Toward an Optimal Combined Load Forecast for System Operations

Authored by

Dr. Frank A. Monforte
Itron, Inc.
12348 High Bluff Drive
Suite 200
San Diego, CA 92130
(858) 724-2681

February 24, 2020

TABLE OF CONTENTS

INTRODUCTION

Deep penetration of non-grid connected renewable generation and storage, electric vehicle charging, smart load control, and time-of-use rates create greater load volatility, which in turn leads to eroding operational load forecast performance. To improve the system operator's confidence with the load forecasting process, there has been a movement toward developing and presenting an ensemble of load forecasts. The ensemble could include forecasts designed to handle the impact of rooftop solar PV and electric vehicle charging, forecasts that incorporate the impact of Time of Use (TOU) pricing and smart load control, and load forecasts produced under alternative weather forecasts. If the alternative load forecasts are clustered closely around each other, then system operations may have greater confidence in the system conditions predicted by the ensemble. On the other, a forecast ensemble with a wide range could raise doubts about the forecasted system conditions leading to system operators taking actions to hedge against the worst-case scenario. In effect, the forecast ensemble quantifies the plausible range of loads given uncertainty around future meteorological conditions such as temperatures, wind, and solar conditions, as well as uncertainty around price sensitive loads and load control actions.

Within this new world of ensemble forecasting there remains the reality that most downstream applications (e.g., transmission and distribution energy management systems and market models) require a single load forecast as an input. This means the load forecasting process needs a way of combing the alternative forecasts into a single "optimal" forecast that is then used for downstream processing. This paper provides a high-level review of some of the econometric/operations research and data science literature on combining forecasts and puts forth a recommendation for how to develop an optimal forecast specific to the problem of operational load forecasting.

ECONOMETRIC AND OPERATIONS RESEARCH APPROACHES

Much of the econometric and operations research literature on combining forecasts starts with the seminal paper *The Combination of Forecasts*, by J. M. Bates and C. W. J. Granger, source Operational Research Society, Vol. 20, No. 4 (Dec. 1969), pp.451-469. In their paper, Bates and Granger use two independently produced forecasts of passenger miles flown prepared by G. A. Barnard in his paper, *New Methods of Quality Control*, Journal of the Royal Statistical Society, Series A, Vol. 126 No. 2 (1963), pp. 255-258 to demonstrate the efficacy of alternative combination methods.[1]

Bates and Granger start their analysis by observing that the Mean Squared Forecast Error (MSE) from a third forecast formed by taking a simple average of the Adaptive and Box-Jenkins forecasts produced by Barnard is smaller than the MSEs of the original two forecasts. They then asked, given that a 50/50 weighting of the two independently generated forecasts improves the overall forecast performance, is it possible to define a method for computing an "optimal" weighting scheme that leads to a combined forecast with the smallest MSE?

Bates and Granger begin by considering the class of linear combinations of alternative forecasts. For example, let there be two alternative forecasts labelled A and B. Further, let time be indexed by t with

[1]See the Appendix for G. A. Barnard's data.

(T) indicating the time at which a new combined forecast is produced. We can compute a combined forecast as:

$$\mathrm{CombinedForecast}_{\mathrm{T}}^{\mathrm{C}} = \mathrm{Weight}^{\mathrm{A}}\ \mathrm{Forecast}_{\mathrm{T}}^{\mathrm{A}} + \mathrm{Weight}^{\mathrm{B}}\ \mathrm{Forecast}_{\mathrm{T}}^{\mathrm{B}}$$

If the weights are constrained to sum to 1.0, this equation can be re-written as:

$$\mathrm{CombinedForecast}_{\mathrm{T}}^{\mathrm{C}} = \mathrm{Weight}^{\mathrm{A}}\ \mathrm{Forecast}_{\mathrm{T}}^{\mathrm{A}} + \left(1 - \mathrm{Weight}^{\mathrm{A}}\right)\mathrm{Forecast}_{\mathrm{T}}^{\mathrm{B}}$$

Bates and Granger define an “optimal” weighting scheme as the set of weights that minimize the variance of the combined forecast. For the above example, the variance of the combined forecast errors is written as:

$$\sigma_{\mathrm{C}}^{2} = \left(\mathrm{Weigth}^{\mathrm{A}}\right)^{2}\sigma_{\mathrm{A}}^{2} + \left(1 - \mathrm{Weigth}^{\mathrm{A}}\right)^{2}\sigma_{\mathrm{B}}^{2} + 2\rho\mathrm{Weight}^{\mathrm{A}}\,\sigma_{\mathrm{A}}\left(1 - \mathrm{Weight}^{\mathrm{A}}\right)\sigma_{\mathrm{B}}$$

Here,

- σ_{C}^{2} is the forecast error variance of the combined (C) forecast
- σ_{A}^{2} is the forecast error variance of the Forecast A forecast
- σ_{B}^{2} is the forecast error variance of the Forecast B forecast
- ρ is the correlation coefficient between the Forecast A and B forecast errors

The optimal weight is derived by taking the derivative of the above equation with respect to $\mathrm{Weight}^{\mathrm{A}}$ and setting the derivative equal to 0. This results in the following formula for computing the optimal weights.

$$\mathrm{Weight}^{\mathrm{A}} = \frac{\sigma_{\mathrm{B}}^{2} - \rho\sigma_{\mathrm{A}}\,\sigma_{\mathrm{B}}}{\sigma_{\mathrm{A}}^{2} + \sigma_{\mathrm{B}}^{2} - \rho\sigma_{\mathrm{A}}\,\sigma_{\mathrm{B}}}$$

That is, the optimal Weight for Forecast A is proportional to the Forecast B forecast error variance less the covariance of the Forecast A and Forecast B errors. If the Forecast A and Forecast B errors are independent (i.e. $\rho = 0$), then the formula for the optimal weight reduces to:

$$\mathrm{Weight}^{\mathrm{A}} = \frac{\sigma_{\mathrm{B}}^{2}}{\sigma_{\mathrm{A}}^{2} + \sigma_{\mathrm{B}}^{2}}$$

In this case, the weight placed on Forecast A depends on the Forecast B forecast error variance. For example, if the Forecast B forecast error variance is twice the size of the Forecast A forecast error variance, the optimal weights would be 2/3 on Forecast A and 1/3 on Forecast B.

The next step in their analysis is the recognition that σ_{A}^{2} and σ_{B}^{2} are unknown. They address this challenge by positing five different methods for estimating these forecast error variances. These methods are summarized below.

Bates and Granger Method 1. Under this method, the Weights are proportional to the sum of the prior forecast squared errors. Specifically,

$$\sigma^2_{B,T} \approx \sum_{t=T-v}^{T-1} (\mathrm{ForecastError}^B_t)^2$$

And

$$\sigma^2_{A,T} \approx \sum_{t=T-v}^{T-1} (\mathrm{ForecastError}^A_t)^2$$

Here, v determines how many historical forecasts are considered when determining the weights. Ideally, v should cover the forecasts developed using the existing forecast models/approaches. If the model or approach is fundamentally altered or a new method/model is introduced, v should be adjusted to capture the performance of the current set of models/methods.

With these approximations, the weight placed on Forecast A at time (T) is computed as:

$$\mathrm{Weight}^A_T = \frac{\sigma^2_{B,T}}{\sigma^2_{A,T} + \sigma^2_{B,T}}$$

And, the weight placed on Forecast B is computed as:

$$\mathrm{Weight}^B_T = 1 - \mathrm{Weight}^A_T$$

The combined forecast at time (T) is then computed as:

$$\mathrm{CombinedForecast}_T = \mathrm{Weight}^A_T \mathrm{Forecast}^A_T + \mathrm{Weight}^B_T \mathrm{Forecast}^B_T$$

It is important to note that the weights can change from one forecast iteration to the next based on the most recent rolling sum of the squared forecast errors. It is also important to note that the weights depend only on historical forecast performance.

Bates and Granger Method 2. This method applies a simple learning scheme to allow the weights to evolve over time instead of reacting instantly to the most recent forecast errors. The weights are computed as:

$$\mathrm{Weight}_{\mathrm{T}}^{\mathrm{A}} = \alpha \mathrm{Weight}_{\mathrm{T}-1}^{\mathrm{A}} + (1-\alpha)\left[\frac{\sigma_{\mathrm{B,T}}^{2}}{\sigma_{\mathrm{A,T}}^{2} + \sigma_{\mathrm{B,T}}^{2}}\right]$$

Here,

> α is the learning rate and is constrained to be $0 \leq \alpha \leq 1$. Large values of α lead to slower evolving weights than small values. At the extremes:
>
> - when $\alpha = 1$, the weights are fixed at their initialized values (for example, 50/50), and
> - when $\alpha = 0$, the weights are the same as the Method 1 weights.

The weight placed on Forecast B is computed in the same way as Method 1, specifically:

$$\mathrm{Weight}_{\mathrm{T}}^{\mathrm{B}} = 1 - \mathrm{Weight}_{\mathrm{T}}^{\mathrm{A}}$$

The combined forecast at time (T) is then computed as:

$$\mathrm{CombinedForecast}_{\mathrm{T}} = \mathrm{Weight}_{\mathrm{T}}^{\mathrm{A}}\mathrm{Forecast}_{\mathrm{T}}^{\mathrm{A}} + \mathrm{Weight}_{\mathrm{T}}^{\mathrm{B}}\mathrm{Forecast}_{\mathrm{T}}^{\mathrm{B}}$$

If the relative forecast performance of the two models remains steady, then the weights from Method 1 and 2 should converge over time.

Bates and Granger Method 3. This method places higher value on recent forecast performance. Under this method, the forecast error variances are approximated by the weighted sum of the prior forecast squared errors. Specifically,

$$\mathrm{s}_{\mathrm{B,T}}^{2} \approx \sum_{\mathrm{t=T-v}}^{\mathrm{T-1}} \omega^{t}\ (\mathrm{ForecastError}_{\mathrm{t}}^{\mathrm{B}})^{2}$$

And

$$\mathrm{s}_{\mathrm{A,T}}^{2} \approx \sum_{\mathrm{t=T-v}}^{\mathrm{T-1}} \omega^{t}\ (\mathrm{ForecastError}_{\mathrm{t}}^{\mathrm{A}})^{2}$$

Here, with $(\omega > 1)$, greater weight is placed on recent forecast performance. Note that (t) runs from a small number for the most distant forecasts to a large number for the most recent forecast. Method 3 is like Method 1 with the exception that greater emphasis is placed on recent forecast performance. We anticipate Method 3 weights would mover quicker with respect to recent trends in forecast performance than either Method 1 or Method 2.

The weight for Forecast A is defined as:

$$\mathrm{Weight}_{\mathrm{T}}^{\mathrm{A}} = \frac{\mathrm{s}_{\mathrm{B,T}}^{2}}{\mathrm{s}_{\mathrm{A,T}}^{2} + \mathrm{s}_{\mathrm{B,T}}^{2}}$$

And, the weight placed on Forecast B is computed as:

$$\mathrm{Weight}_{\mathrm{T}}^{\mathrm{B}} = 1 - \mathrm{Weight}_{\mathrm{T}}^{\mathrm{A}}$$

The combined forecast at time (T) is then computed as:

$$\mathrm{CombinedForecast}_{\mathrm{T}} = \mathrm{Weight}_{\mathrm{T}}^{\mathrm{A}}\mathrm{Forecast}_{\mathrm{T}}^{\mathrm{A}} + \mathrm{Weight}_{\mathrm{T}}^{\mathrm{B}}\mathrm{Forecast}_{\mathrm{T}}^{\mathrm{B}}$$

Like the first two methods, this method is driven only by historical forecast performance.

Bates and Granger Method 4. Unlike the first three methods, Method 4 does not assume that the Forecast A and B forecast errors are independent. Under this method the forecast error variances and forecast error covariance are approximated as follows:

$$\mathrm{s}_{\mathrm{B,T}}^{2} \approx \sum_{\mathrm{t=T-v}}^{\mathrm{T-1}} \omega^{t}\ (\mathrm{ForecastError}_{\mathrm{t}}^{\mathrm{B}})^{2}$$

$$\mathrm{s}_{\mathrm{A,T}}^{2} \approx \sum_{\mathrm{t=T-v}}^{\mathrm{T-1}} \omega^{t}\ (\mathrm{ForecastError}_{\mathrm{t}}^{\mathrm{A}})^{2}$$

$$\mathrm{Covariance}_{\mathrm{T}}^{\mathrm{A,B}} \approx \sum_{\mathrm{t=T-v}}^{\mathrm{T-1}} \omega^{\mathrm{t}}\ \mathrm{ForecastError}_{\mathrm{t}}^{\mathrm{A}}\mathrm{ForecastError}_{\mathrm{t}}^{\mathrm{B}}$$

The weight for Forecast A is defined as:

$$\mathrm{Weight}_{\mathrm{T}}^{\mathrm{A}} = \frac{\mathrm{s}_{\mathrm{B,T}}^{2} - \mathrm{Covariance}_{\mathrm{T}}^{\mathrm{A,B}}}{\mathrm{s}_{\mathrm{A,T}}^{2} + \mathrm{s}_{\mathrm{B,T}}^{2} - 2\mathrm{Covariance}_{\mathrm{T}}^{\mathrm{A,B}}}$$

And, the weight placed on Forecast B is computed as:

$$\mathrm{Weight}_{\mathrm{T}}^{\mathrm{B}} = 1 - \mathrm{Weight}_{\mathrm{T}}^{\mathrm{A}}$$

The combined forecast at time (T) is then computed as:

$$\mathrm{CombinedForecast}_{\mathrm{T}} = \mathrm{Weight}_{\mathrm{T}}^{\mathrm{A}}\mathrm{Forecast}_{\mathrm{T}}^{\mathrm{A}} + \mathrm{Weight}_{\mathrm{T}}^{\mathrm{B}}\mathrm{Forecast}_{\mathrm{T}}^{\mathrm{B}}$$

Like the first three methods, this method is driven only by historical forecast performance but does not impose a constraint of independence. Because independence is not imposed, it is possible that the weights will be both positive and negative, depending on the magnitude of the forecast error covariance.

Bates and Granger Method 5. This method applies a simple learning scheme to use the most recent absolute forecast errors to determine the weight placed on each forecast. The weight placed on Forecast A is computed as:

$$\mathrm{Weight}_{\mathrm{T}}^{\mathrm{A}} = \alpha\mathrm{Weight}_{\mathrm{T-1}}^{\mathrm{A}} + (1-\alpha)\left[\frac{\left|\mathrm{ForecastError}_{\mathrm{T-1}}^{\mathrm{B}}\right|}{\left|\mathrm{ForecastError}_{\mathrm{T-1}}^{\mathrm{A}}\right| + \left|\mathrm{ForecastError}_{\mathrm{T-1}}^{\mathrm{B}}\right|}\right]$$

Here,

$\left|\mathrm{ForecastError}_{\mathrm{T-1}}^{\mathrm{B}}\right|$ is the absolute value of the Forecast B error at time (T-1) and
$\left|\mathrm{ForecastError}_{\mathrm{T-1}}^{\mathrm{A}}\right|$ is the absolute value of the Forecast A error at time (T-1).

