USING CLOUD COMPUTING TO ANALYZE MODEL OUTPUT ARCHIVED IN ZARR FORMAT

Taylor A. Gowan, John D. Horel, and Alexander A. Jacques

University of Utah Department of Atmospheric Sciences

Salt Lake City, UT

Submitted to: Journal of Atmospheric and Oceanic Technology

Corresponding Author: Taylor A. Gowan, taylor.mccorkle@utah.edu

Abstract

Numerical weather prediction centers rely on the GRIdded Binary Second Edition (GRIB2) file format to efficiently compress and disseminate model output as two-dimensional grids. User processing time and storage requirements are high if many GRIB2 files with size O(100 MB) need to be accessed routinely. We illustrate one approach to overcome such bottlenecks by reformatting GRIB2 model output from the High-Resolution Rapid Refresh (HRRR) model of the National Centers for Environmental Prediction to a cloud-compatible file type, Zarr. The resulting data archive (HRRR-Zarr) is stored using the Amazon Web Service (AWS) Simple Storage Service (S3) and available publicly through the Amazon Sustainability Data Initiative.

Currently, four file types (surface, subhourly, isobaric, and native) of HRRR output are generated for every hour of the day and forecast lead time in GRIB2 format. A HRRR GRIB2 surface file of size O(100 mb) for the contiguous United States region consists currently of 173 grids for a mix of variables and vertical levels with each grid containing 1.9 million grid points. To simplify access to the grids in the surface files, we reorganize the HRRR model output for each variable and vertical level into O(1 MB) Zarr files containing all forecast lead times for 150 x150 grid point subdomains. Open source library routines provide efficient access to the compressed Zarr files using low-memory cloud or local computing resources. The HRRR-Zarr approach is illustrated for machine-learning type applications of sensible weather parameters, including real-time alerts for high-impact situations and retrospective access to output from 100’s-1000’s of model runs.

Significance Statement

The rapid evolution of computing power and data storage have enabled numerical weather prediction forecasts to be generated faster and with more detail than ever before. The increased temporal and spatial resolution of forecast model output can force end users with finite memory and storage capabilities to make pragmatic decisions about which data to retrieve, archive, and process for their applications. We illustrate an approach to alleviate this access bottleneck for common weather analysis and forecasting applications by using the Amazon Web Services (AWS) Simple Storage Service (S3) to store output from the High-Resolution Rapid Refresh (HRRR) model in Zarr format. Zarr is a relatively new file type that is flexible, compressible, and designed to be accessed with open-source software either using cloud or local computing resources. The HRRR-Zarr dataset is publicly available as part of the AWS Sustainability Data Initiative.

I. Introduction

The global weather enterprise relies on millions of large, two-dimensional data fields created each day by operational numerical weather prediction (NWP) models (Benjamin et al. 2018). The perceptions, uses, and values for that vast amount of information depend in part on its accessibility and how it is disseminated by end users (Lazo et al. 2009). Advances in computing processing power and storage have allowed operational centers to run models at finer spatial scales and higher temporal frequency, yet only a small fraction of the information available from the models are typically available to end users (Benjamin et al. 2018). Pragmatic decisions are made by operational forecast centers in order to disseminate global and regional model output for dozens of parameters by restricting ranges and frequencies of valid times and horizontal and vertical grid spacings. Those decisions have been heavily influenced by internal and external limitations on storing and accessing the hundreds of gigabytes (GB) of model output generated by each model run. These challenges are not unique to the weather sector; many disciplines are struggling to overcome the “Volume, Variety, and Velocity” of data cubes (datasets in space and time) available from earth observation systems (Giuliani et al. 2020; Yao et al. 2020). Improved data cube cyber-infrastructures are recognized to be needed for environmental datasets to allow the ingestion, storage, access, analysis, and use of data elements ordered by geolocation and other shared attributes (Nativi et al. 2017).

The National Oceanic and Atmospheric Administration (NOAA) Big Data Program (BDP) began in 2015 to address agency-wide issues to access the tens of terabytes (TB) of observations and model output created each day within the agency (Ansari et al. 2018; NOAA 2020). With support from the NOAA BDP, the NOAA Cooperative Institute for Climate and Satellite-North Carolina (CICS-NC) has implemented a data hub architecture to facilitate transfer of key NOAA environmental datasets to infrastructure-as-a-service (IaaS) providers for data storage. This includes over 130 data streams, including such high-demand data sets as current and historical Next Generation Weather Radar (NEXRAD) products from 160 sites in the United States (Ansari et al. 2018). IaaS providers (e.g., Amazon Web Services [AWS]; Google Cloud Platform; IBM; and Microsoft Azure) have the capacity to store enormous datasets and provide public access and computing resources for end users to post-process these data streams within their IaaS environment to reduce the time and cost to access the information using cloud or local compute resources (Molthan et al. 2015; Siuta et al. 2016).

The Google Cloud Platform and AWS Simple Storage Service (S3) began providing public access during 2020 to output from the High-Resolution Rapid Refresh (HRRR) model of the National Centers for Environmental Prediction (NCEP). The HRRR is a convection-allowing model that was developed by the Earth Systems Research Lab (ESRL) and is run operationally every hour by the NCEP's Environmental Modeling Center. HRRR output is available for dozens of surface and upper-atmospheric variables at 3-km grid spacing for a 1.9 million grid-point domain that covers the contiguous United States (CONUS; Benjamin et al. 2016; Blaylock et al. 2017). As of July 2021, the HRRR archives provided by Google and AWS are now approaching 2 petabytes (PB) in total storage and are growing at a rate of over 700 GB per day.

From 2016-2020, more than 1000 registered operational and research users relied on the only publicly-accessible archive of HRRR model output that was managed by researchers at the University of Utah. This archive utilized S3-type storage procedures provided by the Center for High Performance Computing (Blaylock et al. 2017; Blaylock et al. 2018). By 2020, the archive grew to over 160 TB and the ability to continue to maintain and expand the HRRR archive at the University of Utah was no longer feasible. We began exploring alternative approaches and formats to store HRRR model output for machine learning (ML) applications that currently rely on the GRIB2 model output available now from the IaaS providers.

International standards were established by the World Meteorological Organization (WMO) to efficiently store and disseminate NWP model output in hypercube-structured file formats with built-in compression algorithms. The GRIdded Binary Second Edition (GRIB2) format has been in use during the past several decades to archive two-dimensional files that are efficiently compressed using a method similar to JPEG image compression (Silver and Zender 2017). While GRIB2 files effectively help store and transmit large amounts of meteorological data as two-dimensional slices, they can be cumbersome to work with and rely on WMO-defined tables that are unfamiliar to users in other disciplines (Wang 2014). Many users rely on software tools to transform GRIB2 files into other self-describing formats such as netCDF (Silver and Zender 2017). Decoding the two-dimensional slices in GRIB2 format leads to expanded file sizes that contribute to inefficiencies when, for example, an end user may only be interested in certain parameters for all forecast times available from a specific model run within a local or regional subdomain. However, it is possible to access individual variables within GRIB2 files by selecting their byte range or specifying a bounding box for domain subsets, but doing so requires loading each file into memory and performing additional post-processing (Blaylock et al. 2017).

Researchers generally use high-level programming environments that rely on Matlab, Interactive Data Language (IDL), or Python to examine, post-process, and visualize operational model data. For open-source languages such as Python, few libraries exist that read GRIB2 files efficiently and the hundreds of encoded variables that they contain. Data science and ML techniques applied to operational model output typically require multivariate training datasets with long periods of record for which alternative model data structures beyond GRIB2 are necessary (Vannitsem et al. 2020). As summarized by McGovern et al. (2017), these big data and ML methods have been used to improve forecasts of high-impact weather parameters such as storm duration (Cintineo et al. 2014), severe wind (Lagerquist 2016), large hail (Adams-Selin and Ziegler 2016), precipitation type (Reeves et al. 2014; Elmore et al. 2015) and aviation turbulence (Sharman 2016). To continue applying ML and artificial intelligence techniques to the ever-growing model output repositories, it will be critical to have data in structures that allow for flexible dissection in space, time, and across many forecast model runs or ensemble members (McGovern et al. 2017).

An alternative file format, Zarr, is described in this study as a means to archive HRRR files. Zarr is a relatively new file format, developed in 2016 for use in a Malaria genome project, which chunks and compresses N-dimensional datasets for flexible storage in memory, on disk, or within cloud platforms (Vance et al. 2019; Miles et al. 2020). The Zarr format is being used for promising ML and big data applications in other disciplines, e.g., Lyft Level 5 self-driving dataset (Houston et al. 2020), the MalariaGEN project (Pearson et al. 2019), and the Pangeo project (Eynards-Bontemps et al. 2019; Signell and Pothina 2019).

In the weather enterprise, the United Kingdom's Met Office has adopted Zarr as its file storage format of choice for the over 200 TB of data produced by high-resolution NWP models each day (McCaie 2019). Additionally, Unidata developers of netCDF have extended its netcdf-c library to access Zarr data in a storage format referred to as NCZarr (Heimbigner 2021). While recognizing the potential for Zarr as a file format is of high interest, the Open Geospatial Consortium has not approved Zarr Version 2 yet as an official Community Standard (https://www.ogc.org/pressroom/pressreleases/3275).

The HRRR model output in Zarr format developed in this study (hereafter HRRR-Zarr) is one approach to extract and disseminate model output intended for common ML workflows that may require specific variables from 1-1000s of model runs at specific locations. HRRR-Zarr makes it practical to access relatively small fractions of data rather than attempting to retrieve that data from the original GRIB2 formatted files. The capability to do so is possible since HRRR-Zarr formatted files are being created by our group, stored in the AWS S3 environment, and made publicly accessible as part of the AWS Sustainability Data Initiative, complementing the HRRR GRIB2 model archive available there.

The remainder of this manuscript will be organized in the following manner. We first detail the HRRR model specifications, Zarr capabilities and limitations, and the AWS HRRR-Zarr archive structure. Next, we explore potential use cases for the HRRR-Zarr dataset, for both research or operational applications. We will detail the benefits of the HRRR-Zarr format in a general sense, as well as demonstrate its utility in analyzing a high-impact meteorological event from September 2020 that included record-breaking downslope windstorms in two states, devastating wildfire spread, and an early season snowstorm. Finally, a summary and future work are presented.

