhrs, J., Gusdal, Y., Rikardsen, E., Moro, M. D., Brændshøi, J., Kristensen, N. M., Fritzner, S., Wang, K., Sperrevik, A. K., Idžanović, M., Lavergne, T., Debernard, J., and Christensen, K. H.: "in prep for GMD" An operational data-assimilative coupled ocean and sea ice ensembleprediction model for the Barents Sea and Svalbard, p. 20, 2022.

Sakov, P., Counillon, F., Bertino, L., Lisæter, K. A., Oke, P. R., and Korablev, A.: TOPAZ4: an ocean-sea ice data assimilation system for the North Atlantic and Arctic, Ocean Science, 8, 633–656, https://doi.org/10.5194/os-8-633-2012, 2012.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D.: Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Lo-

calization, International Journal of Computer Vision, 128, 336–359, https://doi.org/10.1007/s11263-019-01228-7, 2016.

Serreze, M. C. and Meier, W. N.: The Arctic's sea ice cover: trends, variability, predictability, and comparisons to the Antarctic, Annals of the New York Academy of Sciences, 1436, 36–53, https://doi.org/https://doi.org/10.1111/nyas.13856, 2019.

Shorten, C. and Khoshgoftaar, T. M.: A survey on Image Data Augmentation for Deep Learning, Journal of Big Data, 6, https://doi.org/10.1186/s40537-019-0197-0, 2019.

Simonyan, K. and Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition, https://doi.org/10.48550/ARXIV.1409.1556, 2014.

Simonyan, K., Vedaldi, A., and Zisserman, A.: Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, https://doi.org/10.48550/ARXIV.1312.6034, 2013.

Smith, D. M.: Extraction of winter total sea-ice concentration in the Greenland and Barents Seas from SSM/I data, International Journal of Remote Sensing, 17, 2625–2646, https://doi.org/10.1080/01431169608949096, 1996.

Snow, A. D., Whitaker, J., Cochran, M., Van Den Bossche, J., Mayo, C., Miara, I., Cochrane, P., De Kloe, J., Karney, C., , Filipe, Couwenberg, B., Lostis, G., Dearing, J., Ouzounoudis, G., Jurd, B., Gohlke, C., Hoese, D., Itkin, M., May, R., Little, B., , Heitor, Shadchin, A., Wiedemann, B. M., Barker, C., Willoughby, C., DWesl, Hemberger, D., Haberthür, D., and Popov, E.: pyproj4/pyproj: 3.4.1 Release, https://doi.org/10.5281/ZENODO.2592232, 2022.

Spreen, G., Kaleschke, L., and Heygster, G.: Sea ice remote sensing using AMSR-E 89-GHz channels, Journal of Geophysical Research, 113, https://doi.org/10.1029/2005jc003384, 2008.

Spreen, G., Kwok, R., and Menemenlis, D.: Trends in Arctic sea ice drift and role of wind forcing: 1992-2009, Geophysical Research Letters, 38, n/a–n/a, https://doi.org/10.1029/2011gl048970, 2011.

Strong, C.: Atmospheric influence on Arctic marginal ice zone position and width in the Atlantic sector, February–April 1979–2010, Climate Dynamics, 39, 3091–3102, https://doi.org/10.1007/s00382-012-1356-6, 2012.

Sundararajan, M., Taly, A., and Yan, Q.: Axiomatic Attribution for Deep Networks, https://doi.org/10.48550/ARXIV.1703.01365, 2017.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A.: Going Deeper with Convolutions, https://doi.org/10.48550/ARXIV.1409.4842, 2014.

Sørensen, A. M., Lavergne, T., and Eastwood, S.: Global Sea Ice Concentration Climate Data Record Product Uses Manual Product OSI-450 & OSI-430-b, Tech. Rep. 2.1, Norwegian Meteorological Institute, 2021.

Tonboe, R., Lavelle, J., Pfeiffer, R.-H., and Howe, E.: Product User Manual for OSI SAF Global Sea Ice Concentration, Tech. Rep. 1.6, Danish Meteorological Institute, 2017.