The weight placed on Forecast B is computed as:

$$\mathrm{Weight}_{\mathrm{T}}^{\mathrm{B}} = 1 - \mathrm{Weight}_{\mathrm{T}}^{\mathrm{A}}$$

The combined forecast at time (T) is then computed as:

$$\text{CombinedForecast}_{\text{T}} = \text{Weight}_{\text{T}}^{\text{A}}\text{Forecast}_{\text{T}}^{\text{A}} + \text{Weight}_{\text{T}}^{\text{B}}\text{Forecast}_{\text{T}}^{\text{B}}$$

This last method has the appeal that the weights are driven solely by recent forecast errors (in absolute value), which feels like an autoregressive correction mechanism.

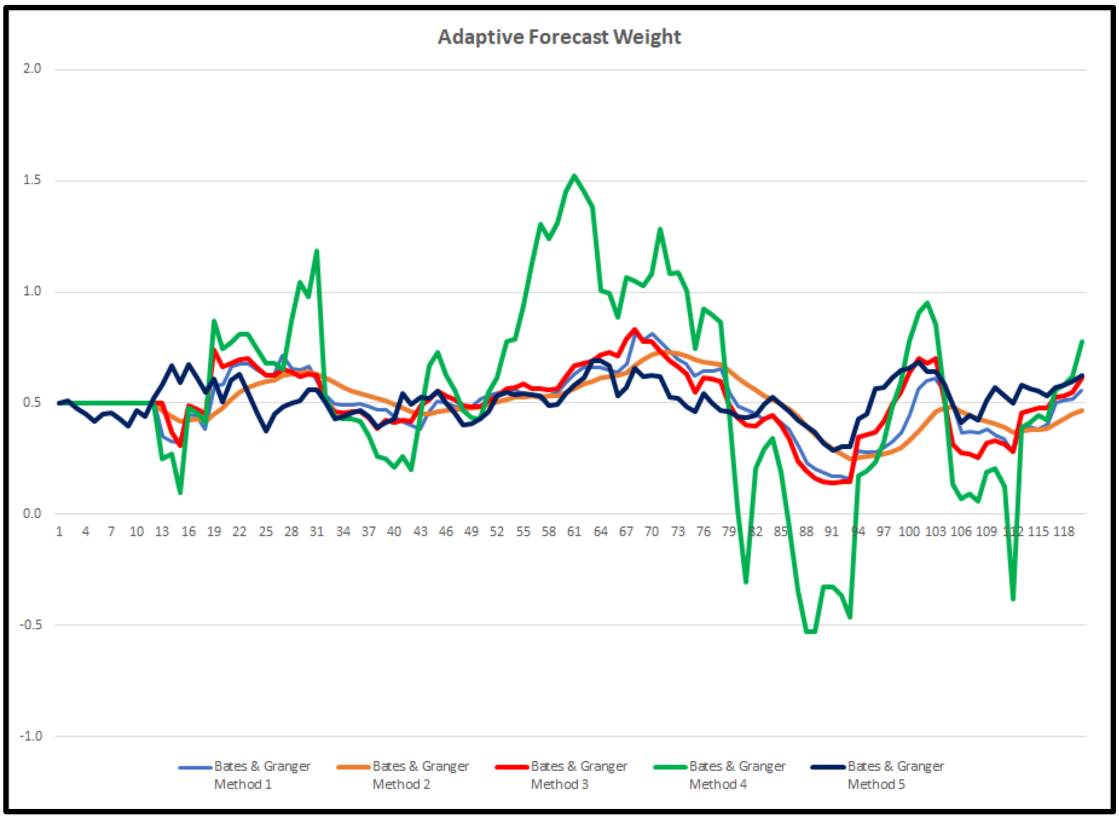
Bates and Granger Combining Forecast Summary. The table below presents forecast performance statistics for the original two forecasts, the simple combined forecast, and each of the Bates and Granger combination forecasts. Focusing on the MSE statistic, we see that, except for Method 4 which did away with the assumption of independence, each combined forecast outperforms both the Adaptive Forecast and the Box-Jenkins forecast. Method 3, which weights recent squared forecast errors heavier, performs the best with MSE of 128.9.

Shown in the figure are the estimated weights placed on the Adaptive Forecast. Method 2 (gold line) shows the smoothing advantage of using the learning framework. Method 4 (green line) shows how the weights will range significantly given the form of how the forecast covariance is estimated. For these data in general, Method 4 is not recommended.

FORECAST PERFORMANCE STATISTICS

	Mean Forecast Error	Median Forecast Error	Maximum Forecast Error	Minimum Forecast Error	Mean Squared Forecast Error	Skewness	Kurtosis
Adaptive Forecasting Forecast	0.3	1.0	40.0	-55.0	177.8	-0.6	2.7
Box-Jenkins Forecast	-0.1	-1.0	43.0	-35.0	148.6	0.1	0.9
Straight Average	0.1	-1.0	24.5	-44.0	130.4	-0.3	1.1
Bates and Granger Method 1	0.3	-0.6	32.0	-40.4	131.3	-0.1	0.9
Bates and Granger Method 2	0.1	-0.9	28.6	-41.6	132.8	-0.2	0.8
Bates and Granger Method 3	0.2	-0.7	32.1	-39.9	**128.9**	-0.1	1.0
Bates and Granger Method 4	0.4	-0.1	57.9	-49.8	162.4	0.2	4.5
Bates and Granger Method 5	0.1	-1.1	26.3	-44.7	130.3	-0.3	1.3

ESTIMATED ADAPTIVE FORECAST WEIGHT BY METHOD



Bunn Outperformance Method. Over the next three decades, several variations to the Bates and Granger methods were introduced with the goal of improving upon their initial work. In a 1975 paper, D. W. Bunn, *A Bayesian Approach to the Linear Combination Forecasts*, Operational Research Quarterly 26, 325-329, introduced an alternative combination method called Outperformance. In this paper, Bunn develops a measure of prior forecast performance that leads to weights that represent the probability of a forecast outperforming another forecast. This represents a third measure of forecast performance that goes beyond absolute and squared forecast errors.

The Outperformance method is defined as:

$$\mathrm{Weight}_T^A = \frac{\sum_{t=T-v}^{T-1}\left(\left|\mathrm{ForecastError}_t^B\right| \geq \left|\mathrm{ForecastError}_t^A\right|\right)}{\sum_{t=T-v}^{T} 1}$$

Here, the Weight placed on Forecast A at time (T) is the portion of time that the Forecast A error is less than or equal to in absolute value to the Forecast B error (i.e., Forecast A outperforms Forecast B). The logical expression in the numerator on the right-hand side of the above equation returns the number of times in the historical data range of T-v to T-1 that the Forecast A error is less than or equal to in absolute value to the Forecast B error. The denominator is the total number of forecasts evaluated.

In effect, the weight represents the probability that Forecast A will outperform Forecast B. For example, let's say 80% of the time over the past three weeks Forecast A outperforms Forecast B. As a result, it is expected that at time (T), Forecast A has an 80% chance of outperforming Forecast B.

The weight placed on Forecast B is computed as:

$$\mathrm{Weight}_{\mathrm{T}}^{\mathrm{B}} = 1 - \mathrm{Weight}_{\mathrm{T}}^{\mathrm{A}}$$

The combined forecast at time (T) is then computed as:

$$\mathrm{CombinedForecast}_{\mathrm{T}} = \mathrm{Weight}_{\mathrm{T}}^{\mathrm{A}}\mathrm{Forecast}_{\mathrm{T}}^{\mathrm{A}} + \mathrm{Weight}_{\mathrm{T}}^{\mathrm{B}}\mathrm{Forecast}_{\mathrm{T}}^{\mathrm{B}}$$

Like the Bates and Granger methods, the Outperformance method relies solely on historical forecast performance. However, the Bunn Outperformance Method is more direct in terms of predicting the relative performance of each forecast.

Regression Methods. In a 1984 paper, C. W. J. Granger and R. Ramanathan, *Improved Methods of Forecasting*, Journal of Forecasting 3, 197-204, introduced the use of linear regression as a means for optimally weighting the alternative forecasts. Under this approach, a combining regression for the Barnard miles flown data and forecasts can be written as follows:

$$\mathrm{PassengerFlownMiles}_{\mathrm{y,m}} = \beta_0 + \beta_1\ \mathrm{AdaptiveForecast}_{\mathrm{y,m}} + \beta_2\ \mathrm{BoxJenkinsForecast}_{\mathrm{y,m}} + \mathrm{e}_{\mathrm{y,m}}$$

In this general case, not suppressing the intercept in the above regression allows for the possibility that the raw forecasts, Adaptive Forecast and Box-Jenkins Forecast, could be biased. The estimated coefficient on the intercept term would then control for the bias.

How do these extensions perform in practice? Applying the Outperformance and Regression methods to the Bernard data leads to the following results. Based on the MSE, the Bunn Outperformance Method outperforms slightly all the Bates and Granger methods. The Granger and Ramanathan regression approach, at least for these data, leads to the best overall combined forecast.

	Mean Forecast Error	Median Forecast Error	Maximum Forecast Error	Minimum Forecast Error	Mean Squared Forecast Error	Skewness	Kurtosis
Adaptive Forecasting Forecast	0.3	1.0	40.0	-55.0	177.8	-0.6	2.7
Box-Jenkins Forecast	-0.1	-1.0	43.0	-35.0	148.6	0.1	0.9
Straight Average	0.1	-1.0	24.5	-44.0	130.4	-0.3	1.1
Bates and Granger Method 1	0.3	-0.6	32.0	-40.4	131.3	-0.1	0.9
Bates and Granger Method 2	0.1	-0.9	28.6	-41.6	132.8	-0.2	0.8
Bates and Granger Method 3	0.2	-0.7	32.1	-39.9	128.9	-0.1	1.0
Bates and Granger Method 4	0.4	-0.1	57.9	-49.8	162.4	0.2	4.5
Bates and Granger Method 5	0.1	-1.1	26.3	-44.7	130.3	-0.3	1.3
Bunn Outperformance Method	0.1	-0.8	25.1	-44.7	130.2	-0.3	1.3
Granger and Ramanathan Regression	0.0	-1.5	26.8	-42.6	**128.5**	-0.3	0.9

Seasonal Outperformance. All the combination weighting methods presented so far have relied on a rolling sum of prior period forecast errors to predict the next period forecast error. Presented below is a plot of the Barnard passenger miles flown data and forecasts. The obvious upward trend in passenger miles flown supports the idea of relying on recent forecast performance since the forecast errors made in the early 1950s are smaller in magnitude than the most recent errors.

The seasonal pattern in miles flown, however, suggests that heavy reliance on recent forecast performance could miss systematic seasonal forecast biases. For example, the forecast errors made in July may not be strong indicators of forecast errors in January. Further, one forecast approach may be better in the summer months while the other approach is better in the winter months.

Building upon Bunn's Outperformance Method, we use the historical forecast performance by Month as an a priori weight. We can rewrite Bunn's Outperformance weighting scheme as follows:

$$\mathrm{Weight}^{A}_{T,m} = \frac{\sum_{t=T-v}^{T-1}\left(\left|\mathrm{ForecastError}^{B}_{t}\right| \geq \left|\mathrm{ForecastError}^{A}_{t}\right|\right) \times (t = m)}{\sum_{t=T-v}^{T}(t = m)}$$

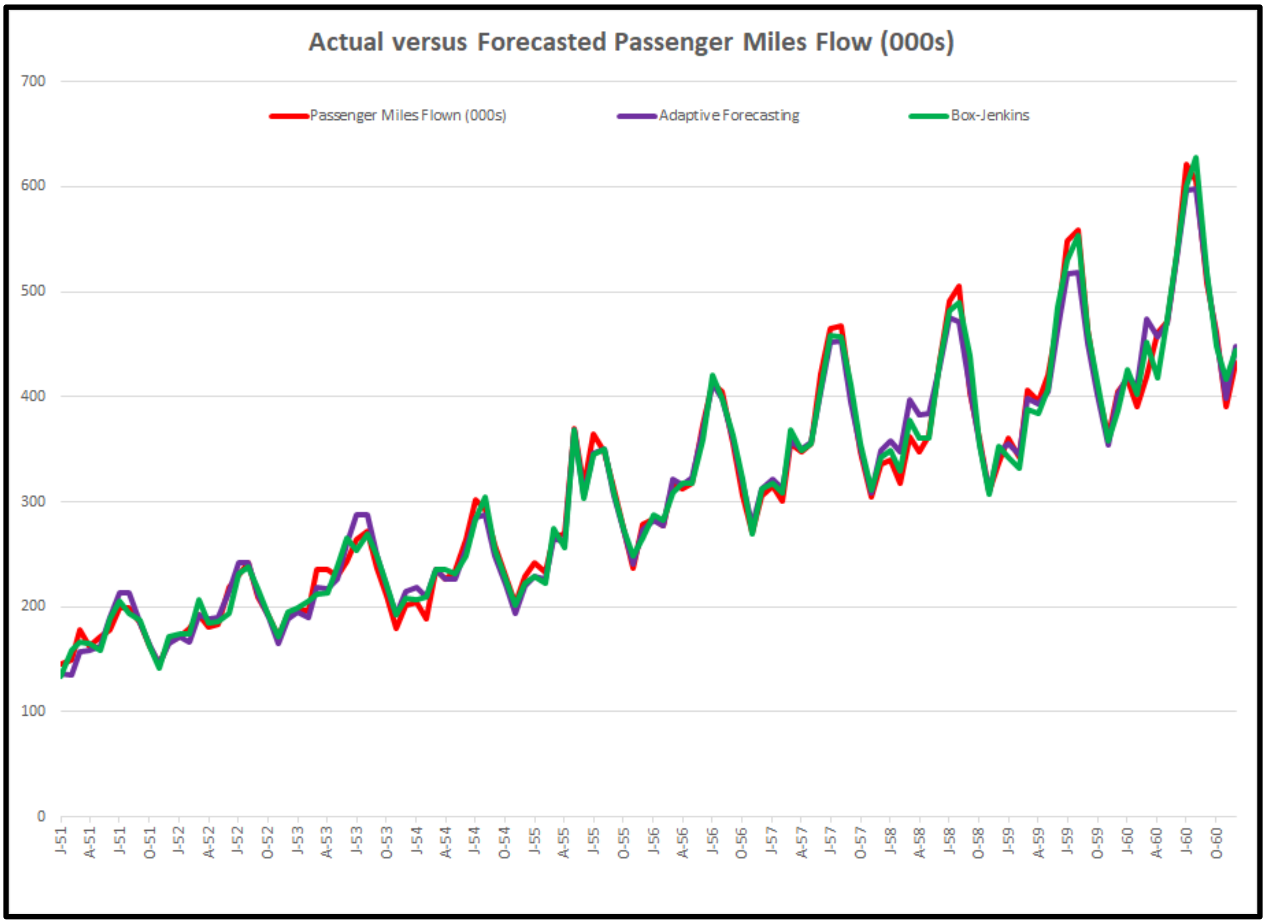
Here, we added a second logical expression (t=m), which will return a value of 1.0 if t equals the target month (m); otherwise the logical expression returns a value of 0.0. The weight for a month (m) is the fraction of times Forecast A outperformed Forecast B in that month.

