Towards Modeling and Predicting Water Table Levels in California

KIEFFER THOMAS, EDWARD SALINAS, AND EJ HASELDEN

University of California, Berkeley School of Information

ABSTRACT

Water is precious, and current trends suggest it may become increasingly scarce. Predicting future water table levels can help manage this vital resource, so we have developed a modeling protocol to predict water table levels from late Spring to early Fall, coinciding with the dry season and heavy agricultural water consumption. We model using data sourced from NASA's GRACE satellites, direct measurements, and LANDSAT snowpack images.

Our protocol combines two independent approaches: 1) an ARIMA method predicting future water table levels based on previous values and historical trends, and 2) a composite image-processing ConvLSTM method which predicts future water table levels based on snowpack data from the LANDSAT in previous months, as well as other data gathered from GRACE satellites. The predictions from these methods are then input to a linear model, which produces a single final prediction. This protocol yielded accuracy over 96% - a marked improvement over previous work. The method successfully maintains error rates that do not trend upward within the prediction window.

I. Introduction

In recent decades, a changing climate and lack of precipitation in California have led to unprecedented drought conditions, marked by decreasing groundwater levels and a water table that struggles to meet agricultural, private, and public consumption needs (Bathke and Riganti 2019) (Mann and Gleick 2015). Studies from the Public Policy Institute of California have outlined major gaps in both the understanding of groundwater availability and groundwater usage, and highlight the need to develop groundwater modeling standards and authoritative groundwater models (Escriva-Bou et al. 2016).

Accurately measuring the true supply of groundwater and forecasting future groundwater levels is an immense challenge. The NASA Gravity Recovery and Climate Experiment (GRACE) (Beaudoing et al. 2021) provides estimates of groundwater storage based on complex modeling of GPS data from satellites and direct measurements. Researchers have corroborated and built on these measurements with methods such as leveraging remote-sensing technology to establish accurate gauges of aquifer storage (Ahamed Et al.,2022) and using infrared measurements to model precipitation (Funk et al. 2015). Machine learning techniques have shown promise in providing accurate forecasts for hydrologic phenomena, for instance using a Bayesian ensemble approach to quantify groundwater storage uncertainty (Yin et al., 2021) and artificial neural networks to predict water resource variables in river systems (Maier et al. 2010).

Our research aims to produce an accurate forecast of groundwater deficits for California over the duration of the "dry season", defined here as April through September, using data about precipitation and snowpack in the preceding months. We leverage two complimentary modeling approaches to arrive at our forecast. First, we built ARIMA models with historic measures of the water table, incorporating seasonality and broader trends in water table levels to predict groundwater levels for

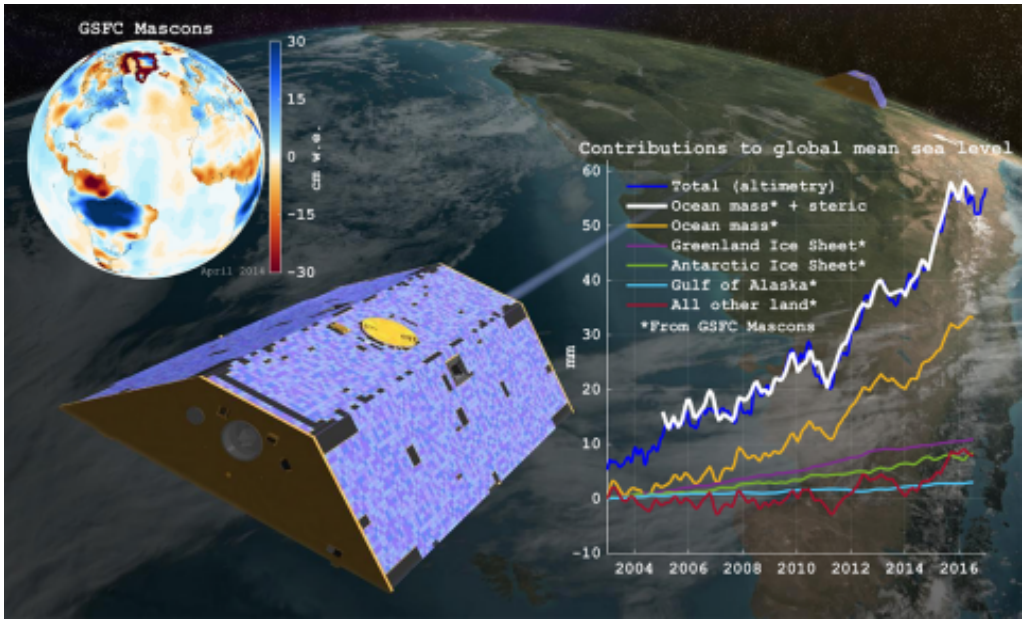


FIG. 1. GRACE Satellite Array

an upcoming dry season. Second, we make predictions for the water table using region-based Convolutional LSTM models that take as input snowpack imagery and key hydrologic variables during the winter months. We then combine these two predictions using linear regression models for each calendar month to arrive at a full seasonal forecast for the water table during the dry months.

Our model is able to predict the water table deficit with a high level of accuracy, on average within 4.3 percentage points of the measurement recorded by the NASA GRACE/GLDAS hydrologic models for a predicted month (with a range of 2.7% to 7.5% average error across our predicted time frames). This model can be used as a helpful prognostic, informing important decisions related to agricultural water use, well-metering, usage restrictions, and even residential and commercial zoning.

II. Data

In this analysis, we use measurements from the NASA Global Land Data Assimilation System (GLDAS) (Li et al. 2020) and the Gravity Recovery and Climate Experiment - Terrestrial Water Storage (GRACE-TWS) (Beaudoing et al. 2021) (Girotto et al., 2016) models. The GLDAS model estimates hydrological measurements such as groundwater level, snowpack, and precipitation by assimilating satellite imagery with a "huge amount" of direct observational data. GRACE data are based on terrestrial water storage observations and integrated with other data sources using a sophisticated model of land surface water and energy processes.

Data was provided to our team by Dr. Manuela Girotto, and consists of monthly hydrologic measurements for the years 2003-2016; measurements are granularly defined in a 33x37 gridded array that captures measurements for the entirety of California (as well as the Sierra Nevada Mountains). Each grid location covers an area of roughly 36 km^2. We used the following features to make predictions of the key target variable *catdef*, an abbreviation for catchment deficit:

GRACE/GLDAS Measurements:

- Catchment deficit (*catdef*, also referred to as "water table"): average depth of water that would need to be added to bring the catchment to saturation
- Root-Zone Moisture Content (*rzmc*): amount of water contained in the root-zone soil layer (0-200cm depth)
- Surface Excess Water (*srfexc*): amount of water in the surface soil layer (0-5cm depth) in excess of equilibrium

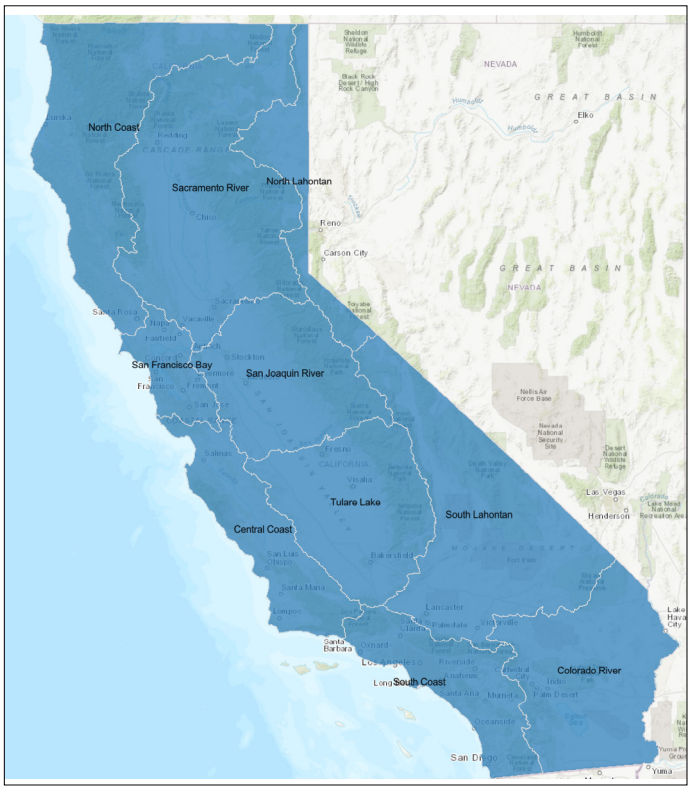
Landsat:

- Snowpack: snow accumulation in the Sierra Nevada mountains based on “remotely sensed fractional snow-covered area data over the Landsat 5–8 record” (Margulis Et al. 2016)

A primary research objective for this study is to determine the relationship between snowpack in the Sierra Nevada mountains through the winter and groundwater measurements throughout California during the subsequent dry season of April through September. As a measure of snowpack and the snow-water equivalent (SWE) therein, our models incorporate input images generated from a Landsat snowpack reanalysis of the Sierras (Margulis Et al. 2016). The study corroborates satellite imagery with in situ snow-level measurements to get highly accurate visual representations of total snowpack.

Further, it is meaningful to differentiate the hydrologic regions in California, where a wide range of different climates mean that snowpack may have little to no relationship with groundwater levels. For example, the arid deserts in Southern California possess little groundwater year-round, and the forested regions in Northern California rely mainly on rainwater to replenish groundwater stores. By contrast, regions like the Central Valley, an agricultural lynchpin that relies heavily on groundwater for irrigation during the summer months, rely on runoff from the snowpack through the drier months. As such, we have used the hydrologic regions defined by the California Department of Water Resources to segment our geographic data and produce unique predictions for each distinct region.

A third objective of our research was to explore any relationship between the GLDAS/GRACE data and data acquired from direct groundwater well measurements. Primarily, we used ARIMA models to predict well measurements from the California Natural Resources Agency (CNRA) (CNRA 2022) in a similar manner to our water table catdef predictions for comparison. Additionally, we correlated well measurements from CNRA to water table measurements from the nearest geographical point in the GRACE/GLDAS data, to see how the well measurements align with the assimilated/integrated data from those models. Our analysis of CNRA data primarily focuses on the measurement of Ground Surface Elevation to Groundwater Elevation (GSE_GWE), the depth to groundwater elevation in feet below ground surface (ft), for the subset of 36 wells that had complete measurements for the time frame from 2003 to 2016.



Source: https://gis.data.ca.gov/datasets/2a572a181e094020bdaeb5203162de15_0/explore

FIG. 2. California hydrologic regions

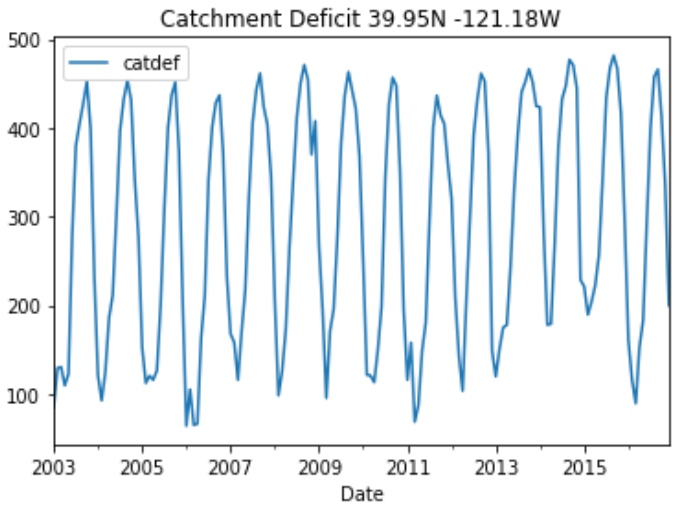


FIG. 3. Catdef time-series data

III. Models and Methods

ARIMA

We modeled the catdef time series using ARIMA models. ARIMA is an acronym for Auto-Regressive Integrated Moving Average. ARIMA models take past values from a time series to predict new values using fitted coefficients (Hyndman and Athanasopoulos 2018). The data's inherent time-series nature with its discrete, equi-spaced intervals permits ARIMA models to be considered and used (Box et al. 2015). ARIMA is linear regression at its core; such regressions are ideal for this project because they do not require extensive data. ARIMA models were designed specifically with time series in mind as well. The relatively constrained number of data points (144 months of training data) are sufficient to train ARIMA models. Additional notes about ARIMA models are offered in the appendix.

Each of the 440 grid locations in the dataset for California were subjected to a scripted automated ARIMA modeling protocol. For each grid location, a series of ARIMA models were fit and used to predict catdef from 1 to 6 months out (April through September). Starting with the subsequence of the first 12 months, an individual ARIMA model was fit for that subsequence and for each subsequence starting with the first twelve months and ending at each month after. This way, the first 12 months led to a model, the first 13 led to a model, and so on. Generally, the first $12 + i$, for $i \geq 12$ months of data, led to an ARIMA model being fit. Forecasts were generated for each of these models' 6-month lead time (Cryer and Chan 2018). We trained 152 models for each California grid location. With a total of 440 locations, that resulted in 66,880 models.

The models were fit with the auto.arima function in R using the forecast package (Hyndman and Khandakar 2008). This routine was used for several reasons. First, with so many series to estimate, a scripted protocol is necessary. Second, auto.arima can estimate many ARIMA models with various p, d, and q hyperparameters and seasonal P, D, and Q hyperparameters and carries out a selection routine using the AIC_c to choose among the models it considers. Third, auto.arima can work with data that has trends, and address those trends by performing and incorporating any pre-processing into the model.

By default, auto.arima can consider non-stationary data and data with trends, but it will not perform any Box-Cox transformations (Hyndman and Athanasopoulos 2018). To better explore models and permit Box-Cox transformations to be considered, we ran auto.arima in two modes, one with lambda = *NULL* (the default) and one with lambda = "auto". With the default mode, no Box-Cox transformation was done. With the auto-mode, a transformation was done which automatically selected a λ-value for the transformation when the model was fit. See (Hyndman and Athanasopoulos 2018) for more details on the transformations.

To begin modeling the CNRA data, we first plotted the well measurements of each site and plotted the locations of these wells on a state map using the field *GSE_GWE*. Observations of the plots revealed that the curves could be clustered based on their shapes. In addition to their shapes, they could also be clustered based on their geographical locations in the state. Based on both visual inspections of the curves and their locations and cluster memberships, four wells were chosen. Each well was chosen as a characteristic example of the wells in its cluster. These four wells, identified by their site IDs (340033N1170693W001, 353890N1191471W001, 373177N1219435W005, and 373922N1183430W001) were subsequently subjected to ARIMA modeling.

The CNRA well measurements were modeled in two ways. First, they were modeled with auto.arima, as were the other data. Additionally, they were subjected to a "grid-search" methodology that explicitly explores ARIMA approaches with different p, d, q, P, D, and Q parameters and selects the model with the best "corrected" AIC. The corrected AIC compensates for the small sample size.

