Machine learning-based cloud forecast corrections for fusions of numerical weather prediction model and satellite data

Chuyen Nguyen, Jason E. Nachamkin, David Sidoti, Jacob Gull, Adam Bienkowski, Rich Bankert, and Melinda Surratt

a American Associations for Engineering Education (ASEE), Monterey, California
b Naval Research Laboratory, Monterey, California
c DeVine Consulting, Fremont, CA
d University of Connecticut, Storrs, Connecticut
e Naval Research Laboratory, Monterey, California - Retired

Corresponding author: Chuyen Nguyen, chuyen.nguyen.ctr@nrlmry.navy.mil

Early Online Release: This preliminary version has been accepted for publication in Artificial Intelligence for the Earth Systems, may be fully cited, and has been assigned DOI 10.1175/AIES-D-22-0072.1. The final typeset copyedited article will replace the EOR at the above DOI when it is published.

© 2023 American Meteorological Society
ABSTRACT: Given the diversity of cloud forcing mechanisms, it is difficult to classify and characterize all cloud types through the depth of a specific troposphere. Importantly, different cloud families often coexist even at the same atmospheric level. The Naval Research Laboratory (NRL) is developing machine learning-based cloud forecast models to fuse numerical weather prediction model and satellite data. These models were built for the dual purpose of understanding numerical weather prediction model error trends as well as improving the accuracy and sensitivity of the forecasts. The framework implements a Unet-Convolutional Neural Network (UNet-CNN) with features extracted from clouds observed by the Geostationary Operational Environmental Satellite (GOES-16) as well as clouds predicted by the Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS). The fundamental idea behind this novel framework is the application of UNet-CNN for separate variable sets extracted from GOES-16 and COAMPS to characterize and predict broad families of clouds that share similar physical characteristics. A quantitative assessment and evaluation based on an independent dataset for upper tropospheric (high) clouds suggests that UNet-CNN models capture the complexity and error trends of combined data from GOES-16 and COAMPS, and also improve forecast accuracy and sensitivity for different lead time forecasts (3-12 hours). This paper includes an overview of the machine learning frameworks as well as an illustrative example of their application, a comparative assessment of results for upper tropospheric clouds.
SIGNIFICANCE STATEMENT: Clouds are difficult to forecast because they require, in addition to spatial location, accurate height, depth, and cloud type. Satellite imagery is useful for verifying geographical location, but is limited by 2D technology. Multiple cloud types can coexist at various heights within the same pixel. In this situation, cloud/no cloud verification does not convey much information about why the forecast went wrong. Sorting clouds by physical attributes such as cloud top height, atmospheric stability and cloud thickness contributes to a better understanding since very different physical mechanisms produce various types of clouds. Using a fusion of numerical model outputs and GOES-16 observations, we derive variables related to atmospheric conditions that form and move the clouds for our Machine learning-based cloud forecast. The resulting verification over the US Mid-Atlantic region revealed our machine learning-based cloud forecast corrects systematic errors associated with high atmospheric clouds, and provides accurate and consistent cloud forecasts from 3 – 12 hour lead times.

1. Introduction

Even with recent advances in satellite imagery and numerical weather prediction models, accurate and rapid acquisition of cloud forecasts through the depth of the troposphere is still a challenge. Different cloud families often coexist at the same atmospheric level, and even within the same region, forcing mechanisms can vary significantly. Cloud forecasting is difficult because the 3D nature of clouds requires understanding both horizontal displacements and vertical cloud-height errors to accurately characterize cloud regimes and quantify their physical properties such as coverage, thickness, top height, and base height (Wood et al. (2011); Miller et al. (2014); Ryu et al. (2017); Minnis et al. (2008)).

Traditional cloud nowcasting approaches mainly focus on three methods: Numerical Weather Prediction (NWP), extrapolation methods such as Optical Flow (OF) that are based on radar or satellite images (Apke et al. (2016)), and knowledge-based expert systems that blend NWP and extrapolation techniques (Fritsch et al. (1998); Isaac et al. (2014); Zhang et al. (2019)).

Numerical Weather Prediction (NWP) models are based on equations pertaining to physical and dynamic atmospheric processes which are integrated forward in time on a three dimensional finite grid. NWP models typically carry systematic errors due to coarse grid resolution, uncertainties in physical parameterization and initial/boundary conditions (Veillette et al. (2018)). It is important...
to develop automatic post-processing and correction of the model output to reduce the systematic error and improve accuracy and operational capacity of NWP models.

Model output statistics (MOS) (Glahn and Lowry (1972)) is a common formulation for statistical interpretation of model output. MOS corrects persistent model biases by quantifying a statistical relationship between observed quantities as predictands and model-derived variables as predictors at specific lead times (Ebert et al. (2004)). There are two main drawbacks to MOS. Any changes in the statistics of the NWP output variables could affect not only the bias but also the variance correlation structure among model derived variables, especially the covariance structure with respect to the observations (Wilson and Vallée (2002)). In an operational setting, MOS is expensive in both time and resources. NWP model bias characteristics vary with location and separate MOS equations must be developed for each region (Wilson et al. (2007)).

These shortfalls motivated the development of the machine learning-based forecast correction models. Machine learning provides a promising alternative due to advantages in transferability, adaptability and capability of handling large datasets. Recently, machine learning has enabled advancements in diverse research areas including neuroscience (Ibtehaz and Rahman (2020); Richiardi et al. (2013)), biomedical signal analysis (Arbelle and Raviv (2019); Theis and Meyer-Bäse (2010)), weather forecasting (Grover et al. (2015); Chantry et al. (2021)) and dynamical systems (Brunton and Kutz (2019)) among others. Convolutional Neural Networks (CNNs) are one of the most popular algorithms used in computer vision (Alom et al. (2019)), recently achieving state-of-the-art performance in various weather forecast applications (Kim et al. (2019); Scher and Messori (2018); Boonyuen et al. (2018); Zhang and Dong (2020); Zhang et al. (2021)). Kim et al. (2019) developed a recurrent inception convolutional neural network (RICNN) that combines a recurrent neural network (RNN) and a CNN for daily electric load forecasting in addressing the challenges associated with climate change and energy crises. Scher and Messori (2018) built a deep CNN system to predict weather forecast uncertainty from past forecasts. A convolutional network sufficiently forecasts rainfall based on the correlation between satellite images and historical rainfall data (Boonyuen et al. (2018)). Zhang and Dong (2020) designed a convolutional recurrent neural network (CRNN) for temperature prediction based on the temporal and spatial correlations of temperature changes from historical data. A combination of C-Convolutional Neural Network and Long Short-Term Memory (LSTM) networks successfully achieve radar echo prediction based
on the shape of weather radar echo (Zhang et al. (2021)). In addition, there are a variety of CNN architectures such as AlexNet (Iandola et al. (2016), ResNet (Wu et al. (2019)) and LinkNet (Chaurasia and Culurciello (2017)) that have shown significant improvements in classification, identification and recognition tasks. As autoencoders have developed, the UNet (Ronneberger et al. (2015)) became one of the most powerful and versatile approaches in both supervised (Berthomier et al. (2020); Fernández et al. (2020)) and unsupervised learning (Chung et al. (2016); Awad and Lauteri (2021)).

In general, UNets are comprised of encoder and decoder pathways, with skip connections between the corresponding layers (Li et al. (2018)). The contracting path is designed to extract features, while the expanding path reconstructs a segmented image through a convolution kernel. A set of residual connections are made between the two paths that enable extracting high resolution features from the contracting path that are combined with information from the upsampled output. A convolution layer can then learn to assemble a more precise localization based on this information (Fernández et al. (2020)). In the last 2 years, UNets have proven to be a powerful and novel tool to detect cloud coverage, especially in multi-scale feature learning for nowcasting tasks. Fernández and Mehrkanoon (2021) and Ahmed and Sabab (2022) introduced various extended UNet architectures for weather forecasting applications. A Broad-UNet model sufficiently nowcasted precipitation and cloud cover based on an image-to-image translation using satellite imagery (Fernández and Mehrkanoon (2021)). Their EfficientNet was successfully designed to understand and classify cloud structures, and was developed based on UNet architecture to extract and reconstruct fine grained features of cloud images (Ahmed and Sabab (2022)).

In this research, UNet models were designed for the fusion of NWP model output and satellite data to capture NWP model error trends as well as improve the accuracy and sensitivity of the cloud forecasts. One major difference between this study’s approach and recent related work is that the satellite and NWP inputs are fused and projected onto a future state, correcting errors in cloud representation and forecast position. The NWP output provides information pertaining to nonlinear atmospheric evolution that is not well sensed by radar or satellite, especially at lead times beyond six hours. Both radiance-based and retrieval-based cloud observations are limited by the top-down nature of the passive satellite sensors (Bodas-Salcedo et al. (2011)). Identification of 3D cloud structure and properties requires physical conditional samples based on properties inherent
in the observations (Schuddeboom et al. (2018)). Although NWP output contains systematic error and bias, it is still the foremost method providing the atmospheric and physical conditions critical to identifying cloud regimes and locations in a 3D view. This approach provides a bridge to utilize the combination of physical model and satellite observations for improving 3D cloud forecasts for multiple lead times (3-12 hours) as well as understanding the error trends of both technologies.