II. Data and Methods

a) The High-Resolution Rapid Refresh Model

The High-Resolution Rapid Refresh (HRRR) is a 3-km, convection-allowing model that is run operationally by NCEP's Environmental Modeling Center (Benjamin et al. 2016). It was developed by the Earth Systems Research Laboratory and was first run operationally September 2014. The latest version of the HRRR model (version 4, deployed 2 December 2020), is initialized each hour, with hourly forecasts out to either 18 or 48 hours depending on the initialization time (Table 1). The operational HRRR domain covers the entire CONUS (Fig. 1), with a separate domain for the state of Alaska (McCorkle et al. 2018). The HRRR is nested within the larger domain of the Rapid Refresh model (RAP), from which it receives its initial and boundary conditions for each model run. The RAP employs identical parameterization schemes, with the exception of a convection parameterization, and assimilates data using the NOAA Gridpoint Statistical Interpolation system, which was modified to include hourly radar data, boundary layer observations, and other cloud processes (Kleist et al. 2009). As discussed by McCorkle et al. (2018), the HRRR model is initialized one hour prior to its analysis time, known as a pre-forecast, in order to assimilate three-dimensional radar reflectivity data, which impacts latent heating estimations and thus the HRRR's ability to forecast convection (James and Benjamin 2017).

NOAA BDP and CICS-NC staff manage the distribution of HRRR model output to IaaS providers Google and AWS. We rely on the archive and real-time HRRR GRIB2 files available publicly as part of the AWS Sustainability Data Initiative (https://registry.opendata.aws/noaa-hrrr-pds/). The HRRR GRIB2 files are publicly accessible via the AWS S3 using the unique identifier "noaa-hrrr-bdp-pds".

HRRR output, accessible from IaaS providers contains eight dimensions, listed below (dimensions contained in each GRIB2 file are listed in bold-face text):

- File type: surface, subhourly, isobaric, native
- Domain: CONUS, Alaska
- Initialization time: hourly from 2014 to the present
- Forecast lead time: 15- or 60-min intervals out to 48 h
- **Level: pressure, height, layer**
- **Variable: sensible weather parameters and many model-specific fields**
- **X-position**

- **Y-position**

Table 1 summarizes the evolution of the HRRR model from its initial operational release in 2014 to the present. We only post-process into Zarr format a limited amount of the output from the HRRR (Table 2). We have focused on reformatting the GRIB2 surface files as many ML use cases require surface sensible weather parameters or meteorological parameters at "standard" levels in the vertical that are stored in those GRIB2 files. At present, the volume of HRRR output in Zarr format accessible from AWS exceeds 120 TB.

We focus our description regarding HRRR-Zarr files on the processing of the CONUS surface files, each of size ~140 MB, that contain 173 grids representing a mix of variables at levels in the vertical of highest interest for many applications (Table 1). Output from HRRR model runs initialized at 00, 06, 12, and 18 UTC are available hourly from the analysis time (F00) and hourly forecast lead times out to 48 h (F48). The HRRR model runs initialized at other hours of the days are available from F00-F18.

The subhourly files are similar to the surface files and contain variables with output available at 15-minute forecast lead times. The isobaric and native files contain meteorological variables at fixed pressure or terrain-following levels, respectively, that are most relevant for users who need the HRRR output for initial and boundary conditions to initialize high resolution forecasts or research simulations (e.g., Crosman and Horel 2017; Foster et al. 2017).

b) *Zarr*

Zarr is a flexible file format for storing N-dimensional data arrays that are chunked (divided into subdomains) and compressed with metadata described in separate JSON-formatted files. The Zarr protocol is similar to the Hierarchical Data Format version 5 (HDF5; Delaunay et al. 2019). Zarr files are read and written with the Zarr Python library that depends on the widely-used NumPy library (Harris et al. 2020). The Zarr format is becoming a desirable file structure for data scientists and researchers because of its seamless ability to read and write to cloud platforms. Other benefits include its library of compression options, multithreading and multiprocessing capabilities, and its backend compatibility with format-agnostic, array-manipulation Python libraries (e.g., xarray, iris, and dask).

A Zarr file is initialized using Python by first defining the file store, which can be in memory, as a directory on local disk, in distributed or cloud storage, or as a zip file. Next, Zarr arrays (hereafter, zarrays) are created and filled in a similar manner to NumPy arrays by defining a data type and shape, and then assigning values and defining zarray attributes (zattrs) described in JSON files that will serve as the key references for that zarray. These zarrays can be chunked along any specified dimension and in any shape, which allows a dataset to be manipulated and stored efficiently for use in specific applications. All chunks in a zarray are uniform in shape and stored as individual objects that are identified by their integer index location in the array (e.g., row and column). The process of defining an optimal chunk structure for the HRRR-Zarr dataset is outlined in the next subsection.

Before sending the chunked zarrays to the Zarr store, they can be encoded and compressed for optimized storage. The encoding instructions are located in the json metadata for the zarray and define the data type's byte order (little endian or big endian), character code (integer, floating point, Boolean, etc.), and the number of bytes. All data types in the NumPy array protocol are acceptable for zarray encoding. Once the encoding parameters have been defined, the zarrays can be compressed using a number of compression algorithms and data filters. The Numcodecs library was designed specifically for data storage applications like Zarr, and serves as an interface to other compressor libraries such as Blosc, Zstandard, LZ4, Zlib, and LZMA. This allows the user to choose the primary compressor, the compression algorithm, and the compression level that will perform best based on the applications of the dataset. When choosing a compression codec and level, the user takes into consideration the potential compression ratio (Eqn. 1) and decoding speed.

$$Compression\ Ratio = \frac{Uncompressed\ Data\ Volume}{Compressed\ Data\ Volume} \qquad (3.1)$$

There exists a plethora of literature that details compression algorithm performance and benchmark test results that aid choosing appropriate compression schemes for a particular use case (Donoho 1993; Alted 2010; Almeida et al. 2014; Wang et al. 2015; Kuhn et al. 2016). In addition to choosing a compression scheme, the Numcodecs library also offers a number of data filters that can be implemented. The filter sorts the data and transforms it in a way that would streamline compression, such as shuffling bytes and bits when adjacent values in an array are correlated. The compatibility of the Zarr protocol and Numcodecs libraries allows for the configuration of an external filter for use with any chosen compressor, even if the filter is not an option by default.

c) The HRRR-Zarr Archive

The HRRR-Zarr archive with the unique AWS S3 bucket identifier “hrrrzarr” is available publicly as part of the AWS Sustainability Data Initiative. That archive was designed to be relevant for users less familiar with environmental dataset formats while supporting a familiar environment for users who routinely use model output in netCDF or GRIB2 format. The HRRR-Zarr conversion workflow follows that of the United Kingdom’s Met Office Informatics Lab, where they are actively using Zarr to store large datasets (Donkers 2020). Due to the challenges that surround manipulating data cubes in various file formats, the Met Office developed the Iris Python library, a format-agnostic library for processing datasets and converting between file formats (Iris 2020). Unlike other Python libraries, Iris and its companion package, Iris-grib, were built to read data cube formats such as GRIB2 and recognize the Climate and Forecast (CF; Eaton et al. 2020) metadata conventions used in numerical model data. For this reason, the Iris libraries maintain the same metadata for the HRRR archive as relied upon for GRIB2 files.

The HRRR-Zarr archive files are built using the Iris and Iris-grib libraries. HRRR-Zarr data files rely on the same self-describing metadata (keywords and CF naming scheme) as the corresponding GRIB2 files obtained from their associated index (.idx) files. As an example, consider the workflow required to process the 48 GRIB2 surface forecast files containing 173 grids from the HRRR model runs initialized at 00 UTC. (The CF names for all 173 HRRR variables are available online at https://mesowest.utah.edu/html/hrrr/zarr_documentation/html/zarr_variables.html). All of the grids from the 48 hourly forecast files are read into memory and then organized into unique Iris data cubes containing data and metadata. The Iris data cubes are then converted to zarrays that are subdivided (chunked), encoded, and output into separate files identified by the parameter's CF name and atmospheric level or layer (e.g., 2-m, 500 mb).

As shown in Fig. 1, the 1799 x 1059 grid is subdivided into 96 chunks of size 150 x 150 based on recommendations for optimal data compression. It should be noted, the 12 chunks along the domain's northern boundary contain data in only the southernmost nine rows. HRRR CONUS analysis (F00) files, whether for surface or isobaric files, are simply subdivided into 96 tiny 2-D files each containing one 150 x150 grid point array. HRRR CONUS forecast (F01-FXX) files are stored as 96 3-D cubes (XX,150,150) where the forecast duration, FXX, depends on HRRR version and time of day (Table 1). Although the data are chunked, the entire domain can still be accessed efficiently, in terms of memory and processing time, with the Zarr Python library. Processing time may be increased when parsing the entire domain, but accessing a single variable for many times using Zarr files is five times faster than attempting the same operation from the original GRIB2 files.

To consolidate the dataset, we chose the LZ4 compression codec, which is a lossless compression algorithm with the ability to quickly and efficiently compress large amounts of data (Collett 2020). The zarrays are encoded as 16-bit little-endian floats, with the exception of the surface pressure parameter, which requires 32-bit little-endian floats. We access the LZ4 algorithm using the Numcodecs library class, Blosc, which is a meta-compressor that imitates the utility of the built-in Python library zlib. When using the LZ4 compression algorithm, additional modifications can be made to tailor the scheme for a particular use. We chose byte shuffling and a compression level of 9 within a range of 1-12 where levels 1 and 12 provide the fastest compression speed and highest compression ratio, respectively. While other compression codecs have been shown to produce higher data compression ratios, their decompression speeds are much slower (Collett 2020).

It is widely recognized that optimal use of cloud resources requires having data processing and analysis within the same compute environment as the data archive. The HRRR-Zarr files are created shortly after the entire model run is accessible as objects in the AWS GRIB2 S3 archive using AWS Elastic Cloud Compute resources in the same region (West-1) as our hrrrzarr bucket. Because of the time required to wait until all GRIB2 forecast fields are available, the most recent Zarr files are typically available 3 hours after the initialization time, e.g., 00 UTC analysis and forecast files are available by 03 UTC. This is dependent on the availability of the GRIB2 data in the AWS archive managed by NOAA BDP.

