Turner, D.D., and coauthors, 2020: A verification approach used in developing the Rapid Refresh and other numerical weather prediction models. *J. Operational Meteor.*, **8** (3), 39-53, doi: https://doi.org/10.15191/nwajom.2020.0803.

Journal of Operational Meteorology
Article

A Verification Approach Used in Developing the Rapid Refresh and Other Numerical Weather Prediction Models

D. D. TURNER
Global Systems Division/Earth System Research Laboratory/NOAA, Boulder, Colorado

J. HAMILTON, W. MONINGER, and M. SMITH
Cooperative Institute for Research in Environmental Sciences/GSD/ESRL/NOAA, Boulder, Colorado

B. STRONG, R. PIERCE, and V. HAGERTY
Cooperative Institute for Research in the Atmosphere/GSD/ESRL/NOAA, Boulder, Colorado

K. HOLUB and S. G. BENJAMIN
Global Systems Division/Earth System Research Laboratory/NOAA, Boulder, Colorado

(Manuscript received 11 June 2019; review completed 7 October 2019)

ABSTRACT

Developing and improving numerical weather prediction models such as the Rapid Refresh (RAP) and High-Resolution Rapid Refresh (HRRR) requires a well-designed, easy-to-use evaluation capability using observations. Owing to the very complex nonlinear interactions between the data assimilation system and the representation of various physics components in the model, changes to one aspect of the modeling system to address a particular shortcoming within the model may have detrimental impacts in another area. Thus, the model verification approach used in the Global Systems Division of the NOAA Earth System Research Laboratory—which actively develops the RAP and HRRR models and other forecasting systems—is designed to allow hypothesis-driven testing of different aspects of the model using observations. In this approach, model changes easily and quickly can be quantified by automatically comparing simulated geophysical variables against many different types of observations that are collected operationally by various agencies, including the National Weather Service. We have implemented this approach in the Model Analysis Tool Suite (MATS). A key aspect of MATS is the use of a database-driven system that stores partial sums of model minus observation pairs over specified geographical regions in order to reduce the dimensionality of the data and, thus, improve the response time of the system. These partial sums are created and stored in a manner that allows the data to be visualized in different ways, thereby providing new insights into the ability of that particular version of the model to replicate the observed atmospheric conditions.

1. Introduction

Numerical weather prediction (NWP) model systems[1] must be verified against observations to demonstrate that the model is able to properly simulate the evolution of the sensible weather component of the atmosphere. This evaluation typically is done both objectively, using statistics to quantify the model's performance, and subjectively (i.e., eyeball comparisons) to ascertain whether the model is getting the atmosphere's evolution "correct." Both the objective and subjective approaches are very valuable to characterize the conditions where the model performs well and to highlight areas where the model needs improvement.

[1] A "model system" is all aspects of the model that are needed to generate a forecast, and in particular includes both the model with its representation of physical and dynamical processes and the data assimilation system used to initialize the model. In this paper, we use "model" and "model system" interchangeably.

Corresponding author address: Dave Turner, Global Systems Division, Earth System Research Laboratory, 325 Broadway, Boulder, CO 80305.
E-mail: dave.turner@noaa.gov

The Rapid Refresh (RAP) and High-Resolution Rapid Refresh (HRRR) model-assimilation systems (Benjamin et al. 2016) are both run operationally in the National Centers for Environmental Prediction (NCEP). These models were developed by the Global Systems Division (GSD) in the NOAA Earth System Research Laboratory. GSD scientists continue to improve these models, with new versions released to run operationally at NCEP approximately every 2 yr. An overview of the RAP and HRRR models, including a summary of their evolution, is given by Benjamin et al. (2016).

The process of preparing an updated version of an operational model—demonstrating that it is indeed superior to or at least matches the current version in all aspects—and releasing it to National Weather Service (NWS) operations is a multi-year challenging task. This paper focuses on the verification approach and tools used within GSD to guide developments that improve the model. We will include a brief history of the tools used over the last decade for RAP and HRRR development and show multiple examples demonstrating how these tools are used by the model developers to gain insight into the model's behavior.

2. Philosophy

Forecasters in the NWS provide excellent subjective feedback on the performance of the RAP and HRRR models. These forecasters evaluate the model's forecasts over their particular areas of responsibility and during particular weather events, and pass notable forecast successes and failures back to GSD in both formal and informal ways. For example, in 2017–18, several NWS forecast offices informed GSD scientists that the operational HRRR seemed to mix out shallow near-surface cold pools too quickly, leading to forecasts of near-surface air temperatures that were much too warm in the daytime. This subjective feedback from the operational forecaster community, as well as from other communities (e.g., aviation and renewable energy), is extremely useful in guiding some of the development work at GSD, as the model developers use this as a starting point in identifying what physical processes are not being represented properly and how these shortcomings could be addressed.

Although we desire perfect forecasts at all times and for all conditions, this is a very challenging goal, especially when the model domain is large and the range of weather conditions broad. The RAP's domain covers the entire North American continent while the HRRR's domain covers all of the continental United States (CONUS, see Fig. 1 in Benjamin et al. 2016), and both models must be able to simulate weather that ranges from quiescent winter periods to vigorous summertime convective storms. Centers that develop operational weather prediction models effectively live the equivalent of the medical community's "first do no harm" principle; modifications to forecast models to address problems in one area should not (markedly) degrade the forecasts of other phenomena or in other regions. Thus, while model developers hope to see an improvement in the representation of the phenomenon they are targeting with the change to the model, they must also look at the impact of model changes both in all regions and on all geophysical variables.

Ultimately, the model developers are performing hypothesis-driven research. They are trying to address a shortcoming within the model and modify the model system to improve the accuracy of the model's forecast. Thus, the testable hypothesis is that the modified model demonstrates improved forecast accuracy relative to the baseline model. Ideally, model developers only attack one issue at a time (i.e., well-controlled experiments) and evaluate each change as they are made, because understanding the impact of multiple simultaneous changes is extremely challenging owing to the complex nonlinear interactions that are present in all NWP models.

The model verification system developed at GSD was designed to support the investigation of these hypotheses by providing tools that allow the model developer to quickly compare the modified model with many different observations, and to gauge how their

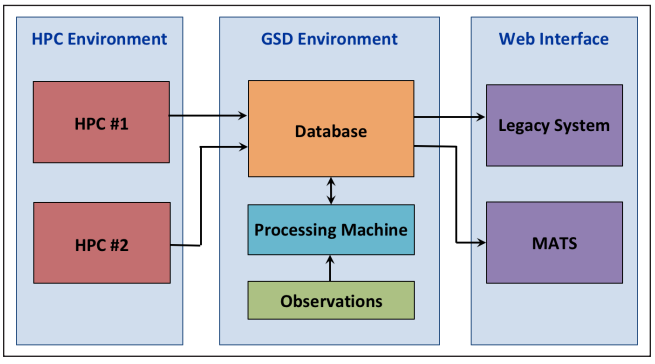

Figure 1. A schematic overview of the GSD model verification system. The HPC and GSD environments constitute the back end of the system, whereas the front end is composed of the database and the web interface. *Click image for an external version; this applies to all figures and hereafter.*

modified model is behaving relative to the baseline model (e.g., the operational model running at NCEP). The comparison against multiple different observed geophysical variables is key; does the modified model give consistently better results? This has led to several requirements for the GSD verification tools:

- ability to easily compare verification results from an experimental model with a reference (e.g., operational) model and have very fast response (i.e., visualize the desired comparison in seconds);
- ability to view the same data in different ways (e.g., time series, diurnally averaged, as a profile, etc.);
- ability to verify forecasts against different kinds of observations, allowing assessment of consistency;
- ability to add new experimental models within a day, including both retrospective runs and new real-time runs;
- ability to add new types of verification (e.g., new observations or new statistical views of the model–observation comparisons) within a month;
- ability to look at verification results within a few hours of real-time;
- ability to check for consistency between different geophysical variables within a model; and
- support both grid-to-observation (i.e., station) and grid-to-grid verification.