The weight placed on Forecast B is computed as:

$$\mathrm{Weight}^{B}_{T} = 1 - \mathrm{Weight}^{A}_{T}$$

The combined forecast at time (T) is then computed as:

$$\mathrm{CombinedForecast}_{T} = \mathrm{Weight}^{A}_{T}\mathrm{Forecast}^{A}_{T} + \mathrm{Weight}^{B}_{T}\mathrm{Forecast}^{B}_{T}$$



This weighting scheme, like the original Bunn weighting scheme, relies on historical performance, but in this case the focus is on the forecast performance in the month (or season) that is being forecasted. The forecast performance of the Seasonal Outperformance approach is presented below. The results suggest that significant value can be gained by developing weights that are stronger predictors of the forecast performance for the period under study.

	Mean Forecast Error	Median Forecast Error	Maximum Forecast Error	Minimum Forecast Error	Mean Squared Forecast Error	Skewness	Kurtosis
Adaptive Forecasting Forecast	0.3	1.0	40.0	-55.0	177.8	-0.6	2.7
Box-Jenkins Forecast	-0.1	-1.0	43.0	-35.0	148.6	0.1	0.9
Straight Average	0.1	-1.0	24.5	-44.0	130.4	-0.3	1.1
Bates and Granger Method 1	0.3	-0.6	32.0	-40.4	131.3	-0.1	0.9
Bates and Granger Method 2	0.1	-0.9	28.6	-41.6	132.8	-0.2	0.8
Bates and Granger Method 3	0.2	-0.7	32.1	-39.9	128.9	-0.1	1.0
Bates and Granger Method 4	0.4	-0.1	57.9	-49.8	162.4	0.2	4.5
Bates and Granger Method 5	0.1	-1.1	26.3	-44.7	130.3	-0.3	1.3
Bunn Outperformance Method	0.1	-0.8	25.1	-44.7	130.2	-0.3	1.3
Granger and Ramanathan Regression	0.0	-1.5	26.8	-42.6	128.5	-0.3	0.9
Seasonal Outperformance	0.2	-0.8	22.7	-40.8	**115.7**	-0.4	1.0

Extended Regression. If regressing the original two forecasts against actual passenger flown miles led to improved forecast performance, it is fair to ask: what would happen if all the alternative forecasts were included in the regression? In this case, the following regression is estimated and the forecast errors from this regression are evaluated.

$$\begin{aligned}\text{PassengerFlownMiles}_{y,m} &= \beta_0 + \beta_1\ \text{AdaptiveForecast}_{y,m} + \beta_2\ \text{BoxJenkinsForecast}_{y,m} \\ &+ \beta_3\ \text{BatesGrangerMethod1}_{y,m} + \beta_4\ \text{BatesGrangerMethod2}_{y,m} \\ &+ \beta_5\ \text{BatesGrangerMethod3}_{y,m} + \beta_6\ \text{BatesGrangerMethod4}_{y,m} \\ &+ \beta_7\ \text{BatesGrangerMethod5}_{y,m} + \beta_8\ \text{BunnOutPerformance}_{y,m} \\ &+ \beta_9\ \text{SeasonalOutPerformance}_{y,m} + e_{y,m}\end{aligned}$$

The results demonstrate the advantage of the extended regression approach with the overall smallest MSE of 99.5.

	Mean Forecast Error	Median Forecast Error	Maximum Forecast Error	Minimum Forecast Error	Mean Squared Forecast Error	Skewness	Kurtosis
Adaptive Forecasting Forecast	0.3	1.0	40.0	-55.0	177.8	-0.6	2.7
Box-Jenkins Forecast	-0.1	-1.0	43.0	-35.0	148.6	0.1	0.9
Straight Average	0.1	-1.0	24.5	-44.0	130.4	-0.3	1.1
Bates and Granger Method 1	0.3	-0.6	32.0	-40.4	131.3	-0.1	0.9
Bates and Granger Method 2	0.1	-0.9	28.6	-41.6	132.8	-0.2	0.8
Bates and Granger Method 3	0.2	-0.7	32.1	-39.9	128.9	-0.1	1.0
Bates and Granger Method 4	0.4	-0.1	57.9	-49.8	162.4	0.2	4.5
Bates and Granger Method 5	0.1	-1.1	26.3	-44.7	130.3	-0.3	1.3
Bunn Outperformance Method	0.1	-0.8	25.1	-44.7	130.2	-0.3	1.3
Granger and Ramanathan Regression	0.0	-1.5	26.8	-42.6	128.5	-0.3	0.9
Seasonal Outperformance	0.2	-0.8	22.7	-40.8	115.7	-0.4	1.0
Aggregate Regression	0.0	-0.8	24.7	-32.7	**99.5**	-0.1	0.5

Summary. The above results give us the following general principles for improving forecast performance.

- It is possible to construct a linear combination of a set of alternative forecasts that will outperform the individual forecasts upon which the combination is built.
- Three ways of measuring historical forecast performance have been suggested:
    - Squared Forecast Errors,
    - Absolute Forecast Errors, and
    - Outperformance
- There is benefit from weighting recent forecast performance heavier using adaptive learning or observation weighting.
- There is benefit from using weights designed to predict performance under forecasted conditions such as season.
- Linear regression approaches can be extended to include not only the original alternative forecasts, but also weighted combinations of the original forecasts.

DATA SCIENCE APPROACHES

The weighting schemes described above assume an ensemble of alternative forecast models exist. In this case, the forecast objective is to design a weighting scheme that delivers a combined forecast that outperforms the individual alternative forecasts. The Data Science literature turns the problem around by designing an optimal set of alternative forecast models. The optimal weighting scheme is then a function of how the ensemble is constructed. From the perspective of load forecasting, the Data Science approaches presented below can be used to help stress test existing model specifications, as well as provide a path for predicting loads and load forecast errors.

The general ideas for this section can be found in several papers. Ye Ren, P.N. Suganthan, and N. Srikanth, *Ensemble Methods for Wind and Solar Power Forecasting – A State-of-the-Art Review*, Renewable and Sustainable Energy Reviews 50 (2015) pp. 82-91, provides robust coverage of the application of ensemble learning to forecasting wind and solar generation. The following discussion culls from their paper some of the key approaches that can be readily applied to load forecasting.

According to Opitz d., and Maclin R., *Popular Ensemble Methods: an Empirical Study*, Journal of Artificial Intelligence Research (1999), 11:169-196, ensemble algorithms can be widely classified as competitive or cooperative in the approach the algorithm takes to form an optimal forecast.

- A *competitive ensemble* method takes multiple independently produced forecasts as inputs and forms a combined forecast as a weighted average of the ensemble forecasts. The weighting schemes introduced earlier fall within this broad class of competitive ensemble methods. The Random Forest and Adaptive Boosting methods that will be discussed shortly are considered competitive ensemble methods.
- A *cooperative ensemble* method segments the forecasting problem into sub tasks, builds the best model or models for each sub task, and then adds the forecasts from each sub task together to form a combined forecast. The Gradient Boost method that is introduced below is a form of a cooperative ensemble method.

Competitive Load Forecast Ensemble Models. Within the spirit of competitive ensemble methods, it is not uncommon for a control room to operate against a series of alternative load forecast models. Generally, the alternative models reflect different combinations of explanatory variables, different nesting of models (e.g., hourly models only, daily energy driving hourly models, or daily peak driving hourly models), different model structures (e.g., regression versus neural networks), and different weather forecasts. Random Forest and Adaptive Boosting methods introduce the idea of creating an ensemble of models by changing the data over which the model parameters are estimated. This is achieved by weighting the observations differently and then using weighted least squares to estimate the model parameters.

Much of the Data Science literature on competitive ensemble learning falls into general categories: Bagging and Boosting. Bagging trains an ensemble of models independently of each other. The forecasts from the ensemble are then weighted together to form a combined forecast. Boosting trains an ensemble of models sequentially in that the errors from the previous model in the sequence inform the estimates of the next model in the sequence. Like Bagging, the forecasts from the Boosting ensemble models are weighted together to form a combined forecast.

Presented below are pseudo Bagging and Boosting Algorithms as they might be applied to load forecasting. This is followed by a discussion of how these pseudo algorithms could be incorporated into the weighting schemes discussed earlier.

Random Forest. Random Forest is a form of a Bagging algorithm that builds an ensemble of models independently of each other. The relative in-sample performance of each ensemble model is then used to form a combined forecast that is a weighted sum of the individual model forecasts. From the perspective of load forecasting, Random Forecast offers a means of creating an ensemble of models that could be identical in specification (e.g., same set of explanatory variables, same nesting of models, and same model structure) but have different estimated parameters. This is done by repeatedly using weighted least squares to fit a candidate model specification to the same historical load data, but at each model iteration, the weights placed on the observations vary randomly. If a sufficiently large set of ensemble models are developed, it is expected the combined forecast will be well behaved over a range of load conditions.

The steps in a *pseudo* Random Forest algorithm as applied to load forecasting are:

Initialization

- Define the estimation date range
- Compute over the estimation date range the variance (σ^2) of the load being model
- Define a candidate load forecast model specification
- Initialize the data observation weights to 1.0 for all observations in the estimation period

Create Ensemble Model Number 1

- Use weighted least squares to estimate the parameters of the candidate load forecast model specification over the selected estimation date range.
- Store the estimated model as Ensemble Model Number 1.
- Compute the unnormalized ensemble model weight as:

$$\text{UnNormalizedEnsembleModelWeight}^1 = e^{-\left(\frac{\sum_{d=1}^{D}\left(\text{EnsembleModelError}_d^1\right)^2}{2\sigma^2}\right)}$$

Here, an observation or day in the estimation period is indexed by (d) with the total number of observations being (D). This weighting scheme places more weight on models that have the best performance as measured by their in-sample fit.

Create Ensemble Model Number 2 to J

- Use weighted least squares to estimate (J-1) candidate models. For each model, the estimation date range stays the same, but the observation weights are set randomly using a random number generator. By assigning observation weights randomly, each model in the ensemble should have slighting different estimated parameters. This in turn will lead to different predicted values.

- Store the estimated model as Ensemble Model Number j.
- Compute the unnormalized ensemble model weight as:

$$\mathrm{UnNormalizedEnsembleModelWeight}^{j} = e^{-\left(\frac{\sum_{d=1}^{D}\left(\mathrm{EnsembleModelError}_{d}^{j}\right)^{2}}{2\sigma^{2}}\right)}$$

Compute Normalized Ensemble Model Weights

Under this step, the weight placed on each forecast from the ensemble of models is normalized to sum to 1.0. The normalization step is:

$$\mathrm{NormalizedEnsembleModelWeight}^{j} = \frac{\mathrm{UnNormalizedEnsembleModelWeight}^{j}}{\sum_{k=1}^{K}\mathrm{UnNormalizedEnsembleModelWeight}^{k}}$$

Here, (K) is the total number of models in the ensemble.

Compute Combined Forecast

Given the (K) estimated ensemble models and their associated normalized ensemble weight, the combined forecast is computed as:

$$\mathrm{CombinedForecast}_{D+h} = \sum_{k=1}^{K}\mathrm{NormalizedEnsembleModelWeight}^{k}\ \mathrm{ModelForecast}_{D+h}^{k}$$

Here, the h-step ahead (D+h) Combined Forecast from the Random Forest ensemble is the weighted sum of h-step ahead ensemble model forecasts.

The forecasts from the combined Random Forest ensemble can then be used as is or can be added to another list of alternative forecast models. An interesting aspect of the Random Forest method is it provides a means of stress-testing the estimated model parameters. This approach could be applied to estimating both regression and neural network models.

Adaptive Boosting. Adaptive Boosting (AdaBoost) is form of a Boosting algorithm that builds an ensemble of models in a sequential manner and then constructs a combined forecast as a weighted sum of the individual ensemble forecasts. The interesting elements of AdaBoost are:

- The AdaBoost algorithm provides a procedure for creating alternative forecast models by weighting the in-sample training data differently for each model in the ensemble,
- The models are constructed sequentially with each subsequent model designed to improve upon those observations the previous model performed poorly on, and
- The algorithm provides a set of normalized weights that are applied to the ensemble forecasts that form a combined forecast that in principle should outperform the individual ensemble forecasts.

Outlined below is a variation of the AdaBoost algorithm applied to the problem of load forecasting. These are not the exact steps of the AdaBoost algorithm as applied to a classification problem since, in those cases, the AdaBoost algorithm builds a sequence of decision trees. In the extreme case, each decision tree is defined by one and only one feature.

In the steps outlined below, the decision trees are replaced with a fully specified regression or neural network model. That is, instead of constructing a series of regression and/or neural network models that are a function of one or few features (e.g., a model with day-of-the-week binary variables only, followed by a model with just average temperature), one candidate model that is fully specified is used. The candidate model will then be estimated multiple times using weighted least squares. At each step in the algorithm, the observation weights that are used will be based on the in-sample fit errors from the previously estimated model. That is, if Model (n) in the sequence of ensemble models performed poorly (i.e., had large model errors) on Summer peak days, then the next model in the sequence (n+1) will be tuned to improve the fit on Summer peak days by weighting those days higher than other days. In principle, the resulting model ensemble should perform well under varying load conditions.

The steps in the *pseudo* AdaBoost algorithm are:

Initialization

- Define the estimation date range
- Compute over the estimation date range the variance (σ^2) of the load being model
- Define a candidate load forecast model specification
- Initialize the data observation weights to 1.0 for all observations in the estimation period

PRENOM Classification Algorithm. The specific weighting scheme presented above is taken from Vassilios Petridis and Athanasios Kehagias, *Predictive Modular Neural Networks – Applications to Time Series*, Kluwer Academic Publishers (1998). In their book, the authors present a basic PRENOM Classification Algorithm that uses normalized sum of squared errors to weight the alternative forecasts.

While this weighting scheme is convenient, there is nothing written in stone that this is the only weighting scheme that could be used. For example, in many load forecasting applications, forecast errors that fall outside of an acceptable operating range (e.g., $\pm\ 250\ \mathrm{MW}$) count significantly more than forecast errors that fall within acceptable tolerances. This suggests an unnormalized weighting scheme that "scores" an ensemble method by the fraction of times the forecast error is within tolerance. For example,

$$\begin{aligned}&\mathrm{UnnormalizedEnsembleModelWeight}^{k}\\&=\frac{\sum_{d=1}^{D}\left(\left|\mathrm{EnsembleModelError}_{d}^{k}\right|\leq\mathrm{Tolerance}\right)}{\sum_{d=1}^{D}\left(\left|\mathrm{EnsembleModelError}_{d}^{k}\right|\leq\mathrm{Tolerance}\right)+\sum_{d=1}^{D}\left(\left|\mathrm{EnsembleModelError}_{d}^{k}\right|>\mathrm{Tolerance}\right)}\end{aligned}$$

Here, the numerator is the total number of forecasts from ensemble model (k) where the forecast error was within tolerance. The denominator is the total number of forecasts produced by ensemble model (k).