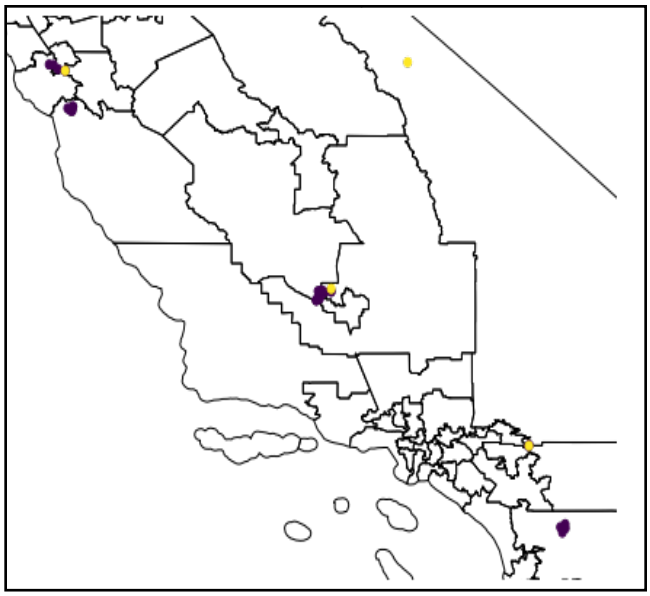


FIG. 4. CNRA well locations across CA (selections in yellow)