Within the hrrrzarr S3 bucket, all files (or objects) are contained in a flat structure where the concept of folder or directory structure is provided by using shared name prefixes or suffixes for objects that mimic traditional directory structure. The zarr files derived from the surface and isobaric sets of HRRR GRIB2 files are stored with the prefixes “sfc/” and “prs/”, respectively (Fig. 2). Model runs are accessible by date, using suffixes for analysis (anl.zarr) and forecast (fcst.zarr) files for each model run, e.g., files with the prefix sfc/20200907_12z_fcst.zarr/ were generated from the F01-F48 HRRR GRIB2 surface files initialized at 1200 UTC 7 Sept 2020. Prefixes follow then based on level (e.g., 700 mb/ or 10m_above_ground/) and CF naming conventions for variables a (e.g., TMP/ or UGRD/) with the final part of the file name being the chunk identifier. A full list of variables (abbreviation and full name) available in the HRRR v4 output and HRRR-Zarr files is available online (https://mesowest.utah.edu/html/hrrr/zarr_documentation/html/zarr_variables.html). Users can download the specific files of interest by accessing them by full name using web tools or from within programs in Python or other languages.

III. HRRR-Zarr Applications and Discussion

The HRRR-Zarr archive was developed with the intention of expanding its utility for ML and other applications that require high velocity file throughput. While demonstrating a full ML scenario is outside the scope here, this section illustrates examples of situations where the Zarr file format may be optimal in terms of efficiency and ease of use. We will use a high-impact meteorological event from September 2020 to showcase the utility of model data in Zarr format for not only research applications, but operational decision making and forecasting use cases as well. This section of the paper will be comprised of subsections that detail the event we are analyzing, followed by example use cases for HRRR model output in Zarr format.

a) Labor Day Weather Event (7-9 September 2020)

In the days leading up to the historic 2020 Labor Day weather event, forecasters in the western half of the United States were on high alert for the extratopical transition of Typhoon Julian as it began recurving poleward and easterward. When extratropical transitions occur, tropical cyclones may interact with the midlatitude flow such that the mid-latitude ridge-wave patterns amplify and high-impact weather occurs downstream (Bosart and Carr 1978; Cordeira et al. 2013; Feser et al. 2015; Keller et al. 2019). In this case, Typhoon Julian did modify the midlatitude wave pattern by amplifying both the anticyclone over the Gulf of Alaska and the midlatitude cyclone situated over Western Canada. The rapid intensification of this ridge-trough pair ultimately produced farreaching effects including historic windstorms and unrelenting wildfire spread (Fig. 3) in the Pacific Northwest, strong downslope winds in Utah and a snowstorm in Colorado. We focus here on the data and model forecasts pertaining to the events that occurred in Oregon, west of the Cascade Mountains.

The Labor Day weather event was synoptically-driven and well-forecasted several days in advance. Prior to the trough arrival and onset of the downslope windstorm, the Pacific Northwest was experiencing extreme fire danger due to warm and dry conditions, with several fires already burning in Washington and Oregon. By 12 UTC on September 7, a thermal trough was situated over coastal Oregon with a tightening pressure gradient orthogonal to it. These conditions are indicative of impending strong northeast and easterly winds in western Oregon. As forecasted, strong easterly winds arrived on the western side of the Oregon Cascades by 00 UTC on September 8. In a near worst case scenario, wind gusts along the western slopes of the Oregon Cascades ignited new fires (Riverside Fire) and significantly intensified existing wildfires (Beechie Creek). For nearly a week after the onset of the downslope winds, persistent easterly flow propagated wildfire smoke west, resulting in historic PM2.5 measurements in excess of 500 micrograms per cubic meter in Portland, Salem, and Eugene, OR (Green 2020). Suppression efforts were minimal given the steep terrain surrounding the wildfires, making it too dangerous for fire crews to extinguish them safely. Ultimately, the Riverside and Beechie Creek fires burned over 1,300 km^2. The following subsections use the data during this weather event to illustrate use cases for future ML applications for operational forecasting and research.

b) Forecast Time Series for a Specific Location

Time series are one of the most straightforward and widely understood visualizations used to show how a given parameter evolves over a period. Scientists and consumers alike are exposed to time series every day when looking at stock market trends, weather forecasts, and health tracking applications. Despite their inherent simplicity, requiring only time and a dependent variable as input, they can be time consuming and challenging to create when starting from data files that represent a single time in space for millions of locations, as is the case with NWP model output in GRIB2 format. As discussed earlier, a HRRR GRIB2 file of size O(100 MB) contains hundreds of two-dimensional forecast fields for a single valid time. Retrieving, storing, and unpacking 18, 36, or 48 such files up to 24 times a day is beyond what many users can deal with in terms of compute power and storage.

Efficient access to model output as time series for specific locations was a key objective leading to the structure and organization of the HRRR-Zarr format. While identical time series can be constructed from both GRIB2 and Zarr file formats, the process and requirements are quite different. As discussed in the Data and Methods section, the tiny two-dimensional analysis HRRR-Zarr files can be easily accessed to estimate prior conditions at a location while the three-dimensional forecast HRRR-Zarr files contain all forecast hours from a model run to assess how future conditions at that location may unfold.

To illustrate the utility of the Zarr format for this use case, we plot time series of forecast wind gusts from the 00, 06, 12, and 18 UTC HRRR model runs for a single point from 12 UTC 6 Sept-18 UTC 8 Sept 2020 (Fig. 3). For this case, we chose the HRRR grid point nearest to the Horse

Creek (Station ID: HSFO3) Remote Automated Weather Station (44.940806^{o}N, 122.400806^{o}W) located downwind of the Beechie Creek Fire. However, since HSFO3 is located in a clearing within a densely forested region, the wind reports from this location tended to be lower than what was evident by the rapid advance of the fire line in that region. To create this visualization of forecasted wind gust, ten small chunks of data (one from each model run) totaling ~ 10 MB were retrieved from the hrrrzarr bucket to obtain all necessary model output. In contrast, 360 GRIB2 files totaling ~54 GB would have been needed to replicate this process or else values within byte ranges in each of those files would need to be determined and accessed. The single access point to all forecast hours in a model run reduces processing time and optimizes workflows for applications such as creating training datasets for a ML model.

Plotting sequentially the model runs available every hour creates a time-lagged ensemble (TLE) for a given valid time. A TLE from HRRR output can provide useful diagnostics for evaluating the uncertainty or spread in values among recent forecasts for which the most recent forecast provides only deterministic guidance (Xu et al. 2019). TLEs with sufficient lead time to be potentially useful operationally can be constructed using a set of sequential HRRR forecasts, with each model run treated as an ensemble member. In this case, we use F06-F18 forecasts from all model runs initialized from 06 UTC 6 September – 06 UTC 9 September to calculate statistics at valid times from 00 UTC 7 September – 12 UTC 9 September. Diagnostic values such as median, minimum, and maximum forecasted wind gusts provide a simple evaluation of the model's uncertainty as the event unfolded (Fig. 4). The unrepresentativeness of the lighter HSFO3 observations relative to that analyzed and forecasted by the HRRR is evident in Fig. 4.

c) Spatial Analysis of Forecast Data

Many applications requiring HRRR model output need only a fraction of the 1.9 million grid points in the HRRR CONUS domain. Hence, users typically implement methods to subset areas of interest from the complete grids. Accessing one or more HRRR-Zarr chunks of size 450 km^2 may help simplify that process for many local applications while adjacent chunks can be stitched together to evaluate conditions for regional scales.

Model analyses are often used as proxies for observations, especially in areas of complex terrain where in-situ measurements may not be available or unrepresentative of prevailing conditions (e.g., Fig. 5). As a further example, we use the HRRR-Zarr analysis files to determine the onset time of wind gusts exceeding 10 m s^{-1} for every point within the Western Oregon chunk encompassing the large fires underway on 7-8 September 2020 (Fig. 6). This wind gust threshold was chosen based on criteria commonly used for red flag warnings issued by the National Weather Service. The filled contours in Fig. 6 depict the approximate onset time of the downslope windstorm event across western Oregon, with the event beginning along the highest reaches of the Cascade Range and then progressing westward later. Such diagnostics can then be related to available wind observations and damage reports to help evaluate the ability of the HRRR model to forecast the temporal evolution of the event.

Building on the TLE concept available from consecutive HRRR forecasts, we calculate the probability of a HRRR wind gust forecast exceeding 10 m s^{-1} at a given time during the downslope windstorm for all grid points within the Western Oregon chunk. The 21 model runs (F01-F18, F24, F30, and F36) available from forecasts valid at 00 and 06 UTC 8 September 2020 are used to calculate the fraction of wind gusts forecasts exceeding that threshold in this subregion (Fig. 7). Using such probabilistic guidance as the event developed, forecasters might have higher confidence that the HRRR model forecasts issued earlier are being confirmed by more recent forecasts as the downslope winds continued. For a single valid time, this metric utilized wind gust values within 20 HRRR-Zarr files, which required less than 20 MB of storage capacity, an amount easily manageable in computer memory. Actual forecast applications might limit the TLE members to those available at least 12 hours in advance, e.g., forecasts with lead times from F12-F18 and those available every 6 h out to 48 h from the HRRRv4 model output now available.

d) Empirical Cumulative Distributions

Empirical cumulative distributions of model data and observations are often utilized to better understand the range of possible values for a given parameter as a function of time and/or location and can be used to correct for model biases (Blaylock et al. 2018, Gowan and Horel 2020). If enough data are available over an adequate period of time, these cumulative distributions can be thought of as a climatology and used for comparison to a parameter at an equivalent time or location in order to recognize conditions that are likely anomalous. Creating distributions from observations or model output typically requires data from thousands of input times and files for the information to be considered useful. This can be a daunting and time-consuming task since a large amount of storage and compute power are needed to efficiently process thousands of GRIB2 data files.

Blaylock et al. (2018) presented an approach to compute empirical cumulative distributions of HRRR model output at all 1.9 million grid points that required harnessing the Open Science Grid

(OSG). The OSG allows users to send jobs that are repetitive in nature (e.g., statistical calculations using large datasets, data mining, etc.) to unused or idle computing resources at hundreds of locations within the OSG consortium, reducing the overall processing time for a given workflow. The OSG method enables large amounts of data to be simultaneously processed, but its complexities can be a drawback for most users without a thorough understanding of the system. Continually updating cumulative distributions using this approach is also difficult to sustain.

A quick and efficient method is illustrated here to generate empirical cumulative distributions of atmospheric parameters from the HRRR-Zarr archive. To assess the anomalous nature of the downslope wind event during September 2020 in northern Oregon, we generated cumulative distributions of wind gust data for each grid point in that region by accessing all HRRR hourly analyses during the month of September during the preceding years 2016-2019. Each grid point's cumulative distribution is derived then from 2,880 wind gust values, one from every hourly HRRR analysis during the four calendar months. A range of percentiles can be derived from the empirical distributions to estimate normal and above normal wind gusts in this area during the month of September.