In addition to these scientific requirements, there are some infrastructure-based requirements that also need to be considered. These include:

- recognition that the verification system will be used daily by potentially dozens of model developers and managers, so the software must be computationally efficient;
- recognition that the verification results will be used in key management decisions in the research-to-operations process, and thus the results must be repeatable—especially as new versions of the software are released; and
- the figures will be used by scientists in refereed journals and, thus, the figures must be of good quality.

3. Technical details

GSD has chosen to address these requirements using a suite of software that is based upon a MySQL database[2]. The verification system routinely acquires the NWP model output, extracts the desired geophysical variables at the location of the observations, pairs the model output with the observations, downsizes the dataset by performing some partial sums of the matched pairs, and then stores these partial sums in the database for future recall, analysis, and display. This database system is continually growing, and already has >1 TB of data with >10 yr of aggregated statistics. Appropriate partitioning of the database is needed to maintain good responsiveness. [Note that a single 36-h forecast for the HRRR is >500 GB given a domain of 1000×1500×50 (x, y, and z) points, >20 vertical profiles, dozens of 2-dimensional variables, and output every 1 h; thus, the verification greatly reduces the amount of data that need to be stored in the database.]

The key to the entire system is the metadata. The users (i.e., the model developers) interact with web-based tools that will ultimately submit a query to the database and then display the results. The parameters for the query depend on what variables are being investigated (e.g., whether the user desire to compare precipitation or temperature), the choice of the statistic desired [e.g., bias, root-mean-square (RMS) error, etc.], the type of plot desired (e.g., time series, diurnally averaged, etc.), the time range, and more. Some of these choices are handled by the actual executable (or in our language, the "app") chosen by the user. Note that all of the apps are accessed via a single web page. However, most of the user choices that are provided by the app are guided by metadata in the database itself, requiring that the web application communicate with the database and, thus, only display appropriate options to the user.

The GSD verification system can be simplified into two components: the back end that populates the database, and the front end that serves the data to the users. Both parts both generate and use metadata in the database, and together provide the full functionality of the GSD system. Figure 1 shows a schematic view of the GSD verification system [with both the original "legacy" and new Model Analysis Tool Suite (MATS) interfaces; these are described below], with the high-performance computing (HPC) and GSD environment

[2] MySQL is an open source relational (i.e., table-based) database system.

composing that back end and the web interface and the database composing the front end.

The GSD verification system has many apps that have been developed to provide specific insights into the model's forecasting ability. These apps include anomaly correlation, upper-air using radiosondes and aircraft, surface meteorology (temperature, dewpoint, wind, and pressure), cloud-related observations (ceiling, visibility, and solar radiation), precipitation, and reflectivity-based observations (Table 1). Additional details on these apps will be given below. An important point is that each of these apps has its own database instance (i.e., populates a unique portion of the MySQL database and is governed by its own set of metadata and processing scripts). However, each of these apps and database instances—from a high-level point of view—is essentially the same and thus we will use the surface meteorology app to illustrate how the back and front ends of the verification system work.

GSD's model verification system is used daily for monitoring the accuracy of the current operational models, evaluating improvements in model physics (e.g., Benjamin et al. 2016; Olson et al. 2019), and evaluating the impact of different datasets in the data assimilation system (e.g., James and Benjamin 2017).

a. The back end of the GSD verification system

The ultimate objective of the back end of an app is to populate the partial sums component of that app's database. There are several steps used in this process, which are illustrated in Fig. 2.

The first is to collect the model output grids. This frequently requires moving model output (e.g., grib2-formatted data files) around, as different HPC systems are used for the operational and experimental RAP and HRRR runs. A suite of scripts extracts the desired variables at the desired locations from the model's gridded output. For the surface meteorology app, the locations of the observation sites are known, and these scripts extract the near-surface values (2-m and 10-m values, as appropriate) for temperature, humidity, and wind at these locations from the model. This is done for both the model's initialization time and all forecast hours. Because the forecasts (especially for real-time runs) are in the future and the observations at that time are not yet collected, the extracted data are placed in the staging area. Depending on the size of the dataset being staged, the data may be stored either on a local disk or in the database (Fig. 2).

Eventually, the observations are made and are delivered to the GSD computing system (Fig. 1). Different scripts match the observations with model output that is valid at the observation time, and these model–observation pairs are stored in the "pairs database" (Fig. 2). After the model data are paired with an observation, they are removed from the staging area to keep the data volume in the staging area manageable. [However, for a very limited set of some models, one or two months of model–observation data are saved in the pairs database to facilitate detailed comparisons at individual stations.]

The individual model–observation pairs in the pairs database are too voluminous to maintain for the long-term and require extensive computational power if the user desires to visualize long time series. To increase the efficiency of our verification system, we compute "partial sums" from the individual model–observation pairs over specified regions of the model domain (e.g., western CONUS) for the given valid time, and only the partial sums are saved to the long-term database. These partial sums include statistics such as $\sum(m_i - o_i)$, $\sum(m_i - o_i)^2$, $\sum m_i$, $\sum o_i$, N, etc. for continuous variables (where m_i and o_i are the model value and observation at point i, and N is the number of points in the summation) and the number of points for each of the four areas of a contingency table (i.e., correct positive, false positive, false negative, correct negative) for discrete variables (see Table 1). From these statistics we can easily derive the bias, RMS error, and other statistics over any longer time window or over unions of different spatial regions. Thus, creating partial sums greatly reduces the disk space needed to store the verification data and subsequently increases the speed and responsiveness of the system when the user requests a particular plot type.

In our surface meteorology example, the observations are collected by Meteorological Aerodrome Report (METAR) and mesonet stations at >2300 locations. Thus, for the operational RAP there are 2300 model–observation pairs in the pairs database for each model forecast hour (the RAP is currently producing out to 39-h forecasts) for a given validation time. It is impractical to save all of these model–observation pairs at all locations and valid times for multiple model versions over multiple years in the database; doing so would result in the database growing to many terabytes very rapidly. Instead, partial sums are computed over defined geographical regions (e.g., western CONUS, northeastern CONUS, etc.), thus reducing the 2300 model–observation pairs for each model validation time

Table 1. Current MATS verification applications for different observation types and plot types available for each.

Observation type for different types of verification	Contingency Table (y/n)	Data Source	Plot Type Available								
			Time series	Diurnal	Die-off	Profile	Map	Threshold	Histogram	Contour	Contour Diff
			Upper-air								
Upper-air	No	NWS Radiosondes	Yes	No	Yes	Yes	No	No	Yes	Yes	Yes
Aircraft	No	AMDAR	Yes	No	No	Yes	No	No	Yes	Yes	Yes
Anomaly correlation	No	GFS Analysis	Yes	No	Yes	Yes	No	No	Yes	Yes	Yes
			Surface								
Surface met	No	METAR	Yes	Yes	Yes	No	Yes	No	Yes	Yes	Yes
Surface met by land use	No	METAR	Yes	Yes	Yes	No	No	No	Yes	Yes	Yes
			Cloud-related								
Ceiling*	Yes	METAR	Yes	Yes	Yes	No	No	No	Yes	Yes	Yes
Visibility*	Yes	METAR	Yes	Yes	Yes	No	No	No	Yes	Yes	Yes
Solar radiation	No	SURFRAD	Yes	Yes	No	No	No	No	Yes	Yes	Yes
			Precipitation								
Precip**	Yes	Stage IV	Yes	No	No	No	No	Yes	Yes	Yes	Yes
			Reflectivity relates								
Composite reflectivity	Yes	MRMS	Yes	Yes	Yes	No	No	No	Yes	Yes	Yes
Echo tops	Yes	MRMS	Yes	Yes	Yes	No	No	No	Yes	Yes	Yes
Vertically integrated liquid	Yes	MRMS	Yes	Yes	Yes	No	No	No	Yes	Yes	Yes

* Also available for sub-hourly (ceiling and visibility).
** Available for 24-h total, 6-h forecast for 6-h period, and 1-h forecasts.