Convolutional LSTM

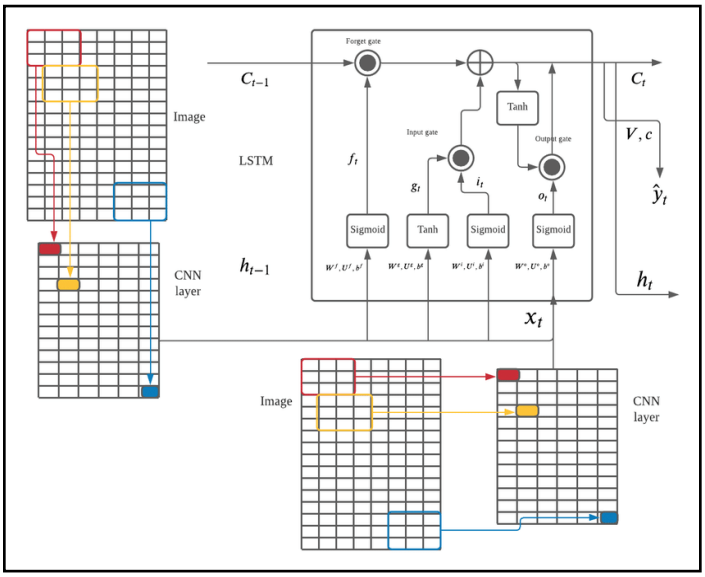
As a complement to the ARIMA prediction, we opted to build upon previous work that achieved promising results using Convolutional LSTM models (Wong et al. 2021), which are defined as Long Short-Term Memory models with convolutional input and recurrent transformations. Our models used a mean absolute error loss function and Adam optimization. For further reference, see the LSTM and Convolutional LSTM descriptions in the Appendix.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv_lst_m2d (ConvLSTM2D)    (None, 1, 150, 322, 64)   156928
_________________________________________________________________
max_pooling3d (MaxPooling3D) (None, 1, 75, 161, 64)    0
_________________________________________________________________
conv_lst_m2d_1 (ConvLSTM2D)  (None, 1, 73, 159, 64)    295168
_________________________________________________________________
max_pooling3d_1 (MaxPooling3 (None, 1, 36, 79, 64)     0
_________________________________________________________________
flatten (Flatten)            (None, 182016)            0
_________________________________________________________________
dense (Dense)                (None, 1300)              236622100
_________________________________________________________________
dense_1 (Dense)              (None, 1300)              1691300
_________________________________________________________________
dropout (Dropout)            (None, 1300)              0
_________________________________________________________________
dense_2 (Dense)              (None, 1221)              1588521
=================================================================
Total params: 240,354,017
Trainable params: 240,354,017
Non-trainable params: 0
```

FIG. 5. ConvLSTM model summary

The previous study established that the most effective training technique combined two features (snowpack plus at least one selected feature from the GRACE dataset) into a single image for each month represented in the data. The resulting sequence of images constituted an ideal input for the ConvLSTM architecture, which was then used to predict catdef for each grid coordinate in the entire sampled area.

We reasoned that each of California's ten unique hydrologic regions would be best served by a different selection of features, so we decided to train a dedicated ConvLSTM model for each hydrologic region. As in the previous study, each model was trained on a composite two-dimensional image (x-values) that represented the entire sampled area, but we provided catdef (y-values) for only one hydrologic region at a time. Values for grid locations outside the selected region were set to null in the training data. We then evaluated each model's accuracy (mean absolute percentage error) based only on that region. This allowed us to tailor composite images using different features for different regions (e.g., rzmc and snowpack for Region 1, srfexc and snowpack for Region 2, etc.; see Figure 18). We tested a total of 44 models: rzmc, srfexc, RainfC (rain from convection), and evapotranspiration for each of the ten regions, as well as one instance each of rzmc, srfexc, RainfC, and evap trained on the entire array of catdef measurements for California.



Source: https://www.researchgate.net/figure/Architecture-of-ConvLSTM2D-in-one-layer_fig2_356851239

FIG. 6. Example ConvLSTM diagram

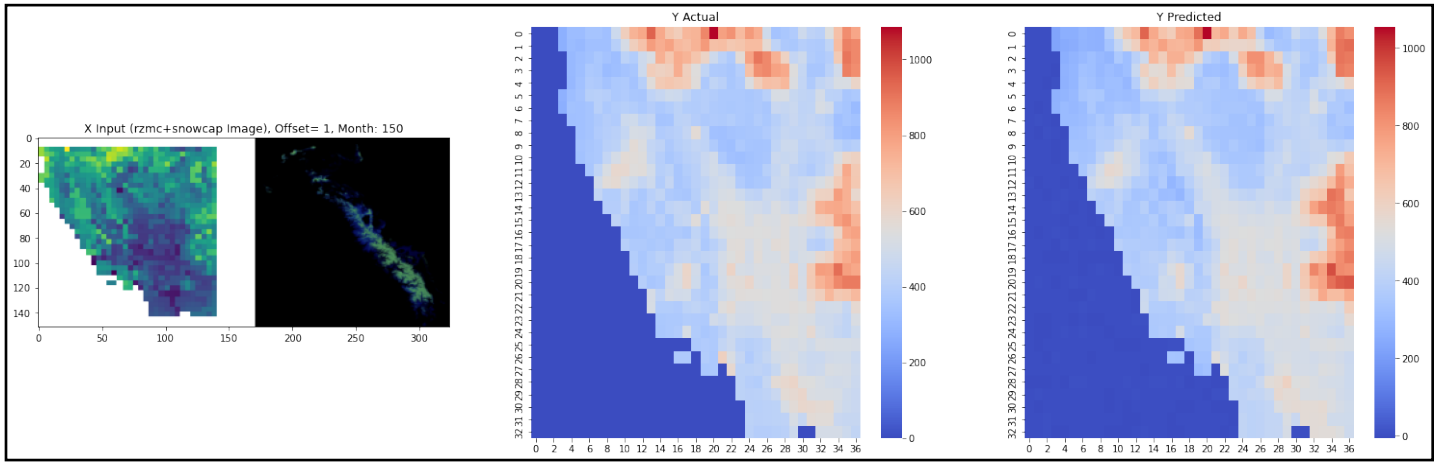


FIG. 7 Prediction example from prior study showing composite feature image (left), actual catdef visualization (center), and predicted catdef visualization (right).

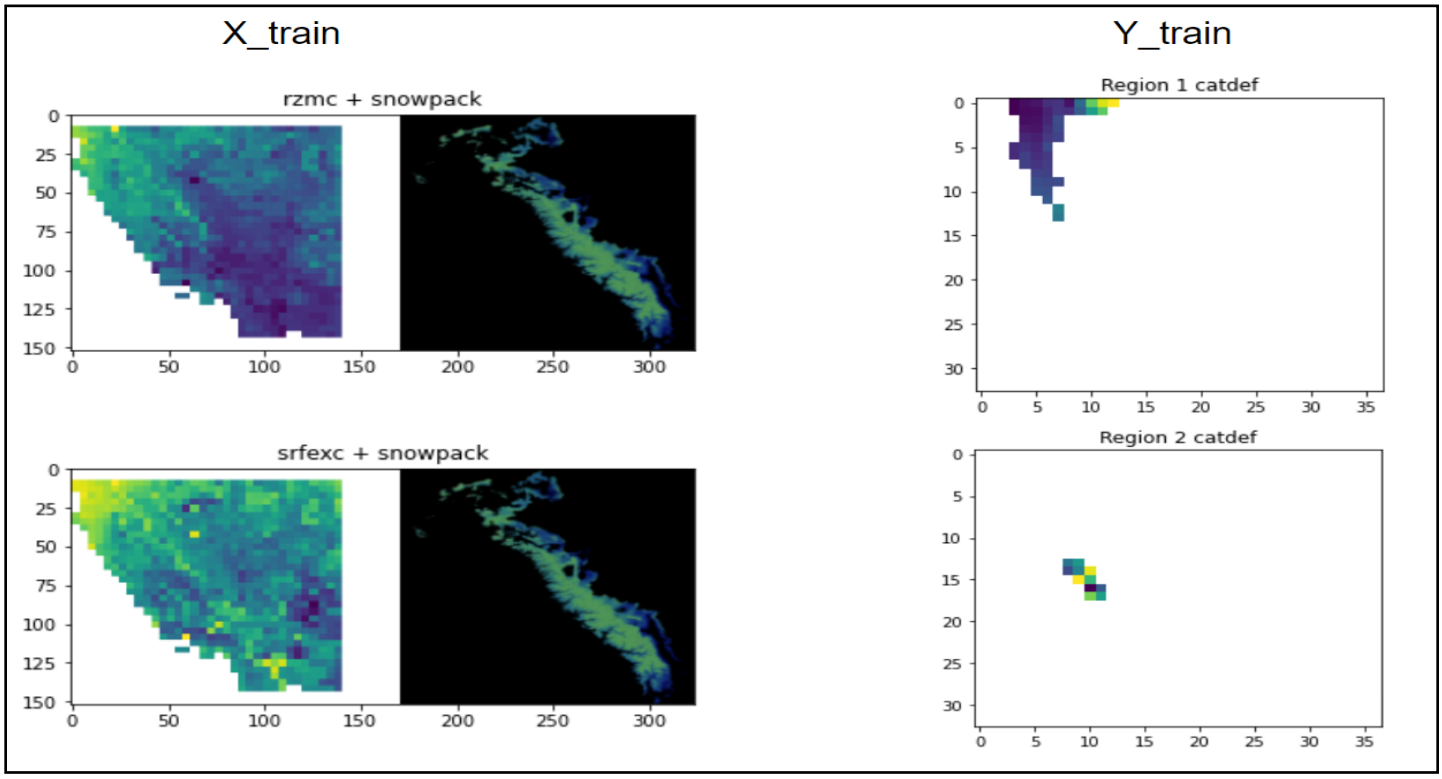


FIG. 8. Targeted training data configuration for regions 1 and 2 show Region 1 trained with rzmc + snowpack composite image (top left) and Region 1 catdef data (top right); Region 2 trained with srfexc + snowpack composite image (bottom left) and Region 2 catdef data (bottom right). Each model then predicted catdef specifically for its respective region.

Ensemble

This study aimed to incorporate both the ARIMA predictions that capture only the seasonality and overall trends seen in the catchment deficit measure, as well as the tailored predictions for catdef from the series of snowpack and hydrologic measurements captured in the ConvLSTM models.

In order to effectively model the combined predictions from the two models, we input the ARIMA and ConvLSTM catdef predictions for the months of April through September in 2013 and 2014 to fit our ensemble models using actual catdef measurements as the target. We then assessed the fit of the ensemble predictions on the months of April through September in the years 2015 to 2016.

We combined the ARIMA and ConvLSTM predictions in a simple linear model that can be defined as follows:

$$prediction_{month} = arima\_pred_{month} * w1_{month} + clstm\_pred_{month} * w2_{month} + bias_{month}$$

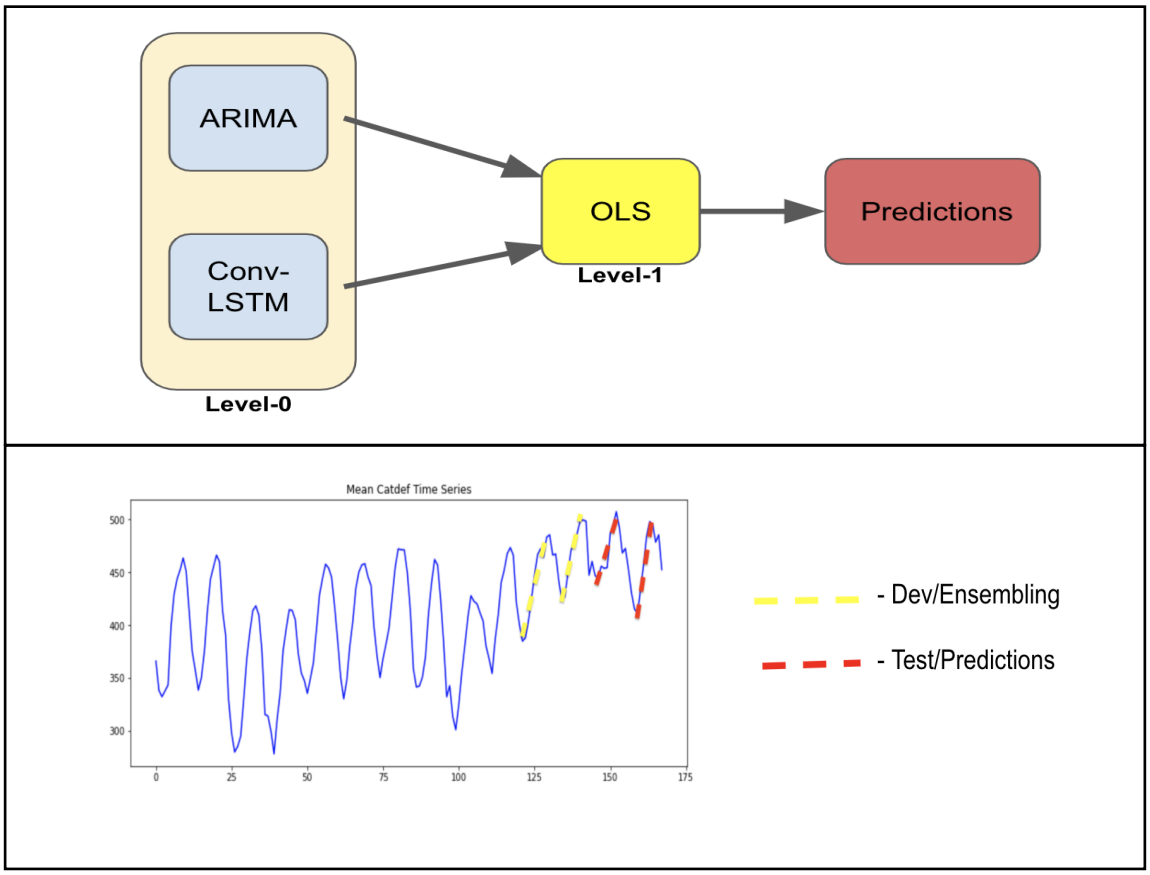


FIG. 9 (top) ARIMA and ConvLSTM predictions combined in an OLS ensemble; (bottom) catdef predictions from ensemble model compared to actual aggregate catdef

We used three different methods of segmenting the data to fit the linear models with the scikit-learn package in Python:

1. Using all prediction data over the development data timeframe (5280 total observations over the two six-month time periods in 2013 and 2014) and built one comprehensive model for all predictions.
2. Subsets of data for each individual gridpoint (12 observations used to fit a linear model for each individual grid location).