In this research, UNet models were applied to fuse NWP and satellite data to capture NWP error trends and improve forecast accuracy. A generalized statistical model was created to produce forecasts for physically similar cloud families. Reliance on cloud physical attributes necessitates that the cloud types be isolated in both the NWP and satellite input. While variables such as direct radiance measurements are alluring for their simplicity, they provide limited physical information. For example, low brightness temperatures may indicate cold surface temperatures or upper tropospheric clouds. Unique representation requires additional information provided by satellite retrievals. Though imperfect, Nachamkin et al. (2022) showed the retrievals effectively identified lower tropospheric stable and unstable cloud regimes. Similar methods were applied to identify other cloud regimes used in our forecasts. Each regime was effectively converted to a binary mask in both the observations and the forecasts. These binary masks were used to train and test the UNet CNN. Since this paper focuses on the machine learning aspects of the work results from the upper tropospheric (high) clouds will be presented for demonstration purposes. upper tropospheric clouds were the most straightforward to identify, but the same UNet CNN architecture was used for all cloud families.

This paper is organized as follows. Section 2 describes the data used as well as the pre-processing steps involved in constructing data sets. Section 3 provides an overview of methodology including the data splitting procedure (training, validating, and independent testing), feature selection, UNet-CNN architectures and implementation, and model evaluation process. Section 4 demonstrates results of independent datasets for upper tropospheric clouds, and Section 5 offers discussion, concluding remarks and potential future work.

2. Data Description

This Section describes the data used in training, validating, and independent testing of the UNet CNN model as well as pre-processing involved in data preparation and fusion. Two data sources
were used: GOES-16 observations and retrievals, and derived variables from COAMPS. All data were mapped onto a common map projection using bilinear interpolation. The output horizontal spacing of the remapped grid is 5km. Data mapped to this common grid consist of a time stamp and the image obtained from GOES-16 and the NWP derived variables.

The following subsections provide more detail on the data sources and the construction of this dataset.

a. GOES-16 Observations

Satellite-based cloud observations originated from the GOES-16 Advanced Baseline Imager (ABI). These include (1) measurements of the 0.65 µm normalized reflectance (visible channel) and (2) the 10.3 µm brightness temperatures as well as retrievals of (3) cloud top height, (4) cloud base height, (5) total condensed water path, and (6) cloud top phase. Reflectance and brightness temperature data were used for viewing and interpretation of the direct satellite observations. The predictand cloud masks were derived from the physical properties, such as cloud top height, depicted in the retrievals. The upper tropospheric cloud masks, which are the primary focus in this work, relied on the cloud top heights, cloud thickness and cloud phase retrievals. Cloud base heights were used for identifying some of the other cloud masks used in our cloud forecast system.

Two sets of gridded GOES-16 data containing all of the fields listed above were used. The first originated from the Cooperative Institute for Research in the Atmosphere (CIRA) and consisted of full-disk data on a ∼3 km spherical grid. The second originated from the National Aeronautics and Space Administration (NASA) Langley Research Center (LARC) and consisted of output from their Clouds and the Earth’s Radiant Energy System (CERES) (Minnis et al. (2020)). These were also full-disk data, but on a ∼10 km spherical grid. CIRA data were primarily used due to the finer grid spacing compared to the NASA. The NASA data were used during the approximately 3% of the period when CIRA data were unavailable.

In both datasets, cloud top heights were retrieved by matching long-wave infrared channel equivalent blackbody temperatures with corresponding numerical model output (Baum et al. (2012); Yost et al. (2020)). For both NASA and CIRA, most height errors are less than 1 km except for regions of optically thin ice and multi-layered clouds. In these regions, cloud top height (Yost et al. (2020); Noh et al. (2017)) underestimates from 5-10 km due to the integrated signal received.
from cloud top and lower layers, or the surface. Cloud base heights derive from subtracting the retrieved cloud geometric thickness from the retrieved cloud top height. Seaman et al. (2017) found the initial CIRA algorithms were highly error prone. However, corrections to geometric thickness based on CloudSat and Cloud-Aerosol Lidar and Infrared Pathfinder Satellite Observations (CALIPSO) significantly reduced errors in single-layer and deep convective clouds (Noh et al. (2017)). Noh also reported mean cloud base height errors of +0.3 km and RMS errors from 1-2 km for most clouds when compared to CloudSat measurements. However, 3-5 km positive cloud base biases commonly found in both datasets for daytime low cloud base heights due mostly to missed clouds in overlapping situations (Yost et al. (2020); Noh et al. (2017)). Case studies indicated single-layered and deep convective clouds (aside from overshooting tops) performed the best. Nachamkin et al. (2022) found that cloud base retrievals produced consistent estimates of the existence of lower-tropospheric cloud cover for clouds with tops at or below 8 km above ground level (AGL). Including deeper clouds resulted in low cloud cover underestimates of approximately 20 percent overall.

b. COAMPS Forecasts

COAMPS Numerical weather forecasts from the United States Mid-Atlantic region were collected for the two-year period from 1 January 2018 – 31 December 2019. The domains consisted of three one-way nested grids with horizontal spacings of 45, 15, and 5 km centered over Norfolk, VA. This study focused on forecasts from the 5 km (277 x 229 grid point) domain. In the vertical, 60 sigma-z levels extended from 10 m to a model top at approximately 50 km. Forecasts were initialized daily at 0000, 0600, 1200, and 1800 UTC, using the Naval Research Laboratory’s Atmospheric Variational Data Assimilation System (NAVDAS) (Daley and Barker (2001)) and run to 12-hours. The previous 6-hour forecast served as a first guess. Forecasts from the Navy Global Environmental Model (NAVGEM) provided boundary conditions at three-hour intervals using a Davies scheme (Davies-Jones et al. (1976)). The explicit microphysics parameterization, used on all grids, is a modified single-moment bulk (Rutledge and Hobbs (1984); Lang et al. (2011); Houze Jr et al. (1989)) scheme described by Smith et al. (2010) which predicts the mixing ratios of cloud droplets, cloud ice, rain, snow, and graupel. The Kain-Fritsch scheme (Kain and Fritsch (1993)) parameterized moist convection on the 15 and 45 km grids. The Fu-Liou scheme (Ma and Tan
parameterized shortwave and longwave radiative transfer. Boundary layer turbulence was parameterized with a 1.5-order turbulence closure method (Mellor and Yamada (1982)) where turbulent kinetic energy was predicted. Land surface processes were parameterized with the Noah land surface model (Niu et al. (2011)), initialized at each data assimilation cycle with the 0.25° NASA Land Information System (LIS) analyses (Kumar et al. (2006)) provided by the US Air Force Weather Agency.

c. Advected GOES-16 Clouds

GOES-16 advected cloud forecasts were constructed from the retrieved cloud height fields valid at the initial time of each COAMPS forecast. Observed cloud top and cloud base heights were interpolated to the nearest three-dimensional point (i,j,k) on all three COAMPS grids, and all points between cloud base and cloud top were assumed to be solid cloud. This initial volumetric field was then advected forward in time as a passive scalar with no sources or sinks using the COAMPS dust and smoke aerosol package (Liu et al. (2003)). Boundary conditions on each nested grid were supplied by advected clouds from the parent grid. Boundaries on the 45 km coarse grid were assumed to be clear. Given the short duration of the forecasts, these clear values did not have sufficient time to influence the inner 5 km grid.

d. Derived Inputs

Characterizing the cloud field was one of the greatest challenges of this research due to uncertainties in the satellite observations. Early attempts to predict the cloud cover based on direct radiance measurements were unsuccessful. Satellite radiance is less error-prone than the physical retrievals, but it contains no innate cloud cover information. Although COAMPS radiance forecasts were correlated with the predictand radiance, other relevant variables such as boundary layer moisture were not. As a result, the ML routines could not resolve the cloud field. For example, stratus and cumulus clouds possess similar brightness temperatures but very different forcing. To make full use of the physical model data, the cloud predictands must be correlated with specific sets of predictors.

To accommodate a physics-based approach, clouds were separated into groups based on physical forcing mechanisms. Five separate cloud families are predicted by our system: lower tropospheric
stable and unstable clouds, mid-tropospheric clouds, deep precipitating clouds, and upper tropospheric clouds. Though the definitions for each family were distinct, some overlap existed. For example, thunderstorms fall into the deep precipitating, lower tropospheric unstable, and upper tropospheric cloud categories. For each type, a combination of GOES-16 physical retrievals and COAMPS analysis output were applied as classifiers. The added uncertainty in classifying the predictand field presents unique challenges due to reduced predictor-predictand correlations. As such, systematic biases are likely incorporated in the statistical equations. Owing to the imperfect observations, this study focuses on the ability of the ML to represent the cloud field in terms of spatial coverage alone. Specific properties such as cloud top, base or thickness were not predicted.