This idea can be extended to how the data observations are weighted across the sequence of ensemble models. For example, the following rule could be used to weight observations with out of tolerance errors greater than within tolerance errors.

$$\mathrm{ObservationWeight}_{d}^{j}=\mathrm{f(x)}=\begin{cases}1, & \left(\left|\mathrm{EnsembleModelError}_{d}^{j-1}\right|\leq\mathrm{Tolerance}\right)\\1+\delta, & \left(\left|\mathrm{EnsembleModelError}_{d}^{j-1}\right|>\mathrm{Tolerance}\right)\end{cases}$$

Here, Tolerance is an acceptable error and δ is a user-specified penalty placed on out of tolerance errors.

Placing greater weight on out-of-tolerance errors is like Support Vector Regression, which weights outliers higher than within tolerance errors. Another variation would be to weight over-forecast errors differently than under-forecast errors.

Create Ensemble Model Number 1

- Use weighted least squares to estimate the parameters of the candidate load forecast model specification over the selected estimation date range.
- Store the estimated model as Ensemble Model Number 1.
- Compute the unnormalized ensemble model weight as:

$$\mathrm{UnNormalizedEnsembleModelWeight}^{1} = \mathrm{e}^{-\left(\frac{\sum_{\mathrm{d}=1}^{\mathrm{D}}\left(\mathrm{EnsembleModelError}_{\mathrm{d}}^{1}\right)^{2}}{2\sigma^{2}}\right)}$$

Here, an observation or day in the estimation period is indexed by (d) with the total number of observations being (D). Note: This weighting scheme places more weight on models that have the best performance as measured by their in-sample fit.

Create Ensemble Model Number 2

- Use weighted least squares to estimate the candidate model over the selected estimation date range where the observation weights are derived from the in-sample model errors of Ensemble Model Number 1 as follows:

$$\mathrm{ObservationWeight}_{\mathrm{d}}^{2} = (\mathrm{EnsembleModelError}_{\mathrm{d}}^{1})^{2}$$

- Store the estimated model as Ensemble Model Number 2.
- Compute the unnormalized ensemble model weight as:

$$\mathrm{UnNormalizedEnsembleModelWeight}^{2} = \mathrm{e}^{-\left(\frac{\sum_{\mathrm{d}=1}^{\mathrm{D}}\left(\mathrm{EnsembleModelError}_{\mathrm{d}}^{2}\right)^{2}}{2\sigma^{2}}\right)}$$

Create Ensemble Model Number 3 through J

Repeat Step 2 to derive ensemble models 3 through J. At each step, use the in-sample errors from the ensemble model in the previous step to form the observation weights. The steps are written generally as:

- Compute observation weights

$$\mathrm{ObservationWeight}_{\mathrm{d}}^{\mathrm{j}} = \left(\mathrm{EnsembleModelError}_{\mathrm{d}}^{\mathrm{j}-1}\right)^{2}$$

- Estimate the candidate model using weighted least squares
- Store the estimated model as Ensemble Model Number (j)
- Compute the unnormalized ensemble model weight as:

$$\mathrm{UnNormalizedEnsembleModelWeight}^{j} = e^{-\left(\frac{\sum_{d=1}^{D}\left(\mathrm{EnsembleModelError}_{d}^{j}\right)^{2}}{2\sigma^{2}}\right)}$$

- Continue to the next ensemble model until a stopping criterion is met. For example, stop after the 10th ensemble model is estimated. Stop if the change in sum of square errors for ensemble model (j) deviates from the sum of squared errors for ensemble model (j-1) by more than a user-specified threshold.

Compute Normalized Ensemble Model Weights

Under this step, the weight placed on each forecast from the ensemble of models is normalized to sum to 1.0. The normalization step is:

$$\mathrm{NormalizedEnsembleModelWeight}^{j} = \frac{\mathrm{UnNormalizedEnsembleModelWeight}^{j}}{\sum_{k=1}^{K}\mathrm{UnNormalizedEnsembleModelWeight}^{k}}$$

Here, (K) is the total number of models in the ensemble.

Compute Combined Forecast

Given the (K) estimated ensemble models and their associated normalized ensemble weight, the combined forecast is computed as:

$$\mathrm{CombinedForecast}_{D+h} = \sum_{k=1}^{K}\mathrm{NormalizedEnsembleModelWeight}^{k}\ \mathrm{ModelForecast}_{D+h}^{k}$$

Here, the h-step ahead (D+h) Combined Forecast from the AdaBoost ensemble is the weighted sum of h-step ahead ensemble model forecasts

The forecasts from the combined AdaBoost ensemble can then be used as is or can be added to another list of alternative forecast models.

Gradient Boosting. A common application of cooperative load forecasting is the use of seasonal models. This builds upon the premise that a model estimated on data for one season should perform well in forecasting loads for that season. For example, a model estimated using data from June, July, and August should perform well forecasting loads on hot sunny, summer days. A cooperative ensemble is then formed by leveraging the load forecasts from a Winter model, a Spring model, a Summer model, and a Fall model.

Gradient Boosting (XBoost) is another form of a boosting algorithm that builds an ensemble of decision trees or models in a sequential manner and then constructs a combined forecast as a sum of the individual ensemble forecasts. XBoost differs from AdaBoost in two fundamental ways.

- AdaBoost uses weighted least squares to fit in sequence a candidate model to the same (load) data, but for each sequence the observation weights depend on the in-sample errors from the previous model in the sequence. XBoost fits a sequence of models where the dependent variable for each sequence is the model residuals from the previous model in the sequence.
- AdaBoost could use the same candidate model specification as the basis for each model in the ensemble. In this case, the models will produce different predicted and forecast values because the estimated parameters will differ across the models. In contrast, the models that result from XBoost will necessarily differ in specification from one sequence to the next.

The XBoost algorithm could be utilized as a load disaggregation algorithm. For example, consider the following ensemble of models developed using a variant of the XBoost algorithm.

- Regress loads on the set of seven (7) day-of-the-week binary variables. This model in the ensemble will predict the average day-of-the-week variation in loads. Call it the Day-of-the-Week model.
- Regress the residuals from Day-of-the-Week model on Cooling Degree Day and Heating Degree Day splines. This model will predict the weather sensitive portion of the load. Call it the Weather model.
- The combined forecast is then computed as the sum of the Day-of-the-Week and Weather model forecasts.

From the perspective of load forecasting, the challenge of the XBoost method is designing a sequence of alternative model specifications that complement each other while maintaining the integrity of the underlying time series of the loads. One option is to have a primary model that is a fully specified regression or neural network model of loads. A secondary model is then designed to provide forecasts of the load forecast bias or error. This two-model configuration could be written generally as follows:

$$\text{Primary Forecast Model: } \mathrm{Load_d} = \mathrm{F}(\beta \mathrm{X_d}) + \mathrm{u_d}$$

$$\text{Bias Forecast Model: } \mathrm{u_d} = \mathrm{G}(\rho \mathrm{Z_d}) + \mathrm{e_d}$$

Here, the Load on day (d) is a function $\mathrm{F}(\beta \mathrm{X_d})$ containing a matrix of explanatory variables ($\mathrm{X_d}$) and a vector of unknown coefficients (β). This model is designed to provide an unbiased forecast of loads. To account for possible systematic forecast error, an auxiliary model of the residuals from the primary model is estimated. This model is a function $\mathrm{G}(\rho \mathrm{Z_d})$ containing a matrix of explanatory variables ($\mathrm{Z_d}$) and a vector of unknown coefficients (ρ). It is expected that the auxiliary forecast bias model would include autoregressive and non-autoregressive explanatory variables. The combined forecast is then computed as:

$$\mathrm{Combined}_{D+h} = F(\hat{\beta}X_{D+h}) + G(\hat{\rho}Z_{D+h})$$

Here, the Combined h-step ahead forecast made on day (D) is the sum of the predicted load and the predicted forecast error bias.

There are many possible ways of leveraging the XBoost algorithm to produce a strong ensemble forecast. To help fix ideas, a *pseudo* XBoost algorithm is described below.

Initialization

- Define the estimation date range
- Define a sequence of ensemble model specifications

Create Ensemble Model Number 1

- Use least squares to estimate the parameters of the first ensemble model over the selected estimation date range.
- Store the estimated model as Ensemble Model Number 1.
- Compute the model residuals

$$\mathrm{Residual}_d^1 = \mathrm{Load}_d - \mathrm{ModelPredicted}_d^1$$

Here, the residual for the first ensemble model on day (d) in the estimation data range is the difference between actual loads on day (d) and predicted value from the first model in the ensemble.

Create Ensemble Model Number 2

- Use least squares to estimate the parameters of the second ensemble model by regressing the Residuals from Ensemble Model 1 on the set of explanatory variables included in the Ensemble Model 2 specification.
- Store the estimated model as Ensemble Model Number 2.
- Compute the model residuals

$$\mathrm{Residual}_d^2 = \mathrm{Residual}_d^1 - \mathrm{ModelPredicted}_d^2$$

Here, the residual for model number 1 on day (d) in the in-sample data range is the difference between actual loads on day (d) and predicted value from the first model in the ensemble.

Create Ensemble Model Number 3 through J

Repeat Step 2 to derive ensemble models 3 through J. At each step, use the in-sample errors from the ensemble model in the previous step as the dependent variables. The steps are written generally as:

- Estimate the parameters of Ensemble Model Number j.
- Store the estimated model as Ensemble Model Number j.

- Compute the model residuals

$$\text{Residual}_{\text{d}}^{\text{j}} = \text{Residual}_{\text{d}}^{\text{j}-1} - \text{ModelPredicted}_{\text{d}}^{\text{j}}$$

Here, the residual for the Jth ensemble model on day (d) in the estimation period is the difference between the residual for the J-1 ensemble model on day (d) and predicted value from the Jth model.

Compute Combined Forecast

Given (K) estimated ensemble models the combined forecast is computed as:

$$\text{CombinedForecast}_{\text{D+h}} = \sum_{\text{k}=1}^{\text{K}} \text{ModelForecast}_{\text{D+h}}^{\text{k}}$$

Here, the h-step ahead (D+h) Combined Forecast from the XBoost ensemble is the sum of h-step ahead ensemble model forecasts.

The forecasts from the combined XBoost ensemble can then be used as is or can be added to another list of alternative forecast models.

Summary. The Data Science literature provides the following general principles for improving forecast performance.

- It is possible to convert a single model into a sequence of models where the estimated parameters of each model in the sequence are informed by the errors from the previous model. The resulting ensemble can lead to a combined forecast that outperforms the original fixed weighted model.
- Building a set of complementary models can lead to improved forecast performance. The challenge is being clever in the design of the complementary models. Modelling the errors from a primary model and then using the auxiliary model to forecast the error expected from the primary model would replace the need for manual bias adjustments.

A GENERAL FORMULA-BASED COMBINED LOAD FORECAST

This section presents guidelines for developing an operational load forecast framework that encompasses the following general principles.

- It is possible to construct a weighted combination of a set of alternative forecasts that will outperform the individual forecasts upon which the combination is built.
- It is possible to convert a single model into an ensemble of competitive and/or complementary models.
- There are multiple ways of measuring historical forecast performance and multiple ways to weight an ensemble of forecasts together to form a combined forecast. How forecast performance is measured should inform the weighting scheme that is implemented.
- There is benefit from placing higher weight on recent forecast performance.
- There is benefit from weighting forecast performance under similar conditions as the forecast period.

The first step in developing a combined load forecast is designing an ensemble of load forecast models that avoid consternation on a control room floor. For many operational load forecasters, the following forecast errors should be avoided at all cost.

Within-Day Forecast Bias Trend. This occurs when the sequence of real-time forecasts is consistently over (under) forecasting real-time loads. Possible causes are:

- Model misspecification (e.g., poor treatment of special event days like public holidays),
- Poor weather and behind-the-meter solar PV generation forecasts, and
- Unexpected load loss (e.g., dropped meter reads or end point outages).

Within-Day Forecast Instability. Because most within-day load forecast models place a heavy reliance on autoregressive load values, it is not uncommon for a control area with volatile loads to experience a sequence of load forecasts made for a particular time of the day (e.g., 10:00AM) to show instability. This is a growing problem in control areas that have significant penetration of behind-the-meter solar PV generation driving increased load volatility.

- An example of forecast instability is a comparison of the 10:00AM forecast made at 02:00AM, 04:00AM, 06:00AM, 08:00AM and 09:00AM. For a control area with stable loads, the 10:00AM forecasts may range from a low of 950MW to a high of 1,050 MW. The range for a control area with volatile loads may range from a low of 500MW to a high of 1,500MW. Most system operators can work with a load forecast that varies +/- 50MW. In contrast, most system operators will not be pleased with a forecast range of +/- 500MW.

Day-Ahead Peak Forecast Bias. Most load forecast models tend to under-forecast peak loads during extreme temperature conditions. There are two factors that lead to the under-forecasts.

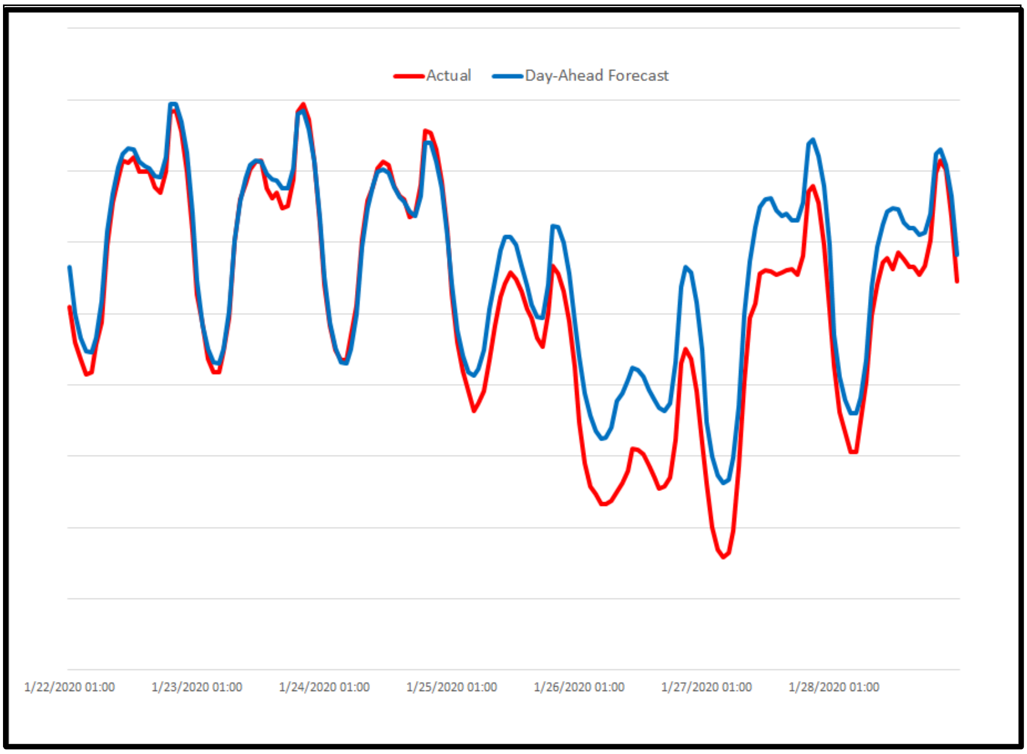
- By construction, load forecast models are segmented averages of historical load data. The segments upon which the average is computed are defined by the mix of explanatory variables. A combination of explanatory variables that isolate peak loads from non-peak loads will lead to a predicted value that is the average of the isolated peak loads. As result, a peak forecast which will be close to the historical average peak has the potential for under-forecasting peak loads on days when the weather conditions are more extreme than those that existed in the historical period.