AMDAR: Aircraft Meteorological Data Relay (Moninger et al. 2003)
METAR: Meteorological Aerodrome Report
SURFRAD: Surface Radiation Network (Augustine et al. 2000)
Stage IV (Nelson et al. 2016)
MRMS: Multi-Radar Multi-Sensor (Smith et al. 2016)

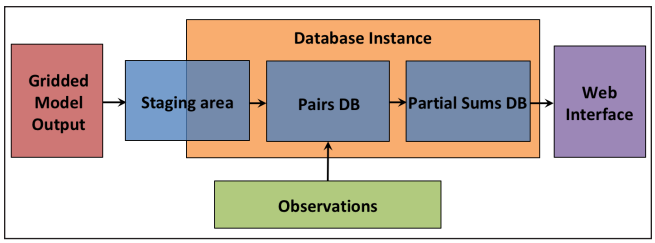

Figure 2. The dataflow through the GSD verification system for an example app (DB stands for database). See the text for details.

into a set of five numbers for each region: the mean observation value, the mean model value, the mean difference, the mean sum of squares difference, and the number of points in the summation. This reduced set of data is placed in the partial sums database (Fig. 2) for subsequent analysis and visualization.

The software that runs on the back end of the verification system is a collection of scripts written in C, Fortran, Perl, Java, and Python. These scripts are run on both the HPC machines and on the processing machine in the GSD environment (Fig. 1). The scripts are run automatically using crontab entries and the locally built workflow software management package; the latter allows dependencies to be specified and provides markedly better management of the workflow.

b. The front end of the GSD verification system

The users (e.g., model developers) interact with the verification system with web-based applications for different observation types. These applications, grouped into related sections as shown in Table 1, are all accessed via a single web page. After the user selects an application, the app then solicits information from

the user on the type of plot to create. The selectable parameters are part of the metadata associated with that application and are stored in the database.

There are many parameters that are common among the different applications, with the most common being the model to evaluate, the time range that the user desires to be analyzed and displayed, and the type of statistic [e.g., bias, RMS error difference, critical success index (CSI), true skill score (TSS), etc.]. There are application-specific selectors too, which are guided by the metadata associated with that application (e.g., only contingency table applications will have the TSS). Because the verification system is designed for hypothesis testing, multiple curves can be added to the display showing results from different models, different time periods, and/or different spatial regions.

Originally, the front end was a suite of different Java applet applications; GSD refers to these applets as the "legacy system." The earliest use of the legacy system in a peer-reviewed paper was Moninger et al. (2010). There was a unique applet for each observation type (e.g., upper-air radiosonde, radar reflectivity observations, etc.) and plot type (e.g., time series, profile, die-off curve[3], etc.), and each applet was developed largely independently of the others. Each applet communicated with its own database instance to determine what user-selectable parameters were possible. The user would specify the first "curve" (i.e., plot on the soon-to-be-created graph) by identifying the model to view, the spatial region desired, the statistic to display, the forecast length to evaluate (unless it was a die-off plot, which does not have this option), the time period over which to average the statistics (e.g., 1 day, 7 days, or 30 days), and (if it is a contingency table statistic) the discrimination threshold desired. The user could then add additional curves using the same process to select different models, spatial regions, thresholds, etc. All of these selections were done using a point-and-click process in the web interface created by the Java applet. After the user finished selecting the desired parameters, the user made one final decision: to plot the data at "matched" or "unmatched." Selecting the former would require that the data displayed in the different curves come from exactly the same time instances, while a temporal hole in one dataset (e.g., due to missing model output at a particular time) would result in data from the other curves being removed from the comparison for that same time period.

When the user presses the "plot matched" or "plot unmatched" button (for either the legacy system or MATS), the verification system uses the options selected to construct a set of queries that are submitted to the database. These queries extract the data from the appropriate partial sums table in the database, perform any desired processing (e.g., temporal averaging) and compute the desired statistic, and return the resulting data to be displayed. The applet then plots these data in the various curves desired by the user in a new popup window on the screen. The user-selected parameters are still visible in the original web browser window, so the user is able to easily modify the parameters of the various curves, add new curves or remove curves, and then replot it. This functionality allows the user to easily interrogate the model–observation statistics in different ways and thus gain insight into how well the model is performing relative to the observations. Generally, for time-averaged verification, matching of the same events ("plot matched") is critical for a comparison of two models. For time-series verification without averaging, use of the "plot unmatched" option can be used effectively.

However, the Java applet-based front end in the legacy system grew organically. When a new plot type was desired, the code from a previous applet was copied and modified. This was done to quickly facilitate the development of new verification types, which was one of the philosophical goals of the GSD verification system. However, as security concerns associated with Java applets grew, a decision was made to port the front end to a new, more modern code base.

The new system is called the Model Analysis Tool Suite, or MATS. There were several requirements for MATS. It had to have the same look and feel as the legacy Java applet applications, which were heavily used by the GSD model developers. The software framework selected was the open source JavaScript-based Meteor web framework. Meteor enables rapid prototyping of the apps, and works across platforms on multiple web browsers, different computer architectures including tablets, and on smart phones. MATS development follows best practices for software development, uses common code across the different MATS apps, eliminates all hard-coding of parameters associated with the particular app, and introduces a true development cycle that had separated development, integration, and testing sites. It uses a formalized automated testing system, which ensures that upgrades to MATS do not

[3] A "die-off" plot shows the desired statistic (e.g., RMS error) plotted as a function of the forecast length.

break previous functionality. MATS also takes better advantage of the metadata in the database and modifies the various curve selectors to make them appropriate for the specific model dataset queried. For example, many retrospective runs are limited in time (i.e., are only run for a period of weeks) and the MATS system adjusts automatically to indicate the date range possible for a selected model dataset.

Another feature available in the GSD verification system is the ability to easily include a plot of the difference between any two curves on the graph. This is done with a simple button selection, thereby making it very easy to see where improvements have been made. Because of the large amount of common code shared among the MATS apps, adding new features or improving its performance is very straightforward and efficient. As the legacy Java apps were basically clones of one another, any desired software changes had to be implemented individually in each app. With MATS, such changes only need to be made once, in the common code library. For example, the MATS graphing package was recently moved from flot.js to plotly.js in the common code, with no changes necessary in any app-specific routines.

c. Adding a new experimental model to the GSD verification system

So how does this process really work? Suppose a GSD scientist would like to perform a retrospective experiment to evaluate the impact of a modified data assimilation technique on the HRRR initialization and its ability to forecast convection. The scientist would want the output from this retrospective run to be included in the verification system, so that she/he is able to compare how the model performed against both observations and the original model system using the various MATS apps. The primary activity that needs to be done is to get the model output into the verification system itself. The process is relatively straightforward if the observations are available (if not, then the observations first must be restaged from the mass-store data archive). First, metadata about the retrospective run need to be created and put into the database; however, a large fraction of the metadata needed are specified by the model that was run (e.g., HRRR, RAP, etc.). Then, one of the GSD verification team members ensures that the observation data are properly staged, modifies some scripts to create the model–observation pairs, and creates the partial sums. The MATS system is then able to immediately display results from this new retrospective run as a new curve on any of the MATS apps, and added to curves from previous experiments for comparison.