3. Subsets of data for each month in the prediction periods–all April observations for both years in the dev data, all May observations etc.—and built six distinct linear models for each month predicted (880 observations for each month in the development data).

The best performance was found in the third segmentation method, with a distinct model for each month in a prediction period.

We also experimented with using neural networks to ensemble the ARIMA and ConvLSTM predictions, incorporating temporal elements (month of measurement) and spatial information (grid locations measured in x and y coordinates). The neural network predictions did not produce demonstrably better results than those from the linear models. As the neural network ensemble approach is more prone to overfitting, less interpretable, and built with a development dataset that is relatively small with only 5280 observations, we chose the monthly linear ensemble models as the most parsimonious and accurate method for combining predictions.

IV. Results

ARIMA

We calculated residuals from the ARIMA models and computed an average RMSE for each grid location. This was done 440 times, once for each grid location in CA and at each of the lead times (one through six months). These grid-level average RMSE values are plotted in the heatmaps below. The same assessments of error were carried out with the ARIMA models that used Automatic Box-Cox transformations. For each plot, the total RMSE was tabulated. For each of the six-month horizons, the total RMSE was lower for the models with the transformations applied.

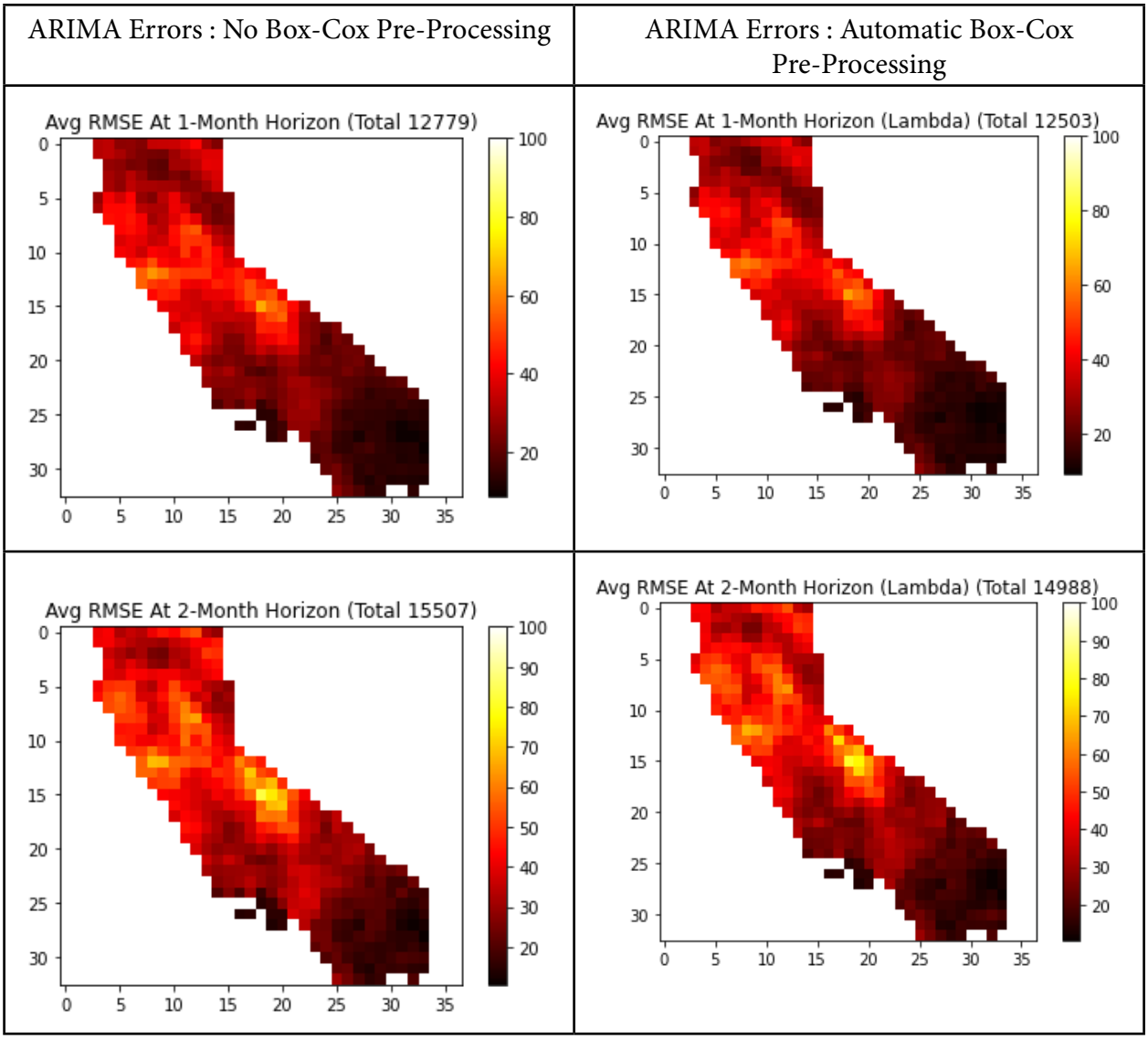


FIG. 10. The improved RMSE in the Box-Cox transformed data (right) becomes more apparent over time vs the untransformed data (left).

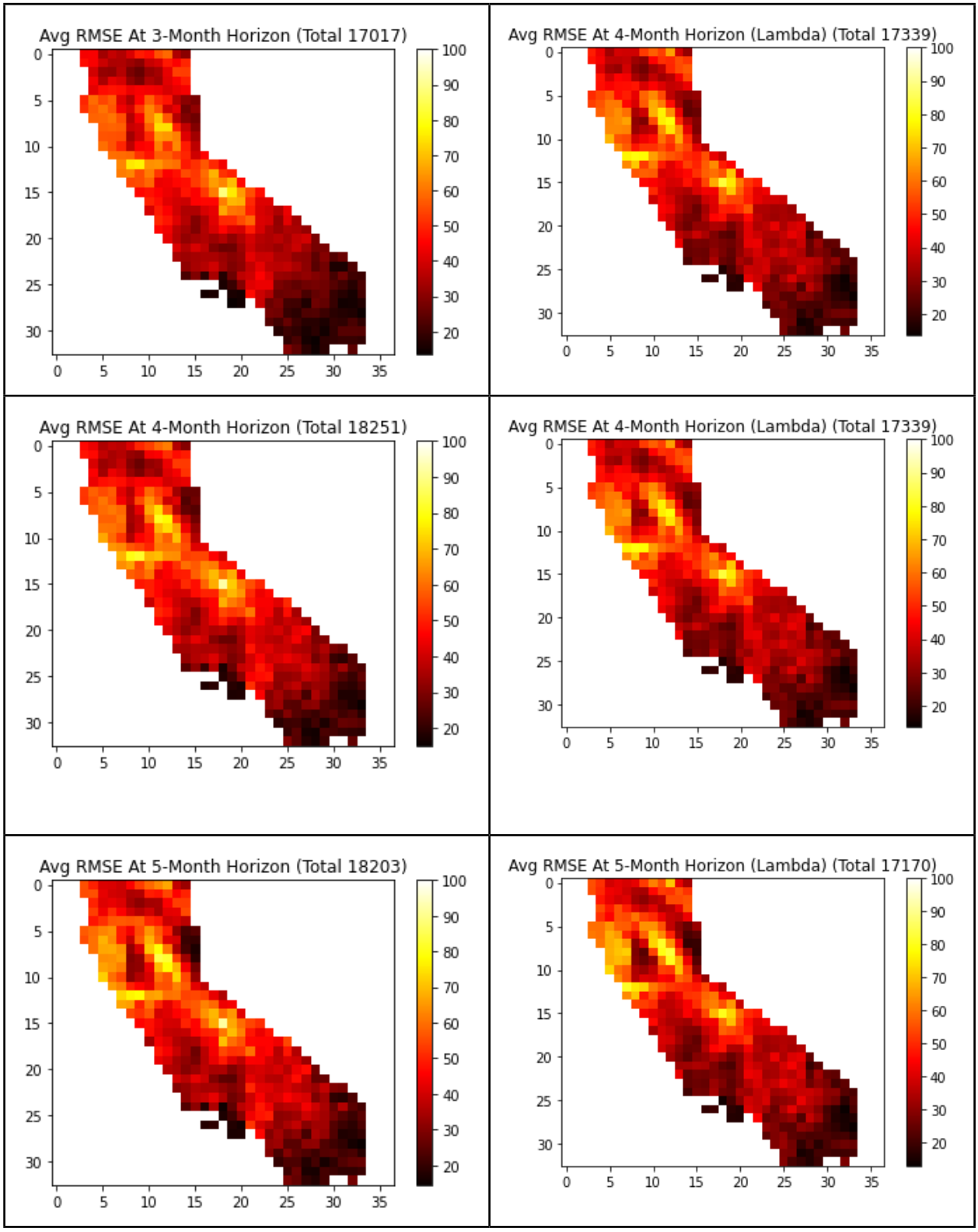


FIG. 10 (continued). The improved RMSE in the Box-Cox transformed data (right) becomes more apparent over time vs the untransformed data (left).

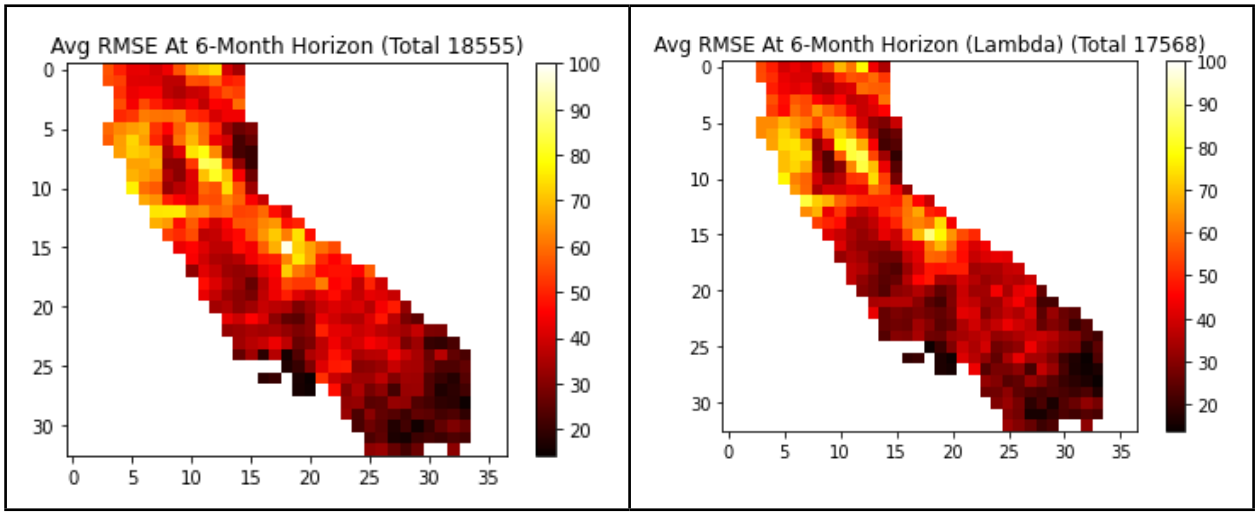


FIG. 10 (continued). The improved RMSE in the Box-Cox transformed data (right) becomes more apparent over time vs the untransformed data (left).

As seen in the plots, regions in or near the San Joaquin valley and the Sierra Nevada mountains are where many of the errors are concentrated. We believe these errors may be related to agricultural water consumption and the dynamic effects from water runoff or snowmelt in the mountains.

The four aforementioned CNRA well-site time-series datasets were subjected to both auto.arima and a trial-and-error approach that selected a "best" ARIMA model determined by its AIC_c. Models with lower AIC_cs were preferred. For each well, for each year 2013-2016, data from January 2003 through and including March of the most recent year were input to both ARIMA modeling protocols. For example, a model for 2013 was fit on data from January 2003 to March of 2013. Six-month-ahead forecasts for April through September covered the Summer season, but part of the Spring and Fall as well.

We estimated 16 total models this way: two protocols across four sites for four years. Plots for 2016 are shown in Figure 11. In these plots, accompanied with RMSE for each forecast, the auto.arima performs better according to its overall lower RMSE scores. For the well with prefix "353…" (Fig. 11b), the auto.arima forecast computed is a "naive" forecast, meaning that the six months are forecasted simply to be the most recent value in the series repeated six times. This was due to some technical difficulties in the auto.arima library. For the well with prefix "3731…" (Fig. 11c), both models predicted the well measurement to increase. In contrast, for the well with prefix "3739…" (Fig. 11d), both approaches correctly predicted an upturn and subsequent downturn.

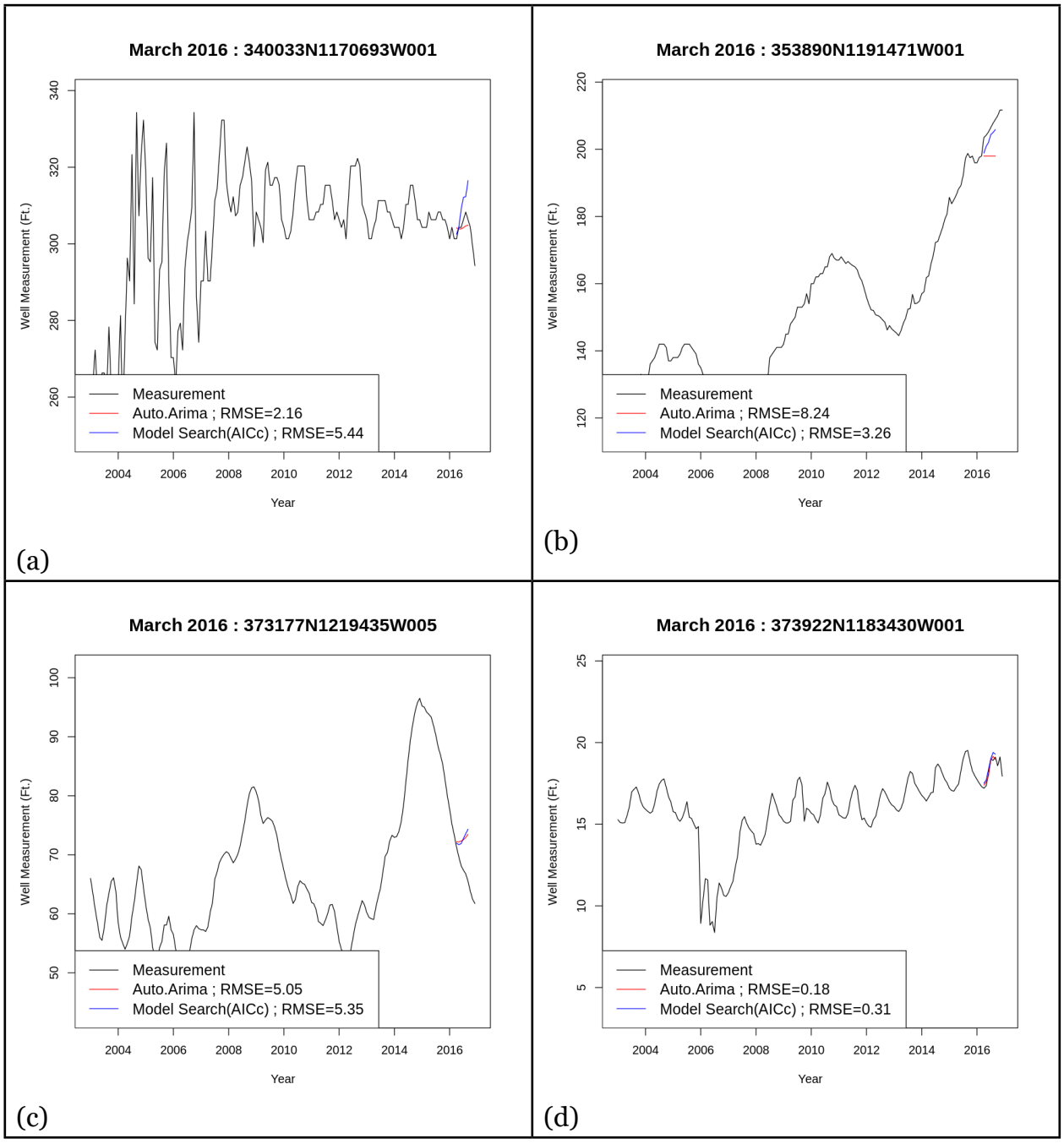


FIG. 11. Selected CNRA well sites. The auto.arima forecast computed in (b) is a "naive" forecast (meaning that the six months are forecasted simply to be the most recent value in the series repeated six times). Both models predicted the well measurement from (c) to increase. Both models correctly predicted an upturn, peak, and subsequent downturn in the well measurement from (d).

Convolutional LSTM

After training all 44 ConvLSTM models on 123 months of data (Jan. 2003 through Mar. 2013), we chose the best configuration for each region (i.e., the model that yielded the lowest mean absolute percentage error versus the actual catdef) for the subsequent 24-month period (Apr. 2013 through Jan. 2015). While limiting the catdef training y-value to a single region improved the performance for most models, some regions (4, 8, and 9) were actually better predicted by a model trained using the entire array of catdef measurements for California. As a result, our final suite of models consisted of seven region-specific models and the region-independent "base" model that was used to predict regions 4, 8, and 9. See Table 1.

We then assembled the regional output predictions from all eight models into a statewide prediction (see Fig 12). Due to time and computational constraints, we chose this 'static' selection over a more complex approach that could potentially weighted the output of each model across each region dynamically over time.

TABLE. 1. Final model configuration for each region

Region	Feature used to train model, in addition to snowpack (x-value)	Catdef region used to train model (y-value)
1	rzmc	1
2	srfexc	2