- Weather forecasts tend to under-forecast extreme hot temperatures and over-forecast extreme cold temperatures because they tend to be constructed as the average of an ensemble of weather forecasts.
- The combination of a load model that tends to the average and weather forecasts that tend toward milder temperatures leads to under-forecasting of peak loads (or over-forecasting of minimum loads).

Across Day Forecast Bias Trend. This occurs when the sequence of day-ahead load forecasts run consistently high (low). An example is depicted in Figure 2. Here, the sequence of one Day-Ahead Forecasts (blue line) track actual loads (red line) well at the beginning of the period, but then switch to over-forecasting loads systematically at the end of the period.

With these forecast errors in mind, we turn toward building a formula-based combined forecast.

FIGURE 2. ACROSS DAY LOAD FORECAST BIAS TREND



Forecast Ensemble. Since the focus of this paper is on methods for constructing a combined forecast from an existing ensemble of forecasts, we assume that the load forecaster has a set of forecast models to work with. These models might have been hand crafted or were developed using a combination of Random Forest, AdaBoost, and Gradient Boosting methods. Ideally, there is at least one model that is designed to avoid the four situations described above.

To help fix ideas, let's assume the forecast analyst has developed a candidate model and then applied the AdaBoost method to form two additional alternative models. This leads to the following three ensemble models:

1. Ensemble Model A. The original candidate model estimated using unweighted least squares.
2. Ensemble Model B. The original candidate model estimated using weighted least squares where the observation weights equal the squared residuals from Ensemble Model A.
3. Ensemble Model C. The original candidate model estimated using weighted least squares where the observation weights equal the squared residuals from Ensemble Model B.

Forecast Performance Target. The next component in the forecast framework is the target that will be used to judge forecast performance. This in turn will guide which weighting scheme is deployed. The data that Bates and Granger worked with had one forecast error, specifically the difference between actual and forecasted monthly passenger miles flown. For hourly load forecasting there are several forecast errors that are candidates for judging performance. For example, the combination weighting scheme can be designed to minimize:

- Peak load forecast errors,
- Minimum load forecast errors,
- Daily energy forecast errors, or
- Night, morning, mid-day, afternoon, or evening hours (i.e., time-of-day) load forecast errors.

In principle, a separate combination forecast could be constructed for each hour or time interval of the day, but that would then introduce a potential issue that the resulting combined forecast would not be as smooth across the hours of the day as a combination scheme that weights a complete day of forecast values equally. To avoid the complexity of having to smooth the resulting combined forecast we restrict the discussion to developing a set of weights that are applied equally to all intervals of the day. We will relax this restriction later in this paper.

For illustrative purposes, we assume the target is to reduce the sequence of day-ahead energy forecast errors. In this case, one set of weights will be developed that can be applied to all hours or intervals of the alternative day-ahead load forecasts. For example,

$$\mathrm{CombinedForecast}_{\mathrm{D+1,i}} = \mathrm{Weight}^{\mathrm{A}}_{\mathrm{D+1}}\mathrm{Forecast}^{\mathrm{A}}_{\mathrm{D+1,i}} + \mathrm{Weight}^{\mathrm{B}}_{\mathrm{D+1}}\mathrm{Forecast}^{\mathrm{B}}_{\mathrm{D+1,i}} + \mathrm{Weight}^{\mathrm{C}}_{\mathrm{D+1}}\mathrm{Forecast}^{\mathrm{C}}_{\mathrm{D+1,i}}$$

Here,

$\mathrm{Weight}^{\mathrm{A}}_{\mathrm{D+1}}$ is the weight placed on the Ensemble Model A day-ahead (D+1) load forecast

$\mathrm{Weight}^{\mathrm{B}}_{\mathrm{D+1}}$ is the weight placed on the Ensemble Model B day-ahead (D+1) load forecast

$\mathrm{Weight}^{\mathrm{C}}_{\mathrm{D+1}}$ is the weight placed on the Ensemble Model C day-ahead (D+1) load forecast

$\mathrm{Forecast}^{\mathrm{A}}_{\mathrm{D+1,i}}$ is the Ensemble Model A day-ahead (D+1) load forecast for time (i)

$\mathrm{Forecast}^{\mathrm{B}}_{\mathrm{D+1,i}}$ is the Ensemble Model B day-ahead (D+1) load forecast for time (i)

$\mathrm{Forecast}^{\mathrm{C}}_{\mathrm{D+1,i}}$ is the Ensemble Model C day-ahead (D+1) load forecast for time (i)

Weighting Scheme. The next question to ask is which weighting scheme should be implemented. The weighting schemes discussed in this paper are:

- Weighted linear average using equal and fixed weights on the alternative load forecasts
- Weighted linear average using dynamic weights based on squared forecast errors
- Weighted linear average using dynamic weights based on absolute forecast errors
- Weighted linear average using dynamic weights based on outperformance indicators
- Weighted linear average using dynamic weights based on seasonal outperformance indicators
- Weighted linear average using regression-based weights

For illustrative purposes, we will assume a weighting scheme that is based on the squared day-ahead energy forecast errors. The day-ahead daily energy forecast errors are defined as:

$$\mathrm{DailyEnergyError}_{\mathrm{d}}^{\mathrm{A,D+1}} = \sum_{\mathrm{i=1,i \in d}}^{\mathrm{I}} \mathrm{Load}_{\mathrm{d,i}} - \sum_{\mathrm{i=1,i \in d}}^{\mathrm{I}} \mathrm{Forecast}_{\mathrm{d,i}}^{\mathrm{A,D+1}}$$

$$\mathrm{DailyEnergyError}_{\mathrm{d}}^{\mathrm{B,D+1}} = \sum_{\mathrm{i=1,i \in d}}^{\mathrm{I}} \mathrm{Load}_{\mathrm{d,i}} - \sum_{\mathrm{i=1,i \in d}}^{\mathrm{I}} \mathrm{Forecast}_{\mathrm{d,i}}^{\mathrm{B,D+1}}$$

$$\mathrm{DailyEnergyError}_{\mathrm{d}}^{\mathrm{C,D+1}} = \sum_{\mathrm{i=1,i \in d}}^{\mathrm{I}} \mathrm{Load}_{\mathrm{d,i}} - \sum_{\mathrm{i=1,i \in d}}^{\mathrm{I}} \mathrm{Forecast}_{\mathrm{d,i}}^{\mathrm{C,D+1}}$$

Here,

$\mathrm{DailyEnergyError}_{\mathrm{d}}^{\mathrm{A,D+1}}$is the day-ahead (D+1) daily energy forecast error on day (d) from the Ensemble Model A load forecast

$\mathrm{DailyEnergyError}_{\mathrm{d}}^{\mathrm{B,D+1}}$is the day-ahead (D+1) daily energy forecast error on day (d) from the Ensemble Model B load forecast

$\mathrm{DailyEnergyError}_{\mathrm{d}}^{\mathrm{C,D+1}}$is the day-ahead (D+1) daily energy forecast error on day (d) from the Ensemble Model C load forecast

$\mathrm{Load}_{\mathrm{d,i}}$ is the actual load on day (d) for time interval (i) where $i \in d$ indicates that the sum is over all time intervals within day (d)

$\mathrm{Forecast}_{\mathrm{d,i}}^{\mathrm{A,D+1}}$ is the Ensemble Model A Day-Ahead (D+1) load forecast made for day (d) and time (i)

$\mathrm{Forecast}_{\mathrm{d,i}}^{\mathrm{B,D+1}}$ is the Ensemble Model B Day-Ahead (D+1) load forecast made for day (d) and time (i)

$\mathrm{Forecast}_{\mathrm{d,i}}^{\mathrm{C,D+1}}$ is the Ensemble Model C Day-Ahead (D+1) load forecast made for day (d) and time (i)

Learning Framework. Bates and Granger introduced the idea of placing higher value on recent forecast performance by either using a weighted rolling sum or a simple learning rate framework. As a first pass, we will use a weighted rolling sum with ($\omega > 1$) to ensure the most recent forecast performance has the biggest weight. Further, we can define (v) to control how many days back over which the rolling sum will be computed. For example, if v = 14, then the rolling sum is over the prior 14 days of day-ahead forecast errors.

We can now define the ensemble model weights as follows:

Ensemble Model A Errors. The weighted rolling sum of the squared Day-Ahead energy forecast errors from Ensemble Model A is computed as:

$$\mathrm{s}_{\mathrm{A,D+1}}^{2} = \sum_{\mathrm{d=D-v}}^{\mathrm{D}} \omega^{d}\left(\mathrm{DailyEnergyError}_{\mathrm{d}}^{\mathrm{A,D+1}}\right)^{2}$$

Ensemble Model B Errors. The weighted rolling sum of the squared Day-Ahead energy forecast errors from Ensemble Model B is computed as:

$$\mathrm{s}_{\mathrm{B,D+1}}^{2} = \sum_{\mathrm{d=D-v}}^{\mathrm{D}} \omega^{d}\left(\mathrm{DailyEnergyError}_{\mathrm{d}}^{\mathrm{B,D+1}}\right)^{2}$$

Ensemble Model C Errors. The weighted rolling sum of the squared Day-Ahead energy forecast errors from Ensemble Model C is computed as:

$$\mathrm{s}_{\mathrm{C,D+1}}^{2} = \sum_{\mathrm{d=D-v}}^{\mathrm{D}} \omega^{\mathrm{d}}\left(\mathrm{DailyEnergyError}_{\mathrm{d}}^{\mathrm{C,D+1}}\right)^{2}$$

The corresponding weights are computed as:

$$\mathrm{EnsembleWeight}_{\mathrm{A}}^{\mathrm{D+1}} = \frac{\mathrm{s}_{\mathrm{B,D+1}}^{2} + \mathrm{s}_{\mathrm{C,D+1}}^{2}}{\mathrm{s}_{\mathrm{A,D+1}}^{2} + \mathrm{s}_{\mathrm{B,D+1}}^{2} + \mathrm{s}_{\mathrm{C,D+1}}^{2}}$$

$$\mathrm{EnsembleWeight}_{\mathrm{B}}^{\mathrm{D+1}} = \frac{\mathrm{s}_{\mathrm{A,D+1}}^{2} + \mathrm{s}_{\mathrm{C,D+1}}^{2}}{\mathrm{s}_{\mathrm{A,D+1}}^{2} + \mathrm{s}_{\mathrm{B,D+1}}^{2} + \mathrm{s}_{\mathrm{C,D+1}}^{2}}$$

$$\mathrm{EnsembleWeight}_{\mathrm{C}}^{\mathrm{D+1}} = \frac{\mathrm{s}_{\mathrm{A,D+1}}^{2} + \mathrm{s}_{\mathrm{B,D+1}}^{2}}{\mathrm{s}_{\mathrm{A,D+1}}^{2} + \mathrm{s}_{\mathrm{B,D+1}}^{2} + \mathrm{s}_{\mathrm{C,D+1}}^{2}}$$

This leads to a combined forecast that is based on the weighted rolling sum of the daily energy forecast errors as:

$$\begin{aligned} &\mathrm{CombinedForecast}_{\mathrm{D+1,i}} \\ &= \mathrm{EnsembleWeight}_{\mathrm{D+1}}^{\mathrm{A}}\mathrm{Forecast}_{\mathrm{D+1,i}}^{\mathrm{A}} + \mathrm{EnsembleWeight}_{\mathrm{D+1}}^{\mathrm{B}}\mathrm{Forecast}_{\mathrm{D+1,i}}^{\mathrm{B}} \\ &+ \mathrm{EnsembleWeight}_{\mathrm{D+1}}^{\mathrm{C}}\mathrm{Forecast}_{\mathrm{D+1,i}}^{\mathrm{C}} \end{aligned}$$

Predictive Performance. The combined forecast described above relies on historical forecast errors to predict future forecast performance. The seasonal outperformance method and the Gradient Boost method demonstrate the value of using forecast errors that are specific to the conditions prevailing on the period being forecasted. We can extend this idea to operational load forecasting by focusing on the forecast performance for days with similar weather and calendar conditions as the forecast day. For example, temperature forecasts can be used to weight one load forecast more than another based on the relative forecast performance of the models to predict loads under different temperature conditions. Cloud cover forecasts can be used to select the forecast model approach that predicts loads the best under forecasted cloud conditions. Calendar conditions can be used to predict relative forecast performance on weekdays versus weekends, or spring season versus winter season.

To add predictive performance to the combining framework, we can leverage modelling approaches like regression, neural network models, and decision trees to predict the relative forecast performance of each ensemble model given the day-ahead weather and calendar condition forecasts. For example, if the weighted rolling sum of squared forecast errors are used for the combination weight, it is possible to predict future squared forecast errors as a function of forecasted Heating Degree Days (HDD) and Cooling Degree Days (CDD) by estimating the following regression model:

$$\mathrm{e}_{\mathrm{n,d}}^2 = \theta_0 + \theta_1\ \mathrm{HDD_d} + \theta_2\ \mathrm{CDD_d} + \mathrm{u_d}$$

Here, the time series of squared day-ahead forecast errors ($\mathrm{e}_{\mathrm{n,d}}^2$) for Ensemble Forecast (n) is regressed on an intercept term and historical HDD and CDD. The estimated regression model can then be used to predict the day-ahead (D+1) squared forecast error as follows:

$$\hat{e}_{\mathrm{n,D+1}}^2 = \hat{\theta}_0 + \hat{\theta}_1\ \mathrm{HDD}_{\mathrm{D+1}}^f + \hat{\theta}_2\ \mathrm{CDD}_{\mathrm{D+1}}^f$$

Here,

$\mathrm{HDD}_{\mathrm{D+1}}^f$ is the day-ahead forecast of Heating Degree Days

$\mathrm{CDD}_{\mathrm{D+1}}^f$ is the day-ahead forecast of Cooling Degree Days

In a similar fashion, a regression model of the absolute forecast errors can be estimated and used to predict future absolute forecast errors, which can then feed the Outperformance Indicator. Referring to the example of three alternative load forecast models, we can apply the idea of developing predictive forecast errors by estimating the following three auxiliary regression models.