As expected, the highest wind gusts evident from the 95th percentile values during September 2016-2019 tend to occur over the Cascade Range and offshore (Fig. 8). Using this four-year distribution, we then compare the 95th percentile values to the analysis and F06, F12, and F18 forecasts valid at 06 UTC on 8 September 2020 (Fig. 9). To emphasize the severity of the event across the region, the excess magnitude of wind gusts values above the 95th percentile are shown. Comparing these forecasts and analysis to the cumulative distribution is a simple way to show how anomalous this event was, with wind gusts exceeding the 95th percentile values by 15-30 m s^{-1} over the Cascade and portions of the Coast Ranges and extending into sections of the Willamette Valley.

The empirical distributions computed using four months of data for a single variable and chunk required less than a minute on a typical workstation. We compare this to the method used by Blaylock et al. (2018), which calculated empirical cumulative distributions for all HRRR model grid points. These distributions were then used to output wind speed values at 19 percentiles at all HRRR grid points for each day of the year. As previously stated, this was a rigorous and time-intensive endeavor that required an enormous amount of model output. Ultimately, calculating these distributions resulted in the need to store 6,935 additional files containing the percentiles at each of the 1.9 million HRRR grid points.

Calculating empirical cumulative distributions, as well as other large-scale statistical metrics, with data in Zarr format gives the end user the ability to continually update their statistics as new information is received. This method especially benefits users who are interested in time-sensitive datasets, like those from numerical weather prediction models. Using the HRRR-Zarr method, a user will be able to efficiently compute statistics that are tailored to a specific application or workflow, without dealing with the overhead of many GB of excess data.

IV. Summary

Vast amounts of output produced by numerical weather prediction models are accessed and processed every day for applications ranging from operational forecasting to research and machine learning. As advancements in technology allow for finer time and spatial resolution model output, users may struggle to keep up even if they are only interested in accessing a small fraction of the data available. Much of this model output is currently available in GRIB2-formatted files containing hundreds of two-dimensional variable fields for a single valid time. Despite the highly compressible nature of GRIB2 files, they are often O(100 MB) each, making high-volume input/output applications challenging due to the memory and compute resources needed to parse them.

We present an approach that reorganizes HRRR analyses (F00) from the surface and isobaric HRRR file types into tiny two-dimensional (150, 150) files in Zarr format for each variable/vertical level combination and 96 subdomains of the CONUS grid. HRRR forecasts from the surface and isobaric files are stored as data cubes (XX, 150,150) where the forecast dimension XX is either 48 for initialization times of 00, 06, 12, and 18 UTC or 18 for all other hours. We create the Zarr formatted files from the HRRR GRIB2 files provided by the NOAA BDP with support provided by the Amazon Sustainability Data Initiative. Our supplementary S3 bucket, hrrrzarr, is publicly accessible as part of the Amazon Initiative.

The structure of the HRRR-Zarr files was designed to allow users the flexibility to access only the data they need through selecting subdomains and parameters of interest without the overhead of memory and processing requirements that comes from accessing numerous large GRIB2 files. Users may retrieve the analysis files needed to diagnose prior conditions or retrieve the forecast files in combination with the analysis files to evaluate future conditions or validate prior forecasts.

Using a high-impact weather event from September 2020, we present workflow examples for analyzing large amounts of sensible weather parameters from the HRRR-Zarr data archive: assembling time series for a specific grid point of forecast conditions over a range of model runs; examining similarities and differences among samples of model forecasts for the same valid times from successive model runs; calculating empirical cumulative distributions over multiyear periods; and detecting forecasts of extreme conditions relative to conditions during other recent years. The small, compressed chunks of data are ideal for high-throughput workflows where minimizing processing time or accessing files corresponding to many different valid times is critical. However, relying on the GRIB2 HRRR files accessible from AWS and Google remains the best option for initializing high resolution model simulations that require many variables at multiple levels over a limited sample of valid times.

The GRIB2-to-Zarr conversion of the HRRR model archive is only one of the many research endeavors that aim to make model data more accessible to end users. As technology and computational power continues to advance, the inevitable path of numerical weather prediction models is towards probabilistic guidance from ensemble forecasting systems (Frogner et al. 2019; Schwartz et al. 2019). Ensemble models introduce an additional data dimension (number of members), which compounds the volume of data produced by each model run. Plans are being developed for GRIB2-to-Zarr conversion of model output from the NOAA Global Ensemble Forecast System (GEFS), which are also available through the NOAA Big Data Program and the Amazon Sustainability Initiative (https://registry.opendata.aws/noaa-gefs/; https://registry.opendata.aws/noaa-gefs-reforecast/). Great utilization of Zarr within the broader community will likely follow if the Open Geospatial Consortium adopts Zarr as an official community data standard. That will likely lead to diverse efforts to evaluate Zarr in the cloud environment relevant to users who want the flexibility to customize a Zarr file structure for their own purposes. Utilizing the Zarr format as an alternative file structure for the vast amount of numerical weather prediction output may help expand its already wide reach to data scientists in other disciplines, while optimizing workflows for end users throughout the weather enterprise.

Acknowledgements

This research is supported by the NOAA/National Weather Service Collaborative Science, Technology, and Applied Research (CSTAR) Program Awards (55500146 and NA20NWS4680046). Creation of the HRRR-Zarr archive would not have been possible without an Amazon Sustainability Data Initiative Promotional Credits Award. The authors would like to thank Zac Flamig for his assistance and time in facilitating the inclusion of the HRRR-Zarr dataset in the AWS Open Data Registry. The authors would also like to recognize James Powell and Adair Kovack, who tested the dataset and provided valuable feedback. The University of Utah Center for High Performance Computing (CHPC) provided computational hardware and software for this work.

Data Availability Statement

The HRRR-Zarr archive is publicly available in the AWS Open Data Registry, made possible by credits awarded from the Amazon Sustainability Data Initiative https://registry.opendata.aws/noaa-hrrr-pds/.

Metadata documentation related to the HRRR model can be found at https://dx.doi.org/10.7278/S5JQ0Z5B.

References

Adams-Selin, R. D., and C. L. Ziegler, 2016: Forecasting hail using a one-dimensional hail growth model within WRF. *Mon. Wea. Rev.*, **144**, 4919–4939, doi:10.1175/MWR-D-16-0027.1.

Almeida, S., V. Oliveira, A. Pina, and M. Melle-Franco, 2014: Two High-Performance Alternatives to ZLIB Scientific-Data Compression. *Proc. of 14th International Conference on Computational Science and Its Applications (ICCSA 2014)*, Guimarães, Portugal, 623–638, doi:10.1007/978-3-319-09147-1_45.

Alted, F., 2014: Why Modern CPUs Are Starving and What Can Be Done About It. *Comput. Sci. Eng.*, **12**, 68–71, doi:10.1109/MCSE.2010.51.

Amazon, 2021: Amazon S3: Object storage built to store and retrieve any amount of data from anywhere. Accessed 2 January 2021, https://aws.amazon.com/s3/.

Ansari S., S. Del Greco, E. Kearns, O. Brown, S. Wilkins, M. Ramamurthy, J. Weber, R. May, J. Sundwall, J. Layton, A. Gold, A. Pasch, and V. Lakshmanan, 2018: Unlocking the potential of NEXRAD data through NOAA's Big Data Partnership. *Bull. Amer. Meteor. Soc.*, **99**, 189–204, doi:10.1175/BAMS-D-16-0021.1.

Benjamin, S. G., J. M. Brown, G. Brunet, P. Lynch, K. Saito, and T. W. Schlatter, 2018: *100 Years of Progress in Forecasting and NWP Applications*, *Meteor. Monogr.*, **59**, 13.1–

13.67, doi:10.1175/AMSMONOGRAPHS-D-18-0020.1.

—, S. S. Weygandt, J. M. Brown, M. Hu, C. R. Alexander, T. G. Smirnova, J. B. Olson, E. P. James, D. C. Dowell, G. A. Grell, H. Lin, S. E. Peckham, T. L. Smith, W. R. Moninger, J. S. Kenyon, and G. S. Manikin, 2016: A North American Hourly Assimilation and Model Forecast Cycle: The Rapid Refresh, 2016. *Mon. Wea. Rev.*, **144**, 1669–1694, doi:10.1175/MWR-D-15-0242.1.

Blaylock, B., J. Horel, E. Crosman, 2017: Impact of Lake Breezes on Summer Ozone Concentrations in the Salt Lake Valley. J. Appl. Meteor. Clim.56, 353-370. http://journals.ametsoc.org/doi/abs/10.1175/JAMC-D-16-0216.1

Blaylock, B. K., J. D. Horel, and S. T. Liston, 2017: Cloud archiving and data mining of High-Resolution Rapid Refresh forecast model output. *Computers and Geosciences*, **109**, 43–50, doi: 10.1016/j.cageo.2017.08.005.

—, —, and C. Galli, 2018: High-Resolution Rapid Refresh Model Data Analytics Derived on the Open Science Grid to Assist Wildland Fire Weather Assessment. *J. Atmos. Oceanic Technol.*, **35**, 2213–2227, doi:10.1175/JTECH-D-18-0073.1.

Bosart, L. F., and F. H. Carr, 1978: A Case Study of Excessive Rainfall Centered Around Wellsville, New York, 20–21 June 1972. *Mon. Wea. Rev.*, **106**, doi:10.1175/1520-

0493(1978)106<0348:ACSOER>2.0.CO;2.

Cintineo, J. L., M. J. Pavolonis, J. M. Sieglaff, and D. T. Lindsey, 2014: An empirical model for assessing the severe weather potential of developing convection. *Wea. Forecasting*, **29**, 639–653, doi:10.1175/WAF-D-13-00113.1.

Collett, Y., 2020: LZ4 – Extremely fast compression, version 1.6.2. Accessed 1 January 2021, https://lz4.github.io/lz4/.

Cordeira, J. M., F. M. Ralph, and B. J. Moore, 2013: The development and evolution of two atmospheric rivers in proximity to western North Pacific tropical cyclones in October 2010. *Mon. Wea. Rev.*, **141**, 4234–4255, doi:10.1175/2010MWR2888.1.

Crosman, E., J. Horel, 2017: Large-eddy simulations of a Salt Lake Valley cold-air pool. Atmospheric Research. 193, 10–25, doi:10.1016/j.atmosres.2017.04.010

Delaunay, X., A. Courtois, and F. Gouillon, 2019: Evaluation of lossless and lossy algorithms for the compression of scientific datasets in netCDF-4 or HDF5 files. *Geosci. Model Dev.*, **12**, 4099–4113, doi:10.5194/gmd-12-4099-2019.