4. Examples

There are two general types of verification: grid-to-observation and grid-to-grid. The first is a point-based verification method using the locations of the observations as the reference. This is very straightforward for the observations that do not move (e.g., the surface stations), but is more complicated for observations collected on moving platforms such as aircraft. Nonetheless, the procedure is still the same: match the model value to the observation time and location when creating the model–observation pair. Grid-to-grid verification uses a gridded analysis as a reference. Examples of observation datasets used in this type of verification are the mosaics of radar reflectivity from the Multi-Radar Multi-Sensor (MRMS) product (Smith et al. 2016), the 2-D distribution of precipitation from the stage IV product (Nelson et al. 2016), and the analysis field of the Global Forecast System (generally used for anomaly correlation). Most of the time, the model grid points do not align perfectly with the observation grid, so either the observations are interpolated to the model grid, or the model output points—within some specified distance of the observation (e.g., 3 km, 13 km, or 20 km)—are paired with the observations.

The verification statistics computed from these model–observation pairs depend on the nature of the geophysical variable. Some variables such as temperature, wind, and humidity are continuous in time and space, and thus the partial sums are constructed such that simple statistics such as bias and RMS error can be computed. However, other geophysical variables such as precipitation, radar reflectivity, and ceiling height are not continuous, and thus the verification is done using contingency tables. For these variables, several preset thresholds are used [e.g., for precipitation we use the thresholds of 0.0254, 2.54, 6.35, 12.7, 25.4, 50.8, and 76.2 mm (0.01, 0.1, 0.25, 0.5, 1.0, 2.0, and 3.0 in) over some time period], and the partial sums for all of these thresholds are stored in the database. From these partial sums, various statistics such as the TSS, CSI, bias, and false alarm ratio can be computed easily. Note that the precipitation thresholds, as well as the neighborhood sizes for the reflectivity and precipitation apps, have predetermined values because these are used in the

generation of the partial sums.

There are many apps within MATS (Table 1), each of which uses a different observation type (e.g., radiosondes, radar reflectivity, surface visibility, etc.), and each app has many different plot types that are available to visualize the data in different ways. The main MATS web page (www.esrl.noaa.gov/gsd/mats; Fig. 3) provides easy access to all of these apps. Each app uses a different observation dataset as truth; these are provided in Table 1.

After selecting the desired app—which will be used to evaluate the model output against the desired observation type—the MATS application will show the "control panel" that allows the various parameters of the curve to be selected. It is generally advisable to work from the top down when selecting the parameters, as MATS is metadata-driven and thus the selection of one parameter will provide different options for the parameters further down in the control panel. The user will first select the plot type (Fig. 4a). Next, the user will select various options for that plot type (Fig. 4e), and after selecting the various options (e.g., "data source" provides a list of the different models that could be evaluated and "statistic" allows the user to select RMS error, bias, mean model value, etc.), the user would then click "add curve" (Fig. 4f). This adds the new curve to the list of curves shown in Fig. 4b–c. In this example, a time series of the 2-m temperature bias between two models (HRRR-GSD in red and HRRR-OPS in blue; this cryptic naming of these two models is described in the next paragraph) is compared to quality controlled METAR observations over the eastern HRRR domain (i.e., for all stations east of 100ºW) with 6-h averages applied. Note that the user easily can select different colors for each curve by specifying different red/green/blue indices in Fig. 4b–c, if desired. The parameters for any curve can be modified using the "edit curve" buttons in Fig. 4b–c, and curves also can be removed if desired. After all of the curves have been specified, the user selects the date range (Fig. 4g), the amount of data completeness, and if a standard deviation filter should be used (Fig. 4i). If the user also would like to see the difference between the curves, this can be specified using the radio buttons to create differences (Fig. 4h). At this stage, the user can select either "plot matched" or "plot unmatched" to trigger MATS to issue the query to the database and create the plot. Generally, MATS generates the plot within 5 s of the trigger.

To illustrate some of the capabilities of the GSD verification system, we will demonstrate how the

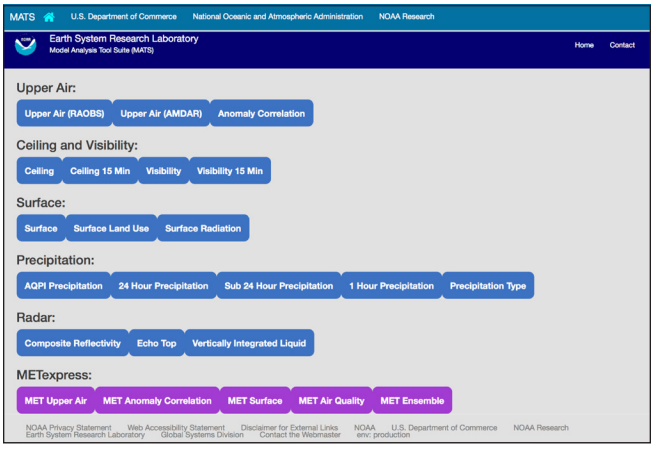

Figure 3. The primary MATS page, which lists all of the currently available (as of October 2019) model–observation verification apps.

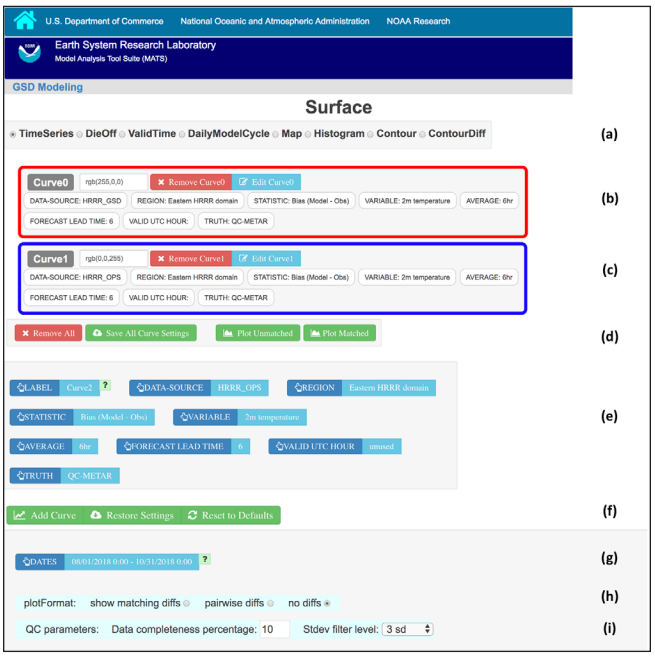

Figure 4. The curve parameter "control panel" for the surface app. Areas marked (a) through (i) are described in the text.

different apps are used to help the model developers evaluate changes made to the experimental HRRR (henceforth called HRRR-GSD) relative to the baseline model currently running at NCEP (HRRR-OPS, which is currently version 3). On 13 September 2018, a change was made to HRRR-GSD whereby the 6th-order diffusion was dramatically decreased, especially for hydrometeors. The developers had noticed some artificial-looking reflectivity echoes in spatial images of reflectivity (not shown) from case studies (which

often serve as an important source of hypotheses). A hypothesis was formed that there was too much diffusion in the model that was producing these artifacts. [Note this is an actual example of model development, and while it is used here, the results shown here are not the final results—but were informative on how to continue to improve the model towards version 4.]. To test this hypothesis, the 6th order diffusion coefficient was markedly reduced in the experimental HRRR (i.e., HRRR-GSD) on 13 September 2018, and this change was evaluated over the next month relative to the same length of time prior to this change to ascertain the impact on the model forecasts.