Each type was treated as an independent binary mask. To further account for observational uncertainty, each binary mask employed a set of linear parameters to identify clouds. Since each cloud family requires a unique set of criteria to identify it, describing them all in full detail would go beyond the scope of a single paper. Here, we focus exclusively on the results of the upper tropospheric cloud forecasts as these clouds were the most straightforward to distinguish.

Upper tropospheric clouds were identified in the GOES-16 retrievals primarily by cloud top due to the lack of intervening cloud layers. All clouds with cloud top heights $\geq 7900$ m were classified as upper tropospheric clouds. Height underestimates in optically thin clouds were accounted for by locating thin clouds and adjusting their classifier criteria. Thin clouds were identified based on the retrieved total condensed water path (TCP), which is a proxy for cloud optical thickness. Ice clouds were identified using the cloud top phase retrievals. Based on Nachamkin et al. (2017) the retrieved heights of most ice clouds with TCPs less than $\sim 25$ g m$^{-2}$ were too low. Following that work, the height threshold used to identify upper tropospheric clouds was linearly reduced from 7900 m for all ice phased cloud tops with TCPs of 100 g m$^{-2}$ to 7000 m at TCPs of 10 g m$^{-2}$. Although height errors can be up to several kilometers for very thin cirrus (Yost et al. (2020)), visual inspections of the resulting mask demonstrate good agreement, as indicated in an example from 1800 UTC (1300 local standard time), 22 February 2018 (Fig. 1). The GOES upper tropospheric cloud mask (Fig. 1c) generally captured the region of thick cirrus extending across the northern portions of the domain (Fig 1a,c,e). A region of lower-topped clouds centered over central Ohio and southwestern Pennsylvania (near 80° W 40° N) appears as a hole in the upper...
Figure 1. Cloud features valid at 1800 UTC, 22 February 2018 are shown. In a) the GOES16 retrieved cloud top heights in km are shaded as indicated by the colorbar. In b) the COAMPS 6-hour predicted cloud top heights in km are shaded as indicated by the colorbar. In c) and d) the corresponding GOES-16 and COAMPS binary upper tropospheric cloud masks are indicated by the shaded regions. In e) the GOES-16 0.65 μm visible image is shaded in greyscale, and in f) the advected binary upper tropospheric cloud mask is shaded.

tropospheric cloud mask. Another isolated region of cirrus overlaying the low, thick cloud cover near the center of the domain (76° W 39° N) is also well depicted.
Upper tropospheric clouds were easily identified in COAMPS because the full condensate field is explicitly predicted. In COAMPS, cloud top is defined as the highest level containing condensate with a total water content greater than or equal to $1 \times 10^{-6} \text{ kg m}^{-3}$. All clouds above 7900 m were identified as upper tropospheric clouds. An example of the COAMPS 6-hour forecast cloud top heights and upper tropospheric cloud mask valid for the same time as the GOES-16 imagery is shown in Fig. 1b,d,f. The COAMPS upper tropospheric clouds are clearly depicted by the mask. Note there is also reasonable agreement between COAMPS and GOES for this case. For the advected clouds, cloud top height was derived directly from the scalar field. Cloud top was defined as the highest level containing scalar “cloud” values greater than or equal to $1 \times 10^{-6} \text{ kg kg}^{-1}$. Like the COAMPS condensate, all points with cloud tops above 7900 m were defined as upper tropospheric clouds. The advected cloud mask for the 22 February case is shown in Fig. 1f. Note the advected mask is also a reasonable forecast of the observed clouds in this case.

The upper tropospheric cloud predictor features consisted of the satellite observed upper tropospheric cloud locations at the COAMPS initialization time (1200 UTC) as well as the advected GOES-16 and COAMPS forecast upper tropospheric clouds at each forecast lead time. To account for uncertainties in cloud top height, a normalized field was created that included clouds with heights up to 10% below the definitions above. Lower clouds were assigned values between 0 and 1 increasing linearly with height. This normalized field, referred to as the cloud top score, insures that clouds with tops a few hundred meters below the 7900 m threshold are still considered. A second field, referred to as the cloud top mask, was a strict binary mask consisting of the subset of the clouds that met or exceeded the specific criteria defined above for the COAMPS, GOES and Advected fields.

3. Methodology

a. Feature Selection

In the past few decades, many feature selection algorithms have been designed for various applications, each with advantages and disadvantages (Zhao et al. (2010)). These algorithms broadly fall into three main categories: filter, wrapper, and embedded methods (Goswami and Chakrabarti (2014)). Filter algorithms rely on the general characteristics and correlations between input and output data and evaluate features without involving any learning algorithm (Cherrington et al.
Table 1. All input features for the 3 hour lead time prediction

<table>
<thead>
<tr>
<th>Lead time</th>
<th>Data sources</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial time</td>
<td><strong>GOES</strong></td>
<td>Cloud top height</td>
</tr>
<tr>
<td>(UTC 12)</td>
<td></td>
<td>Cloud top score</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cloud top binary mask</td>
</tr>
<tr>
<td></td>
<td><strong>COAMPS</strong></td>
<td>Cloud top height</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total condensed water path</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cloud top score</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cloud top binary mask</td>
</tr>
<tr>
<td></td>
<td><strong>Advected</strong></td>
<td>Cloud top score</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cloud top binary mask</td>
</tr>
<tr>
<td>3 hours</td>
<td><strong>COAMPS</strong></td>
<td>Cloud top height</td>
</tr>
<tr>
<td>(UTC 15)</td>
<td></td>
<td>Total condensed water path</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cloud top score</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cloud top binary mask</td>
</tr>
<tr>
<td></td>
<td><strong>Advected</strong></td>
<td>Cloud top score</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cloud top binary mask</td>
</tr>
</tbody>
</table>

(2019)). Wrapper methods require a predetermined learning algorithm and use its performance as evaluation criteria to select features. Embedded model algorithms incorporate variable selection as a part of the training process, and feature relevance is obtained analytically from the objective of the learning model (Liu et al. (2010)).

Due to limited GPU and memory, after the predictor features were generated as mentioned in Section 2d, a univariate statistical feature selection algorithm was applied to minimize potentially redundant features and reduce running time and memory usage (Aggarwal et al. (2018)). This method is a common, simple and fast filter strategy to assess the importance of features by examining the intrinsic relationships between the input variables and the output.

In this research, the Kbest function from the python scikit-learn library was utilized to select the best features based on ANOVA F statistic tests (Pedregosa et al. (2011)). The ANOVA F-statistic compares the ratio of the between-class variance for a feature to the within class variance for that feature. Features with high F-values are highly correlated with the output cloud label, while low F values indicate low correlation. The Kbest method implemented in the scikit-learn library
automatically ranks features according to their ANOVA F-statistic, and then removes all but a selected list of features with the highest ANOVA F-statistic.

First, we implemented Kbest function to all upper tropospheric cloud input features for the 3 hour predictions. These features are described in Section 2d and listed in Table 1. They include:

(i) Initial time (UTC12) GOES cloud top heights, derived cloud top scores and binary masks, COAMPS cloud top heights, derived cloud top scores, cloud top binary masks, and the total condensation water paths, GOES advected cloud top heights, derived cloud top scores and binary masks.

(ii) Prediction time (UTC15) COAMPS output cloud top heights, derived cloud top scores and binary masks, and the total condensation water paths, GOES advected best guesses cloud top heights, derived cloud top scores and binary masks.

For the 6, 9, and 12 hour lead time predictions we added COAMPS output of cloud top heights, derived cloud top scores and binary masks, and GOES advected cloud top scores and binary masks valid at each prediction time. For example, input features for the 6 hour lead time prediction
Table 3. Input features for the 6 hour lead time prediction

<table>
<thead>
<tr>
<th>Lead time</th>
<th>Data sources</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial time (UTC 12)</td>
<td>GOES</td>
<td>Cloud top score  &lt;br&gt; Cloud top binary mask</td>
</tr>
<tr>
<td>3 hours (UTC15)</td>
<td>COAMPS</td>
<td>Cloud top height &lt;br&gt; Cloud top score &lt;br&gt; Cloud top binary mask  &lt;br&gt; Advected</td>
</tr>
<tr>
<td>6 hours (UTC18)</td>
<td>COAMPS</td>
<td>Cloud top height &lt;br&gt; Cloud top score &lt;br&gt; Cloud top binary mask  &lt;br&gt; Advected</td>
</tr>
</tbody>
</table>

included COAMPS output of cloud top heights, derived cloud top scores and binary masks, and GOES advected best guesses of derived cloud top scores and binary masks valid at both 3 and 6 hours (see table 3).

b. Training, Validating and Independent Testing

Once the Kbest feature selection was complete, a temporal splitting process was applied to the 2-year input dataset to monitor and evaluate the machine learning. First, the data were arranged by Julian day with the first day of the year January 1st assigned a value of zero. A splitting function looped through the full dataset at 5-day intervals, selecting the first 3 days of each interval for training, the 4th day for validation, and the 5th day for independent testing. In general, the dataset was split into 60% for training, 20% for validating, and 20% for independent testing. This splitting process was employed for two primary reasons:

(i) It is important to ensure that training, validating, and independent testing data are generalized, containing information evenly spread throughout the year. Since the dataset contained only two yearly cycles, random splitting can lead to imbalances between training, validating, and testing data sets. Cloud occurrence and NWP performance vary systematically through the year, and the Julian
day splitting provided consistent samples through all seasons. It also reduced temporal correlations between successive days.