Donkers, K., 2020: To the cloud and back again. Medium, accessed 28 December 2020, https://medium.com/informatics-lab/create-zarr-from-pp-files-ffa6b7972d6f.

Donoho, D. L., 1993: Unconditional Bases Are Optimal Bases for Data Compression and for Statistical Estimation. *Appl. Comput. Harmonic Anal.*, **1**, 100–115, doi:10.1006/acha.1993.1008.

Eaton, B., J. Gregory, B. Drach, K. Taylor, S. Hankin, J. Blower, J. Caron, R. Signell, P. Bentley, G. Rappa, H. Höck, A. Pamment, M. Juckes, M. Raspaud, R. Horne, T. Whiteaker, D. Blodgett, C. Zender, and D. Lee, 2020: NetCDF Climate and Forecast (CF) Metadata Conventions version 1.8. Accessed 27 December 2020, https://cfconventions.org/Data/cf-conventions/cf-conventions-1.8/cf-conventions.pdf.

Elmore, K. L., Z. L. Flamig, V. Lakshmanan, B. T. Kaney, V. Farmer, H. D. Reeves, and L. P. Rothfusz, 2015: Verifying forecast precipitation type with mPING. *Wea. Forecasting*, **30**, 656–667, doi:10.1175/WAF-D-14-00068.1.

Eynard-Bontemps, G., R. Abernathey, J. Hamman, A. Ponte, and W. Rath, 2019: The Pangeo Big Data Ecosystem and its use at CNES. *Proc. of the 2019 conference on Big Data from Space (BiDS' 2019),* Munich, Germany, 49–52, doi:10.2760/848593.

Feser, F., M. Schubert-Frisus, H. von Storch, M. Zahn, M. Barcikowska, S. Haeseler, C. Lefebvre, and M. Stendel, 2015: Hurrican Gonzalo and its extratropical transition to a strong European storm. *Bull. Amer. Meteor. Soc.*, **96**, S51–S55, doi:10.1175/BAMS-D-15-00122.1.

Foster, C., E. Crosman, J. Horel, 2017: Simulations of a Cold-Air Pool in Utah's Salt Lake Valley: Sensitivity to Land Use and Snow Cover. Boundary Layer Meteorology. 164, 63-87, doi:10.1007/s10546-017-0240-7

Frogner, I- L, A. Singleton, M. Køltzow, U. Andrae, 2019: Convection- permitting ensembles: Challenges related to their design and use. *Q. J. R. Meteorol Soc.*, **2019**, 1–17. doi:10.1002/qj.3525.

Giuliani, G., B. Chatenoux, T. Piller, F. Moser, and P. Lacroix, 2020: Data Cube on Demand (DcoD): Generating an earth observation Data Cube anywhere in the world. *Int. J. Appl. Earth. Obs. Geoinformation*, **87**, 1–6, doi:10.1016/j.jag.2019.102035.

Gowan, T. A., and J. Horel, 2020: Evaluation of IMERG-E Precipitation Estimates for Fire Weather Applications in Alaska. Wea. Forecasting, 35, 1831–1843, doi:10.1175/WAF-D-20-0023.1.

Green, A., 2020: Portland's air quality is off the charts Sunday, and parts of Oregon are just as bad due to wildfires. The Oregonian/Oregon Live. Accessed 15 January 2021, https://www.oregonlive.com/news/2020/09/portlands-air-quality-is-off-the-charts-on-sunday-and-much-of-oregon-is-just-as-bad-due-to-wildfires.html.

Harris, C. R. and Coauthors, 2020: Array programming with NumPy. *Nature*, **585**, 357–362, doi:10.1038/s41586-020-2649-2.

Heimbigner, D., 2021: Overview of Zarr Support in netCDF-C. Unidata and the UCAR Community Programs. Accessed 31 May 2021, https://www.unidata.ucar.edu/blogs/developer/en/entry/overview-of-zarr-support-in.

Houston, J., G. Zuidhof, L. Bergamini, Y. Ye, A. Jain, S. Omari, V. Iglovikov, and P. Ondruska, 2020: One Thousand and One Hours: Self Driving Motion Prediction Dataset. Lyft, accessed 9 December 2020, https://level5.lyft.com/dataset/.

Iris, 2020: Iris: A Python library for analysing and visualising meteorological and oceanographic datasets, version 2.4.0. Met Office, accessed 22 December 2020, https://scitools.org.uk/iris.

James, E. P., and S. G. Benjamin, 2017: Observation System Experiments with the Hourly Updating Rapid Refresh Model Using GSI Hybrid Ensemble–Variational Data Assimilation. *Mon. Wea. Rev.*, **145**, 2897–2918, doi:10.1175/MWR-D-16-0398.1.

Keller, J. H., C. M. Grams, M. Reimer, H. M. Archambault, L. Bosart, J. D. Doyle, J. L. Evans, T. J. Galarneau Jr., K. Griffin, P. A. Harr, N. Kitabatake, R. McTaggart-Cowan, F. Pantillon, J. F. Quinting, C. A. Reynolds, E. A. Ritchie, R. D. Torn, and F. Zhang, 2019: The Extratropical Transition of Tropical Cyclones. Part II: Interaction with the

Midlatitude Flow, Downstream Impacts, and Implications for Predictability. *Mon. Wea. Rev.*, **147**, 1077–1106, doi:10.1175/MWR-D-17-0329.1.

Kleist, D. T., D. F. Parrish, J. C. Derber, R. Treadon, W. Wu, and S. Lord, 2009: Introduction of the GSI into the NCEP Global Data Assimilation System. *Wea. Forecasting*, **24**, 1691–1705, doi:10.1175/2009WAF2222201.1.

Kuhn, M., J. M. Kunkel, and T. Ludwig, 2016: Data compression for climate data. *Supercomput. Front. Innovations*, **3**, 75–94, doi:10.14529/jsfi160105.

Lagerquist, R., 2016: Using machine learning to predict damaging straight-line convective winds. M.S. thesis, School of Meteorology, University of Oklahoma, 251 pp. [Available online at http://hdl.handle.net/11244/44921.]

Lazo, J. K., R. E. Morss, and J. L. Demuth, 2009: 300 billion served: Sources, perceptions, uses, and values of weather forecasts. *Bull. Amer. Meteor. Soc.*, **90**, 785–798, doi:10.1175/2008BAMS2604.1.

McCaie, T., 2019: Creating a data format for high momentum datasets. Medium, accessed 20 December 2020, https://medium.com/informatics-lab/creating-a-data-format-for-high-momentum-datasets-a394fa48b671.

McCorkle, T. A., J. D. Horel, A. A. Jacques, and T. Alcott, 2018: Evaluating the Experimental High-Resolution Rapid Refresh – Alaska Modeling System using USArray Pressure

Observations, *Wea. Forecasting*, **33**, 933–953, doi: 10.1175/WAF-D-17-0155.1.

McGovern, A., K. L. Elmore, D. J. Gagne, S. E. Haupt, C. D. Karstens, R. Lagerquist, T. Smith, and J. K. Williams, 2017: Using artificial intelligence to improve real-time decision-making for high-impact weather. *Bull. Amer. Meteor. Soc.*, **98**, 2073–2090, doi:10.1175/BAMS-D-16-0123.1.

Miles, A., and Coauthors, 2020: zarr-developers/zarr-python: v2.5.0. https://zenodo.org/record/4069231.

Molthan, A. L., J. L. Case, J. Venner, R. Schroeder, M. R. Checchi, B. T. Zavodsky, A. Limaye, and R. G. O'Brien, 2015: Clouds in the Cloud: Weather Forecasts and Applications within Cloud Computing Environments. *Bull. Amer. Meteor. Soc.*, **96**, 1369–1379, doi:10.1175/BAMS-D-14-00013.1.

National Oceanic and Atmospheric Administration (NOAA), 2020: Big Data Program. Accessed 8 December 2020, https://www.noaa.gov/organization/information-technology/big-data-program.

Nativi, S., P. Mazzetti, and M. Craglia, 2017: A view-based model of data-cube to support big earth data systems interoperability. *Big Earth Data*, **1**, 75–99, doi:10.1080/20964471.2017.1404232.

Pearson, R. D., R. Amato, and D. P. Kwiatkowski, and MalariaGEN Plasmodium Falciparum Community Project, 2019: An open dataset of *Plasmodium falciparum* genome variation in 7,000 worldwide samples, *BioRxiv*, doi: 10.1101/824730.

Reeves, H. D., K. L. Elmore, A. Ryzhkov, T. Schuur, and J. Krause, 2014: Sources of uncertainty in precipitation-type forecasting. *Wea. Forecasting*, **29**, 936–953, doi:10.1175/WAF-D-14-00007.1.

Schwartz, C.S., G.S. Romine, R.A. Sobash, K.R. Fossell, and M.L. Weisman, 2019: NCAR's Real-Time Convection-Allowing Ensemble Project. *Bull. Amer. Meteor. Soc.*, **100**, 321–343, doi:10.1175/BAMS-D-17-0297.1.

Sharman, R., 2016: Nature of aviation turbulence. *Aviation Turbulence: Processes, Detection, Prediction*, R. Sharman and T. Lane, Eds., Springer, 3–30, doi:10.1007/978-3-319-23630-8_1.

Signell, R. P., and D. Pothina, 2019: Analysis and Visualization of Coastal Ocean Model Data in the Cloud. *J. Mar. Sci. Eng.*, **7**, 1–12, doi: 10.3390/jmse7040110.

Silver, J. and C. Zender, 2017: The compression-error trade-off for large gridded data sets. *Geosci. Model Dev.*, **10**, 413–423, doi:10.5194/gmd-10-413-2017.

Siuta D., G. West, H. Modzelewski, R. Schigas, and R. Stull, 2016: Viability of Cloud Computing for Real-Time Numerical Prediction. *Wea. Forecasting*, **31**, 1985–1996, doi:10.1175/WAF-D-16-0075.1.

Vance, T. C., M. Wengren, E. Burger, D. Hernandez, T. Kearns, E. Medina-Lopez, N. Merati, K. O'Brien, J. O'Neil, J. T. Potemra, R. P. Signell, and K. Wilcox, 2019: From the Oceans to the Cloud: Opportunities and Challenges for Data, Models, Computation and Workflows. *Front. Mar. Sci.*, **6**, 1–18, doi:10.3389/fmars.2019.00211.