Figure 5a shows a time series of composite reflectivity frequency bias for a threshold of 15 dBZ over a 13-km domain, for both the HRRR-GSD (red) and HRRR-OPS (blue). The change to the model on 13 September is seen clearly in this time series, as the two curves largely overlapped before that date, but afterwards the red curve has larger values than the blue, implying that the modified model now reports more events with composite reflectivity above that 15-dBZ threshold. The developer may want to look at how this physics change impacts different forecast lengths; this is shown for the TSS in a die-off plot (Fig. 5b). This figure demonstrates that the HRRR-GSD has higher TSS values than HRRR-OPS in both periods, and that the amount of improvement is about the same in both periods, thereby suggesting that the change to the diffusion coefficient did not markedly increase or decrease the TSS for reflectivity above 15 dBZ. To gain more insight into this model physics change and see if there is any time-of-day dependence, the developer may look at composite reflectivity plots (Fig. 5c–f) that show the frequency bias (same threshold) as a function of forecast lead time (x axis) versus valid time (y axis). Both the HRRR-OPS and unmodified HRRR-GSD (Figs. 5e and 5c, respectively) show a frequency bias that moved closer to the desired value of 1.0 as the forecast lead time increased, but both also show that there is a significant time-of-day dependence to the bias, with both models underestimating the reflectivity in the afternoon (i.e., between 1500 and 2400 UTC). However, the modified model (Fig. 5d) shows a much different situation, with a frequency bias of about 1.3 that is uniformly constant across both lead time and valid time. However, it is clear that the meteorology also has evolved between the two periods (1 August–13 September versus 14 September–31 October) as the HRRR-OPS contours for the two periods (Fig. 5e and

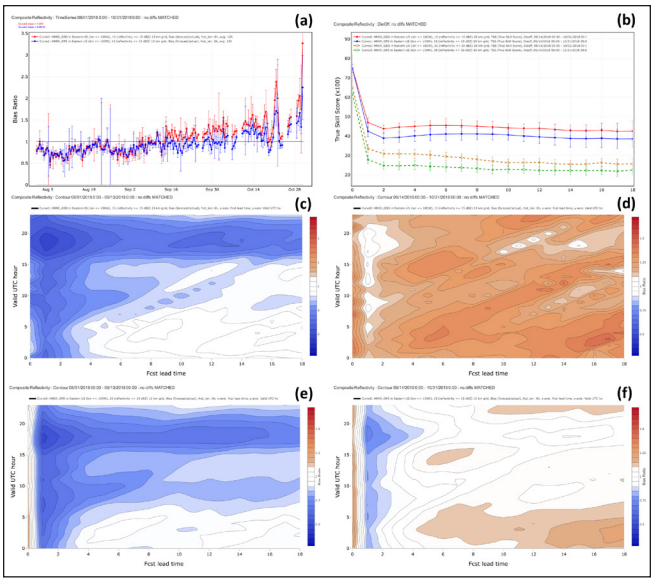

Figure 5. Examples from the reflectivity app, where (a) shows a time series of frequency bias for reflectivity >15 dBZ over a 13-km neighborhood for HRRR-GSD in red and HRRR-OPS in blue, (b) shows a die-off plot for TSS for HRRR-GSD and HRRR-OPS for both the first period of 1 August–13 September (red and blue curves, respectively) and the second period of 14 September–31 October (brown and green lines, respectively), where the forecast length in hours is shown on the x axis, (c) and (d) show contour plots of the reflectivity frequency bias for this same threshold as a function of forecast lead time (x axis) by valid UTC time (y axis) for the HRRR-GSD model for the first and second periods, respectively, and (e) and (f) are the same as panels (c) and (d) but for the HRRR-OPS model.

5f, respectively) also are different from each other. This may spur the model developers to ask additional questions (e.g., "Are the results similar for higher reflectivity thresholds?"), which can be easily and quickly investigated with the MATS system.

The model developer may want to quantify the impact of this change on precipitation. Figure 6 shows results from both the HRRR-GSD and HRRR-OPS using the precipitation app for two periods: before the change (1 August–13 September) and after the change (14 September–31 October). This app uses gridded stage IV precipitation data as the truth, and these statistics are only shown using model–observation matches in the eastern HRRR domain (i.e., east of 100ºW where there is better radar coverage and thus the stage IV data are deemed to have higher quality). The CSI (Fig. 6a), TSS (Fig. 6b), and frequency bias (Fig. 6c) for different

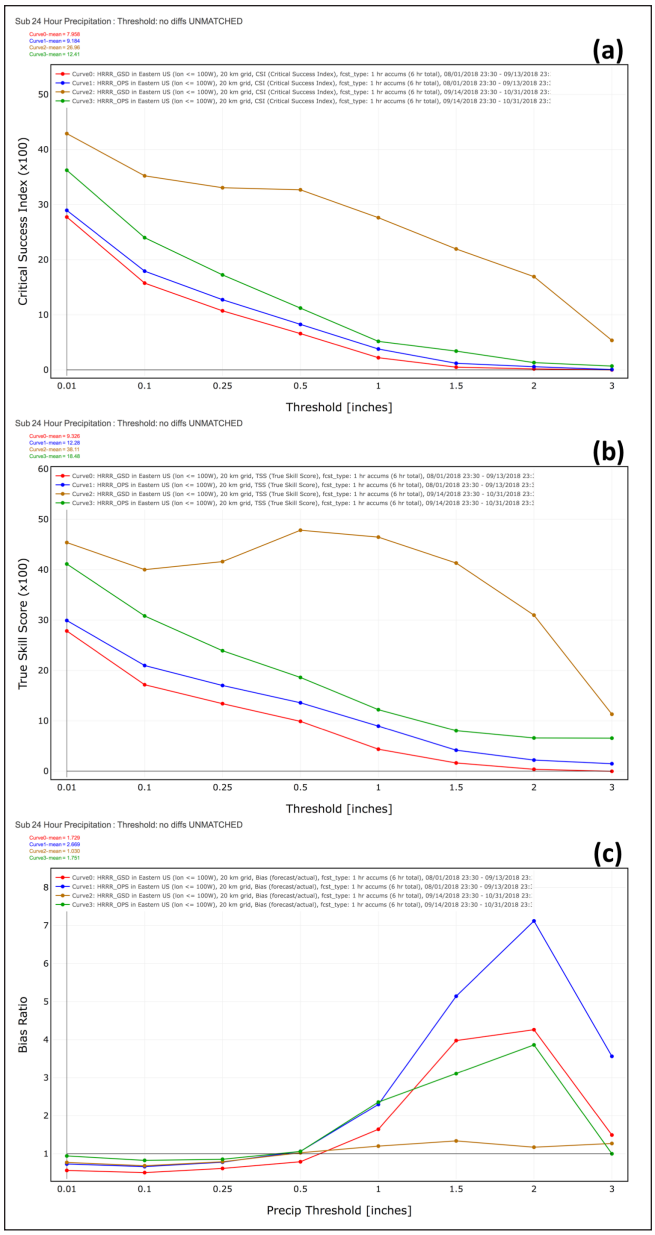

Figure 6. Results for the precipitation app showing (a) critical success index, (b) true skill score, and (c) frequency bias (1 in = 25.4 mm). Results for the HRRR-GSD for 1 August–13 September and 14 September–31 October are given in red and brown, respectively, and the HRRR-OPS for the same two periods are in blue and green, respectively.

precipitation thresholds for a 1-h accumulation are shown. These results clearly demonstrate that the modified model (HRRR-GSD in period 2; brown curves in Fig. 6) has markedly higher CSI and TSS and a much smaller and flatter bias for all precipitation thresholds than the original model.