3	srfexc	3
4	srfexc	All of CA (base model)
5	rzmc	5
6	srfexc	6
7	srfexc	7
8	srfexc	All of CA (base model)
9	srfexc	All of CA (base model)
10	srfexc	10

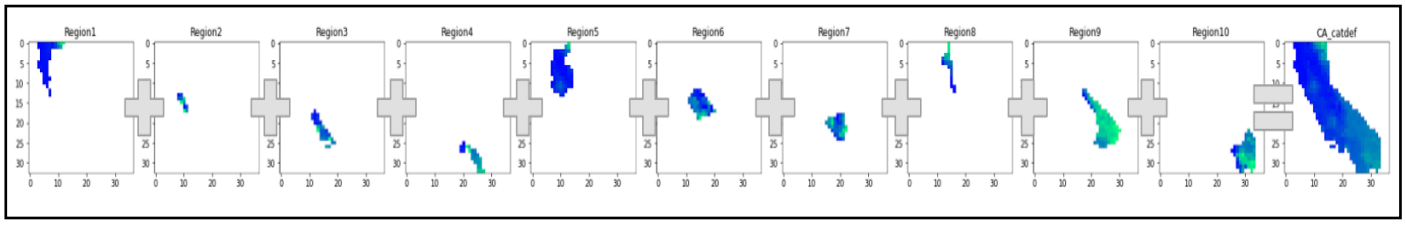


FIG. 12. Regional catdef predictions assembled into statewide prediction

Our final composite ConvLSTM predicted the total California catdef (sum of 440 grid predictions) with an average accuracy of 91.7% (MAPE = 8.3%) vs. 88.1% for the original configuration (single model trained on full catdef, snowpack, and rzmc) over the 27-month testing period used in the final ensemble (Sep. 2014 through Dec. 2016).

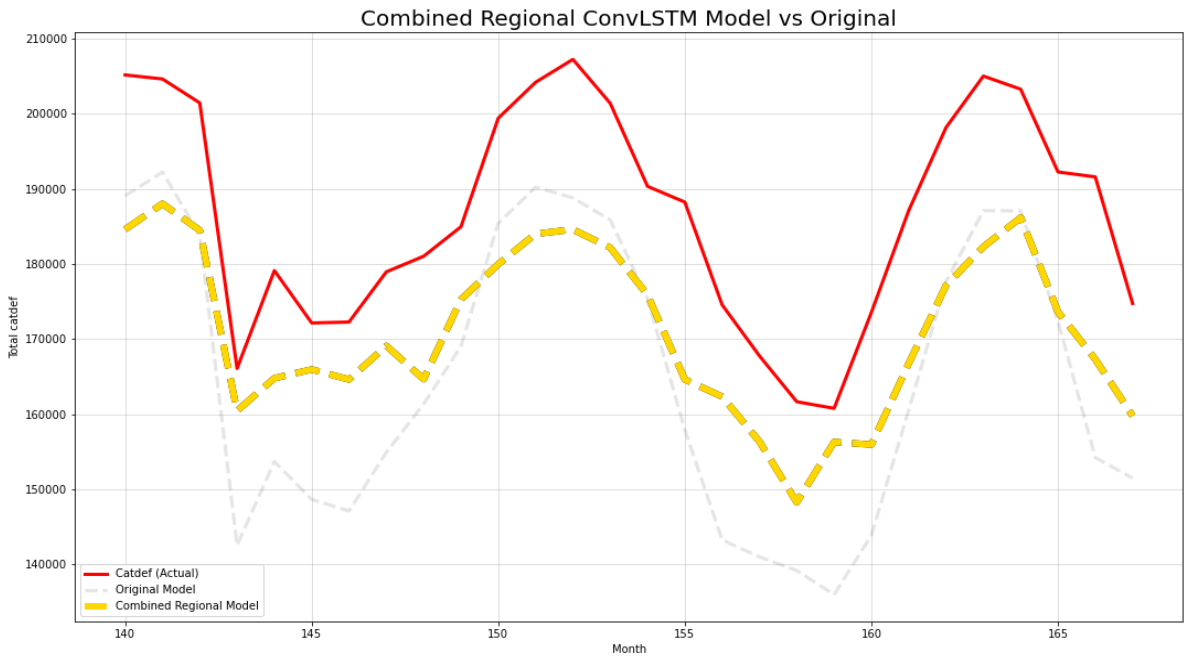


FIG. 12. The combined regional ConvLSTM model shows improved accuracy over a single ConvLSTM used to predict the entire aggregate catdef.

Ensemble

With the monthly linear ensemble model, we see marked improvements in overall prediction accuracy. Using MAPE for the predicted catdef measurement as the evaluation metric, we see in 2015 the ARIMA, ConvLSTM, and ensemble linear model had overall MAPE of 5.03%, 9.15%, and 3.73%, respectively. For 2016, overall MAPE for the ARIMA model is 5.08%, ConvLSTM 9.79%, and ensemble 4.80%.

Figure 12 illustrates the performance of the different models across the prediction periods of April through September in 2015 and 2016. Interestingly, the best estimates are for the later months in the prediction periods; though later in time from when the predictions were made, this suggests the models can predict with seemingly decent accuracy the apex of drought indicators (e.g. catdef and groundwater levels) in a given dry season. See Appendix for further analysis of the mean, minimum, and maximum predictions of catdef.

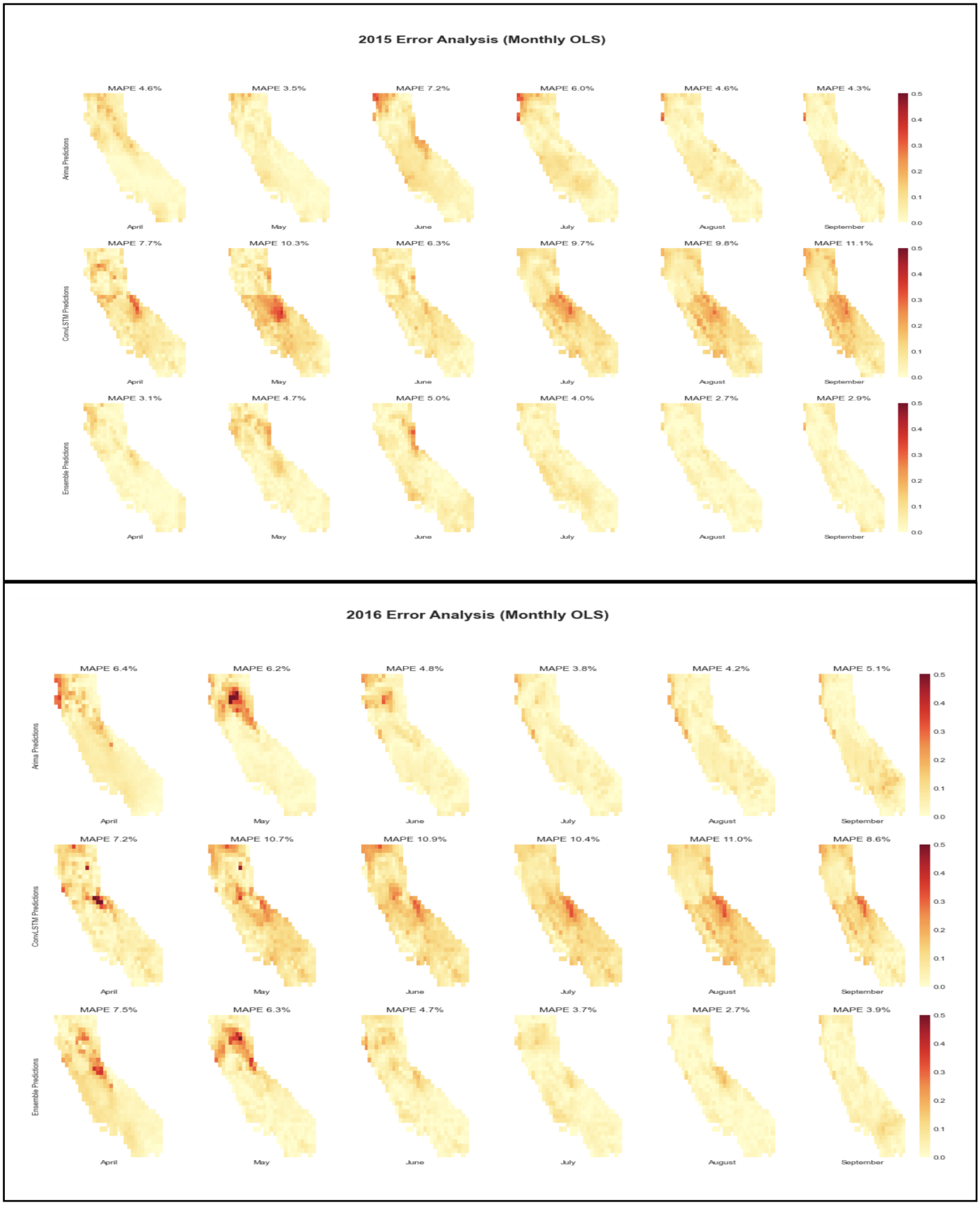


FIG. 12. Mean absolute percent error across California for ARIMA, ConvLSTM, and Ensemble predictions, Apr-Sep 2015 (top) and 2016 (bottom)

Illustrating predictions for specific grid points highlights the ensemble model's improved accuracy relative to the component models. Figure 14 shows catdef predictions compared to actual catdef measurements for grid point [4,10] (lat/long 41.245 N, -121.93 W)—located in the forested regions of Northern California, more reliant on rainfall and with bigger seasonal changes to catdef—and grid point [24,20] (34.138 N, -118.2 W)—located in the arid desert regions of Southern California, with smaller fluctuations in catdef and the underlying groundwater levels.

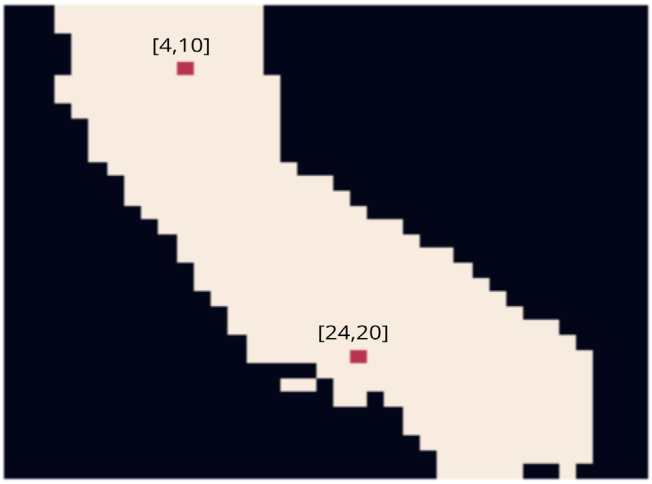


FIG. 13. Geographic location of gridpoints [4,10] and [24,20]

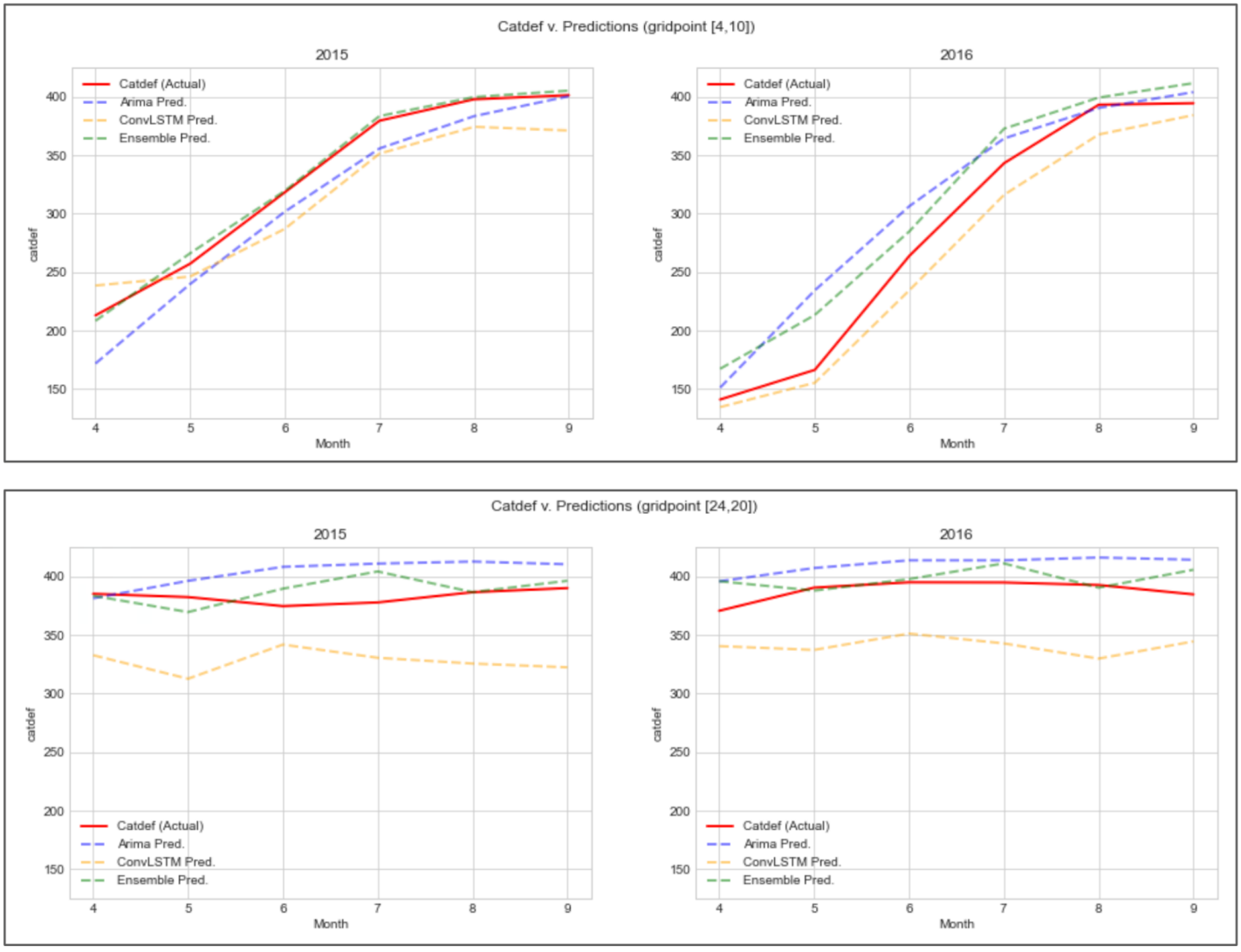


FIG. 14. Comparison of actual catdef and model predictions for gridpoints [4,10] (top) and [24,20] (bottom)

A full summary of the individual monthly ensemble models' coefficients for the ARIMA and ConvLSTM predictions and corresponding R-squared values indicate that in all cases, each model's catdef estimate coefficients are highly significant ($p < .001$), contributing to a more precise overall estimate of catdef, and R-squared values are all above .9, indicating these ensemble models can account for nearly all of the variance observed in catdef measurements.

TABLE. 2. Ensemble model coefficients, standard errors, and R-squared values

OLS Model Month	Intercept	ARIMA_coef	ConvLSTM_coef	R^2
4	24.135 (1.669)	0.723 (0.018)	0.251 (0.02)	0.985856
5	27.423 (2.014)	0.425 (0.016)	0.556 (0.018)	0.984319
6	30.67 (2.632)	0.304 (0.016)	0.688 (0.02)	0.972942
7	36.951 (2.783)	0.541 (0.012)	0.438 (0.014)	0.961528
8	28.027 (2.893)	0.445 (0.016)	0.538 (0.017)	0.959584
9	32.141 (4.067)	0.641 (0.026)	0.314 (0.028)	0.921162

Results Summary

Prediction average percent error was brought from as high as 11% down to 2.9% for September 2015. Performance was relatively poor in 2016, however; for April of that year, the ARIMA, ConvLSTM, and linear predictor MAPE values were 6.4%, 7.2%, and 7.5% respectively, with the linear model MAPE being between the other two, and closer to the worse of the two. MAPE trends lower over time for that year with an overall value of 3.9% for the linear predictor vs 5.1% and 8.6% for the ARIMA and ConvLSTM models, respectively. Figure 15 presents coefficients for the linear model over time. The bias is log10 transformed before plotting to help the scale, but its true value resides in the range ~(25-40). A scatterplot from the test data shows residuals of the ConvLSTM plotted against residuals of the ARIMA model. 36%, 55%, 6%, and 3% of the points are distributed among the four quadrants, respectively. 55 + 36 = 91% of the time, the ConvLSTM model underestimated the true catdef value resulting in positive residuals. In contrast, the ARIMA model underestimated 39% of the time and overestimated 61% of the time.

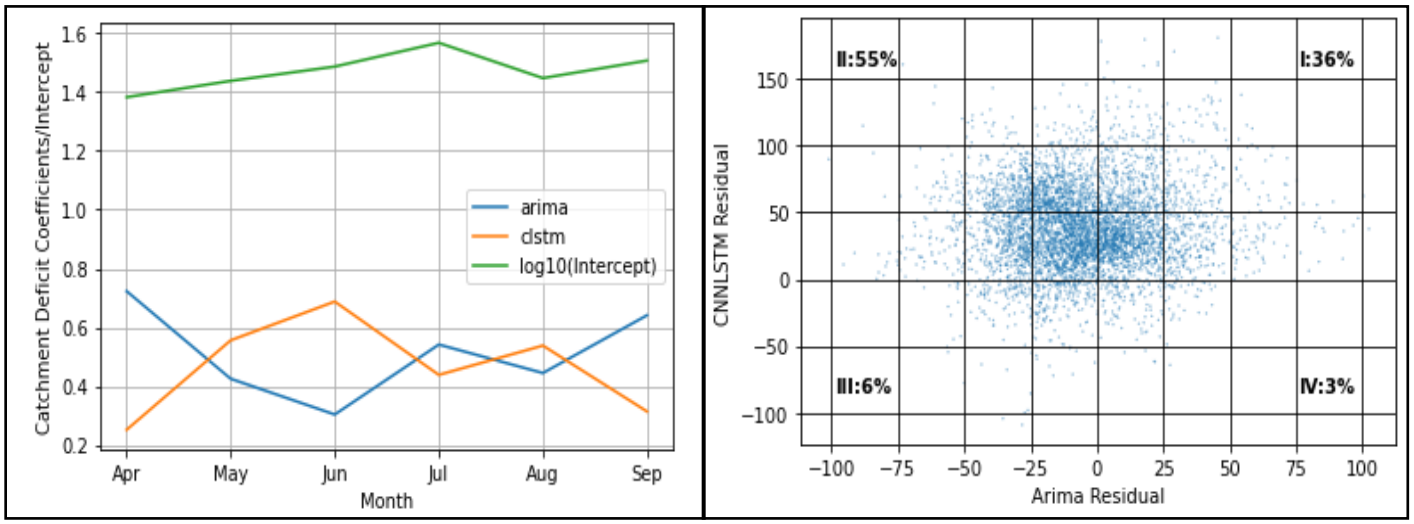


FIG. 15. Ensemble coefficients (left); model residuals in test data (right)

One goal of this study was to produce potentially actionable data for policymakers who allot water distribution based on, among other things, the predicted state of the water table across future dry seasons. Through discussion with multiple hydrologists, we determined that a minimum/maximun predicted catdef envelope over a given dry season may be more valuable than a series of point estimates.

We produced such a report for April through September of 2015 and 2016, respectively. Figure 16 illustrates our prediction envelopes for each hydrologic region, compared to the actual aggregated catdef values across that region for each month, and Figure 17 shows MAPE for the minimum, mean, and maximum value predictions for each grid point estimate.