Vannitsem, S., J. B. Bremnes, J. Demaeyer, G. R. Evans, J. Flowerdew, S. Hemri, S. Lerch, N. Roberts, S. Theis, A. Atencia, Z. Ben Bouallègue, J. Bhend, M. Dabernig, L. De Cruz, L., Hieta, O. Mestre, L. Moret, I. O. Plenković, M. Schmeits, M., Taillardat, J. Van den Bergh, B. Van Schaeybroeck, K. Whan, and J. Ylhaisi, 2020: Statistical Postprocessing for Weather Forecasts – Review, Challenges and Avenues in a Big Data World. *Bull. Amer. Meteor. Soc.*, Early Online Release,1–44, doi:10.1175/BAMS-D-19-0308.1.

Wang, N., J. Bao, J. Lee, F. Moeng, and C. Matsumoto, 2015: Wavelet Compression Technique for High-Resolution Global Model Data on an Icosahedral Grid. *J. Atmos. Oceanic Technol.*, **32**, 1650–1667, doi:10.1175/JTECH-D-14-00217.1.

Wang, Y. Q., 2014: MeteoInfo: GIS Software for meteorological data visualization and analysis. *Meteor. Appl.*, **21**, 360–368, doi:10.1002/met.1345.

Xu, M., G. Thompson, D. R. Adriaansen, and S. D. Landolt, 2019: On the value of time-lag-ensemble averaging to improve numerical model predictions of aircraft icing conditions. Wea. Forecasting, 34, 507–519, doi:10.1175/WAF-D-18-0087.1.

Yao, X., G. Li, J. Xia, J. Ben, Q. Cao, L. Zhao, Y. Ma, L. Zhang, and D. Zhu, 2020: Enabling the Big Earth Observation Data via Cloud Computing and DGGS: Opportunities and Challenges. *Remote Sens.*, **12**, 1–15, doi:10.3390/rs12010062.

HRRR CONUS		Forecast Length for Initialization Times		Number of GRIB2 Output Files			
v.	First Date	0, 6, 12, 18 UTC	Other Hours	Surface	Pressure	Native	Subhourly
1	9/30/2014	15	15	102	659	778	26
2	8/23/2016	18	18	135	687	1110	41
3	7/12/2018	36	18	151	701	1126	44
4	12/2/2020	48	18	173	711	1136	196

Table 1: Selected Characteristics of HRRR CONUS Versions from 2014-present available from IaaS cloud providers.

HRRR CONUS		File Type	
v.	First Date	Analyses	Forecasts
2	8/23/2016	Surface	N/A
3	7/12/2018	Surface, Isobaric	Surface
4	12/2/2020	Surface, Isobaric	Surface

Table 2: Availability of Zarr analysis and forecast files (as of July 2021) in AWS S3 for surface and isobaric file types.

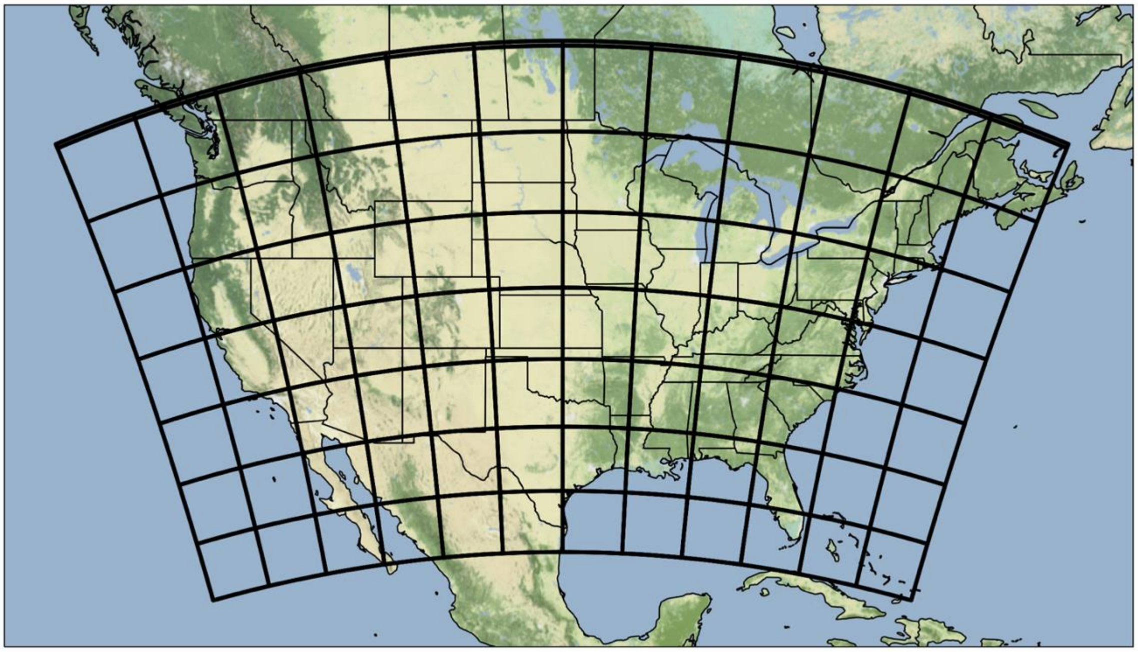

Figure 1. HRRR domain (1799 x 1059 grid points) divided into 96 chunks of size 150 x 150 grid points with the northernmost 12 chunks containing 9 rows of non-NaN data.

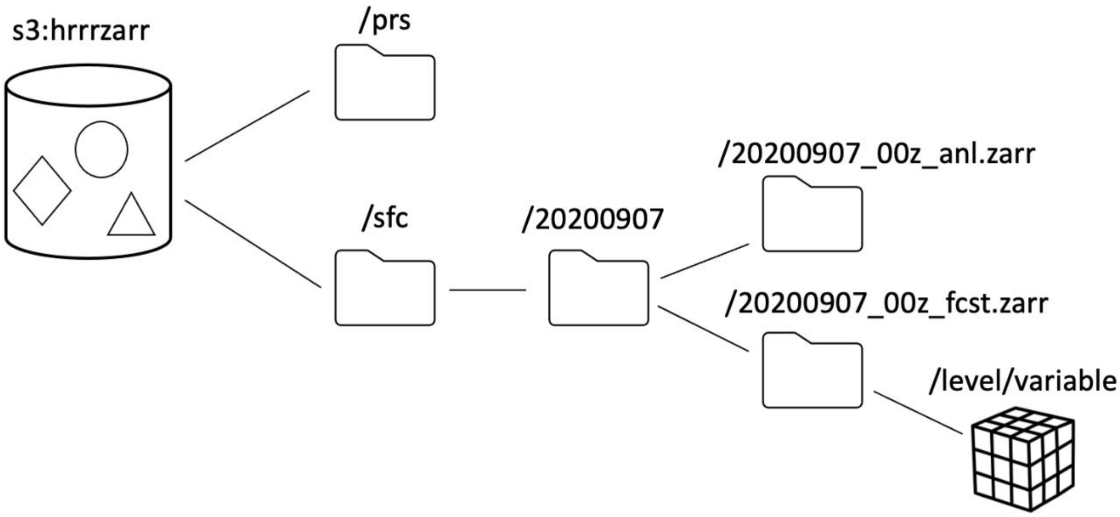

Figure 2. Files within the AWS S3 bucket hrrrzarr are named to emulate a hierarchical data structure.

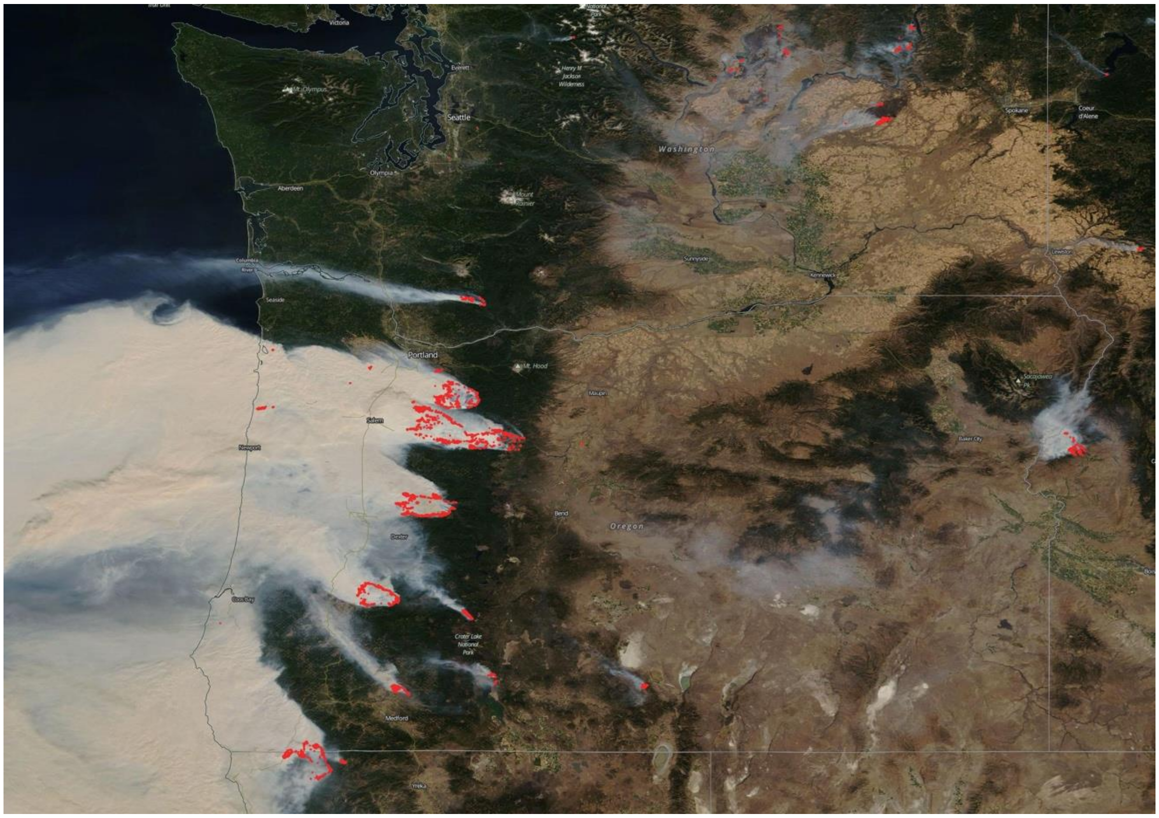

Figure 3. Boundaries of active fires (red outlines), estimated using VIIRS 375 m thermal anomalies, and smoke from wildfires in the Pacific Northwest on 9 September 2020 (source: https://worldview.earthdata.nasa.gov).