$$\begin{aligned}\mathrm{e}_{\mathrm{A,d}}^2 = \theta_0 &+ \theta_1\ \mathrm{Weekend_d} + \theta_2\ \mathrm{HDD_d} + \theta_3\ \mathrm{CDD_d} + \theta_4\ \mathrm{Cloudy_d} + \theta_5\ \mathrm{Sunny_d} + \theta_6\ \mathrm{HDD_d\ Cloudy_d} \\ &+ \theta_7\ \mathrm{HDD_d\ Sunny_d} + \theta_8\ \mathrm{CDD_d\ Cloudy_d} + \theta_9\ \mathrm{CDD_d\ Sunny_d} + \mathrm{u_d}\end{aligned}$$

$$\begin{aligned}\mathrm{e}_{\mathrm{B,d}}^2 = \theta_0 &+ \theta_1\ \mathrm{Weekend_d} + \theta_2\ \mathrm{HDD_d} + \theta_3\ \mathrm{CDD_d} + \theta_4\ \mathrm{Cloudy_d} + \theta_5\ \mathrm{Sunny_d} + \theta_6\ \mathrm{HDD_d\ Cloudy_d} \\ &+ \theta_7\ \mathrm{HDD_d\ Sunny_d} + \theta_8\ \mathrm{CDD_d\ Cloudy_d} + \theta_9\ \mathrm{CDD_d\ Sunny_d} + \mathrm{u_d}\end{aligned}$$

$$\begin{aligned}\mathrm{e}_{\mathrm{C,d}}^2 = \theta_0 &+ \theta_1\ \mathrm{Weekend_d} + \theta_2\ \mathrm{HDD_d} + \theta_3\ \mathrm{CDD_d} + \theta_4\ \mathrm{Cloudy_d} + \theta_5\ \mathrm{Sunny_d} + \theta_6\ \mathrm{HDD_d\ Cloudy_d} \\ &+ \theta_7\ \mathrm{HDD_d\ Sunny_d} + \theta_8\ \mathrm{CDD_d\ Cloudy_d} + \theta_9\ \mathrm{CDD_d\ Sunny_d} + \mathrm{u_d}\end{aligned}$$

Here,

$\mathrm{Weekend_d}$ is a binary variable that takes on a value of 1.0 if day (d) is a Saturday or Sunday, else 0

$\mathrm{Cloudy_d}$ is a binary variable that takes on a value of 1.0 if day (d) had heavy cloud cover, else 0

$\mathrm{Sunny_d}$ is a binary variable that takes on a value of 1.0 if day (d) had clear skies, else 0

The predictive forecast error models can take on any form including regression, neural networks or Decision tree and utilize any set of explanatory variables. We can use the predicted squared errors to augment the weighting scheme as:

$$\mathrm{EnsembleWeight}^{\mathrm{A}}_{\mathrm{D+1}} = \frac{\mathrm{s}^2_{\mathrm{B,D+1}} + \hat{e}^2_{\mathrm{B,D+1}} + \mathrm{s}^2_{\mathrm{C,D+1}} + \hat{e}^2_{\mathrm{C,D+1}}}{\mathrm{s}^2_{\mathrm{A,D+1}} + \mathrm{s}^2_{\mathrm{B,D+1}} + \mathrm{s}^2_{\mathrm{C,D+1}} + \hat{e}^2_{\mathrm{A,D+1}} + \hat{e}^2_{\mathrm{B,D+1}} + \hat{e}^2_{\mathrm{C,D+1}}}$$

$$\mathrm{EnsembleWeight}^{\mathrm{B}}_{\mathrm{D+1}} = \frac{\mathrm{s}^2_{\mathrm{A,D+1}} + \hat{e}^2_{\mathrm{A,D+1}} + \mathrm{s}^2_{\mathrm{C,D+1}} + \hat{e}^2_{\mathrm{C,D+1}}}{\mathrm{s}^2_{\mathrm{A,D+1}} + \mathrm{s}^2_{\mathrm{B,D+1}} + \mathrm{s}^2_{\mathrm{C,D+1}} + \hat{e}^2_{\mathrm{A,D+1}} + \hat{e}^2_{\mathrm{B,D+1}} + \hat{e}^2_{\mathrm{C,D+1}}}$$

$$\mathrm{EnsembleWeight}^{\mathrm{C}}_{\mathrm{D+1}} = \frac{\mathrm{s}^2_{\mathrm{A,D+1}} + \hat{e}^2_{\mathrm{A,D+1}} + \mathrm{s}^2_{\mathrm{B,D+1}} + \hat{e}^2_{\mathrm{B,D+1}}}{\mathrm{s}^2_{\mathrm{A,D+1}} + \mathrm{s}^2_{\mathrm{B,D+1}} + \mathrm{s}^2_{\mathrm{C,D+1}} + \hat{e}^2_{\mathrm{A,D+1}} + \hat{e}^2_{\mathrm{B,D+1}} + \hat{e}^2_{\mathrm{C,D+1}}}$$

Here,

$\hat{e}^2_{\mathrm{A,D+1}}$ is the forecasted squared forecast error for Ensemble Model A on day (D+1)

$\hat{e}^2_{\mathrm{B,D+1}}$ is the forecasted squared forecast error for Ensemble Model B on day (D+1)

${\hat{e}^2_{\mathrm{C,D+1}}}^2$ is the forecasted squared forecast error for Ensemble Model C on day (D+1)

General Formula-Based Combined Load Forecast. Pulling all the pieces together gives the following general formula-based ensemble weighting framework.

$$\mathrm{EnsembleWeight}^{n}_{D+1} = \alpha \mathrm{EnsembleWeight}^{n}_{D} + (1-\alpha)\left[\frac{\sum_{j=1,j\neq n}^{N-1}\left(\delta \mathrm{Score}^{j}_{D+1} + (1-\delta)\widehat{\mathrm{Score}}^{j}_{D+1}\right)}{\sum_{k=1}^{N}\left(\delta \mathrm{Score}^{k}_{D+1} + (1-\delta)\widehat{\mathrm{Score}}^{k}_{D+1}\right)}\right]$$

Here,

$\mathrm{EnsembleWeight}^{n}_{D+1}$ is the ensemble forecast (n) weight on day (D+1)

$(\mathrm{Score}^{j}_{D+1})$ is the historical forecast performance of ensemble forecast (j) for time (D+1)

$(\widehat{\mathrm{Score}}^{j}_{D+1})$ is the predicted forecast performance of ensemble forecast (j) for time (D+1)

α is a learning rate which controls how quickly the ensemble forecast weight evolves over time

δ is a blending weight that determines the trade-off between historical forecast and forecasted performance

j indexes the full list of N ensemble forecasts except for ensemble forecast (n)

k indexes the full list of N ensemble forecasts

In this case, the historical and predicted forecast performance scores (Score^{j}_{D+1} and $\widehat{\mathrm{Score}}^{j}_{D+1}$) represent whatever forecast performance metric the analyst chooses to implement. Other performance metrics including Bunn's Outperformance Measure and Bates and Granger's Absolute Error Measure can be substituted for the squared error metric depicted above.

Different values for the meta parameters (α, δ, ω, and v) define alternative ensemble model weights. For example,

- Learning Rate (α close to 0) or ($\omega > 1$) implies the ensemble weight reacts quickly to most recent observed and predictive forecast performance trends
- Learning Rate (α close to 1) or ($\omega < 1$) implies the ensemble weight reacts slowly to recent observed and predictive forecast performance trends
- Blending weight ($\delta = 0$) implies the ensemble weight is driven only by predictive forecast performance
- Blending weight ($\delta = 1$) implies the ensemble weight is driven only by observed forecast performance
- Historical range (v) sets the scope over which historical and predicted performance is measured

The final combined forecast can then be written generally as:

$$\mathrm{CombinedForecast}_{D+1,i} = \sum_{n=1}^{N} \mathrm{EnsembleWeight}^{n}_{D+1}\mathrm{Forecast}^{n}_{D+1,i}$$

Here, the combined forecast on day (D+1) for time interval (i) is the weighted sum of the ensemble forecasts.

A strength of the learning framework is the ensemble weights can be re-computed on the fly using a minimal set of information. This will prove beneficial when considering recreating a combined forecast every hour or more frequently to support within day forecasting.

A MODEL-BASED COMBINED LOAD FORECASTING

We return now to Granger and Ramanathan, who put forth an alternative to formula-based weighting schemes by constructing a statistically weighted combined forecast using regression techniques. Building from the example of three ensemble forecasts, we can define a statistically weighted combined forecast regression as follows:

$$\begin{aligned}\mathrm{DailyEnergy_d} &= \omega_0\ \mathrm{Intercept_d} + \omega_A\ \mathrm{DayAheadForecast_d^A} + \omega_B\ \mathrm{DayAheadForecast_d^B}\\ &\quad + \omega_C\ \mathrm{DayAheadForecast_d^C} + \mathrm{e_d}\end{aligned}$$

Here,

$\mathrm{DailyEnergy_d}$ is the total energy for the day (d)

$\mathrm{Intercept_d}$ takes on a value of 1.0 for all days (d)

$\mathrm{DayAheadForecast_d^A}$ is the day-ahead forecast from Ensemble Model A for day (d)

$\mathrm{DayAheadForecast_d^B}$ is the day-ahead forecast from Ensemble Model B for day (d)

$\mathrm{DayAheadForecast_d^C}$ is the day-ahead forecast from Ensemble Model C for day (d)

ω_0 is the weight placed on the Intercept

ω_A is the weight placed on the day-ahead forecast from Ensemble Model A

ω_B is the weight placed on the day-ahead forecast from Ensemble Model A

ω_C is the weight placed on the day-ahead forecast from Ensemble Model A

$\mathrm{e_d}$ is the model error on day (d)

The intercept term is added to model specification to account for the possibility that all the ensemble forecasts are biased low (high). For this model, the weights ($\omega_A, \omega_B, \omega_C$) are not constrained to sum to 1.0. Granger and Ramanathan address this concern by offering a constrained regression approach where the ensemble forecast weights are constrained to sum to 1.0, but it is unclear that imposing that type of constraint is necessary. We will continue with the unconstrained version of the combined model and assume that setting the weights to values that minimize the sum of the squared day-ahead forecast errors is a good thing.

There are limitations with this simple model specification that need to be addressed. First, the ensemble forecast weights are fixed. A clear advantage of the formula-based weighting schemes is the fact the weights evolve over time. Fixed weights could lead to consistent under-forecasting of peak loads. Further, fixed weights most likely will not help correct for ensemble forecasts exhibiting runs of across day under (over) forecast biases.

To free up the weights, interaction terms can be added that will allow the weights to vary by forecasted weather and calendar conditions. For example,

$$
\begin{aligned}
\mathrm{DailyEnergy_d} \;=\; & \omega_0\ \mathrm{Intercept_d} \;+ \omega_A\ \mathrm{DayAheadForecast_d^A} + \omega_B\ \mathrm{DayAheadForecast_d^B} \\
& + \omega_C\ \mathrm{DayAheadForecast_d^C} + \omega_{A,CDD}\mathrm{DayAheadForeast_d^A CDD_d} \\
& + \omega_{A,HDD}\mathrm{DayAheadForeast_d^A HDD_d} \;+ \omega_{B,CDD}\mathrm{DayAheadForeast_d^B CDD_d} \\
& + \omega_{B,HDD}\mathrm{DayAheadForeast_d^B HDD_d} \;+ \omega_{C,CDD}\mathrm{DayAheadForeast_d^C CDD_d} \\
& + \omega_{C,HDD}\mathrm{DayAheadForeast_d^C HDD_d} \;+ \omega_{A,Wkend}\mathrm{DayAheadForeast_d^A Weekend_d} \\
& + \omega_{B,WkEnd}\mathrm{DayAheadForeast_d^B Weekend_d} \;+ \omega_{C,WkEnd}\mathrm{DayAheadForeast_d^C Weekend_d} \;+ e_d
\end{aligned}
$$

Here, the weather conditions (i.e., CDD and HDD) and day-of-the-week (i.e., Weekend binary) variables are interacted with the ensemble forecasts to allow the contribution of each ensemble forecast to vary with respect to calendar and weather conditions. If one (or more) of the ensemble forecasts is designed to predict loads under peak conditions, then the interaction terms can be designed to swing weight toward this/these forecast(s) during peak producing weather and calendar conditions. This will help improve the combined day-ahead peak forecast.

There are a couple of ways to help correct for short-term across-day forecast error bias trends. The first option is to include a lagged dependent variable as an explanatory variable. In this case, any consistent rise or fall in the loads will be captured by the one-day lagged autoregressive term. The second option is to augment the ensemble model forecasts with their respective day-ahead forecast errors. For example,

$$\mathrm{AugmentedDayAheadForecast_d^{A,AR1}} = \mathrm{DayAheadForecast_d^A} + \mathrm{DayAheadForecastError_{d-1}^A}$$

Here, an autoregressive (AR) augmented Model A ensemble forecast is constructed by adding to the raw Model A day-ahead forecast ($\mathrm{DayAheadForecast_d^A}$) the error from the previous day ($\mathrm{DayAheadForecastError_{d-1}^A}$). This idea can be extended to include the two-day back forecast errors and the three-day back forecast errors as follows:

$$\mathrm{AugmentedDayAheadForecast_d^{A,AR2}} = \mathrm{DayAheadForecast_d^A} + \mathrm{DayAheadForecastError_{d-2}^A}$$

$$\mathrm{AugmentedDayAheadForecast_d^{A,AR3}} = \mathrm{DayAheadForecast_d^A} + \mathrm{DayAheadForecastError_{d-3}^A}$$

Rather than including the lagged errors in one ensemble forecast, we construct three separate augmented forecasts so that the weight placed on the one-day, two-day, and three-day back errors can vary. This follows the theme of a regression with ARMA error corrections, but instead we are treating the ARMA error component as alternative ensemble forecasts. The intent of adding these autoregressive augmented forecasts is to control for short-term forecast error bias trends. A similar set of augmented ensemble forecasts can be added to the formula-based weighting schemes to help control for short-term forecast bias trends.

Since the goal of the combined model is to produce an accurate load forecast, there is no reason not to add other explanatory variables that could help improve the forecast performance of the combining equation. In fact, the sky is the limit in terms of how to best define the combination model. Nonlinear combinations of the alternative forecasts can be included, or the regression model can be replaced with a neural network specification. Since there is no theoretical framework for choosing the optimal set of explanatory variables or features, the forecast analyst is left with trial and error to determine a set that works well for the problem of interest. Alternatively, or in addition to, AdaBoost can be applied to the estimation process as a means of stress testing the estimated coefficient under alternative observation weighting.

STATISTICALLY ADJUSTED FORMULA-BASED COMBINED LOAD FORECAST

Let's return to the general formula-based combined forecast definition.

$$\mathrm{CombinedForecast}_{D+1,i} = \sum_{n=1}^{N} \mathrm{EnsembleWeight}_{D+1}^{n} \mathrm{Forecast}_{D+1,i}^{n}$$

The strength of this approach is the ensemble weights are dynamic and can be designed to reflect different forecast performance metrics. The Achilles Heel of this framework is the ensemble forecasts. If all the ensemble forecasts are weak predictors of peak loads or are prone to short-run forecast error bias trends, then the resulting combined forecast will also be subject to these errors.

Now let's return to the regression-based approach offered by Granger and Ramanathan, which can be written generally as follows:

$$\mathrm{CombinedForecast}_{D+1,i} = \mathrm{F}\left(\mathrm{EnsembleForecast}_{D+1,i}, \mathrm{X}_{D+1,i}\right)$$

Here, the day-ahead (D+1) Combined Forecast for time interval (i) is a statistical combination of the (N) ensemble forecasts and a set of explanatory variables that describe the calendar and weather conditions for forecast day (D+1) and interval (i).

The strength of this approach is that a set of explanatory variables can be added to the regression specification that could account for potential peak forecast errors and possible forecast error bias trends. The challenge of this approach is defining a set of interaction terms to ensure the weights placed on the ensemble forecasts are not static.