(ii) Since Unet-CNN architecture primarily relies on spatial information, a Julian day-based splitting process also provides an opportunity to understand and quantify temporal trends such as seasonal effects on model performance. Since the data only contained two yearly cycles, the default random shuffle routine could unevenly sample a particular season. Such imbalances lead to biases in the statistical model. Seasonal and temporal results are discussed in Section 4.

Separate UNet models, each with its own set of input features, were trained for every forecast lead time. Each model uniquely accounts for the time evolution of the COAMPS and GOES-16 advected cloud forecast errors as well as the waning influence of the GOES-16 cloud observations valid at the COAMPS model initiation time. For each lead time, features consisted of forecasts valid at the lead time combined with the GOES-16 retrievals valid at the COAMPS initialization time. At lead times beyond 3 hours all forecasts valid at the previous lead times were also included in the feature set to produce a highly simplified time-based ensemble (table 3).

c. UNet-CNN Architecture

Inspired by the success of UNet and its variants in medical image segmentation (Ronneberger et al. 2015), we used a similar architecture as our backbone network and applied it to all cloud types and lead times. A specific example of the architecture for the upper tropospheric clouds at the 6 hour lead time forecast is illustrated in Figure 2.

The network input is of shape \((N \times H \times W \times F)\) where \(N\) is the number of samples, \(H\) and \(W\) are the height and width of the images (north-south and east-west extent), and \(F\) is the number of features or atmospheric variables, which we consider as channels in our network. The output is of shape \((N \times H \times W \times 1)\), where the last dimension corresponds to continuous cloud model output at each grid cell which ranges from 0 to 1.

After Unet-CNN model generated probabilistic output \(\widehat{y}_i \in [0, 1]\). Then, we used 2 different iterative techniques to select threshold converting probabilistic output to binary cloud labels (1 = cloud, 0 = no cloud). Choosing Threshold based on (1) Receiver Operator Characteristic (ROC) curve or (2) maximizing the equitable threat score (ETS). ETS is calculated as shown in Equation 16. The thresholds \((T)\) slightly vary with different lead times; all models perform well with \(T\)
Figure 2. UNet Example for 336x336 pixels with 15 channels. Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. The white boxes with blue stripes represent copied feature maps. The arrows denote the different operations.

between 0.5 and 0.6 for both methods. In this state of research, we chose to use $T = 0.5$ for all models and cloud types. If $\tilde{y} \geq 0.5$, it was set to 1, the value of a cloudy pixel.

The UNet architecture, built as an encoder-decoder with skip connections, enables the extraction of meaningful information and solving image-to-image segmentation. First, encoder components extract local spatial information from the input image, then decoder components perform classification on each pixel to reconstruct the segmented output. A set of skip connections between contracting and expanding components, a state-of-the-art development beyond the original UNet, was an important part of this architecture, providing precise localization in the output image.

Two major differences between this UNet-CNN architecture and the original are the application of a combination dice and cross entropy loss function, and an improved bias initializer functions.

d. Implementation Details

In total, the network is comprised of ten encoder and decoder components. Each component was constructed based on the typical architecture of a UNet-convolutional network (Ronneberger et al. (2015)).
(i) Each component in the encoder path consists of the repeated application of two convolution kernels (same padding convolutions) with a rectified linear unit (ReLU) activation function and a max pooling operation for down-sampling. All convolution kernels are of size $3 \times 3$ with layer depths $(32, 64, 128, 256, 512)$. All max pooling operations are of size $(2 \times 2)$ with stride of 2; at each down-sampling step we doubled the number of feature channels.

(ii) Every step in the expansive path consists of an up-sampling of the feature map followed by a convolution (up-convolution) that reduces the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two convolution kernels with rectified linear unit (ReLU) activation functions. All convolution kernels are also of size $3 \times 3$ with layer depths $(256, 128, 64, 32)$. All concatenation operations are of size $(2 \times 2)$ with stride of 2 along the correspondingly cropped feature map from the contracting path.

e. Training and Loss

The CNN training process relies on back-propagation to calculate and update model parameters such as weight and bias to minimize error or loss. This process is optimized and defined by the loss function (Moltz et al. (2020)). Many loss functions were developed and used in image segmentation; they broadly fall into 4 main categories: distribution-based losses (such as the cross entropy loss) (Ho and Wookey (2019)), region-based losses (such as Dice loss) (Shen et al. (2018)), boundary-based losses (such as the boundary loss) (Kervadec et al. (2019)), and more recently compound losses (Moltz et al. (2020)).

The cross entropy loss is typically distribution-based (Qu et al. (2020)) and it is the most widely used loss function in classification problems such as a UNet (Ronneberger et al. (2015)), 3D UNet (Çiçek et al. (2016)), and SegNet (Pradhan et al. (2019)). In contrast, the Dice loss is considered as a direct loss minimisation, and its optimization is based on the most commonly used metric for evaluating segmentation performance (Jin et al. (2020)). A Dice loss ($D$) (Milletari et al. (2016)) is defined as:

$$D = \frac{2 \sum_i^N \hat{y}_i y_i}{\sum_i^N \hat{y}_i^2 + \sum_i^N y_i^2}$$  \hspace{1cm} (1)

where the sum runs over the $N$ grid cells, of the predicted segmentation image $\hat{y}_i \in [0, 1]$ and the ground true binary image $y_i \in \{0, 1\}$. This Dice loss is a differentiable approximation of the Dice
metric that depends on the continuous network output $\hat{y}_i \in [0,1]$ (Milletari et al. (2016)). This dice loss function has also shown great results in the Attention UNet (Schlemper et al. (2019)) and V-Net (Milletari et al. (2016)).

Recently, Van Beers (2021) compared image segmentation performance between intersection over union loss IoU and Dice loss functions. The intersection over union metric $IoU$ or coefficient is defined as:

$$IoU = \frac{\sum_i^N \hat{y}_i y_i}{(\sum_i^N \hat{y}_i + \sum_i^N y_i - \sum_i^N \hat{y}_iy_i)}$$

(2)

where the sum runs over the $N$ grid cells, of the predicted segmentation image $\hat{y}_i \in [0,1]$ and the ground true binary image $y_i \in \{0,1\}$. This IoU loss is a differentiable approximation of the IoU metric that depends on the continuous network output $\hat{y}_i \in [0,1]$ . Although IoU and Dice both ignore true negatives, IoU is more efficient than Dice at reducing the effects of very imbalanced data sets. The IoU relation of true positives to false positives and false negatives is reduced by a factor of 2. Consequently, it reduces discrepancies between positive labeled pixels and negative background pixels, especially when a detected or positive class is only a small portion of the image.

Taghanaki et al. (2019) and Moltz et al. (2020) developed and applied a combined loss function containing two independent loss functions: Dice and Cross Entropy. This combined function also showed improvements in image segmentation accuracy and reduced the effects of class imbalance.

One of the major challenges in cloud segmentation is that only a small portion of pixels in an image are detected as cloud ($\sim 10 - 15\%$). Inspired by Van Beers (2021) and Moltz et al. (2020), a combination of loss functions, IoU and binary cross entropy, was applied in this research to reduce the effects of class imbalance in cloud segmentation. Detailed explanations and formulae are described below.

(i) Combination of IoU and Cross Entropy Loss Function  
During the training and validation phase, the input images and their corresponding segmentation maps are used to train and validate the network with the stochastic gradient descent implementation similar to (Jia et al. (2014)). The objective of training the UNet-CNN is to minimize the output loss through back-propagation.

The loss function is an optimisation mechanism that directly affects model convergence during the training process. In this research, we use a combination of two loss functions: binary cross entropy and IoU are operated simultaneously through the skip connect between layers of UNet.
Binary cross entropy measures the difference between two probability distributions for given random variables in a set of events: ground truth and model prediction. In a binary classification scenario, it is equivalent to the negative log likelihood loss (Yeung et al. (2022)). The binary cross entropy loss is calculated as:

\[ \mathcal{L}_{BCE} = -\sum_{i=1}^{N} y_i \log \hat{y}_i + \sum_{i=1}^{N} (1 - y_i) \log (1 - \hat{y}_i) \] (3)

where the sum run over the \( N \) kernels \( y_i \in \{0, 1\} \) and \( \hat{y}_i \in [0, 1] \); \( y_i \) refers to the ground truth label and \( \hat{y}_i \) represents the model predicted values.