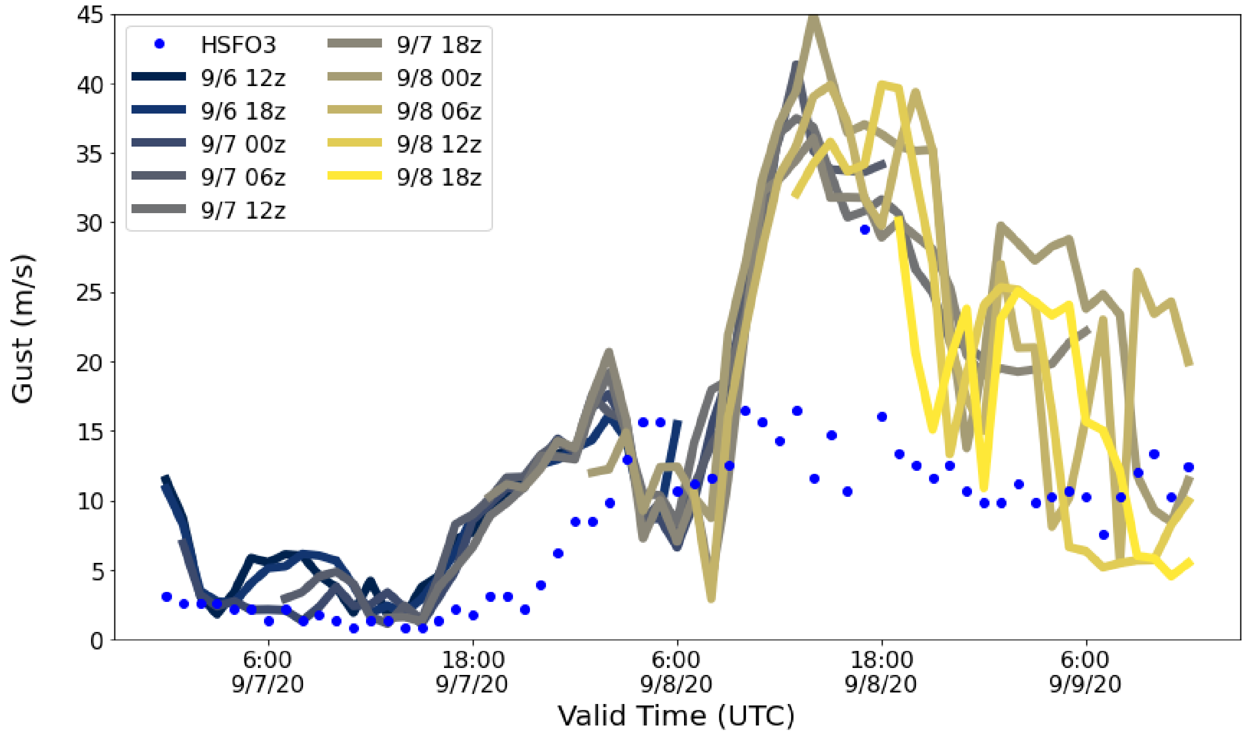

Figure 4. Wind gusts (m s^{-1}) from HSFO3 (blue dots) and HRRR wind gust forecasts near HSFO3 colored corresponding to initialization time.

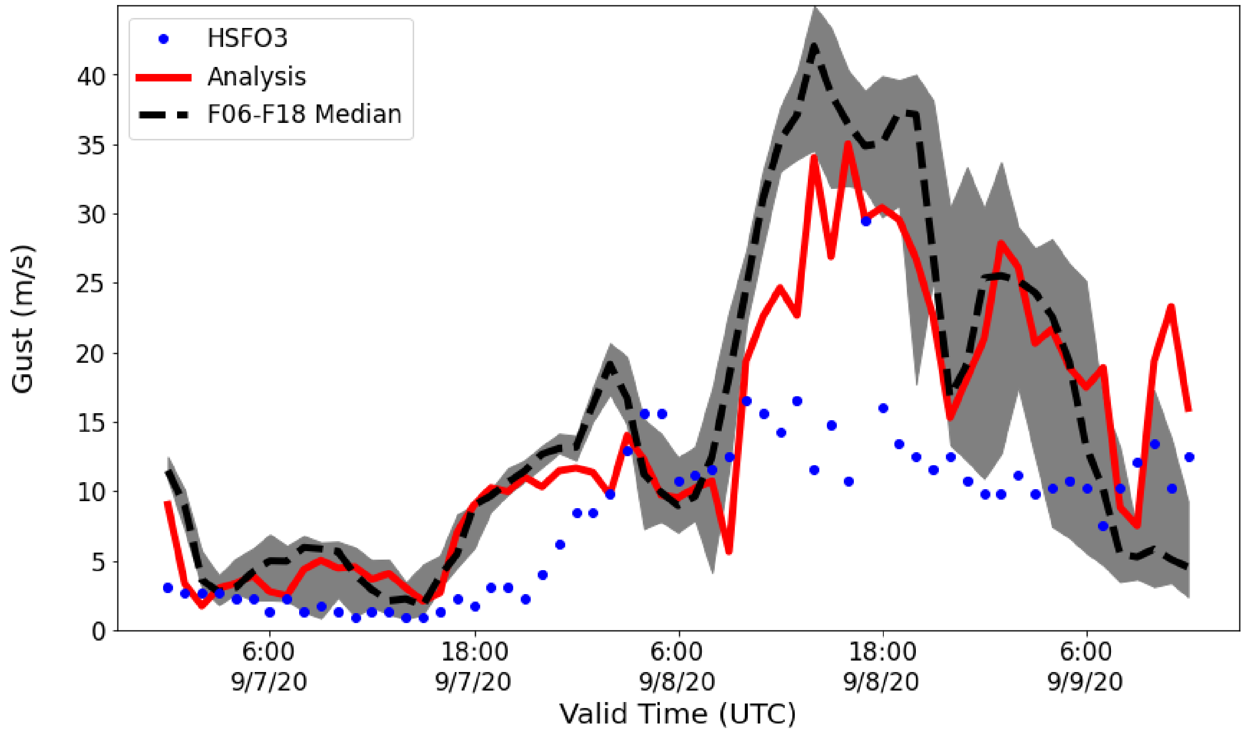

Figure 5. Wind gusts (m s^{-1}) from HSFO3 (blue dots), HRRR analyses (red line) and median of F06-F18 forecasts (dashed black line) near HFSO3 for valid times from 00 UTC 7 Sept – 12 UTC 9 Sept. The shading indicates the range between the maximum and minimum wind gusts from the F06-F18 Time-Lagged Ensemble.

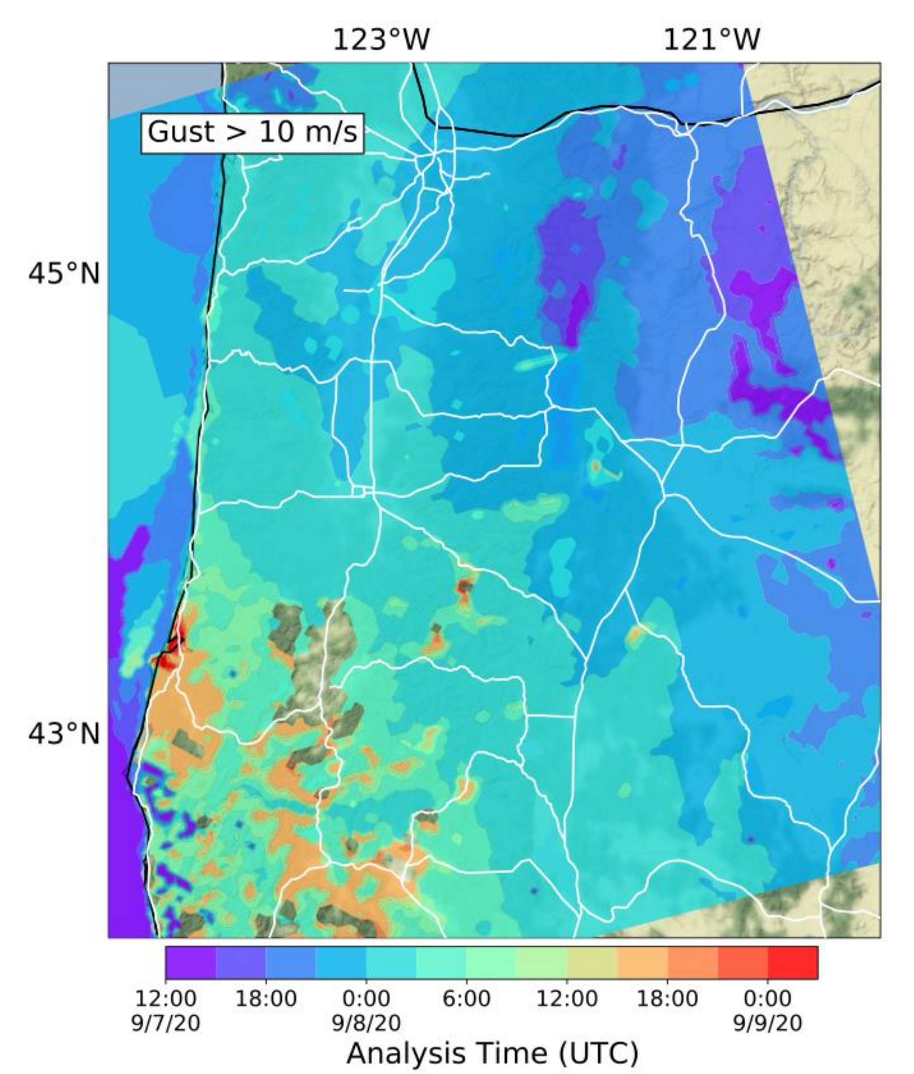

Figure 6. Time of first HRRR analysis (F00) with a wind gust exceeding 10 m s^{-1} (shaded according to the scale) at each grid point for model runs initialized between 12 UTC 7 September – 03 UTC 9 September.