It would be nice to have a framework that combines the strength of both approaches. To do this, let's return to the general formula-based weighting scheme which is rewritten here.

$$\mathrm{EnsembleWeight}_{D+1}^{n} = \alpha \mathrm{EnsembleWeight}_{D}^{n} + (1-\alpha)\left[\frac{\sum_{j=1, j\neq n}^{N-1}\left(\delta \mathrm{Score}_{D+1}^{j} + (1-\delta)\widehat{\mathrm{Score}}_{D+1}^{j}\right)}{\sum_{k=1}^{N}\left(\delta \mathrm{Score}_{D+1}^{k} + (1-\delta)\widehat{\mathrm{Score}}_{D+1}^{k}\right)}\right]$$

We can use this general formula to construct a set of formula-based weighted ensemble forecasts as follows:

$$\mathrm{WeightedEnsembleForecast}_{d,i}^{n} = \mathrm{EnsembleWeight}_{d,i}^{n} \mathrm{Forecast}_{d,i}^{n}\ , \qquad \forall\ n = 1 \text{ to } N$$

We can then redefine the Granger and Ramathan combined regression as a function of the weighted ensemble forecasts instead of the unweighted ensemble forecasts. The combined function can be written generally as:

$$\mathrm{CombinedForecast}_{D+1,i} = F\left(\mathrm{WeightedEnsembleForecast}_{D+1,i}, X_{D+1,i}\right)$$

Following the above example of three ensemble forecasts (A, B, and C), we can write down an illustrative combining equation as:

$$\begin{aligned}\mathrm{CombinedForecast}_{d,i} &= \omega_{0,i}\mathrm{Intercept}_{d,i} + \omega_{A,i}\mathrm{EnsembleWeight}^{A}_{d,i}\mathrm{Forecast}^{A}_{d,i} \\ &+ \omega_{B,i}\mathrm{EnsembleWeight}^{B}_{d,i}\mathrm{Forecast}^{B}_{d,i} + \omega_{C,i}\mathrm{EnsembleWeight}^{C}_{d,i}\mathrm{Forecast}^{C}_{d,i} + \beta_i\ X_{d,i} + e_{d,i}\end{aligned}$$

Here, the parameters $(\omega_{A,i}, \omega_{B,i}, \omega_{C,i})$ provide statistical adjustment to the formula-based ensemble weights for each ensemble forecast and time interval.

In this case, the weight placed on the ensemble forecast continues to evolve as described by the formula-based weighting scheme implemented by the forecast analyst. At the same time, the statistical adjustment parameters correct for systematic forecast bias. As described above, interaction terms can be introduced as needed to help improve the overall performance of the combined forecast.

What do we gain from this approach? First, the weights on the ensemble are dynamically driven by the selected formula-based approach. Second, the combined forecast is ensured to be unbiased even if the ensemble forecasts are biased. Third, factors other than the ensemble forecasts can be included to help the overall performance of the combined forecast. In other words, it exploits the strengths of the formula-based and regression-based weighting schemes.

When estimating the statistically adjusted regression, weighted least squares can be used. This sets up the idea of designing observation weights that place more weight on days with similar calendar and weather conditions as the forecast day. Each day a new set of regression weights can be estimated based on observation weights driven by forecasted calendar and weather conditions. Within a day, the regression weights remain fixed unless there is a significant change in the weather. The formula-based weights can continue to evolve throughout the day as the performance of each forecast ensemble is scored against actual loads. In effect, the regression weights favour forecast ensembles that have strong forecast performance on days like the day being forecasted. The dynamically evolving formula-based weights capture real-time forecast performance and shift weight to those ensemble forecasts that are exhibiting strong within day performance.

A REAL-LIFE EXAMPLE

To help fix ideas, let's walk through a real-life example. In this example, the forecast horizon is for the balance-of-the-current day (d) out two days ahead. The periodicity of the load data is 15-minutes. Over the past six months, the control room has noticed an uptick in load volatility during daylight hours. The current thinking is the ramp up of solar PV installations is the main casual factor driving the volatility.

Until recently, system operations worked against two alternative forecasts. Both forecasts are derived from the same fully specified load forecast model. To quantify the load forecast uncertainty due to weather forecast uncertainty, weather forecasts from two weather service providers are pushed through the load forecast model. This approach has performed well until recently when it has been noticed that on clear sunny days, the forecasts tend to be too high. At the same time, it has been noticed that on heavy cloud days, the forecasts tend to be too low. During days marked by intermittent or partially cloudy days, the load forecasts tend to track the overall load shape well although there are short periods when the forecast errors (both over- and under-forecast errors) can be large.

To account for the impact of rooftop solar PV generation, the forecast team developed two additional model specifications. The first model uses a reconstituted load approach to account for solar PV generation. The second model includes estimates of solar PV generation as additional explanatory variables. Both models utilize solar PV generation estimates and forecasts from a third weather service provider. The reconstituted load model utilizes the weather forecast from weather service A and the direct model uses the weather forecast from weather service provider B. The resulting ensemble of four forecasts are listed below.

1. Ensemble Forecast 1 comes from a fully specified model of 15-minute loads that does not include the impact of behind-the-meter solar PV generation. This forecast is driven by the weather forecasts from Weather Service Provider A.
2. Ensemble Forecast 2 comes from the same fully specified model as Ensemble Forecast 1, but the weather forecasts that drive the forecast comes from Weather Service Provider B.
3. Ensemble Forecast 3 is a variation of the fully specified model from Ensemble Forecast 1, but uses a reconstituted load approach to estimate the model parameters. The weather forecast comes from Weather Service Provider A and the behind-the-meter solar PV generation forecast comes from Weather Service Provider C.
4. Ensemble Forecast 4 is a variation of the fully specified model from Ensemble Forecast 1 where the variation includes forecasts of behind-the-meter solar PV generation as an additional explanatory variable. The weather forecast comes from Weather Service Provider B and the behind-the-meter solar PV generation forecast comes from Weather Service Provider C.

In this example, the load forecasts are updated every 15 minutes and they incorporate the most recent 15-minute load measurement and load forecast errors. The sequence of 15-minute ahead, 30-minute ahead, 45-minute ahead, 60-minute ahead, two-hour ahead, four-ahead and eight-hour ahead forecasts are stored and can be used to develop measures of forecast performance.

Thoughts on Additional Ensemble Forecasts. The challenge with the formula-based combined forecast frameworks is they produce combined forecasts that are an average of the ensemble forecast. For cases like peak day forecasting, unless one or more of the ensemble forecasts is designed to produce a peak forecast, the combined forecast will be too low. In a similar fashion, unless one or more of the ensemble forecasts correct for across-day and within-day forecast biases, the combined forecast will be subject to across day and within day biases.

One option to account for too low of a combined peak forecast is to add an ensemble load forecast that is driven by an augmented weather forecast. For example, the augmented weather forecast could be constructed by adding five degrees to the baseline weather forecast. This would give a load forecast that will have additional weather-driven loads.

To account for across-day forecast error bias, an additional ensemble forecast can be formed by adding to one of the existing ensemble forecasts the prior day combined forecast error. In a similar fashion, the within-day forecast error bias can be used to from another augmented load forecast.

Day-Ahead Combined Load Forecast. Most system operators start with a day-ahead load forecast that supports day-ahead scheduling of grid-connected generation. Typically, the day-ahead load forecast upon which the day-ahead scheduling process runs is computed during the morning of the prior day. Once the day-ahead load forecast is sent to the day-ahead scheduling desk, the forecast focus turns to forecasting loads for the balance-of-the-day. Ensemble forecast weights that predict how well each ensemble forecast will perform given forecasted weather and calendar conditions are used to form the day-ahead combined load forecast.

For this example, it is anticipated that Ensemble Forecast 1 and 2, which ignore the impact of solar PV generation, will over-forecast loads on clear sunny days and under-forecast loads on heavy dark cloudy days. The forecast performance of these two ensemble forecasts is expected to be comparable to Ensemble Forecast 3 and 4 on days marked by intermittent clouds. At the same time, Ensemble Forecast 3 and 4 should perform well on clear sunny days and heavy dark cloud days due to the explicit accounting of the solar PV load impact. This information can be captured statistically by estimating a set of auxiliary day-ahead forecast performance models like the following:

$$\begin{aligned}\mathrm{e}_{\mathrm{n,d}}^2 = \theta_0 &+ \theta_1\,\mathrm{Weekend_d} + \theta_2\,\mathrm{HDD_d} + \theta_3\,\mathrm{CDD_d} + \theta_4\,\mathrm{Cloudy_d} + \theta_5\,\mathrm{Sunny_d} + \theta_6\,\mathrm{HDD_d\,Cloudy_d} \\ &+ \theta_7\,\mathrm{HDD_d\,Sunny_d} + \theta_8\,\mathrm{CDD_d\,Cloudy_d} + \theta_9\,\mathrm{CDD_d\,Sunny_d} + \mathrm{u_d}\end{aligned}$$

Here,

$\mathrm{e}_{\mathrm{n,d}}^2$ is the squared day-ahead daily energy forecast error for forecast ensemble (n) on day (d)

We can then use the estimated equations to forecast the squared day-ahead forecast error for each forecast ensemble. We use these forecasted squared errors to score each forecast. In the following example, the model with the smallest forecasted squared error is assigned a value of 1.0 and the other three are assigned a value of 0.0.

$$\widehat{\mathrm{Score}}_{1,\mathrm{d}+1} = \begin{cases} 1.0, & \mathrm{MIN}\left(\hat{e}^2_{1,\mathrm{d}+1}, \hat{e}^2_{2,\mathrm{d}+1}, \hat{e}^2_{3,\mathrm{d}+1}, \hat{e}^2_{4,\mathrm{d}+1}\right) = \hat{e}^2_{1,\mathrm{d}+1} \\ 0.0, & \mathrm{MIN}\left(\hat{e}^2_{1,\mathrm{d}+1}, \hat{e}^2_{2,\mathrm{d}+1}, \hat{e}^2_{3,\mathrm{d}+1}, \hat{e}^2_{4,\mathrm{d}+1}\right) \neq \hat{e}^2_{1,\mathrm{d}+1} \end{cases}$$

$$\widehat{\mathrm{Score}}_{2,\mathrm{d}+1} = \begin{cases} 1.0, & \mathrm{MIN}\left(\hat{e}^2_{1,\mathrm{d}+1}, \hat{e}^2_{2,\mathrm{d}+1}, \hat{e}^2_{3,\mathrm{d}+1}, \hat{e}^2_{4,\mathrm{d}+1}\right) = \hat{e}^2_{2,\mathrm{d}+1} \\ 0.0, & \mathrm{MIN}\left(\hat{e}^2_{1,\mathrm{d}+1}, \hat{e}^2_{2,\mathrm{d}+1}, \hat{e}^2_{3,\mathrm{d}+1}, \hat{e}^2_{4,\mathrm{d}+1}\right) \neq \hat{e}^2_{2,\mathrm{d}+1} \end{cases}$$

$$\widehat{\mathrm{Score}}_{3,\mathrm{d}+1} = \begin{cases} 1.0, & \mathrm{MIN}\left(\hat{e}^2_{1,\mathrm{d}+1}, \hat{e}^2_{2,\mathrm{d}+1}, \hat{e}^2_{3,\mathrm{d}+1}, \hat{e}^2_{4,\mathrm{d}+1}\right) = \hat{e}^2_{3,\mathrm{d}+1} \\ 0.0, & \mathrm{MIN}\left(\hat{e}^2_{1,\mathrm{d}+1}, \hat{e}^2_{2,\mathrm{d}+1}, \hat{e}^2_{3,\mathrm{d}+1}, \hat{e}^2_{4,\mathrm{d}+1}\right) \neq \hat{e}^2_{3,\mathrm{d}+1} \end{cases}$$

$$\widehat{\mathrm{Score}}_{4,\mathrm{d}+1} = \begin{cases} 1.0, & \mathrm{MIN}\left(\hat{e}^2_{1,\mathrm{d}+1}, \hat{e}^2_{2,\mathrm{d}+1}, \hat{e}^2_{3,\mathrm{d}+1}, \hat{e}^2_{4,\mathrm{d}+1}\right) = \hat{e}^2_{4,\mathrm{d}+1} \\ 0.0, & \mathrm{MIN}\left(\hat{e}^2_{1,\mathrm{d}+1}, \hat{e}^2_{2,\mathrm{d}+1}, \hat{e}^2_{3,\mathrm{d}+1}, \hat{e}^2_{4,\mathrm{d}+1}\right) \neq \hat{e}^2_{4,\mathrm{d}+1} \end{cases}$$

These scores are then used to set the ensemble weights as:

$$\mathrm{EnsembleWeight}^{\mathrm{n}}_{\mathrm{d}+1} = \frac{\widehat{\mathrm{Score}}^{\mathrm{n}}_{\mathrm{d}+1}}{\sum_{\mathrm{k}=1}^{\mathrm{N}} \widehat{\mathrm{Score}}^{\mathrm{k}}_{\mathrm{d}+1}}, \forall\ \mathrm{i}$$

Here, the weight placed on the day-ahead forecast of ensemble (n) is proportional to its score. In this case, the same weight is applied to all time intervals (i).