Veland, S., Wagner, P., Bailey, D., Everett, A., Goldstein, M., Hermann, R., Hjort-Larsen, T., Hovelsrud, G., Hughes, N., Kjøl, A., Li, X., Lynch, A., Müller, M., Olsen, J., Palerme, C., Pedersen, J. L., Rinaldo, ., Stephenson, S., and Storelvmo, T.: Knowledge needs in sea ice forecasting for navigation in Svalbard and the High Arctic, Tech. Rep. NF-rapport 4/2021, Svalbard Strategic Grant, Svalbard Science Forum, 2021.

Vinogradova, K., Dibrov, A., and Myers, G.: Towards Interpretable Semantic Segmentation via Gradient-Weighted Class Activation Mapping (Student Abstract), Proceedings of the AAAI Conference on Artificial Intelligence, 34, 13 943–13 944, https://doi.org/10.1609/aaai.v34i10.7244, 2020.

Wagner, P. M., Hughes, N., Bourbonnais, P., Stroeve, J., Rabenstein, L., Bhatt, U., Little, J., Wiggins, H., and Fleming, A.: Sea-ice information and forecast needs for industry maritime stakeholders, Polar Geography, 43, 160–187, https://doi.org/10.1080/1088937x.2020.1766592, 2020.

Wang, L., Scott, K., and Clausi, D.: Sea Ice Concentration Estimation during Freeze-Up from SAR Imagery Using a Convolutional Neural Network, Remote Sensing, 9, 408, https://doi.org/10.3390/rs9050408, 2017.

Williams, T., Korosov, A., Rampal, P., and Ólason, E.: Presentation and evaluation of the Arctic sea ice forecasting system neXtSIM-F, The Cryosphere, 15, 3207–3227, https://doi.org/10.5194/tc-15-3207-2021, 2021.

Williams, T. D., Bennetts, L. G., Squire, V. A., Dumont, D., and Bertino, L.: Wave–ice interactions in the marginal ice zone. Part 1: Theoretical foundations, Ocean Modelling, 71, 81–91, https://doi.org/10.1016/j.ocemod.2013.05.010, 2013.

Wu, M.-Y., Wu, Y., Yuan, X.-Y., Chen, Z.-H., Wu, W.-T., and Aubry, N.: Fast Prediction of Flow Field around Airfoils Based on Deep Convolutional Neural Network, Applied Sciences, 12, 12 075, https://doi.org/10.3390/app122312075, 2022.

Wu, Y. and He, K.: Group Normalization, 2018.

Yamashita, R., Nishio, M., Do, R. K. G., and Togashi, K.: Convolutional neural networks: an overview and application in radiology, Insights into Imaging, 9, 611–629, https://doi.org/10.1007/s13244-018-0639-9, 2018.

Yu, T. and Zhu, H.: Hyper-Parameter Optimization: A Review of Algorithms and Applications, https://doi.org/10.48550/ARXIV.2003.05689, 2020.

Yu, X., Rinke, A., Dorn, W., Spreen, G., Lüpkes, C., Sumata, H., and Gryanik, V. M.: Evaluation of Arctic sea ice drift and its dependency on near-surface wind and sea ice conditions in the coupled regional climate model HIRHAM–NAOSIM, The Cryosphere, 14, 1727–1746, https://doi.org/10.5194/tc-14-1727-2020, 2020.

Zampieri, L., Goessling, H. F., and Jung, T.: Predictability of Antarctic Sea Ice Edge on Subseasonal Time Scales, Geophysical Research Letters, 46, 9719–9727, https://doi.org/10.1029/2019gl084096, 2019.

Zeiler, M. D., Krishnan, D., Taylor, G. W., and Fergus, R.: Deconvolutional networks, in:

2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, https://doi.org/10.1109/cvpr.2010.5539957, 2010.
Zhang, Z., Luo, P., Loy, C. C., and Tang, X.: Facial Landmark Detection by Deep Multi-task Learning, in: Computer Vision – ECCV 2014, pp. 94–108, Springer International Publishing, https://doi.org/10.1007/978-3-319-10599-4_7, 2014.
Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A.: Learning Deep Features for Discriminative Localization, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, https://doi.org/10.1109/cvpr.2016.319, 2016.