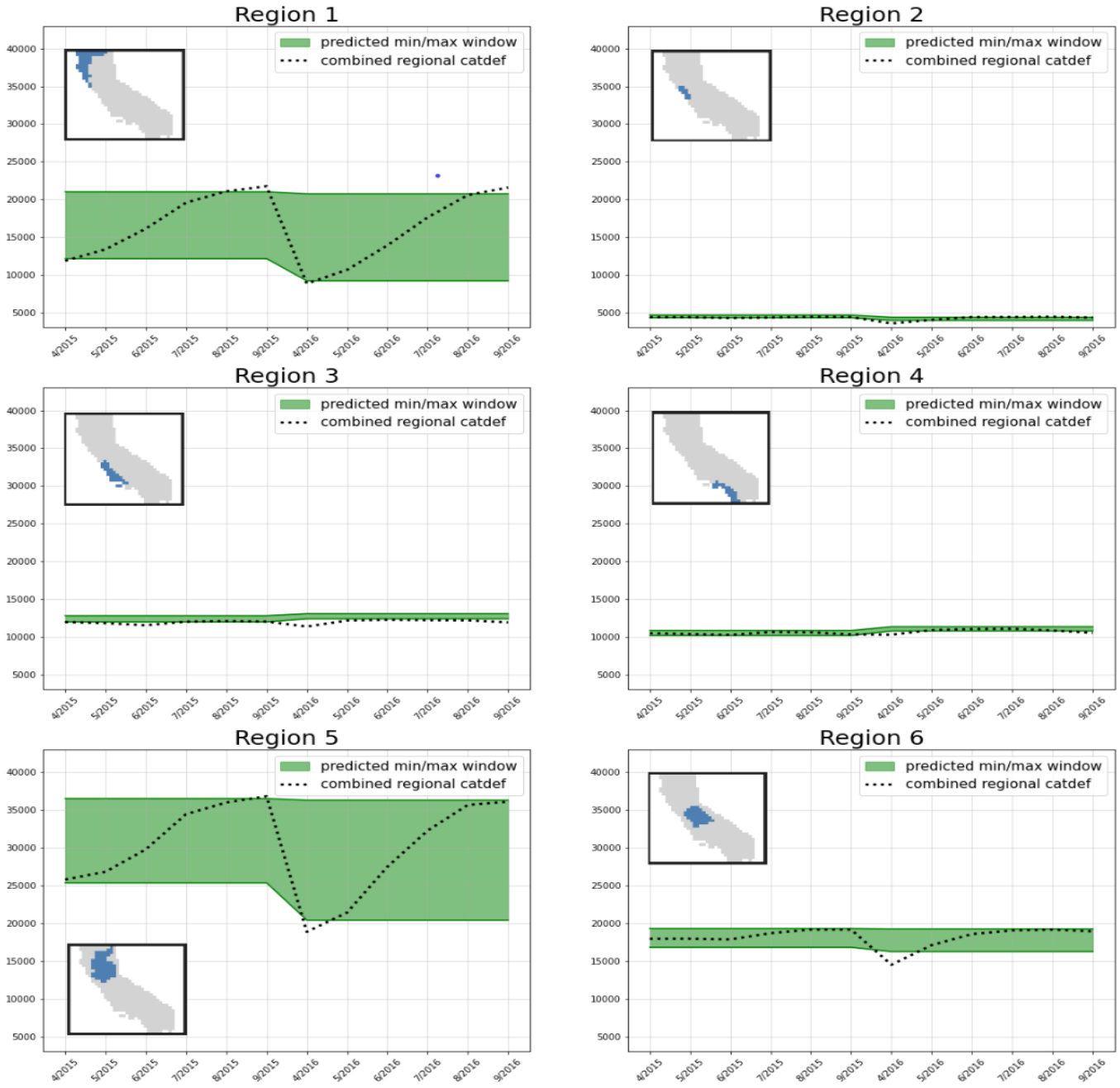


FIG. 16. Catdef minimum/maximum prediction envelope for each hydrologic region

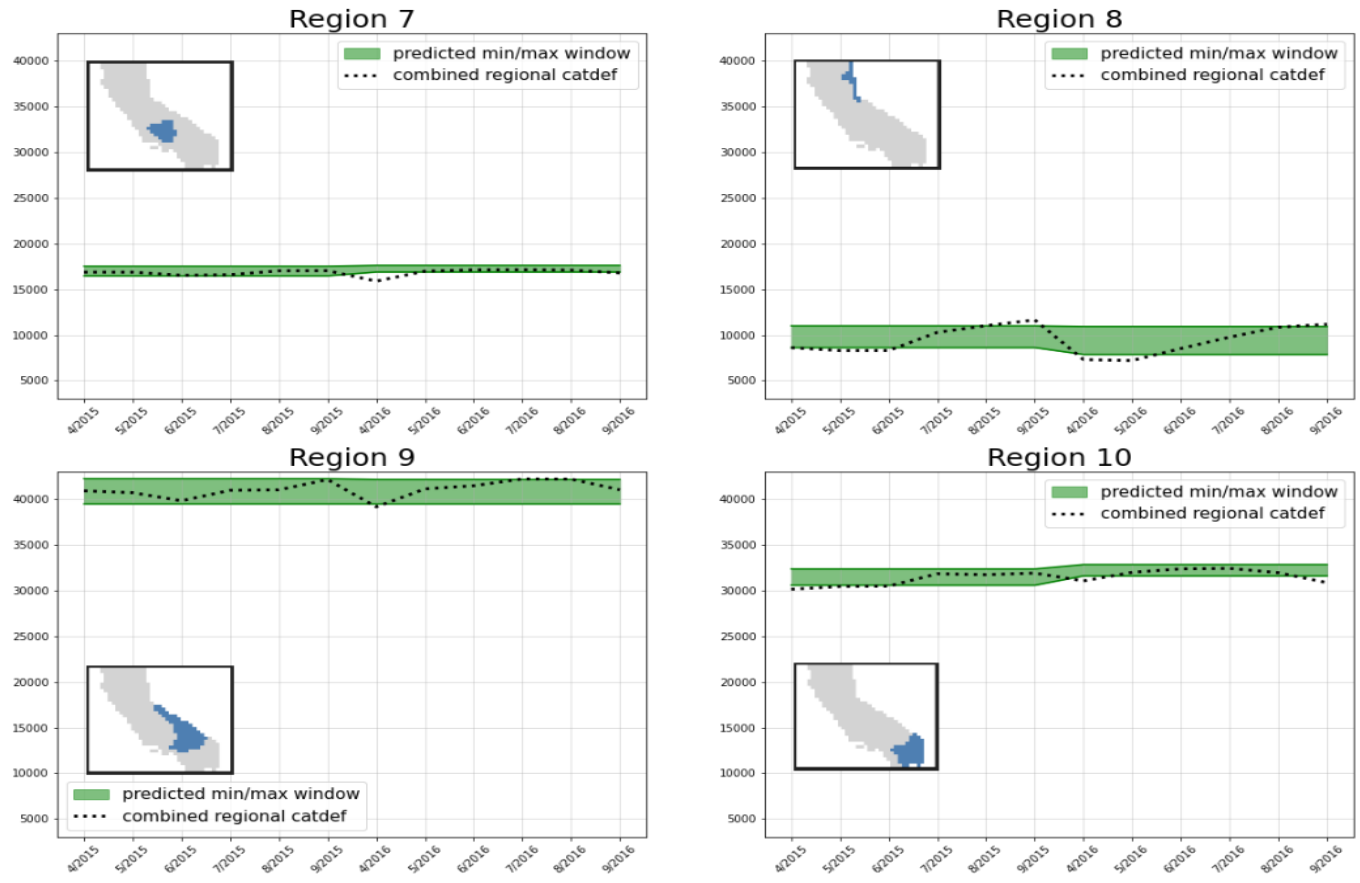


FIG. 16 (continued). Catdef minimum/maximum prediction envelope for each hydrologic region

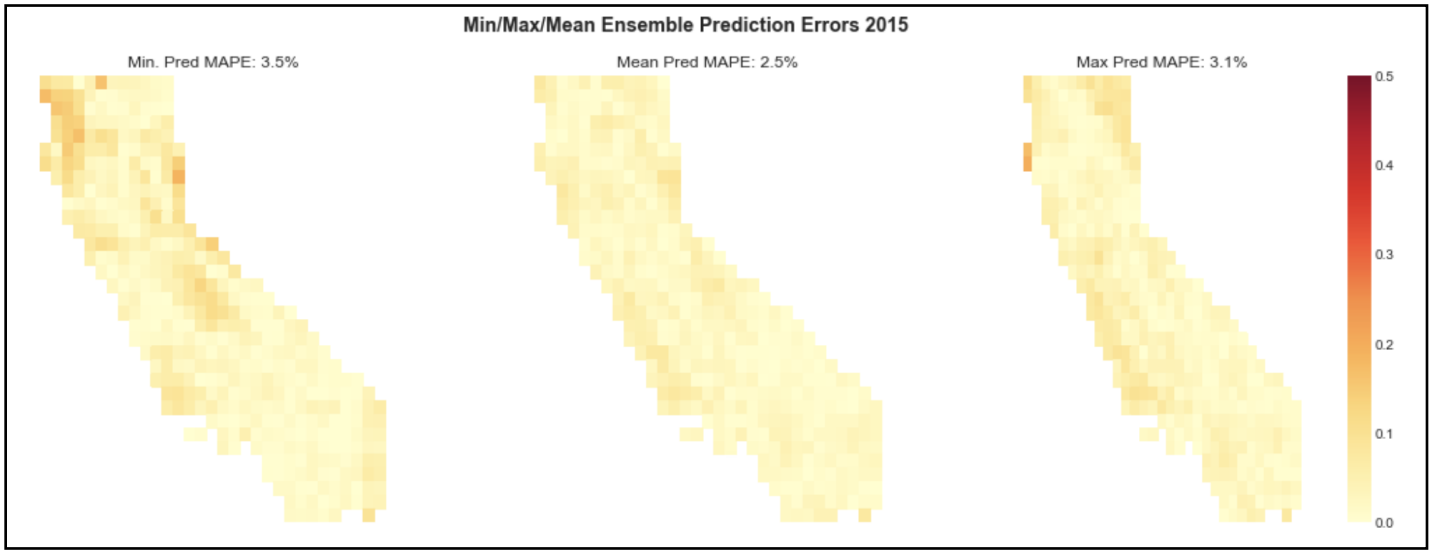


FIG. 17. Catdef minimum/maximum/mean prediction errors

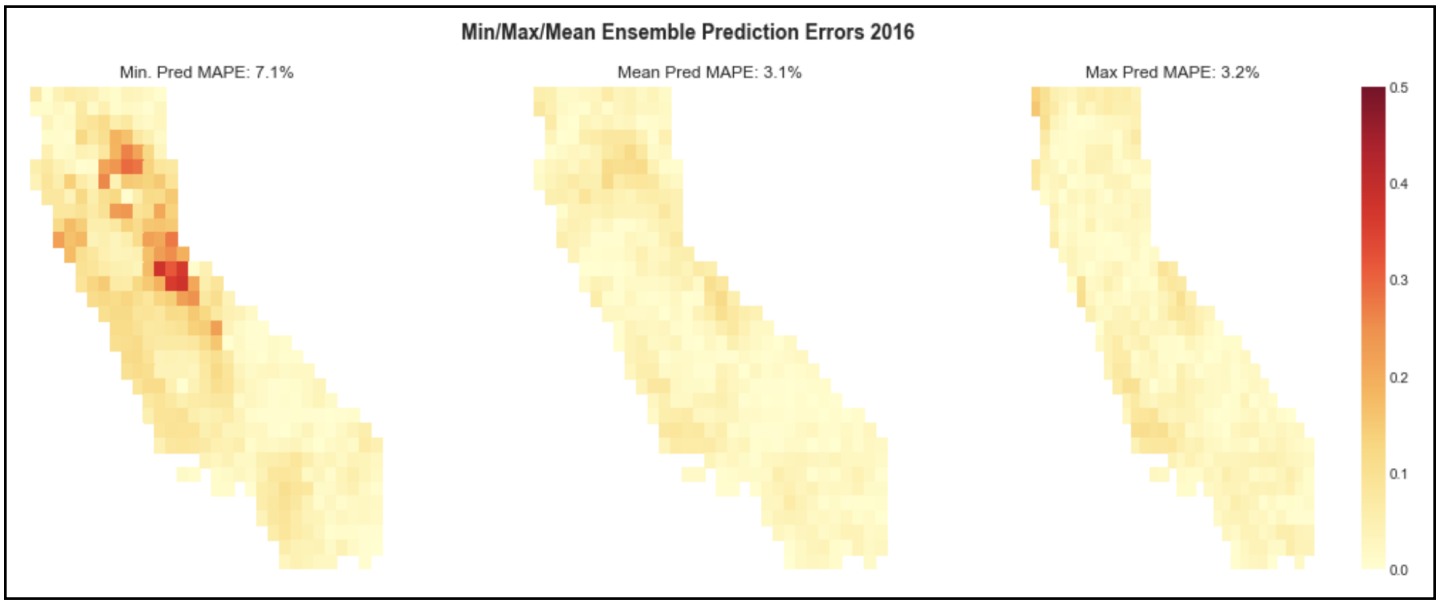


FIG. 17 (continued). Catdef minimum/maximum/mean prediction errors

V. Conclusion and Discussion

We presented results from three models to predict catchment deficit levels at 440 locations across the state of California. Two models (a time-series ARIMA model and a ConvLSTM model) each predicted the continuous value of catchment deficit. These predictions, in turn, were then submitted as features to a relatively simple linear model. Each of the component predictions, in general, produced results consistent with trends in the true value of the deficit. The final linear model then often resulted in an even better prediction.

Reviewing the MAPE plots reveals that MAPE trends down over time for both 2015 and 2016, despite a reasonable expectation that error would increase over time. For 2015, the maximum ensemble MAPE was in June (5.0%) and the minimum was in September(2.9%). In 2016, the maximum MAPE value was even earlier, in April, and the minimum value in August. In terms of geography, we recall that errors are greatest in the central valley where agriculture consumes large amounts of groundwater. We believe agricultural water use increases the variance in the deficit values. Reasoning similarly, we believe that the wet season and runoff melt enter into the water table earlier in the season, leading to possibly higher errors at that time in April and May. By late Summer and early Fall (August and September), the wet season is long past and any snowpack melt has ceased. We believe this decrease in precipitation contributes to a lower variability in the deficit, and in turn an improved error and lower MAPE value. We speculate that serial nature of the six linear models and their superior fit when defined on a monthly basis may have been aided by this larger trend in the water cycle.

Features of the final linear model are collinear. Their collinearity is not surprising, given that they both predict the same target value. With respect to the training data, the two predictions were minimally correlated in July with a coefficient of 0.87; and maximally in April with a coefficient of 0.98. With the exception of July, all the correlations were above 0.90 with both mean and median near 0.95. Collinearity in the features may not necessarily hurt the final estimates from the linear model, but it may contribute to the standard error of the coefficients. See 3.3.3.6 in James et al. 2021. Collinearity can also lead to OLS solutions with equal or near-equal loss but with different

coefficients; this is instability in the optimal coefficients. We present the coefficients in the previous section, but any interpretation should keep the collinearity of the features in mind.
We observed that the ConvLSTM model often underestimated the catdef values. To make the distribution of the residuals more balanced, perhaps the loss function could be adjusted. MAPE has a bias favoring estimates that are below the actual values (Armstrong 1985). Perhaps using an alternative loss function such as UMBRAE (Unscaled Mean Bounded Relative Absolute Error) (Chen and Garibaldi 2017) would address this undesirable trend. In addition, in cases where loss is the most extreme, explainability methods like "integrated gradients" (Sundararajan et al. 2017) could be applied and perhaps yield insight using an empty snowpack and rzmc across the input as a baseline to see what attributes of the input features lead to the underestimation. More broadly, the method could be applied to see which snowpack and rzmc features, their time, and place, lead to various deficit predictions across the state of California. Interestingly, the linear model final step perhaps acts as a kind of "correction" to account for prediction biases from the ConvLSTM model.

In this work, we used the CNRA data from several wells and the GRACE/GLDAS satellites. Future work might seek to incorporate additional data sources whose measurements are taken in situ. This is to avoid any indirect errors or noise that may be introduced when data are acquired from satellites. More data could help corroborate existing models and help zero in on ground truth measurements of ground water storage and aquifer levels.

We would like to gratefully acknowledge help and support from

Puya Vahabi, PhD
Alberto Todeschini, PhD
Manuela Girotto, PhD
Paolo D'Odorico, PhD
Alvar Escriva-Bou, PhD
Jonathan Wong
Randy Moran
Dean Hyunho Kang

VI. Appendix

ARIMA Models

ARIMA models have at their cores ARMA models. ARMA models, having no "I" are Auto-Regressive Moving Average models. ARMA models are characterized by weights on two kinds of features.

First, ARMA models regress values of the time series on past values of the time series. These weights are from "AR" models because they auto-regress one value of the time series on earlier values of the series. The number of non-zero weights is typically defined with the variable p as seen in equation 3.1.16 from Box et al. (2015) of a general AR model where the z values represent values of the time series with t indexing into current and past values here and for other variables, ϕ the weights, and a_t, a "white noise" term:

$$\tilde{z}_t = \phi_1 \tilde{z}_{t-1} + \phi_2 \tilde{z}_{t-2} + \cdots + \phi_p \tilde{z}_{t-p} + a_t$$

Second, the class of ARMA models also includes the class of so-called moving average (MA) models. "Rather than using past values of the forecast variable in a regression, a moving average model uses past forecast errors" (Hyndman and Khandakar 2008). Equation 3.1.18 from Box et al. (2015) exemplifies an MA(q) model:

$$\tilde{z}_t = a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \cdots - \theta_q a_{t-q}$$

where the Θ are the coefficients. The number of non-zero terms for MA models is often denoted as q. Here the a variables refer to the white noise terms, which can be considered not only as random deviations from zero but also as residuals when viewed in the context of previous values in the time series and previous predictions.

When AR and MA models are combined, they are called Auto-Regressive Moving Average Models. An example equation of an ARMA(p, q) model with p AR coefficients and q MA coefficients from Box et al. (2015) 3.1.12 is shown below:

$$\tilde{z}_1 = \phi_1 \tilde{z}_{t-1} + \cdots + \phi_p \tilde{z}_{t-p} + a_t - \theta_1 a_{t-1} - \cdots - \theta_q a_{t-q}$$

ARMA modeling requires that the time series data be "stationary". This means that the average value of the time series is a constant. This requirement not only helps to define boundaries on the ranges of the estimated coefficients, but helps any theoretical series considered during modeling not "explode" to infinity. To help bring about stationarity, data can be transformed. One example of a class of such a transformation is the class of Box-Cox transformations. As discussed in (Hyndman and Khandakar 2008) these are not unlike a log transformation. Besides Box-Cox transformations, additional pre-processing might include any trend removal, smoothing, or seasonal variation removal.

Another transformation is “differencing”. Differencing a time-series refers to the practice of creating a “differenced” series whose values are computed by taking the differences of subsequent values in a regular “undifferenced” series. For example, the series 1, 2, 3, 4, 5 would have as its differenced series 1, 1, 1, 1. This is because 2 - 1 = 1, 3 - 2 = 1, 4 - 3 = 1, and 5 - 4 = 1. Note that the differenced series has one less term. A subsequent difference operation would lead to the second difference series of 0, 0, 0 because 1 - 1 = 0, and of course one less term because of the additional differencing.

When differencing is considered part of the model building process, the number of times it is done is denoted by *d* and this combined with ARMA models can be used to refer to the ARIMA(p, d, q) class of models exemplified by equation 1.2.7 from Box et al., where the *w* terms refer to the differenced series (not the “raw, undifferenced series”) with *d* differencing operations.

$$w_t = \phi_1 w_{t-1} + \cdots \phi_p w_{t-p} + a_t - \theta_1 a_{t-1} - \cdots - \theta_q a_{t-q}$$

Estimation of the ϕ and Θ parameters is done by maximum likelihood analysis by invoking the normal distribution’s density function for the random noise in the model. The maximum likelihood analysis can be carried out in a “conditional” way using initial values for the first values of the time series. It can also be carried out in an “unconditional” way where no specific initial values are assumed. These details are explained in much greater detail in chapter 7 of Box et al.

Once coefficients have been estimated, forecasting can be done using the equation directly. For example consider a time series whose last two values are 1.278 and 0.540 and an ARIMA(1,1,1) model with AR and MA coefficients of 0.5081 and -0.8808 respectively. This way, the last value in the differenced series is -0.737. During the model’s estimation, the final residual was found to be -0.200. Given these values the prediction of the next value in the series can be computed as (0.5080*-0.737) + (-0.8808 * -0.200) + 0.54 = 0.342. Note that the 0.54, the last value in the time series, is added back to go from a “differenced space” back to the original “undifferenced space”. This is as also shown in the *R* snippet and output Figure A.1.

With an ARIMA model that has been estimated using data, one can compute AIC (*Akaike Information Criterion*), AIC-corrected (AIC_C), and BIC values from the model and data. These values can be used to compare separate ARIMA models, and ultimately for model selection.

AIC incorporates the likelihood of a model as well as the number of parameters. AIC is computed as $2k - 2\ln(L)$ where L is the maximized log likelihood of the fitted model and k the number of parameters ; this way, lower AIC values are preferred. Additionally, the AIC_C is is computed from the AIC, but by adding the ratio as seen in this formula:

$$AIC_c = AIC + \frac{2k^2+2k}{n-k-1}$$

where n is the sample size. The corrected AIC can be better for models with smaller sample sizes.

A more expansive ARIMA model, the seasonal ARIMA model (SARIMA) can address auto-regression, differencing, and moving averages at the seasonal levels as well. The seasonal parameters are calculated similar to the regular ones, but at seasonal scales and referred to with the capital letters P, D, and Q for the auto-regressive, differencing, and moving-average components.