One of the most frequently used verification apps is that for surface observations. Figure 7a shows time series of both the 2-m temperature bias (red, blue) and RMS error (brown, green) for the HRRR-GSD and HRRR-OPS models during this period. Again, the difference in bias between the two models seems to change pretty substantially after the changes were made to the diffusion on 13 September, and the characteristics of RMS error also changed. However, there is a clear "sawtooth" pattern in the time series that could be a time-of-day dependence. So, the 2-m temperature bias (Fig. 7b) and RMS error (Fig. 7c) were plotted, again showing the HRRR-GSD (red lines) both before and after the physics change (solid and dashed lines, respectively) as well as the HRRR-OPS (blue lines) for the same two time periods (again, solid and dashed lines). The temperature bias in modified model (red dashed line in Fig. 7b) is markedly different than the other models; the bias is smaller, flatter with time, and close to zero at night (0100 to 1200 UTC), but becomes substantially negative and approaches a 1ºC cold bias during the day (1300 to 2400 UTC). The RMS error of the HRRR-OPS (blue dashed line in Fig. 7c) decreased substantially for the second time period relative to the first (blue solid line in Fig. 7c), indicating that the meteorology has changed in the second period; however, the RMS error for the HRRR-GSD was essentially unchanged for the two periods, suggesting that the RMS error in the 2-m temperature actually got worse with this physics change. Similar plots (not shown) quickly could be made showing the 2-m relative humidity (RH) or 10-m wind bias and RMS error, thereby providing the developer more information on how the evolution of the surface layer was changed with the modified physics.

If the near-surface temperature, humidity, and winds are affected by the modified physics, a very natural question is to ask how the profiles of these quantities are being affected. MATS has two apps that allow investigation of this question: upper-air radiosonde observations launched by national weather services around the world and Aircraft Meteorological Data Relay (AMDAR) commercial aircraft data. Figure 8 shows the bias (red and blue profiles) and RMS error (brown and green profiles) for temperature (Fig. 8a), RH (Fig. 8b), and wind speed (Fig. 8c) for both HRRR-GSD (red and brown) and HRRR-OPS (blue and green). Statistics for the 1 August–13 September period are denoted with solid lines and filled circles, while statistics for the 14 September–31 October period

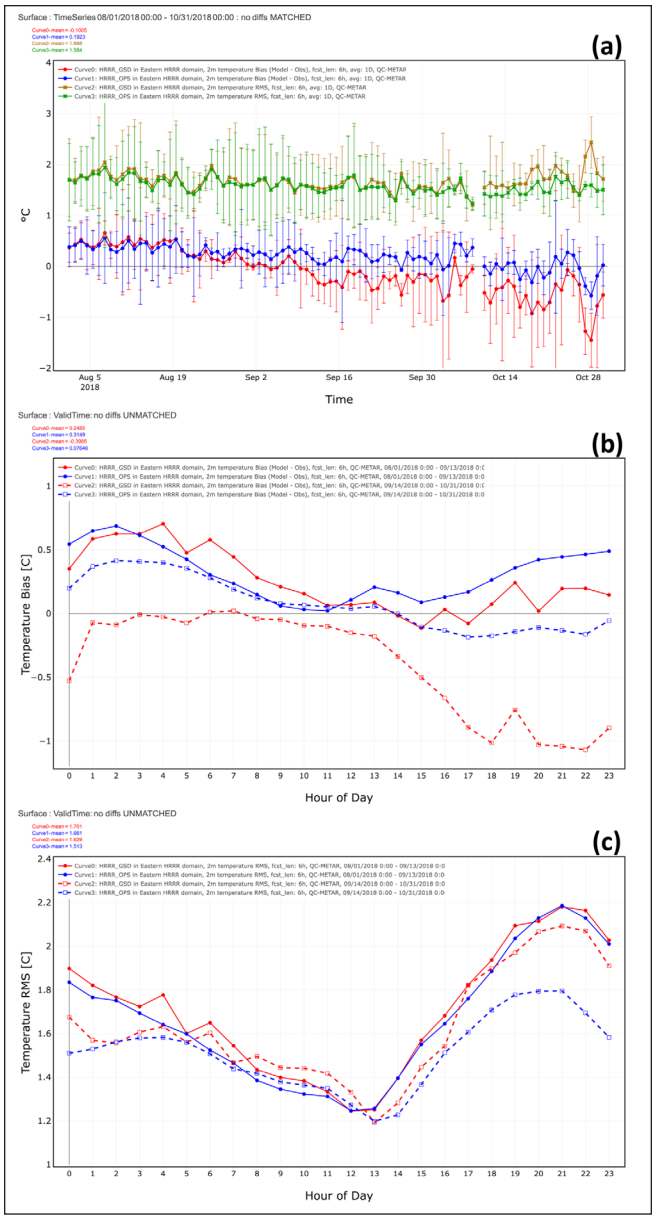

Figure 7. Examples from the surface app showing (a) a time series of RMS error in temperature for the HRRR-GSD and HRRR-OPS models (brown and green, respectively) and temperature bias for the same two models (red and blue, respectively), (b) a valid time plot showing the temperature bias for the HRRR-GSD for 1 August–13 September and 14 September–31 October (solid red and dashed red curves, respectively) and the HRRR-OPS for the same two time periods (solid blue and dashed blue, respectively), and (c) same as panel (b) but showing the temperature RMS error.

are shown with dashed lines and open squares. The wind profiles show no significant difference between the HRRR-GSD and HRRR-OPS for either period (Fig. 8c); this suggests that the modified model did not affect the winds at all. However, the temperature bias for the modified model (dashed red curve in Fig. 8a) is markedly different and much smaller for altitudes below 900 mb than the other three curves (Fig. 8a), which suggests that the changes to the physics affected the entire lower part of the planetary boundary layer. Similarly, the RH bias profile from the modified model also looks better (red dashed curve in Fig. 8b), although it is clear from the change between the two periods that the HRRR-OPS model has some challenges in its ability to capture the water vapor change in the environment moving toward a drier autumn condition, and this might be affecting the experimental model also.

Ultimately, the model developer needs to test the hypothesis that the change in the model diffusion improved the forecast ability of the HRRR-GSD. This is a non-trivial task as our results clearly have shown a marked improvement in the precipitation (Fig. 6) but a degradation of the 2-m temperature bias in the daytime (Fig. 7b). The developer will use information such as from Figs. 5, 6, 7, and 8—as well as figures from the other variables, statistics, and regions for these apps—and plots from the other apps in an attempt to determine if the modification to the model is conforming to the hypothesized behavior that was anticipated before the change was made, and how the change impacted other aspects of the model's behavior. Owing to the daunting complexity inherent in any NWP model, the developer will need to interrogate the data in multiple ways, which led to the requirements for our verification system outlined in section 2.