9 Supporting Figures

10 Poster contribution to The 11th International Workshop on Sea Ice Modelling, Assimilation, Observations, Predictions and Verification

Developing a deep learning model for short term and high resolution prediction of WMO sea ice concentration categories

Are Frode Kvanum[1], Cyril Palerme[1], Malte Müller[1], Jean Rabault[1], Nick Hughes[2]

[1] Norwegian Meteorological Institute, Oslo, Norway (Contact: arefk@met.no)
[2] Ice Service, Norwegian Meteorological Institute, Tromsø, Norway

Summary of forecast production

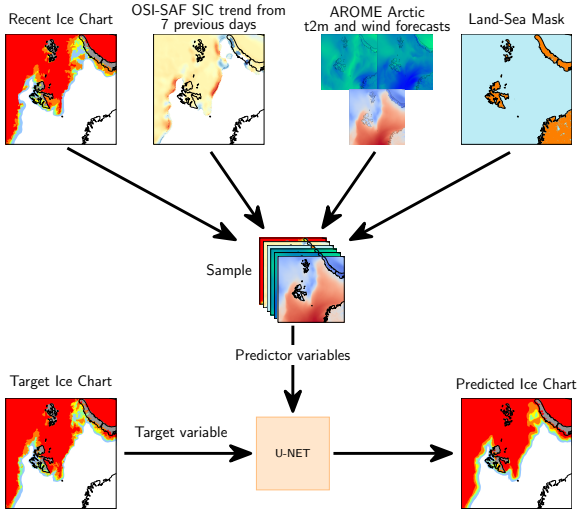


Figure 1: The deep learning forecasting system workflow

Introduction

- Sea ice concentration prediction targeting km-scale resolution is challenging.
- Maritime operators in the Arctic are lacking high resolution and high frequency sea ice forecasts for tactical decision making
- Deep learning systems are computationally lightweight, and can create a forecast on a consumer computer in minutes. Training the deep learning system is done on a cluster, once.

Model development

- The deep learning model is based on the U-Net architecture (image to image, pixelwise prediction).
- Each WMO sea ice concentration category is predicted separately.
- The final forecast is the pixelwise sum of the individually predicted sea ice concentration contours.
- 1km resolution (1792 x 1792 grid points) with (1 – 3) day lead time.
- Training period: 2019 – 2020 (288 samples).
- Testing period: 2022 (147 samples).

Input variables

- Sea ice concentration from the sea ice charts at t_0 produced by the Norwegian Ice Service (1km).
- Linear sea ice trend derived from Ocean and Sea Ice Satellite Apllication Facility passive microwave (SSMIS) using the seven previous days (10km).
- 2 meter temperature and 10 meter winds from a regional NWP system (AROME Arctic) (2.5km). Time steps between forecast bulletin date and target valid date is reduced to a mean-value field that projects temporal information onto a single time step.
- Land sea mask from AROME Arctic (2.5km).

Target variables

- Sea ice charts at time t_0+(1 – 3) days relative to the predictor date.
- The target sea ice concentration is divided into sea ice concentration contours following the WMO sea ice categories

U-Net architecture

- 2,359,047 trainable parameters
- Training is performed on an Nvidia A100 GPU, and takes ~3 hours
- During training, the U-Net uses 52Gb of memory
- After training, a single prediction is made in 6 seconds with a CPU