```
set.seed(102)
ts.sim=arima.sim(n=100,list(ar=c(0.75),ma=c(-0.25)),sd=sqrt(1))
mam=auto.arima(ts.sim)
print("An estimated (1,1,1) ARIMA model:")
print(mam)
last_value=ts.sim[100]
next_to_last_value=ts.sim[99]
print(paste("The last two values of the time series : ",next_to_last_value," and ",last_value))
their_difference=last_value-next_to_last_value
print(paste("Their difference  :",their_difference))
last_residual=mam$residuals[100]
print(paste("The last residual :",last_residual))
ar_coef=as.numeric(mam$coef["ar1"])
ma_coef=as.numeric(mam$coef["ma1"])
manual_prediction=(ar_coef*their_difference)+(ma_coef*last_residual)+last_value
library_prediction=predict(mam,h=1)$pred[1]
print(paste("The manual prediction  : ",manual_prediction))
print(paste("The library prediction : ",library_prediction))
```

```
[1] "An estimated (1,1,1) ARIMA model:"
Series: ts.sim
ARIMA(1,1,1)

Coefficients:
         ar1      ma1
      0.5081  -0.8808
s.e.  0.1707   0.1190

sigma^2 = 1.121:  log likelihood = -145.44
AIC=296.89   AICc=297.14   BIC=304.67
[1] "The last two values of the time series :  1.27796466523913  and  0.540329669614839"
[1] "Their difference  : -0.737634995624294"
[1] "The last residual : -0.200062349103981"
[1] "The manual prediction  :  0.34177295533218"
[1] "The library prediction :  0.341772955332041"
```

FIG. A.1. ARIMA code and output

Convolutional LSTM Models

LSTM Definition

An LSTM is a type of recurrent neural network (RNN) that preserves long-term dependencies across a large number of time steps, thus creating a sort of persistent memory that is ideal for time-series forecasting applications.

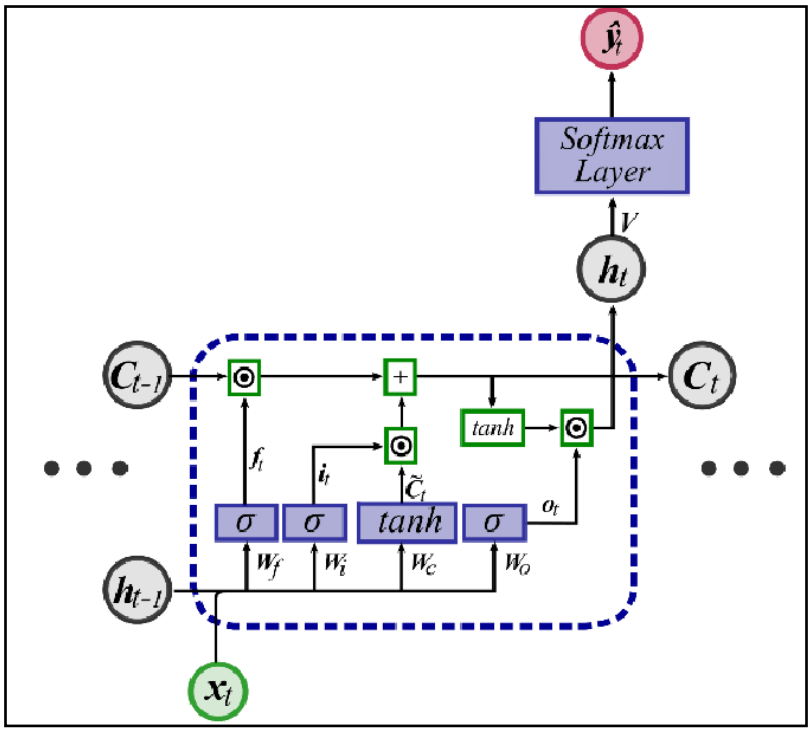


FIG. A.2. LSTM architecture

The idea behind LSTMs consists in controlling the information flow in and out of the network's memory cell $\boldsymbol{C}$ *by means of specialized gate units: forget, input and output. Actually, a gate is a sigmoid (*$\boldsymbol{\sigma}$*) neural network layer followed by a pointwise multiplication operator. Each gate is controlled by the concatenation of the network state at a previous time step* $\boldsymbol{h_{t-1}}$ *and the current input signal* $\boldsymbol{x_t}$ *. The forget gate decides what information will be discarded from the cell state* $\boldsymbol{C}$ *and the input gate what new information is going to be stored in it. The output gate determines the new state* $\boldsymbol{h_t}$ *. [The following equations] describe the internal operations carried out in a LSTM neural unit:*

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_f \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_f)$$

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_i \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_i)$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_o \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_o)$$

$$\boldsymbol{C}_t = \boldsymbol{f}_t * \boldsymbol{C}_{t-1} + \boldsymbol{i}_t * \widetilde{\boldsymbol{C}}_t$$

where,

$$\widetilde{\boldsymbol{C}}_t = tanh(\boldsymbol{W}_C \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_C)$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t * \tanh(\boldsymbol{C}_t)$$

where $\boldsymbol{W}_f$, $\boldsymbol{W}_i$, $\boldsymbol{W}_O$, and $\boldsymbol{W}C$ and $\boldsymbol{b}_f$, $\boldsymbol{b}_i$, $\boldsymbol{b}_O$, and $\boldsymbol{b}_C$ are weight matrices and bias vectors, respectively, to be learned by the network during training (Castro et. al. 2017).

Convolutional LSTM Definition

Put simply, a convolutional LSTM is an LSTM with convolutional input and recurrent transformations. From the seminal paper *Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting* (Shi et al. 2015):

> *The major drawback of FC-LSTM in handling spatiotemporal data is its usage of full connections in input-to-state and state-to-state transitions in which no spatial information is encoded. To overcome this problem, a distinguishing feature of our design is that all the inputs X1, . . . , Xt, cell outputs C1, . . . , Ct, hidden states H1, . . . , Ht, and gates it, ft, ot of the ConvLSTM are 3D tensors whose last two dimensions are spatial dimensions (rows and columns). To get a better picture of the inputs and states, we may imagine them as vectors standing on a spatial grid. The ConvLSTM determines the future state of a certain cell in the grid by the inputs and past states of its local neighbors. This can easily be achieved by using a convolution operator in the state-to-state and input-to-state transitions. The key equations of ConvLSTM are shown … below, where '∗' denotes the convolution operator and '◦' … denotes the Hadamard product:*

$$it = \sigma(Wxi * Xt + Whi * Ht{-}1 + Wci \circ Ct{-}1 + bi)$$
$$ft = \sigma(Wxf * Xt + Whf * Ht{-}1 + Wcf \circ Ct{-}1 + bf)$$
$$Ct = ft \circ Ct{-}1 + it \circ \tanh(Wxc * Xt + Whc * Ht{-}1 + bc)$$
$$ot = \sigma(Wxo * Xt + Who * Ht{-}1 + Wco \circ Ct + bo)$$
$$Ht = ot \circ \tanh(Ct)$$

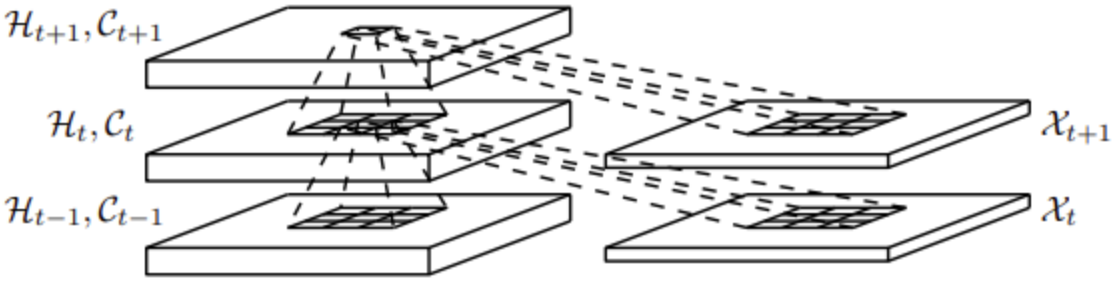


FIG. A.3. ConvLSTM inner structure

Regional Data Analysis

We explored the idea of modeling each hydrologic region independently by analyzing the relative variability of each region's aggregated catdef value across all sampled months, as well as the respective importance of each region's catdef in predicting the statewide catdef (using a linear model). See Figure A.4.

For each region, we trained four unique models using different feature sets: snowpack plus rzmc, snowpack plus srfexc, snowpack plus evap (evapotranspiration), and snowpack plus RainfC (rain from convection). The four features were taken from the GRACE dataset and chosen based on EDA performed by Wong et. al. These models were trained with feature data for the entire southwestern United States used by Wong et. al., but with catdef data for only the target region. We also trained four additional "base" models using the four permutations of feature sets and catdef data for the entire southwest area. Table A.2 shows the most accurate model (in terms of MAPE) for each region for each month in the test period. For each region, we chose the model that performed best in the highest number of months to include in our ensemble.

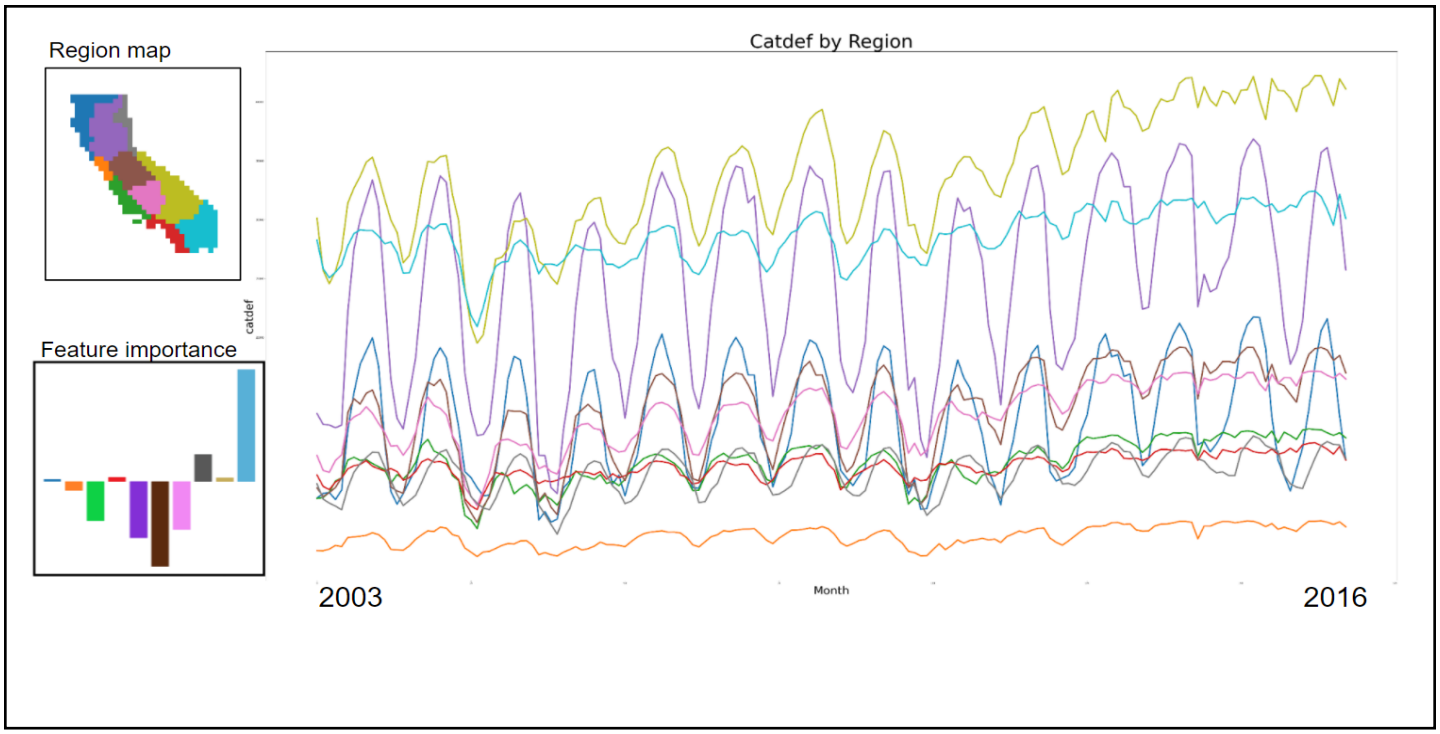


FIG. A.4. LSTM architecture

TABLE. A.2. Best model configuration for each test month

Region	Month 123	Month 124	Month 125	Month 126	Month 127	Month 128	Month 129	Month 130	Month 131	Month 132	Month 133	Month 134	Month 135	Month 136	Month 137	Month 138	Month 139	Month 140	Month 141	Month 142	Month 143	Month 144	Month 145	Month 146
1	base, evap	base, srfexc	r1, rzmc	r1, rzmc	base, evap	base, evap	base, srfexc	base, srfexc	r1, srfexc	r1, evap	r1, rzmc	base, rzmc	base, rzmc	r1, rzmc	base, srfexc	r1, rzmc	r1, rzmc	base, evap	r1, rzmc	r1, srfexc	r1, RainfC	base, evap	r1, RainfC	r1, rzmc
2	base, evap	base, srfexc	r2, srfexc	r2, rzmc	r2, srfexc	r2, rzmc	r2, srfexc	r2, rzmc	base, evap	base, srfexc	r2, srfexc	base, srfexc	r2, srfexc	r2, srfexc	r2, srfexc	r2, srfexc	r2, rzmc	r2, rzmc	r2, rzmc	r2, rzmc	base, rzmc	r2, srfexc	base, srfexc	base, srfexc
3	r3, srfexc	r3, srfexc	r3, srfexc	r3, evap	r3, rzmc	r3, rzmc	r3, srfexc	r3, srfexc	r3, srfexc	r3, srfexc	r3, srfexc	r3, evap	r3, srfexc	r3, srfexc	r3, srfexc	r3, srfexc	r3, rzmc	r3, rzmc	r3, evap	r3, srfexc	base, srfexc	r3, srfexc	r3, srfexc	r3, srfexc
4	base, srfexc	r4, srfexc	r4, srfexc	r4, evap	r4, srfexc	base, srfexc	base, srfexc	base, srfexc	base, srfexc	base, srfexc	base, srfexc	base, srfexc	base, srfexc	base, srfexc	r4, srfexc	r4, srfexc	r4, srfexc	r4, srfexc	base, srfexc	base, srfexc	base, srfexc	base, srfexc	base, srfexc	r4, srfexc
5	base, evap	base, srfexc	base, srfexc	r5, evap	r5, rzmc	r5, RainfC	base, evap	r5, rzmc	r5, evap	base, srfexc	base, srfexc	r5, rzmc	r5, rzmc	base, srfexc	r5, srfexc	r5, rzmc	r5, rzmc	r5, RainfC	r5, rzmc	r5, srfexc	r5, RainfC	r5, evap	r5, rzmc	r5, evap
6	r6, srfexc	r6, srfexc	r6, srfexc	r6, evap	r6, rzmc	r6, rzmc	r6, evap	r6, rzmc	base, evap	base, srfexc	r6, srfexc	r6, srfexc	r6, evap	r6, evap	r6, srfexc	r6, rzmc	r6, rzmc	r6, rzmc	r6, rzmc	r6, rzmc	r6, srfexc	base, srfexc	r6, srfexc	r6, srfexc
7	r7, srfexc	r7, srfexc	r7, srfexc	r7, evap	r7, evap	r7, rzmc	base, srfexc	base, srfexc	r7, srfexc	r7, srfexc	r7, srfexc	base, srfexc	base, srfexc	base, srfexc	r7, srfexc	r7, rzmc	r7, rzmc	base, evap	r7, evap	r7, srfexc	r7, srfexc	base, srfexc	base, srfexc	r7, srfexc
8	r8, srfexc	r8, srfexc	base, srfexc	base, evap	r8, srfexc	r8, srfexc	base, srfexc	base, srfexc	r8, srfexc	base, srfexc	base, srfexc	r8, RainfC	base, rzmc	base, srfexc	r8, srfexc	r8, RainfC	r8, srfexc	r8, RainfC	r8, srfexc	r8, srfexc	base, srfexc	base, srfexc	base, srfexc	base, srfexc
9	base, srfexc	r9, srfexc	base, srfexc	r9, rzmc	r9, rzmc	base, srfexc	base, srfexc	base, srfexc	r9, srfexc	base, srfexc	r9, srfexc	base, srfexc	r9, srfexc	base, srfexc	base, srfexc	r9, rzmc	r9, rzmc	r9, rzmc	r9, rzmc	r9, srfexc	r9, srfexc	base, srfexc	r9, srfexc	r9, srfexc
10	base, srfexc	base, srfexc	base, srfexc	r10, srfexc	r10, srfexc	r10, srfexc	r10, srfexc	r10, srfexc	base, srfexc	base, srfexc	r10, srfexc	base, srfexc	base, srfexc	base, srfexc	r10, srfexc	r10, srfexc	r10, srfexc	r10, srfexc	base, srfexc	r10, srfexc	r10, srfexc	base, srfexc	base, srfexc	r10, srfexc

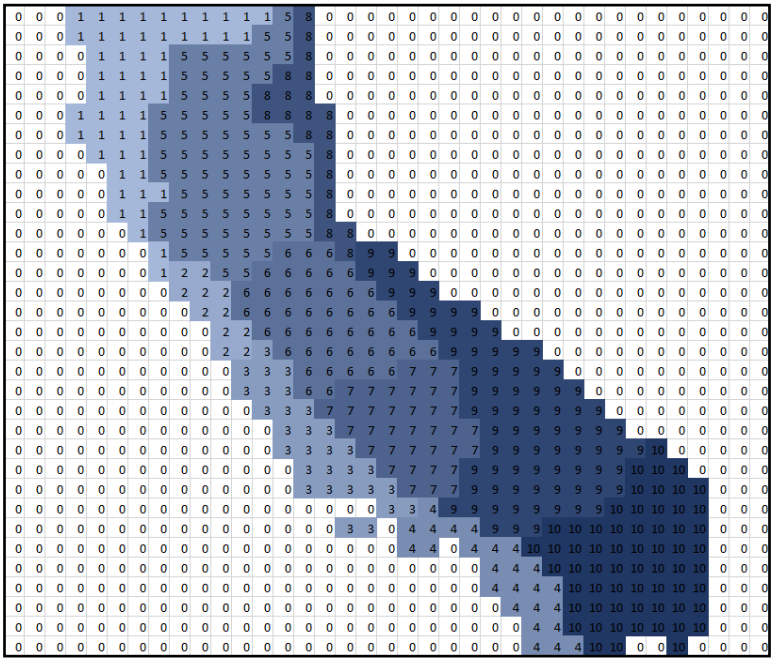


FIG. A.5. Regional catdef data was produced by "masking" grid locations in the 33x37 catdef data, so that all values not part of the specified region were set to null.

Summary Figures for Ensemble, ConvLSTM, and ARIMA Envelopes

The following figures illustrate relative performance of prediction envelopes for each of the ten hydrologic regions, for each of the three models.

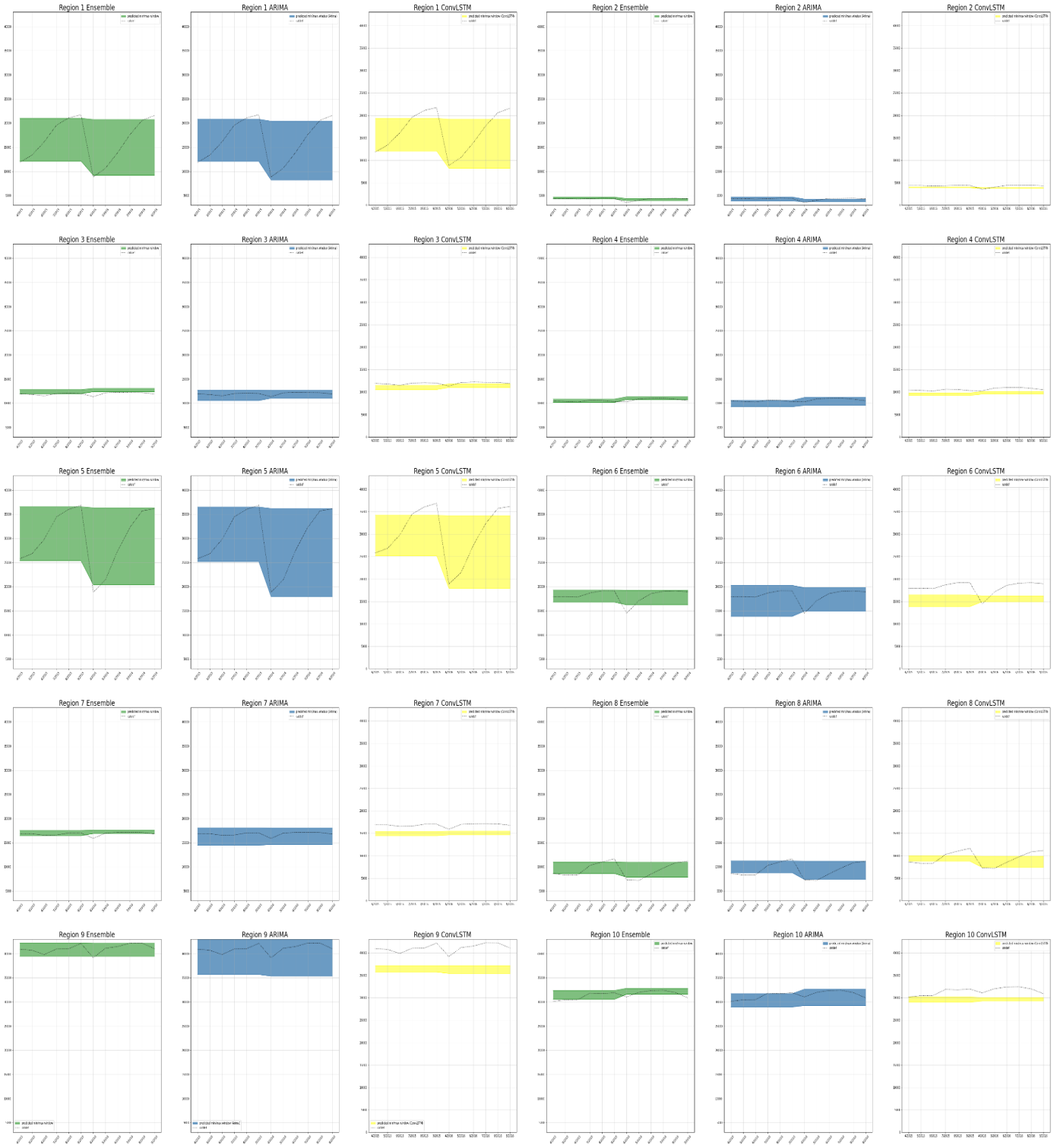


FIG. A.6. Catdef prediction envelopes for ARIMA, ConvLSTM, and Ensemble models

VIII REFERENCES

Ahamed, A., et al. (2022) "Assessing the utility of remote sensing data to accurately estimate changes in groundwater storage." Science of The Total Environment 807 : 150635.

Armstrong JS. (1985) Measures of Accuracy In: Long-Range Forecasting: From Crystal Ball to Computer. A Wiley-Interscience Publication. Wiley; 1985. p. 346–354

Bathke, D., and Riganti, C. (2019) "US Drought Monitor." https://droughtmonitor.unl.edu/CurrentMap/StateDroughtMonitor.aspx?CA

Beaudoing, H., Rodell, M., Getirana, A., Li, B., NASA/GSFC/HSL (2021) Groundwater and Soil Moisture Conditions from GRACE Data Assimilation L4 7-days 0.125 x 0.125 degree V4.0, Greenbelt, MD, USA, Goddard Earth Sciences Data and Information Services Center (GES DISC), 10.5067/UH653SEZR9VQ

Box G.E.P., Jenkins G., Reinsel, G., LJung G. (2015) Time Series Analysis : Forecasting and Control Wiley; 5th edition (June 29, 2015)

CNRA (2022) “Periodic Groundwater Level Measurements”. California Natural Resources Agency. https://data.cnra.ca.gov/dataset/periodic-groundwater-level-measurements

Castro, J., Achanccaray Diaz, P., Sanches, I., Cue La Rosa, L., Nigri Happ, P., Feitosa, R. (2017). EVALUATION OF RECURRENT NEURAL NETWORKS FOR CROP RECOGNITION FROM MULTITEMPORAL REMOTE SENSING IMAGES at XXVII Congresso Brasileiro de Cartografia e XXVI Exposicarta. Retrieved from https://www.researchgate.net/publication/328761192_EVALUATION_OF_RECURRENT_NEURAL_NETWORKS_FOR_CROP_RECOGNITION_FROM_MULTITEMPORAL_REMOTE_SENSING_IMAGES

Chen C, Twycross J, Garibaldi JM. (2017) A new accuracy measure based on bounded relative error for time series forecasting. PLoS One. 2017;12(3):e0174202. Published 2017 Mar 24. doi:10.1371/journal.pone.0174202

Cryer JD., Chan Kung-Sik (2018) Time Series Analysis: With Applications in R (Springer Texts in Statistics) Springer; 2nd edition (April 4, 2008)

Escriva-Bou, A., et al. (2016) "Accounting for California water." California Journal of Politics and Policy 8.3

Funk, C., et al. (2015) "The climate hazards infrared precipitation with stations—a new environmental record for monitoring extremes." Scientific data 2.1 (2015): 1-21.

Girotto, M., et al. (2016) "Assimilation of gridded terrestrial water storage observations from GRACE into a land surface model." Water Resources Research 52.5 (2016): 4164-4183.

Hyndman, R.J. and Athanasopoulos G.(2018) Forecasting: Principles and Practice, OTexts; 2nd edition (August 4, 2018)

Hyndman, R. J., and Khandakar, Y. (2008). Automatic Time Series Forecasting: The forecast Package for R. Journal of Statistical Software, 27(3), 1–22. https://doi.org/10.18637/jss.v027.i03

James G., Witten D., Hastie T., and Tibshirani R. (2021) An Introduction to Statistical Learning, with Applications in R, Second Ed., Springer

Li, B., Beaudoing, H., Rodell, M., NASA/GSFC/HSL (2020) GLDAS Catchment Land Surface Model L4 monthly 1.0 x 1.0 degree V2.1, Greenbelt, Maryland, USA, Goddard Earth Sciences Data and Information Services Center (GES DISC), 10.5067/FOUXNLXFAZNY

Maier, HR., et al. (2010) "Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions." Environmental modelling & software 25.8 (2010): 891-909.

Mann, M., and Gleick, P. (2015) "Climate change and California drought in the 21st century." Proceedings of the National Academy of Sciences 112.13 (2015): 3858-3859.

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

Wong, J., Kang, D., and Moran, R. (2021) Predicting Total Discharge into California’s Central Valley from the Sierra Nevada Mountains, UC Berkeley MIDS Capstone Projects: Fall 2021. Retrieved Jan 18, 2022 from https://www.ischool.berkeley.edu/sites/default/files/sproject_attachments/predicting_the_total_discharge_from_snowpack_data.pdf

Shi, X., Chen, Z., Wang, H., Yeung, D., Wong, W., Woo, W. (2015) Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. NIPS 2015: 802-810. Retrieved Jan 18, 2022 from https://arxiv.org/abs/1506.04214v2

Yin, J., et al. (2021) "Bayesian machine learning ensemble approach to quantify model uncertainty in predicting groundwater storage change." Science of The Total Environment 769 (2021): 144715.