IoU loss based on the relation of true positives to false positives and false negatives is similar to a dice metric for evaluating segmentation performance. IoU loss \( \mathcal{L}_{IoU} \) is defined as:

\[ \mathcal{L}_{IoU} = 1 - IoU \] (4)

where \( IoU \) was defined in Equation (2).

The combination loss \( \mathcal{L}_{combo} \) is defined as a sum of the cross entropy loss and IoU loss:

\[ \mathcal{L}_{combo} = \mathcal{L}_{BCE} + \mathcal{L}_{IoU} \] (5)

where \( \mathcal{L}_{BCE} \) and \( \mathcal{L}_{IoU} \) were defined in Equations (3) and (4).

f. Convolutional Layer Bias Initializer

(i) Bias Initializer Function    In deep networks with many convolution layers and connecting paths, proper initialization of the weights and biases is extremely important (Shanmugamani (2018)). Mukherjee and Yeri (2021) investigated the effect of weight initialization techniques such as Random, Zero, Xavier, and He on the performance of Neural Networks. This research showed that a suitable weight initialization function reduces the exploding or vanishing gradient problem by improving the update routine. These updated and adjusted weight values are applied to the same variance for outputs of activation functions at each layer and back-propagated associated gradients throughout the network. Our cloud forecast dataset inherited large and complex biases from COAMPS. In extreme cases, COAMPS bias values can reach 2.00 (twice the observed cloud
amount). For this reason, we tested a more complex bias initialization method to improve model performance. Inspired by Mukherjee and Yeri (2021), different built-in initialization functions from Keras Tensorflow were compared including random normal, random uniform, truncate normal, He normal, and He uniform (Hanin and Rolnick (2018)). These functions act as a bias initialization to automatically determine the optimal bias to facilitate the stochastic gradient descent to reach global minimal errors for the network. The random normal initializer was selected and applied because it proved to be the most robust and reliable option for our models.

(ii) Bias Initializer Formula  Each neuron in the CNN takes input variable values and calculates the weight and bias sum using a weight and bias matrix. A nonlinear activation function is applied, the weight and bias for an individual neuron is described by:

\[
y = f\left(\sum_{i=0}^{k-1} \sum_{j=0}^{l-1} (\Theta_{i,j} \times x[i+s \times a, j+s \times b] + \beta)\right)
\]

where:

- \(\Theta_{i,j}\) is the element of the convolution filter at position \(i,j\)
- \(x[i+s \times x, j+s \times y]\) is the element of the input tensor at position \(k, l\)
- \(k\) and \(l\) are the kernel dimensions
- \(s\) is the stride
- \(a\) and \(b\) are the sliding indexes
- \(\beta\) is the convolutional layer bias
- \(f(x) = \max(0,x)\) is the ReLU activation function, where \(x\) is the input to the activation function

In this research, we focused on finding an efficient bias initializer to reduce the effects from the very large biases in the COAMPS input variables. Bias initializers are strategies for (1) setting the initial values of a bias matrix for a neural network layer, then (2) adjusting the biases in the bias matrix through optimization algorithms during back-propagation. The Keras CNN algorithm commonly assumes an initial bias of zero by default during the training process. However, systematic biases from our input variables are very high and complex, especially in the warm season. COAMPS cloud coverage fractions range from multiples of 0.75 to 2.00 times the GOES-16 fractions. Keras Tensorflow offers numerous different built-in initializer functions, each representing unique routines for setting the initial values of a neural network layer’s bias matrix (Li et al.
The COAMPS biases require a more complex initialization function that can capture the heterogeneous biases as they evolve temporally and spatially with the changing atmospheric conditions. A random normal initialization was applied in this research because it provides the closest attribution of bias values. This function assumes that all bias matrix values are random numbers selected from a normal distribution (Manaswi (2018)).

g. Model Evaluation

In this research, UNet-CNN performance was validated from an independent test dataset, which was set aside from the beginning of the experiment as mentioned in Section 3b.

1) Evaluation Metrics

A set of standard performance metrics commonly used for validating weather forecasts (Nachamkin et al. (2022)), such as the probability of detection (POD), false alarm ratio (FAR), bias (BIAS), and equitable threat score (ETS) are computed to understand and quantify the UNet-CNN improvements. The predictions output by the UNet were compared to GOES-16 ground truth for all images. In the equations below, the UNet-CNN predictions are denoted by $\hat{y}$, while ground truth GOES-based cloud masks are denoted by $y$.

After Unet-CNN model generated probabilistic output $\hat{y}_i \in [0, 1]$. We use the same threshold $T$ explained in section c. In this state of research, we chose to use $T = 0.5$.

BIAS is defined as the ratio of the cloud coverage area greater than or equal to a threshold $T = 0.5$ for both the UNet and ground truth for all pixels. A BIAS of 1.0 indicates the predictions are unbiased. BIAS values larger than 1 indicate over-prediction, and values smaller than 1 indicate under-prediction:

$$BIAS_T = \frac{\# |\hat{y} \geq T|}{\# |y \geq T|}$$

Similarly, accuracy, POD, FAR, and ETS metrics were computed from the number of pixels correctly classified or misclassified based on $T = 0.5$. True positives are referred to as hits ($H$), false negatives are referred to as misses ($M$), false positives are referred as false alarms ($FA$), and correctly identified no-cloud pixels are referred to as true negatives ($TN$). Hits ($H$), Misses ($M$), False alarm ($FA$), true negative ($TN$) are defined in equations below:
\[ H_T = |\hat{y} \geq T \text{ AND } y \geq T| \]  
(8)

\[ M_T = |\hat{y} < T \text{ AND } y \geq T| \]  
(9)

\[ FA_T = |\hat{y} \geq T \text{ AND } y < T| \]  
(10)

\[ TN_T = |\hat{y} < T \text{ AND } y < T| \]  
(11)

Accuracy measures the fraction of predictions of a model got right over total number of predictions. For our cloud binary classification task, accuracy is defined as:

\[ \text{Accuracy}_T = \frac{H_T + TN_T}{H_T + TN_T + M_T + FA_T} \]  
(12)

The probability of detection (POD) and false alarm rate (FAR) are defined as:

\[ POD_T = \frac{H_T}{H_T + M_T} \]  
(13)

\[ FAR_T = \frac{FA_T}{H_T + M_T} \]  
(14)

The ETS is computed in reference to the expected number of correct cloud forecasts attained by an independent random forecast (H\text{Random}). H\text{Random} as defined by:

\[ H_{\text{Random}} = \frac{(H_T + FA_T)(H_T + M_T)}{H_T + FA_T + M_T + TN_T} \]  
(15)

ETS accounts for the increased probability of correct forecasts during periods of extensive cloud cover and is defined as in Equation 16.

\[ ETS_T = \frac{H_T - H_{\text{Random}}}{H_T + FA_T + M_T - H_{\text{Random}}} \]  
(16)

These metrics were calculated in the aggregate sense from the sums of all hits, misses, false alarms, and true negative values from all forecasts of a given lead time. Doing so avoids extremes associated with low cloud coverage events, such as infinite bias values on clear days. As result, the scores are weighted towards cloudier days as they contribute more to the totals.
2) Fracti-ons Skill Score

The Fractions Skill Score (FSS) (Roberts and Lean (2008)) accounts for near misses by sampling square neighborhoods of length $b$ centered at each point in the verification domain of size $N_x \times N_y$. At a given scale $b$, the fractions of observed $O_b(i, j)$ and forecast $F_b(i, j)$ points are computed for all neighborhoods and combined to calculate the FSS as:

$$FSS_b = 1 - \frac{MSE_b}{MSE_{bref}}$$  \hspace{1cm} (17)

where $MSE_b$ and $MSE_{bref}$ are defined as (18) and (19):

$$MSE_b = \frac{1}{N_x N_y} \sum_{i=1}^{b} \sum_{j=1}^{b} [O_b(i, j)F_b(i, j)]$$  \hspace{1cm} (18)

$$MSE_{bref} = \frac{1}{N_x N_y} \left( \sum_{i=1}^{b} \sum_{j=1}^{b} O_b^2(i, j) + \sum_{i=1}^{b} \sum_{j=1}^{b} F_b^2(i, j) \right)$$  \hspace{1cm} (19)

Like the metrics in Section 3g1, the FSS was calculated from the aggregate sum of all daily $MSE_b$ and $MSE_{bref}$ values from each forecast realization. Additionally, the 90% confidence intervals were calculated using a bootstrap technique to randomly sample the $MSE_b$ and $MSE_{bref}$ pairs and recalculate the FSS 10,000 times. Each sample size was 75% of the original distribution and replacements were allowed. The 90% confidence interval was derived from the resulting distribution of FSS values.