Results

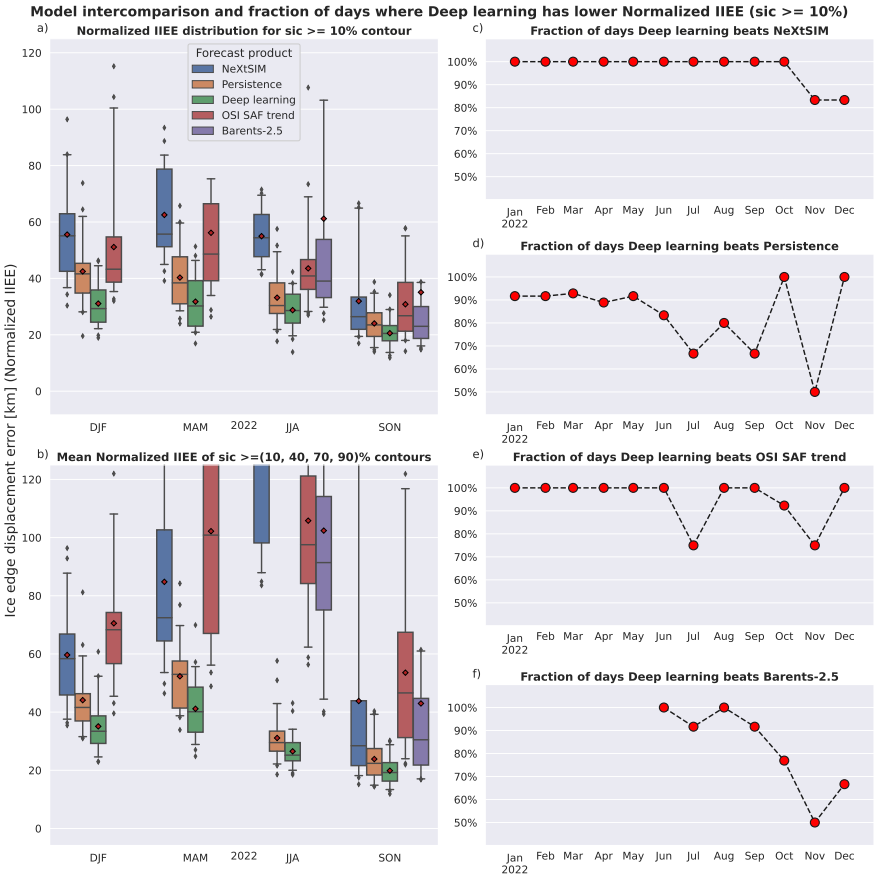


Figure 2: (a, b) Box and whisker plot of Normalized IIEE for different forecasting products as well as two benchmarks (Persistence and OSI SAF linear trend). The boxes cover the interquartile range. Whiskers denote the 5th and 95th percentiles. (c,d,e,f) Percentage of days where the Deep learning forecast achieves a lower Normalized IIEE score than the compared to product. (e) OSI SAF trend is a linear trend computed from the past 7 days. (f) Barents-2.5 is an in-development ocean and sea ice model implemented at MET Norway.

Model intercomparison

- The ice edge displacement error (Normalized Integrated Ice Edge Error) for an ice edge defined at the >=10% WMO sea ice concentration contour has on average been improved by **28%** between the four validational products.
- The deep learning model improves **90%** of the forecasted dates between the four validational products, with regards to achieving a lower Normalized IIEE for the >=10% concentration contour.