Significance levels (i.e., error bars) are critical to assess the importance of verification statistics. Generally, error bars can be shown in most MATS displays (e.g., in Figs. 5a–b and 7a). Comparing statistics for different observation types can be very important to look for consistency. Guidelines for using different verification techniques and their combinations will be described in an upcoming paper.

5. Future

The GSD verification system is continually evolving and growing. We have multiple projects underway now that will greatly extend its capability, providing new insights on the ability of the various models to simulate the atmosphere.

One class of upgrades includes the incorporation of new observations into the verification system. An

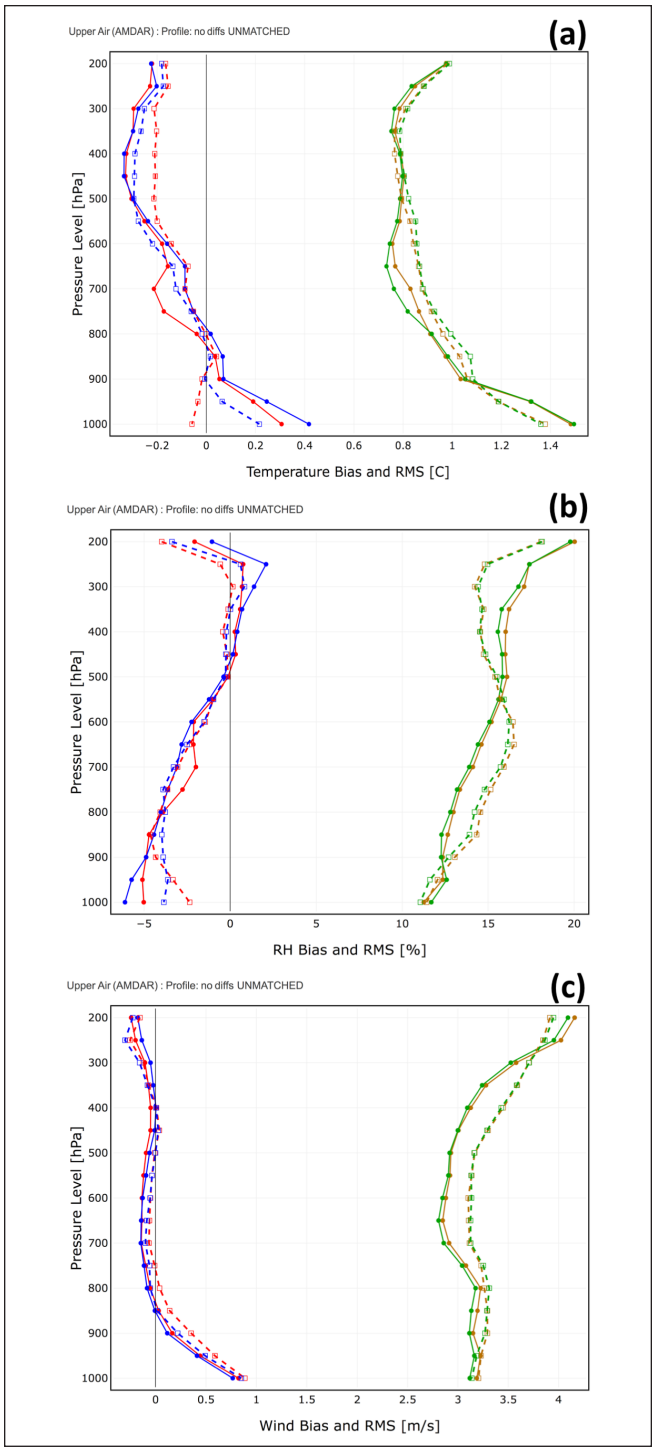

Figure 8. Examples from the aircraft (also called AMDAR) app showing the bias (red and blue lines) and RMS error (green and brown lines) for (a) temperature, (b) RH, and (c) wind. The HRRR-GSD results are shown in red and brown, while the HRRR-OPS results are shown in blue and green. Data for 1 August–13 September use closed circles with solid lines, while results for 14 September–31 October are shown in open squares with dashed lines.

important requirement for the observations is that they be available in near real-time in a production-like manner, and have well-characterized uncertainties. In particular, we are moving to include satellite data into the verification system. The starting point will be to verify top-of-the-atmosphere infrared brightness temperatures; however, we will limit this to channels that are not sensitive to the middle-to-upper stratospheric structure because the RAP/HRRR model top is 10 hPa. We also plan to include new applications to compare model output against satellite precipitation observations, especially over the oceans.

A second class of upgrades includes the ability to verify model ensembles such as the HRRR Ensemble (HRRRE; Dowell et al. 2016) against observations. Thus, MATS tools are needed to create Brier scores, rank histograms, receiver operating characteristic curves, continuous ranked probability scores, and reliability diagrams; these statistics are described by Wilks (1995) and Strauss and Lanzinger (1996). We already have begun this development using some tools from the National Center for Atmospheric Research (NCAR); we discuss this collaboration below.

Currently, almost all of the verification statistics provided by the GSD verification system are specific to the spatial region. Statistics computed over regions—especially large regions like the CONUS—can encompass many very different weather events, thereby making it hard for the model developer to separate out what physical process(es) in the model may need to be improved. For example, while the RMS errors for various variables have consistently decreased with time as the model has improved (e.g., the 2-m temperature RMS error over CONUS was 2.4°C for HRRR version 1, and is now 1.9°C for HRRR version 3), these RMS error values are still stubbornly large. Is this because there are some particular weather events that the model is unable to represent properly? To answer this, we hope to develop some physical-process oriented tools within MATS, which allow the model–observation pairs to be saved with various discriminators that allow the data to be interrogated in different ways (e.g., as a function of stability, wind direction, etc.). Process-oriented verification is becoming more popular in both the weather and climate communities (e.g., Maloney et al. 2019). The challenge here is one of data volume and subsequent system responsivity; some partial sums need to be computed to reduce the dimensionality of the data, but the data should not be reduced so much that we lose the ability to generate statistics for various scenarios

determined by the model's use of the discriminators. We already have one app that separates the surface meteorological statistics as a function of land-surface type; this app has proven very useful in adjusting the HRRR's performance over different surface roughness lengths (for example). Furthermore, funding from the second Wind Forecast Improvement Project (WFIP-2; Olson et al. 2019; Wilczak et al. 2019) has allowed us to develop a prototype app that includes a wide range of discriminators that can be used when generating the verification statistics.

We are considering using machine learning (ML) technologies to replace some of the current geographically constrained partial sums. This will be extremely important when attempting to reduce the dimensionality of satellite observations. For example, we may use an appropriately trained ML algorithm to identify a field of cumulus clouds in satellite data and the corresponding field (or lack thereof) in the model output; this would enable statistics such as 25th, 50th, and 75th percentile of the liquid water path in the cumulus field to be compared.

One of the lessons learned from the WFIP-2 prototype application is that the storage of discriminators with the partial sums results in a significantly larger dataset in the database. The inclusion of satellite data, as well as the continual growth in experimental models, also will grow the database substantially. We believe that we need to move our verification system to a NoSQL database architecture. Using a NoSQL solution—which is a document-based database system versus the current MySQL table-based system—will allow our system to scale more easily as the database continues to grow, and provide much more flexibility for implementing the physical-process oriented tools where the number of discriminators that we are storing may change with time as the model developers and their use of the verification system evolve.