4. Results and Discussion

a. Comparative Assessments

Evaluation metrics, as described in Section 3g1 above, were calculated from the independent test dataset to evaluate UNet-CNN performance. Separate sets of identical but independent statistics were also calculated from the training and validation datasets to check the robustness of the results. The UNet-CNN, COAMPS and Advected GOES-16 forecasts were all verified against GOES-16 observations for the 3, 6, 9, and 12 hour lead times. Results of these evaluations, summarized in Tables 4, 5, and 6, indicate the UNet improves the quality and accuracy of cloud forecast for
Figure 3. Accuracy and ETS of UNet-CNN (red), COAMPS (blue) and Advected GOES-16 (grey) upper tropospheric cloud forecasts for the 3-12 hour lead times. Metrics were derived from the test dataset.

all lead times from 3 to 12 hours. The similarity of the scores between the testing, training, and validation datasets shows the results are robust and that the UNet is stable. Note that the CNN performance is influenced by the accuracy of the COAMPS forecasts as well as any errors in the GOES-16 retrievals. COAMPS forecast errors increase from the initial time, while the retrieval errors vary with cloud type and cloud thickness as mentioned in Section 2a.

The 3-12 hour Accuracy and ETS of the UNet-CNN, COAMPS and Advected GOES-16 upper tropospheric clouds from the test dataset are plotted in Figure 3 to illustrate how the Unet improvements trend with forecast lead time. Compared to COAMPS, the UNet improves the accuracy score by 7% at the 3 hour lead time, and approximately 3% - 4% for the 6, 9, and 12 hour lead times. On average, only about 10 – 15% of the pixels in a given scene are cloudy. Although the UNet only improves the accuracy by 4% - 7%, it significantly improves the ability to correctly forecast cloud against the clear sky background. Accuracy reflects the number of correctly predicted cloudy and clear pixels, while the ETS focuses on the cloudy pixels alone. It ranges from -1/3 for a worse than a random forecast to 1.0 for a perfect forecast. Based on the ETS (Fig. 3), the UNet cloudy forecasts score 54% above COAMPS at 3 hours, 41% at 6 hours, 31% at 9 hours, and 41% for 12 hours. As observed in Figure 3, the UNet performance advantage decreases steadily from 3-9 hours but slightly increases at 12 hours, indicating the improvements likely extend beyond 12 hours.

The advected upper tropospheric clouds also outperformed COAMPS, but the advantage was lost by 9 hours. Both the advected and COAMPS clouds were used as predictor features in the UNet, but the importance of the advected clouds likely waned with time. Advected clouds
Figure 4. POD and FAR scores for the UNet-CNN (red), COAMPS (blue) and Advected GOES (grey) upper tropospheric cloud forecasts from 3-12 hours. Metrics were derived from the test dataset.

do not account for the nonlinear effects associated with cloud formation and dissipation. New clouds often form in regions of rising air associated with large-scale atmospheric disturbances. Developing and dissipating thunderstorms also contribute to nonlinear evolution of the cloud field. Upper tropospheric cloud formation was noted to be very dynamic in this region as convection was common. Even in winter, the Gulf Stream provided ample instability to support deep convection in the southeastern portions of the domain.

As shown in Table 4, UNet biases rank between 0.98 -1.07 for all lead times. Bias values greater than 1 indicate over-prediction, less than 1 under-prediction, with one being a perfect score. The UNet provides well balanced forecasts up to 12 hours lead time. COAMPS and advected GOES-16 upper tropospheric cloud under-prediction errors increase with lead time, with the COAMPS bias of 0.80 at 12 hours being the worst score. Furthermore, POD and FAR scores (Figure 4) demonstrate the UNet increased correct hit rates while maintaining a bias near 1.0. The POD quantifies the likelihood of correctly predicting clouds where they are observed, while the FAR measures the likelihood of predicting clouds in clear areas. Both scores range from 0 to 1, with a perfect POD perfect score of 1 and FAR of 0. Compared to COAMPS, the UNet improves the POD 22% over COAMPS at 3 hours, 20% at 6 hours, 22% at 9 hours, and 36% at 12 hours. Importantly, the UNet is able to maintain a consistent POD of 72−75%, while COAMPS decreases from 61% at 3 hours to 55% at 12 hours. The UNet FAR was 54% lower than COAMPS at 3 hours, 23% at 6 hours, 10% at 9 hours, and 5% at 12 hours.
Table 4. Evaluation metrics derived from the test dataset for the UNet, COAMPS, and Advected GOES-16 upper tropospheric cloud forecasts for the 3-12 hour lead times. Metrics include the accuracy, ETS, Bias, POD, FAR, and 11-pixel (55 km) FSS.

<table>
<thead>
<tr>
<th>Lead Time</th>
<th>Models (Test)</th>
<th>Skill Score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Accuracy</td>
</tr>
<tr>
<td>3 hours</td>
<td>UNet</td>
<td>86%</td>
</tr>
<tr>
<td></td>
<td>COAMPS</td>
<td>79%</td>
</tr>
<tr>
<td></td>
<td>Advected -GOES</td>
<td>83%</td>
</tr>
<tr>
<td>6 hours</td>
<td>UNet</td>
<td>84%</td>
</tr>
<tr>
<td></td>
<td>COAMPS</td>
<td>80%</td>
</tr>
<tr>
<td></td>
<td>Advected -GOES</td>
<td>79%</td>
</tr>
<tr>
<td>9 hours</td>
<td>UNet</td>
<td>82%</td>
</tr>
<tr>
<td></td>
<td>COAMPS</td>
<td>79%</td>
</tr>
<tr>
<td></td>
<td>Advected -GOES</td>
<td>77%</td>
</tr>
<tr>
<td>12 hours</td>
<td>UNet</td>
<td>82%</td>
</tr>
<tr>
<td></td>
<td>COAMPS</td>
<td>77%</td>
</tr>
<tr>
<td></td>
<td>Advected -GOES</td>
<td>73%</td>
</tr>
</tbody>
</table>

As mentioned in Section 3g2, the FSS accounts for near misses through the use of neighborhood samples centered at each pixel. Each neighborhood has its own FSS score which ranges from 0 to 1 with 1 being the best-quality forecast. Imperfect, unbiased forecasts can receive a score of 1 if the errors are small enough to be contained within the neighborhoods. For instance, a cloud mask forecast with small-scale spatial errors will have low FSS scores as long as the spatial scale of the errors is larger than the size of the neighborhood samples. The FSS rapidly increases (improves) with increasing neighborhood size because the same number of observed and predicted pixels is eventually contained within each neighborhood sample. In this way, the horizontal scale of the errors can be estimated from the rate of increase of FSS improvement.

The FSS was calculated for neighborhood scales of 1 and 11 pixels (5 and 55 km) to evaluate how the UNet’s cloud forecast improvements relate to spatial errors in the study region. As shown in Table 4 and Figure 5, the UNet’s 1-pixel and 11-pixel FSS scores are higher than COAMPS, though the gap between the scores is smaller for the 11-pixel neighborhoods. This convergence in the FSS...
Table 5. Same as Table 4 except evaluation metrics were derived from the training dataset.