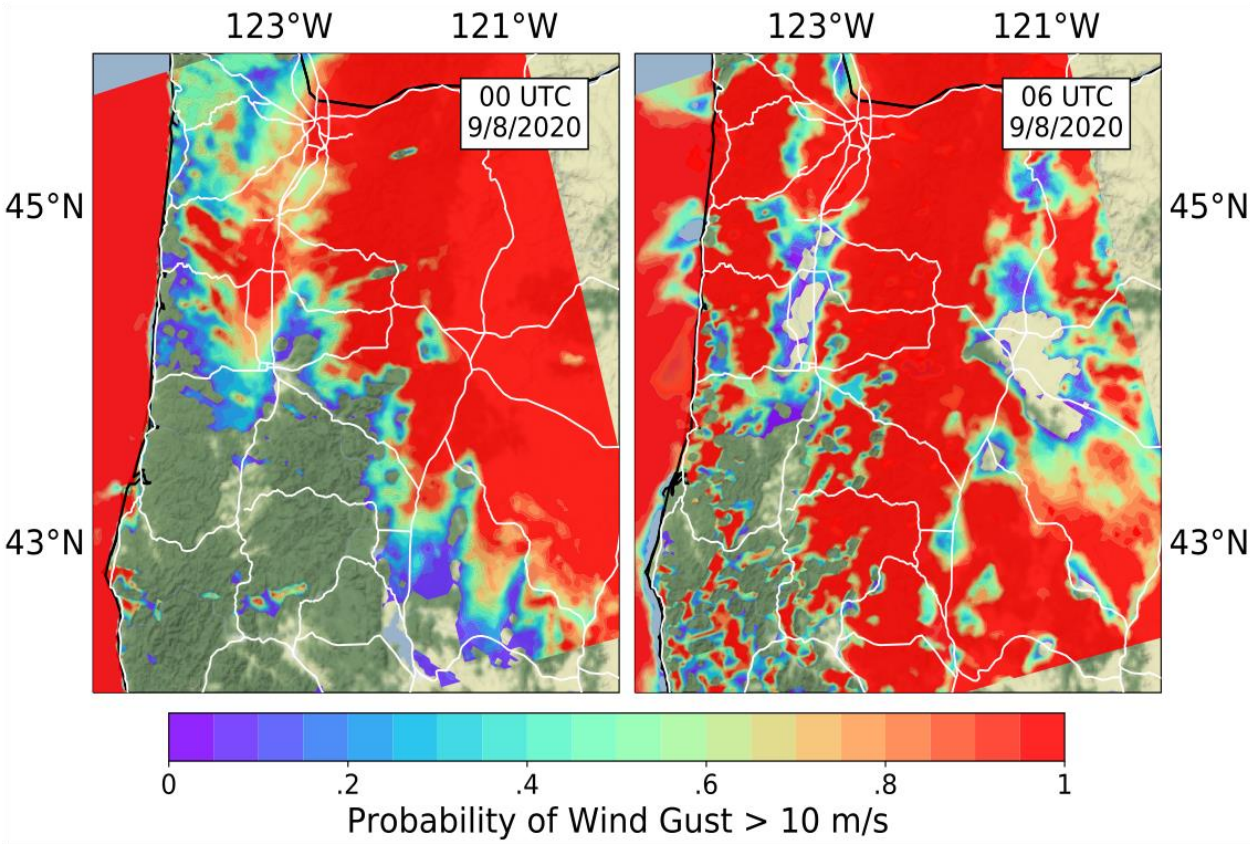

Figure 7. Fraction of HRRR wind gust forecasts exceeding 10 m s^{-1} at valid times 00 UTC (left) and 06 UTC (right) on 8 September 2020. Contours correspond to probability values (0-1) and are shaded according to the scale.

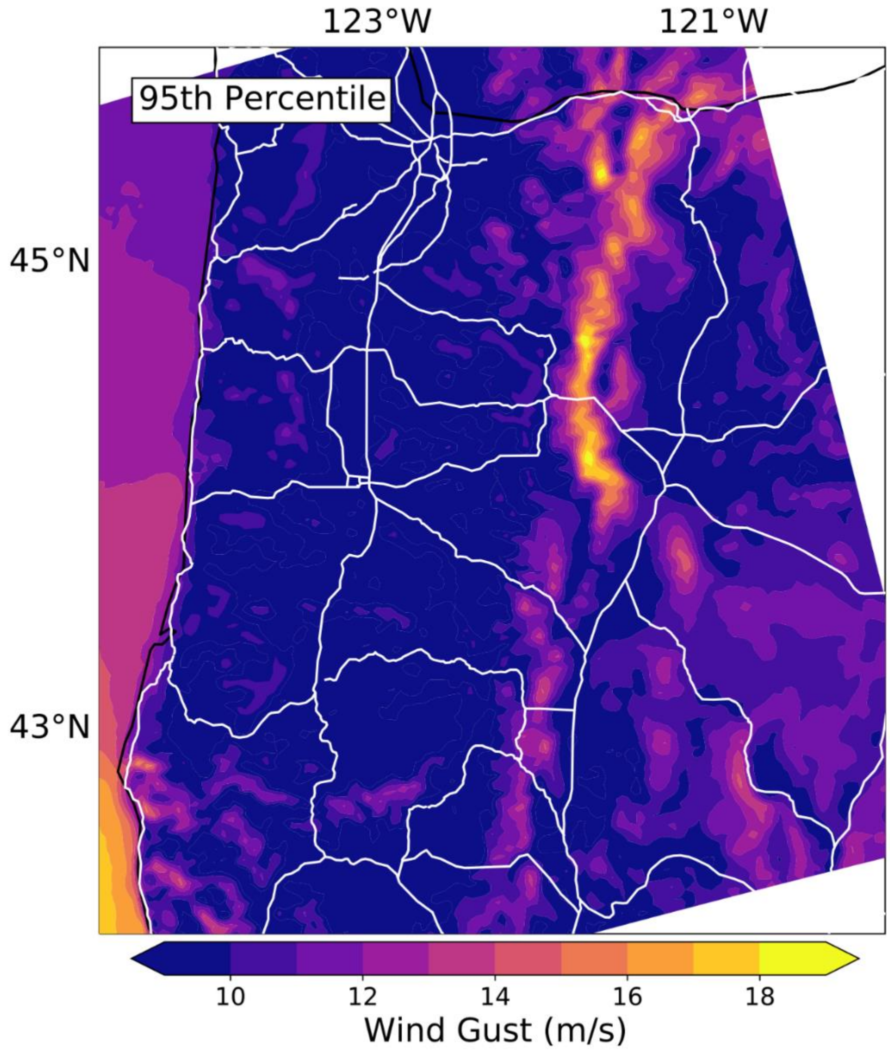

Figure 8. 95th percentile wind gust values (m s^{-1}; shaded according to the scale) calculated at each grid point from empirical cumulative distributions derived from HRRR analyses during September 2016-2019.

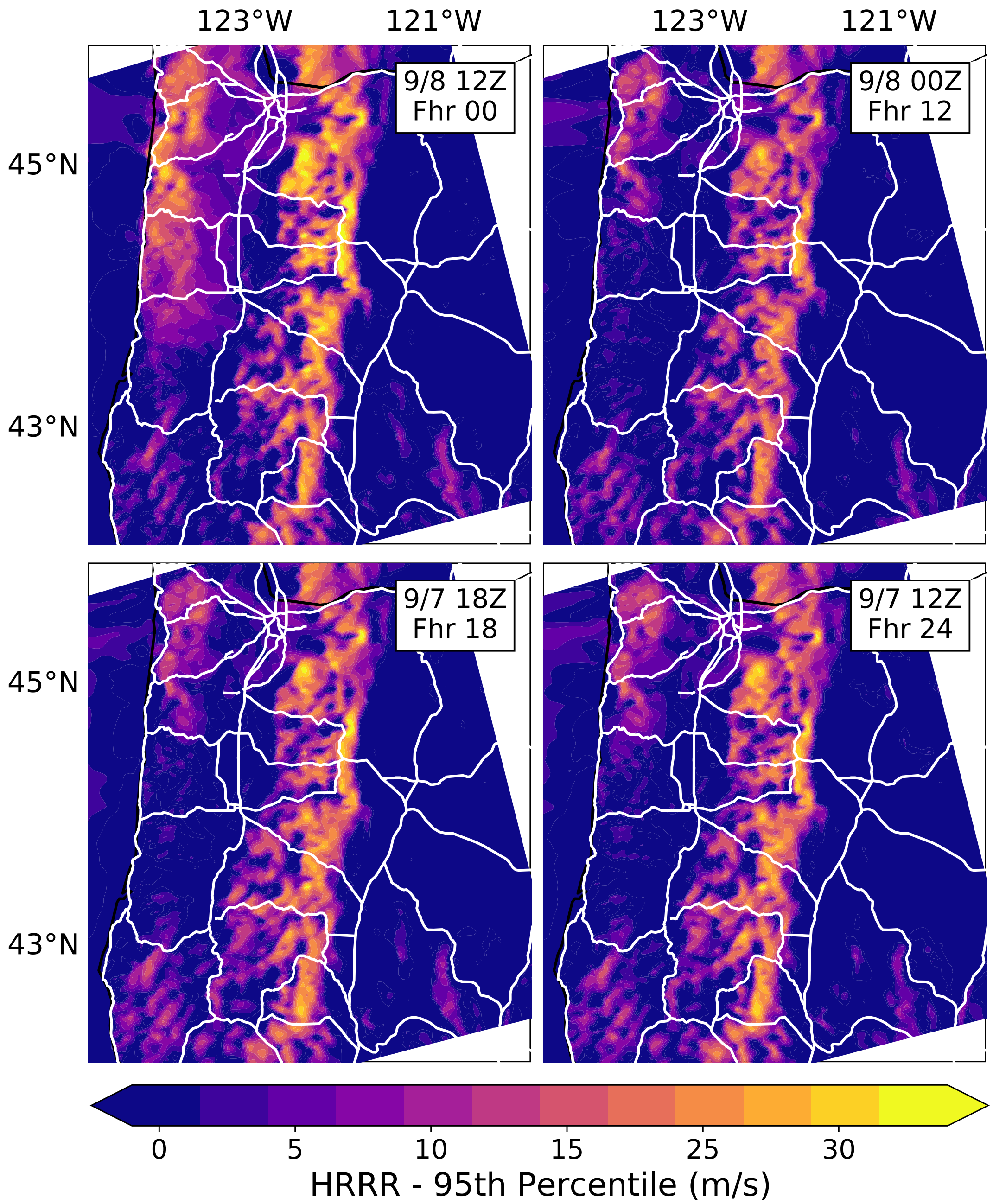

Figure 9. Wind speed (m s^{-1}; shaded according to scale) in excess of the 95th percentile wind gust values at 12 UTC 8 September 2020. Subplots correspond to the verifying analysis (upper left) and F12, F18, and F24 forecasts valid at that time.