While we have used the RAP and HRRR models as examples for this discussion, the GSD verification system is being used to evaluate all models being developed at GSD such as the flow-following finite-volume icosahedral model (Sun et al. 2018) and NOAA's new global forecast system using the finite-volume cubed-sphere (FV3; Putnam and Lin 2007) model. NOAA's current plan is to ultimately replace the RAP and HRRR, which are based upon the advanced research version of the Weather Research and Forecasting (WRF-ARW), with a stand-alone regional version of the FV3 model; this new regional model also will be a rapid-refresh type system with at least 1-h updates and is being called the rapid-refresh forecast system (RRFS). GSD plans to incorporate the FV3 and the RRFS into its verification system as these models develop so that GSD modelers can help evaluate, and ultimately improve, these modeling systems.

Last, scientists at NCAR, with support from several agencies, have been developing a suite of tools called the Model Evaluation Tools (METplus; Gotway et al. 2018). The METplus tools, which have been developed over the last decade, are being increasingly used by the NWS's Environmental Modeling Center (EMC). One of the roles played by the EMC is the final evaluation of new versions of models as these models are moved into production at NCEP. However, the graphical user interface to the METplus data, which is called METviewer, has many options and has proved to have a very difficult learning curve. GSD, with funding from the NWS as part of the Next Generation Global Prediction Program, is working with NCAR to develop a MATS interface to the METplus tools and data. This interface, which is called METexpress, uses the philosophy developed in GSD that the various options in the verification tools should be driven by the metadata associated with each application and that these metadata should be stored in the database. Indeed, five apps already have been developed for METexpress (see the bottom of Fig. 3; METexpress apps are shown with purple buttons), with many more currently under development. Furthermore, GSD is working with NCAR to package METplus, which includes METexpress, into containers that can be run both in cloud-computing environments and on other computing systems. Ultimately, we have come full circle to the beginning of this section, as MET tools are generating the verification statistics that compare model ensembles to observations and, thus, the METexpress interface will be used to make these statistics available to model developers and other users.

The discussion in this paper was focused on the model verification tools developed at GSD and how they are used by developers at GSD to improve both the physics and data assimilation components of modeling systems. However, MATS and METexpress apps are increasingly being used by users outside of GSD to gain insights on how well the different modeling systems perform for different seasons, regions, geophysical variables, and more. For example, forecasters at EMC are using MATS to characterize the Real-Time Mesoscale Analysis product (De Pondeca

et al. 2011), which is an important tool for the NWS. Similarly, operational forecasters can use MATS to gain insight into how the next version of the operational HRRR might differ from the version that is currently in operations, and thus help them to make better forecasts when the model is updated at NCEP. Furthermore, the MATS database is updated in real-time, and thus model forecasts can be evaluated through this interface within 1–2 h after the observations are collected, which could make this tool valuable for real-time decision support.

Acknowledgments: The GSD model verification system is available to users both inside and outside GSD (via www.esrl.noaa.gov/gsd/mats). There are many people who have contributed ideas that have been incorporated into the GSD verification system to improve it over the years. In particular, we thank Eric James, Haidao Lin, Steve Weygandt, Jaymes Kenyon, Joe Olson, Tanya Smirnova, Curtis Alexander, and Terra Ladwig. We also thank Craig Hartsough for serving as an independent evaluator during each release of MATS. Moreover, Bob Lipschutz plays a critical role in managing the observational and model data transfers. We also thank Melinda Marquis, Joseph Olson, and Jaymes Kenyon for their suggestions in the development of the prototype WFIP-2 physical-processed oriented verification app. This effort was primarily supported with internal GSD funds, with additional support from the Federal Aviation Administration (FAA), NOAA's Atmospheric Science for Renewable Energy Program, and the NWS via its Next Generation Global Prediction Program. The views expressed are those of the authors and do not necessarily reflect the official policy or position of NOAA, FAA, or NWS. We greatly appreciate the comments from two anonymous reviewers who helped improve various aspects of this paper.

REFERENCES

Augustine, J. A., J. J. DeLuisi, and C. N. Long, 2000: SURFRAD–A national surface radiation budget network for atmospheric research. *Bull. Amer. Meteor. Soc*., **81**, 2341–2357, Crossref.

Benjamin, S. G., and Coauthors, 2016: A North American hourly assimilation and model forecast cycle: The Rapid Refresh. *Mon. Wea. Rev*., **144**, 1669–1694, Crossref.

De Pondeca, M. S. F. V., and Coauthors, 2011: The Real-Time Mesoscale Analysis at NOAA's National Centers for Environmental Prediction: Current status and development. *Wea. Forecasting*, **26**, 593–612. Crossref.

Dowell, D. C., and Coauthors, 2016: Development of a High-Resolution Rapid Refresh Ensemble (HRRRE) for severe weather forecasting. Preprints, *28th Conf. on Severe Local Storms*, Portland, OR, Amer. Meteor. Soc., 8B.2. [Available online at ams.confex.com/ams/28SLS/webprogram/Paper301555.html.]

Gotway, J. H., K. Newman, T. Jensen, B. Brown, R. Bullock, and T. Fowler, 2018: Model Evaluation Tools version 8.0 (METv8.0) User's Guide. Developmental Testbed Center, Boulder, CO, 432 pp. [Available online at dtcenter.org/met/users/docs/users_guide/MET_Users_Guide_v8.0.pdf.]

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, Crossref.

Maloney, E. D., and Coauthors, 2019: Process-oriented evaluation of climate and weather forecasting models. *Bull. Amer. Meteor. Soc*., **100**, 1665–1686, Crossref.

Moninger, W. R., R. D. Mamrosh, and P. M. Pauley, 2003: Automated meteorological reports from commercial aircraft. *Bull. Amer. Meteor. Soc*., **84**, 203–216, Crossref.

____, S. G. Benjamin, B. D. Jamison, T. W. Schlatter, T. L. Smith, and E. J. Szoke, 2010: Evaluation of regional aircraft observations using TAMDAR. *Wea. Forecasting*, **25**, 627–645, Crossref.

Nelson, B. R., O. P. Prat, D.-J. Seo, and E. Habib, 2016: Assessment and implication of NCEP stage IV quantitative precipitation estimates for product intercomparisons. *Wea. Forecasting*, **31**, 371–394, Crossref.

Olson, J. B., and Coauthors, 2019: Improving wind energy forecasting through numerical weather prediction model development. *Bull. Amer. Meteor. Soc*., **100**, 2201–2220, Crossref.

Putnam, W. M., and S.-J. Lin, 2007: Finite-volume transport on various cubed-sphere grids. *J. Comput. Phys*., **227**, 55–78, Crossref.

Smith, T. M., and Coauthors, 2016: Multi-Radar Multi-Sensor (MRMS) severe weather and aviation products: Initial operating capabilities. *Bull. Amer. Meteor. Soc*., **97**, 1617–1630, Crossref.

Strauss, B., and A. Lanzinger, 1996: Verification of the ensemble prediction system (EPS). *ECMWF Newsletter*, No. 72, ECMWF, Reading, United Kingdom 9–15. [Available online at www.ecmwf.int/sites/default/files/elibrary/1996/14652-newsletter-no72-springsummer-1996.pdf.]

Sun, S., R. Bleck, S. G. Benjamin, B. W. Green, and G. A. Grell, 2018: Subseasonal forecasting with an icosahedral, vertically quasi-Lagrangian coupled model. Part I: Model overview and evaluation of systematic errors. *Mon. Wea. Rev.*, **146**, 1601–1617, Crossref.

Wilczak, J., and Coauthors, 2019: The second Wind Forecast Improvement Project (WFIP2): Observational field campaign. *Bull. Amer. Meteor. Soc.*, **100**, 1701–1723, Crossref.

Wilks, D. S., 1995: *Statistical Methods in the Atmospheric Sciences*. Academic Press, 467 pp.