<table>
<thead>
<tr>
<th>Lead Time</th>
<th>Models (Train)</th>
<th>Skill Score</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Accuracy</td>
<td>ETS</td>
<td>Bias</td>
<td>POD</td>
<td>FAR</td>
<td>FSS</td>
</tr>
<tr>
<td>3 hours</td>
<td><strong>UNet</strong></td>
<td>87%</td>
<td>0.49</td>
<td>0.97</td>
<td>0.74</td>
<td>0.24</td>
<td>0.86</td>
</tr>
<tr>
<td></td>
<td><strong>COAMPS</strong></td>
<td>80%</td>
<td>0.32</td>
<td>0.97</td>
<td>0.61</td>
<td>0.37</td>
<td>0.81</td>
</tr>
<tr>
<td></td>
<td><strong>Advected -GOES</strong></td>
<td>82%</td>
<td>0.38</td>
<td>0.93</td>
<td>0.65</td>
<td>0.31</td>
<td>0.84</td>
</tr>
<tr>
<td>6 hours</td>
<td><strong>UNet</strong></td>
<td>85%</td>
<td>0.46</td>
<td>1.00</td>
<td>0.74</td>
<td>0.26</td>
<td>0.84</td>
</tr>
<tr>
<td></td>
<td><strong>COAMPS</strong></td>
<td>79%</td>
<td>0.32</td>
<td>0.93</td>
<td>0.61</td>
<td>0.34</td>
<td>0.81</td>
</tr>
<tr>
<td></td>
<td><strong>Advected -GOES</strong></td>
<td>79%</td>
<td>0.32</td>
<td>0.95</td>
<td>0.62</td>
<td>0.35</td>
<td>0.78</td>
</tr>
<tr>
<td>9 hours</td>
<td><strong>UNet</strong></td>
<td>81%</td>
<td>0.40</td>
<td>1.00</td>
<td>0.71</td>
<td>0.30</td>
<td>0.81</td>
</tr>
<tr>
<td></td>
<td><strong>COAMPS</strong></td>
<td>81%</td>
<td>0.30</td>
<td>0.87</td>
<td>0.58</td>
<td>0.33</td>
<td>0.80</td>
</tr>
<tr>
<td></td>
<td><strong>Advected -GOES</strong></td>
<td>76%</td>
<td>0.26</td>
<td>0.92</td>
<td>0.57</td>
<td>0.38</td>
<td>0.73</td>
</tr>
<tr>
<td>12 hours</td>
<td><strong>UNet</strong></td>
<td>81%</td>
<td>0.43</td>
<td>1.07</td>
<td>0.77</td>
<td>0.28</td>
<td>0.84</td>
</tr>
<tr>
<td></td>
<td><strong>COAMPS</strong></td>
<td>77%</td>
<td>0.30</td>
<td>0.79</td>
<td>0.56</td>
<td>0.29</td>
<td>0.79</td>
</tr>
<tr>
<td></td>
<td><strong>Advected -GOES</strong></td>
<td>73%</td>
<td>0.24</td>
<td>0.87</td>
<td>0.55</td>
<td>0.37</td>
<td>0.71</td>
</tr>
</tbody>
</table>

Scores at larger scales is due to the reduced detail in the UNet forecasts compared to COAMPS. For example, consider the 6-hour UNet forecast corresponding to the 22 February 2018 case (Fig. 6). The UNet upper tropospheric cloud mask is better on average, especially along the eastern boundary of the main cloud shield where COAMPS predicted insufficient upper tropospheric cloud coverage. This improvement is reflected in the scores. The ETS, 1-pixel FSS and 11-pixel FSS scores for the COAMPS forecast (Fig. 1d) are: 0.40, 0.72, and 0.85 while the corresponding UNet forecast scores are: 0.63, 0.87, and 0.93, respectively. Notably, small-scale features such as the cloud-free region in southwestern Pennsylvania, are absent from the UNet mask. These features are represented in the COAMPS mask, but displacement errors result in reduced overlap between the COAMPS and GOES-16 masks. Thus, the COAMPS ETS and 1-pixel FSS values are considerably below those for the UNet while the difference between the 11-pixel FSS scores is relatively smaller. Increased variance in the predicted and/or observed masks often leads to increased errors at the pixel scale due to displacements. Since most cost functions operate on the pixel scale, solutions with minor offset errors will not be favored, even if they appear more realistic. Filtering the predictors as well as the predictand masks could mitigate this problem as there is likely some...
Table 6. Same as Table 4 except the evaluation metrics were derived from the validation dataset.

<table>
<thead>
<tr>
<th>Lead Time</th>
<th>Models (Validation)</th>
<th>Skill Score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Accuracy</td>
</tr>
<tr>
<td>3 hours</td>
<td>UNet</td>
<td>86%</td>
</tr>
<tr>
<td></td>
<td>COAMPS</td>
<td>78%</td>
</tr>
<tr>
<td></td>
<td>Advected -GOES</td>
<td>82%</td>
</tr>
<tr>
<td>6 hours</td>
<td>UNet</td>
<td>82%</td>
</tr>
<tr>
<td></td>
<td>COAMPS</td>
<td>78%</td>
</tr>
<tr>
<td></td>
<td>Advected -GOES</td>
<td>78%</td>
</tr>
<tr>
<td>9 hours</td>
<td>UNet</td>
<td>80%</td>
</tr>
<tr>
<td></td>
<td>COAMPS</td>
<td>76%</td>
</tr>
<tr>
<td></td>
<td>Advected -GOES</td>
<td>74%</td>
</tr>
<tr>
<td>12 hours</td>
<td>UNet</td>
<td>81%</td>
</tr>
<tr>
<td></td>
<td>COAMPS</td>
<td>77%</td>
</tr>
<tr>
<td></td>
<td>Advected -GOES</td>
<td>72%</td>
</tr>
</tbody>
</table>

intermediate spatial scale that is sufficiently predictable for the UNet to capture. The resolvable scale that can be achieved is likely related to the amount of data available for training. Two years is not sufficient to sample the full degree of atmospheric variability. In summary, these results from training (Table 5), evaluating (Table 6) and independent testing datasets (Table 4), show that despite the filtering effects of the CNN, the UNet has a great potential to capture the complexity and systematic errors from COAMPS and GOES-16 inputs and provide a consistent and accurate cloud prediction over 12 hour period.

### b. Seasonal Effects

The ETS, Bias, POD, and FAR were calculated from the independent testing dataset for warm (15 April - 13 October) and cold (14 October - 14 April) seasons to evaluate seasonal effects on the UNet performance. Temporal coverage fractions for the 6-hour UNet-CNN and COAMPS forecast cloud masks are visually compared in Figures 7 and 8 to demonstrate the UNet seasonal performance. Each image represents the time average upper tropospheric cloud coverage based on
Figure 5. FSS scores for the UNet-CNN (green), COAMPS (blue) and Advected GOES (red) upper tropospheric cloud forecasts from 3-12 hours. FSS scores for the 1-pixel (5 km) neighborhoods are solid lines while the 11-pixel (55 km) neighborhoods are dotted. Shaded regions indicate 90% confidence intervals. Metrics were derived from the test dataset.

the binary masks over the season. A value of 0.5 means that clouds were present at 1800 UTC in 50% of the samples.

COAMPS systematic errors are strongly influenced by season. In the warm season, COAMPS predicts too few clouds (bias = 0.83) with a very low ETS (0.28), low correct hit rate (0.54) and high false alarm rate (0.35). In the cold season, COAMPS performs better with a more balanced bias (0.90); it also doubles the quality of cloud forecasts with higher ETS (0.46), improves correct hit rate (0.69), and improves high false alarm rate (0.23). The UNet shows overall improvements for the 6 hour forecast based on ETS, Bias, POD, and FAR scores for both seasons. During the warm season, the UNet improves the ETS by 54%, bias by 11%, POD by 28%, and also reduces FAR by 29%. In the cold season, the UNet improves the ETS by 33%, bias by 4%, POD by 28%, and also reduces FAR by 29%. It also performs better in cold than warm seasons, like COAMPS. Because the UNet was trained from a dataset that included all seasons, it tends to average out systematic errors in the warm and cold seasons, it slightly over-predicts upper tropospheric clouds for the cold season (bias = 1.06), and slightly under-predicts them for warm season (bias = 0.92).
Figure 6. The UNet CNN 6-hour forecast binary upper tropospheric cloud mask valid at 1800 UTC, 22 February 2018 is shaded. This case was selected from the test dataset.

Temporal cloud fraction images (Figs. 7, 8) indicate better agreement between the UNet and GOES-16 in terms of the domain-wide trends in upper tropospheric cloud cover. Cloud maxima above the Appalachian mountains are distinctly visible during both seasons in COAMPS as southwest-northeast oriented lines through the western portion of the domain, indicating over-prediction errors there. These maxima are not as pronounced in the UNet or GOES-16 fractions. Cold season upper tropospheric cloudiness is also more evenly distributed in both the GOES-16 and UNet images compared to COAMPS.

Individual forecasts provide further details about the UNet performance for specific seasonal weather phenomena. During the cold season, upper tropospheric clouds were most often associated with large-scale fronts and cyclones. These systems produced broad cirrus shields which were
well suited for the UNet to resolve. The 22 February, 2018 case in Figs. 1 and 6 depicts clouds over the northern portion of the domain associated with a stationary front. Although the UNet did not resolve the finer details of the cloud shield, the overall forecast scored better than COAMPS. Another forecast from 9 November 2018 (Fig. 9a-c) shows upper tropospheric clouds associated with a cold front as it progressed from west to east through the central domain, as well as a cluster of clouds...
Figure 9. Binary upper tropospheric cloud masks are shaded for a series of individual cases selected from the test dataset. The dates are indicated at the top center of each row. COAMPS 6-hour forecasts (right column), GOES-16 observations (middle column), and UNet CNN 6-hour forecasts (left column) are displayed. All images are valid at 1800 UTC.
of intense oceanic thunderstorms in the southeastern portion of the domain. The UNet clouds were
more consolidated than COAMPS in both systems. In the case of the front, the consolidation was
an improvement. However, the thunderstorm cloudiness was overly consolidated.