Comparing persistence with a deep learning prediction

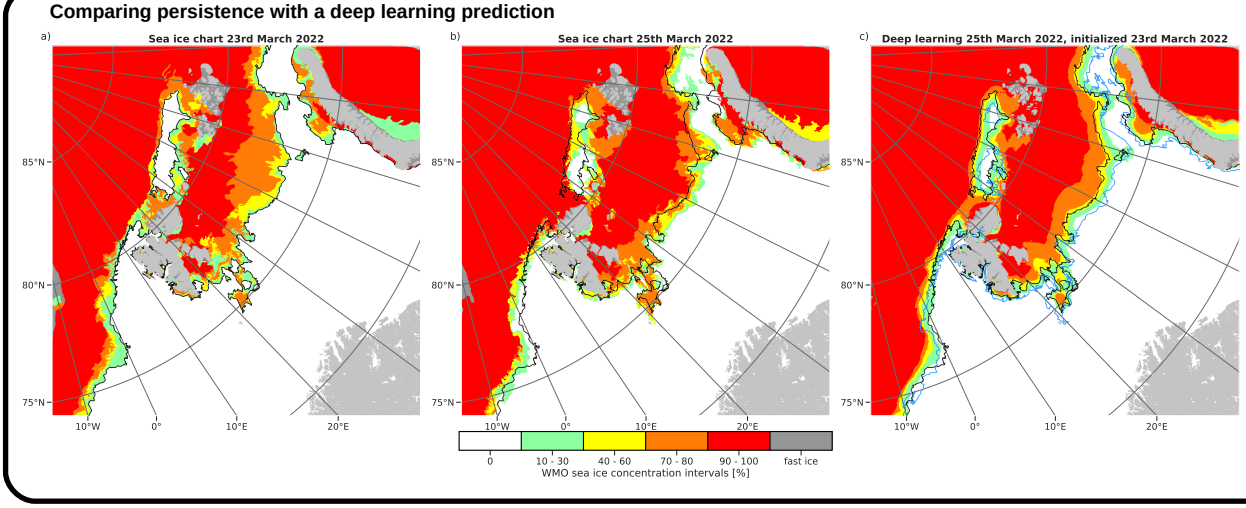


Figure 3: (a) Sea ice chart produced on 23 March 2022. (b) Sea ice chart produced on 25 March 2022. (c) Deep learning prediction for 25 March 2022, with a 2 day lead time. The sea ice chart in (a) was among the input variables for (c). The black line in (a,b,c) is the ice edge for (a) given a >=10% threshold. The blue line in (c) is the ice edge for (b) given a >=10% threshold.

- Persistence sea ice edge displacement error = **65km**.
- Deep learning sea ice edge displacement error = **37km**.
- The displacement error was computed with regards to the >=10% concentration contour.

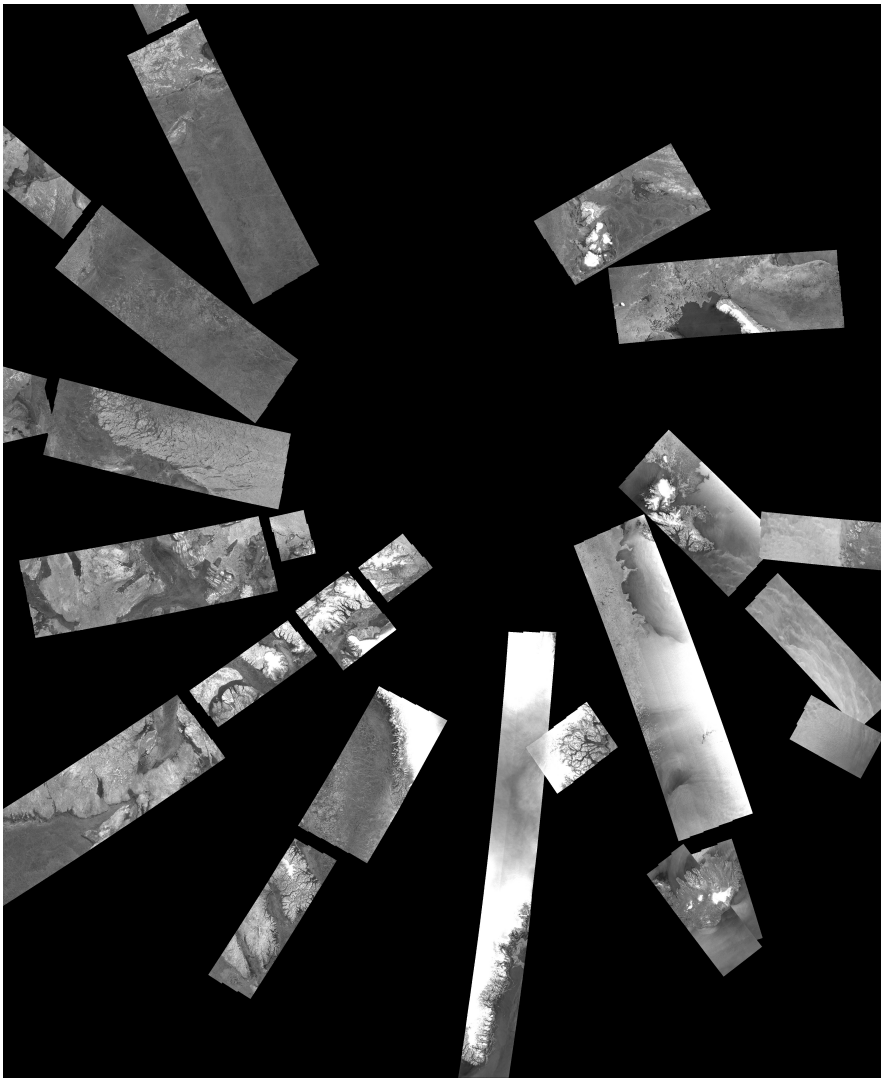


Figure 55: Daily SAR observations of the Arctic from Sentinel 1A 23 Jan 2023