Thunderstorm-generated upper tropospheric clouds were not as well predicted by the UNet,
especially over land during the warm season. Land-based summer thunderstorms often occurred
in the afternoons and tended to be isolated as in the forecast from 8 August 2019 (Fig. 9d-f).
COAMPS sometimes predicted these types of storms, as it did on this day as indicated by the
scattered upper tropospheric clouds over land. However, the clouds were too large and slightly
offset from the GOES-16 clouds. In contrast, the UNet predicted no upper tropospheric clouds at all
over land. The UNet performed better in cases when thunderstorms occurred beneath preexisting
layers of high cirrus clouds, as on 25 August 2018 (Fig. 9g-i). Multiple thunderstorms occurred
in three general groups over the northwestern and north central domain as well as over the ocean.
The UNet identified all three clusters and was more accurate in their placement than COAMPS.
Generally, isolated land-based thunderstorms were often missed by the UNet, likely because
variables indicative of atmospheric instability were not included in the input features. Since upper
tropospheric clouds are common to both stable and unstable environments, atmospheric stability
variables provide conflicting information. Thunderstorms represent a unique cloud family and are
best represented by their own machine learning model. This family of clouds will likely be added
to future generations of our system.

UNet performance is seasonally influenced, but these results show that it is able to improve the
quality of upper tropospheric cloud forecasts based solely on the spatial information extracted from
the convolution layers. This encourages the potential development of a LSTM-UNet in the near
future. the LSTM-UNet can incorporate both spatial and temporal information and potentially
further improve upper tropospheric cloud forecasts.

c. Evaluations of COAMPS biases

To evaluate the UNet performance for various types of COAMPS forecast bias, temporal cloud
fractions and skill scores were derived from the independent test dataset for three separate cate-
gories: COAMPS extreme under-predictions (bias \( \leq 0.75 \)), COAMPS typical biases (0.75 > bias <
Figure 10. Comparison of the 6-hour temporal forecast cloud fractions and skill scores for cases in the test dataset of COAMPS extreme under-prediction

![Comparison of the 6-hour temporal forecast cloud fractions and skill scores for cases in the test dataset of COAMPS extreme under-prediction](image1)

<table>
<thead>
<tr>
<th></th>
<th>ETS</th>
<th>Bias</th>
<th>POD</th>
<th>FAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>COAMPS</td>
<td>0.24</td>
<td>0.66</td>
<td>0.44</td>
<td>0.33</td>
</tr>
<tr>
<td>Unet-CNN</td>
<td>0.37</td>
<td>0.88</td>
<td>0.63</td>
<td>0.28</td>
</tr>
</tbody>
</table>

Aggregate Statistics of 38 days (Bias ≤ 0.75)

Figure 11. Comparison of the 6-hour temporal forecast cloud fractions and skill scores for cases in the test dataset of low COAMPS bias

![Comparison of the 6-hour temporal forecast cloud fractions and skill scores for cases in the test dataset of low COAMPS bias](image2)

<table>
<thead>
<tr>
<th></th>
<th>ETS</th>
<th>Bias</th>
<th>POD</th>
<th>FAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>COAMPS</td>
<td>0.35</td>
<td>0.94</td>
<td>0.66</td>
<td>0.29</td>
</tr>
<tr>
<td>Unet-CNN</td>
<td>0.46</td>
<td>1.05</td>
<td>0.78</td>
<td>0.26</td>
</tr>
</tbody>
</table>

Aggregate Statistics of 101 days (0.75 < Bias < 1.5)

0.75), and COAMPS extreme over-predictions (bias ≥ 1.5). Comparisons between The UNet-CNN and COAMPS for the 6-hour forecasts are displayed in Figures 10 - 12.

The UNet shows a great potential to improve upper tropospheric cloud forecasts even when COAMPS outputs are extremely under and over predicted. However, COAMPS systematic errors still have a strong influence on the UNet performance. In extreme under-predicted cases (bias =
Figure 12. Comparison of the 6-hour temporal forecast cloud fractions and skill scores for cases in the test dataset of COAMPS extreme over-prediction.

0.66), the UNet improves ETS by 54%, bias by 33%, POD by 43%, and also reduces FAR by 15%. In the typical cases (bias = 0.94), the UNet improves ETS by 31%, bias by 12%, POD by 18%, and reduces FAR by 10%. In extreme over-predicted cases (bias = 2.02), The UNet improves ETS by 26%, bias by 6%, PDO by 10%, and reduces FAR by 7%. The UNet is capable of significantly improving cloud forecasts when COAMPS outputs are well predicted or under-predicted, in part due to a systematic under-prediction of thin cirrus layers in COAMPS. For example, on 17 April 2019 (Fig. 9j-l), the GOES-16 observations showed that much of the domain was covered by thin cirrus. Although COAMPS predicted only scattered upper tropospheric clouds, the UNet predicted a more consolidated cloud shield covering much of the same region depicted by GOES. Given the general proclivity for consolidated clouds in the UNet this type of error is easily corrected. Extreme COAMPS over-prediction errors are more difficult to correct, especially because they often occur during nearly clear conditions. On days with only a few scattered clouds, relatively minor over-prediction errors lead to large biases due to the small number of observed pixels in the denominator of Equation 7. On days like this the UNet often removes all small cloud entities as it did in the 1 August case (Fig.9f). Otherwise, the tendency for the production of consolidated clouds limited the ability of the CNN to correct over-predictions of sparse clouds. However, cases when COAMPS predicted areas of spurious clouds were corrected, such as the extraneous region...
of upper tropospheric cloud over the southern Appalachians in Fig. 9m-o. COAMPS commonly predicted too many clouds over the high terrain in this area as evidenced by the maxima in the COAMPS cloud fractions (Figs 10-12).

5. Conclusions

A UNet CNN statistical model was developed to generate 12-hour cloud cover forecasts from COAMPS forecasts, GOES-16 imagery valid at the COAMPS analysis time, and advected GOES-16 clouds using COAMPS winds. Forecasts were generated for five general cloud types using the same generalized UNet architecture, and the results from the upper tropospheric cloud forecasts were discussed here.

Two innovative features of our Unet CNN include the combined binary cross entropy/IoU loss and the bias initializer function. The combined loss function was more effective at training the cloud forecasts which were imbalanced towards true negative values, containing far fewer cloudy pixels than clear ones. The bias initializer function effectively removed large and complex biases introduced by COAMPS while at the same time maintaining high rates of true positive matches. The means for the bias initializer were generated from a validation dataset which was separate from the training and testing sets. These biases are dependent on the COAMPS performance, and will likely need to be regenerated with any improvements in COAMPS.

Verification statistics indicate the CNN was able to remove negative COAMPS upper tropospheric cloud coverage biases while increasing the number of true positive overlapping pixels. Improvements were greatest during the first 3-6 hours, but remained consistent through 12 hours. The advected GOES-16 upper tropospheric clouds performed well during the first 6-9 hours though performance steadily declined. The robust performance of the beyond 6 hours CNN suggests the advected clouds had limited impact compared to COAMPS.

These promising results suggest a number of potential avenues for future work. Since the advected clouds require considerable computing power to generate, a set of denial experiments should be conducted to determine their value as a predictor. The strong performance at 12-hours suggests the benefits may last to longer lead times. The filtered nature of the UNet CNN is an issue that will need to be addressed. Larger training datasets will help, but transformer-based algorithms may also alleviate this problem. Poor performance predicting land-based thunderstorms also suggests
adding a sixth cloud model to represent them would be helpful. Finally, since clouds are 3D, we found it most effective to separate corrections in horizontal position from the vertical extent. The cloud type forecasts represent the corrections to horizontal cloud position alone. Many forecasters rely on vertical properties such as cloud top height, cloud base height, and cloud thickness for aviation and military applications. These vertical properties can now be derived from the corrected positions of the five cloud types using a second set of machine learning models. These models will be trained from a combination of GOES-16 retrievals and active sensor measurements from CloudSat and CALIPSO.
Acknowledgments. This research is supported by a grant from the Naval Research Laboratory under grant number N0001421WX00031. Thanks to Steve Miller, Jeremy Solbrig, and Matt Rogers (CIRA), and Rabi Palikonda and William Smith (NASA) Langley Research Center for help obtaining the satellite retrieval data. Computer resources for the COAMPS simulations and data archival were supported in part by a grant of high performance computing (HPC) time from the Department of Defense Major Shared Resource Center, Stennis Space Center, MS. The work was performed on Cray XC40 and SGI 8600 computing systems.

Data availability statement. The satellite retrieval data were collected daily from NASA and CIRA. NASA LARC daily imagery can be found here: https://satcorps.larc.nasa.gov/ and CIRA daily imagery can be found here: https://rammb.cira.colostate.edu/ramsdis/online/goes-16.asp. The COAMPS forecasts as well as the satellite data interpolated to the analysis grid are stored at the DoD HPC and are considered to be controlled unclassified data and require users to register with the US Government and acquire permission prior to use. More details can be found here: https://www.nrlmry.navy.mil/coamps-web/web/reg.

References