This winner-takes-all scheme could be replaced with any number of schemes. For example, a scheme could be designed that assigns a weight of 0.0 to the ensemble forecast with the largest predicted sum of squared errors and a weight of 1.0 for all other forecasts. This scheme would cull out the worst forecast and then weight equally the remaining forecasts.

$$\widehat{\mathrm{Score}}^{1}_{\mathrm{d}+1} = \begin{cases} 0.0, & \mathrm{MAX}\left(\hat{e}^2_{1,\mathrm{d}+1}, \hat{e}^2_{2,\mathrm{d}+1}, \hat{e}^2_{3,\mathrm{d}+1}, \hat{e}^2_{4,\mathrm{d}+1}\right) = \hat{e}^2_{1,\mathrm{d}+1} \\ 1.0, & \mathrm{MAX}\left(\hat{e}^2_{1,\mathrm{d}+1}, \hat{e}^2_{2,\mathrm{d}+1}, \hat{e}^2_{3,\mathrm{d}+1}, \hat{e}^2_{4,\mathrm{d}+1}\right) \neq \hat{e}^2_{1,\mathrm{d}+1} \end{cases}$$

$$\widehat{\mathrm{Score}}^{2}_{\mathrm{d}+1} = \begin{cases} 0.0, & \mathrm{MAX}\left(\hat{e}^2_{1,\mathrm{d}+1}, \hat{e}^2_{2,\mathrm{d}+1}, \hat{e}^2_{3,\mathrm{d}+1}, \hat{e}^2_{4,\mathrm{d}+1}\right) = \hat{e}^2_{2,\mathrm{d}+1} \\ 1.0, & \mathrm{MAX}\left(\hat{e}^2_{1,\mathrm{d}+1}, \hat{e}^2_{2,\mathrm{d}+1}, \hat{e}^2_{3,\mathrm{d}+1}, \hat{e}^2_{4,\mathrm{d}+1}\right) \neq \hat{e}^2_{2,\mathrm{d}+1} \end{cases}$$

$$\widehat{\mathrm{Score}}^{3}_{\mathrm{d}+1} = \begin{cases} 0.0, & \mathrm{MAX}\left(\hat{e}^2_{1,\mathrm{d}+1}, \hat{e}^2_{2,\mathrm{d}+1}, \hat{e}^2_{3,\mathrm{d}+1}, \hat{e}^2_{4,\mathrm{d}+1}\right) = \hat{e}^2_{3,\mathrm{d}+1} \\ 1.0, & \mathrm{MAX}\left(\hat{e}^2_{1,\mathrm{d}+1}, \hat{e}^2_{2,\mathrm{d}+1}, \hat{e}^2_{3,\mathrm{d}+1}, \hat{e}^2_{4,\mathrm{d}+1}\right) \neq \hat{e}^2_{3,\mathrm{d}+1} \end{cases}$$

$$\widehat{\mathrm{Score}}^{4}_{\mathrm{d}+1} = \begin{cases} 0.0, & \mathrm{MAX}\left(\hat{e}^2_{1,\mathrm{d}+1}, \hat{e}^2_{2,\mathrm{d}+1}, \hat{e}^2_{3,\mathrm{d}+1}, \hat{e}^2_{4,\mathrm{d}+1}\right) = \hat{e}^2_{4,\mathrm{d}+1} \\ 1.0, & \mathrm{MAX}\left(\hat{e}^2_{1,\mathrm{d}+1}, \hat{e}^2_{2,\mathrm{d}+1}, \hat{e}^2_{3,\mathrm{d}+1}, \hat{e}^2_{4,\mathrm{d}+1}\right) \neq \hat{e}^2_{4,\mathrm{d}+1} \end{cases}$$

These ensemble weights can then be used directly to form a combined load forecast as:

$$\mathrm{Combined}_{d+1,i} = \sum_{n=1}^{N} \mathrm{EnsembleWeight}_{d+1}^{n} \mathrm{Forecast}_{d+1,i}^{n}$$

Or included as part of statistically adjusted combined forecast as:

$$\mathrm{Combined}_{d+1,i} = \sum_{n=1}^{N} \omega_i^n \mathrm{EnsembleWeight}_{d+1}^{n} \mathrm{Forecast}_{d+1,i}^{n} + \ \beta_i \ X_{d+1,i}$$

Within-Day Combined Load Forecast. Once the day-ahead load forecast has been sent to the day-ahead scheduling desk, the focus turns to the balance-of-the-day forecasts. Most systems have a time window within which the stack of available grid-connected generation is fixed. For this example, we will assume this window is two hours ahead. This means there is a premium on the accuracy of the sequence of two-hour ahead load forecasts. We define a successful forecast as a two-hour ahead forecast error that is less than an acceptable error tolerance (e.g. +/- 50MW). Specifically,

$$\mathrm{Score}_{d,i-1}^{n} = \left(\left|\mathrm{Load}_{d,i-1} - \mathrm{Forecast}_{d,i-1}^{n,2HrAhead}\right| \leq \tau_{d,i-1}\right)$$

Here,

$\mathrm{Score}_{d,i-1}^{n}$ the forecast performance score for forecast ensemble (n) will take on a value of 1.0 if the two-ahead forecast for time interval (d, i-1) is within acceptable forecast error tolerances as defined by ($\tau_{d,i-1}$)

$\tau_{d,i-1}$ is a user defined acceptable forecast error which is allowed to vary by day (d) and time (i-1)

The ensemble weight is then computed as:

$$\mathrm{EnsembleWeight}_{d,i}^{n} = \alpha \mathrm{EnsembleWeight}_{d,i-1}^{n} + (1-\alpha)\left[\frac{\mathrm{Score}_{d,i-1}^{n}}{\sum_{k=1}^{N} \mathrm{Score}_{d,i-1}^{k}}\right]$$

In this case, if two or more ensemble forecasts have acceptable forecast errors, then the weight is shared across the winning forecasts. This will then be added to the prior ensemble weight based on the value of the learning parameter, α. The forecast analyst is tasked with defining a value for the learning parameter that balances the stability of the sequence of the combined load forecasts and near-term accuracy. As will be discussed below, this balancing act will be tested with deeper saturations of behind-the-meter solar PV.

To initialize the weighting scheme, we can use the ensemble weights used to compute the day-ahead forecast for the current day, or we can re-run the day-ahead predictive models with the most recently available current day forecasted weather and calendar conditions. That is:

$$\mathrm{EnsembleWeight}_{\mathrm{d}}^{\mathrm{n}} = \frac{\widehat{\mathrm{Score}}_{\mathrm{d}}^{\mathrm{n}}}{\sum_{\mathrm{k}=1}^{\mathrm{N}} \widehat{\mathrm{Score}}_{\mathrm{d}}^{\mathrm{k}}}$$

Ideally, the Ensemble Weights would be re-initialized whenever there is a significant change in the current day weather forecast. That way, the weights placed on the ensemble forecasts are given a chance to reset quickly to evolving weather conditions.

For the first 15-minute interval of the current day, we have:

$$\mathrm{EnsembleWeight}_{\mathrm{d,i}}^{\mathrm{n}} = \alpha \mathrm{EnsembleWeight}_{\mathrm{d}}^{\mathrm{n}} + (1-\alpha)\left[\frac{\mathrm{Score}_{\mathrm{d,i-1}}^{\mathrm{n}}}{\sum_{\mathrm{k}=1}^{\mathrm{N}} \mathrm{Score}_{\mathrm{d,i-1}}^{\mathrm{k}}}\right]$$

These ensemble weights can then be used directly to form a within-day combined load forecast as:

$$\mathrm{Combined}_{\mathrm{d,i}} = \sum_{\mathrm{n}=1}^{\mathrm{N}} \mathrm{EnsembleWeight}_{\mathrm{d,i}}^{\mathrm{n}} \mathrm{Forecast}_{\mathrm{d,i}}^{\mathrm{n}}$$

Or included as part of statistically adjusted combined forecast as:

$$\mathrm{Combined}_{\mathrm{d,i}} = \sum_{\mathrm{n}=1}^{\mathrm{N}} \omega_{\mathrm{i}}^{\mathrm{n}} \mathrm{EnsembleWeight}_{\mathrm{d,i}}^{\mathrm{n}} \mathrm{Forecast}_{\mathrm{d,i}}^{\mathrm{n}} + \beta_{\mathrm{i}}\, \mathrm{X}_{\mathrm{d,i}}$$

Does Behind-the-Meter Solar PV Become a Game Changer? The combined forecast approaches described above are sums of several moving parts. The challenge is striking a balance between leveraging the information contained in the most recent load forecast errors versus the information contained in the forecasted weather, solar PV, and calendar conditions. Often the ensemble forecast models are, by design, highly autoregressive (e.g., the model for load at 08:00 is a function of loads at 07:45, 07:30, 07:15). The autoregressive nature of these models means the load forecasts from one iteration to the next (e.g., every 15 minutes a new load forecast is generated) could swing significantly up and down if the measured loads that drive the autoregressive terms are bouncing around. Depending on how far out into the forecast horizon the autoregressive components drive the load forecast values, we can end up in a situation where near-term load volatility could be driving load forecast instability two to 24 hours ahead. This is very much like the tail (e.g., the morning measured loads) wagging the dog (i.e., the afternoon load forecast). We have seen cases of morning load volatility – driven by solar PV generation volatility as a result of rapid movement of cloud cover over a control region – leading to afternoon peak forecasts swinging up and down with each iteration of the load forecast. The resulting instability in the afternoon peak forecasts leads to pre-scheduling of additional grid-connected generation to cover the load uncertainty.

Before deep penetration of behind-the-meter solar PV, morning temperature patterns and associated morning loads were correlated with afternoon temperature patterns and associated afternoon loads. In this case, the relatively stable autoregressive terms from the morning hours worked in a positive fashion to improve the afternoon forecast performance. With deeper penetrations of behind-the-meter solar PV, the temperature patterns remain correlated, but the associated load patterns are not as tightly bound to temperatures on days with partially or intermittent cloud cover. As a result, the autoregressive terms that contributed positively to the afternoon forecasts before solar PV, become liabilities with solar PV.

There are a couple actions that can be taken to improve forecast stability. Smoothing of the measured loads that feed the autoregressive terms will dampen unexpected swings associated with rapid cloud movement. The choice of smoothing window width (e.g., 30-minute smoothing, 45-minute smoothing, 60-minute smoothing) will determine how much of the load volatility driven by fast moving cloud cover is reduced. Wide windows lead to smooth loads. However, wide windows could lead to missing key turning points in the loads. This means, window width is another lever that the forecast analyst can set as part of the combination framework.

In addition to pre-smoothing of measured loads, the ensemble weighting scheme can be designed to dampen the impact that near-term autoregressive terms have on longer forecast horizons. To achieve this end, we re-write the general formula-based weighting scheme as follows:

$$\mathrm{EnsembleWeight}^{n}_{d,i+h} = \alpha \mathrm{EnsembleWeight}^{n}_{d,i-1} + (1-\alpha)\left[\frac{\delta^{h}\,\mathrm{Score}^{n}_{d,i-1} + \left(1-\delta^{h}\right)\widehat{\mathrm{Score}}^{n}_{d,i+h}}{\sum_{k=1}^{N}\delta^{h}\,\mathrm{Score}^{k}_{d,i-1} + \sum_{k=1}^{N}\left(1-\delta^{h}\right)\widehat{\mathrm{Score}}^{k}_{d,i+h}}\right]$$

Here,

> $\mathrm{EnsembleWeight}_{d,i+h}^{n}$ is the weight placed on the h-step ahead forecast from ensemble forecast (n) that was produced at day (d) and time (i)
>
> α is a learning rate that determines how quickly the ensemble weights evolve
>
> $\mathrm{Score}_{d,i-1}^{n}$ is the unnormalized historical performance score for ensemble forecast (n) at day (d) and time interval i-1
>
> $\widehat{\mathrm{Score}}_{d,i+h}^{n}$ is the unnormalized h-step ahead forecast performance score for ensemble forecast (n) made at day (d) and time interval (i)
>
> δ^{h} is the h-step ahead weight placed on the unnormalized historical performance score

Following the previous example, $\widehat{\mathrm{Score}}_{d,i+h}^{n}$ would be set according to the expected day-ahead forecast performance of ensemble forecast (n). This can be extended to include the forecast performance of the two-hour ahead, three-hour ahead, …, to day-ahead forecast performance.

To help build intuition, let's consider the following special cases.

Learning Rate ($\alpha = 0$) and ($\delta \quad = 1$). Under this case, the ensemble weight reduces to:

$$\mathrm{EnsembleWeight}_{d,i+h}^{n} = \frac{\mathrm{Score}_{d,i-1}^{n}}{\sum_{k=1}^{N} \mathrm{Score}_{d,i-1}^{k}}$$

In this case, the h-step ahead ensemble weight depends only on the relative historical performances of the competing ensemble forecasts. Under this case, volatile measured loads will lead to forecast instability that is driven through the autoregressive terms in the models.

Learning Rate ($\alpha = 0$) and ($\delta \quad = 0$). Under this case, the ensemble weight reduces to:

$$\mathrm{EnsembleWeight}_{d,i+h}^{n} = \frac{\widehat{\mathrm{Score}}_{d,i+h}^{n}}{\sum_{k=1}^{N} \widehat{\mathrm{Score}}_{d,i+h}^{k}}$$

In this case, the h-step ahead ensemble weight depends only on the relative forecasted performances of the competing ensemble forecasts. Under this case, volatile measure loads will have limited impact on forecast instability for longer forecast horizons.

Learning Rate ($\alpha = 0$) and ($\delta = 0.5$). Under this case, the ensemble weight reduces to:

$$\mathrm{EnsembleWeight}_{d,i+h}^{n} = \left[\frac{0.5^{h}\ \mathrm{Score}_{d,i-1}^{n} + \left(1 - 0.5^{h}\right)\widehat{\mathrm{Score}}_{d,i+h}^{n}}{\sum_{k=1}^{N} 0.5^{h}\ \mathrm{Score}_{d,i-1}^{k} + \sum_{k=1}^{N}\left(1 - 0.5^{h}\right)\widehat{\mathrm{Score}}_{d,i+h}^{k}}\right]$$

In this example, for forecast horizons of less than six hours ahead, the ensemble weight depends on both the historical and forecasted performances of the competing ensemble forecasts. For longer forecast horizons, only the forecasted performances matter. The trick is to decide values for α and δ that balance the information contained in the recent observed forecast performance and the projected forecast performance.

Behind-the-meter solar PV introduces the very real possibility that the ensemble model forecast used for the morning hours is not the ensemble model forecast used for the afternoon hours. For example, a morning with heavy to partially cloudy skies may best be forecasted with a model that ignores solar PV generation altogether, while a sunny, hot afternoon is best handled with a model that includes solar PV generation as an explanatory variable. This suggests the model scores should be based on Outperformance measures instead of sum of squares. Like cloud cover, which leads to knife edge swings in loads, the weighting scheme needs to adjust quickly to evolving cloud movements.

SUMMARY

The econometric and operations research literature on ensemble learning focuses on how to optimally weight together an ensemble of forecasts. Alternative forecast performance metrics are introduced, but ultimately the weighting schemes place value on recent performance. The Data Science literature takes a different approach by introducing methods for constructing an ensemble of forecasts. How the ensemble is constructed then leads to rules for weighting the forecasts together to form a combined forecast. The key takeaways from the literature review are:

- It is possible to construct a linear combination of a set of alternative forecasts that will outperform the individual forecasts upon which the combination is built.
- Three ways of measuring historical forecast performance have been suggested:
    - Squared Forecast Errors,
    - Absolute Forecast Errors, and
    - Outperformance
- There is benefit from weighting recent forecast performance heavier using adaptive learning or observation weighting.
- There is benefit from using weights designed to predict performance under forecasted conditions such as season.
- Linear regression approaches can be extended to include not only the original alternative forecasts, but also weighted combinations of the original forecasts.
- It is possible to convert a single model into a sequence of models where the estimated parameters of each model in the sequence are informed by the errors from the previous model. The resulting ensemble can lead to a combined forecast that outperforms the original fixed weighted model.
- Building a set of complementary models can lead to improved forecast performance. The challenge is being clever in the design of the complementary models. Modelling the errors from a primary model and then using the auxiliary model to forecast the error expected from the primary model would replace the need for manual bias adjustments.

Taking a step back, the focus of most of the literature reviewed in this paper is on weighting schemes that produce an optimal, one-step ahead forecast. The focus of operational load forecasting is on the full sequence of forecasts that span the balance-of-the-day out several days ahead. To address the unique nature of the load forecasting problem, the following concepts were introduced:

- Measures of predictive performance extend the weighting schemes to balance recent historical forecast performance with expectations about future forecast performance given forecasted weather and calendar conditions.
- Adding additional ensemble forecasts designed to correct for recent forecast bias trends, as well as under-forecasting of peak loads, will improve the overall performance of the combined forecast.
- Load data smoothing and weighting schemes that trade-off historical forecast performance for predictive forecast performance can improve load forecast stability.

Ultimately, what matters is how well the ensemble forecasts predict loads. This requires building powerful forecast models that capture the key drivers of load consumption patterns.

Bibliography

Bates, J. and Granger, C. W. J., (1969), “The combination of forecasts”, Operations Research Quarterly 20, pp. 451-468.

G. A. Barnard in his paper, *New methods of quality control*, Journal of the Royal Statistical Society, Series A, Vol. 126 No. 2 (1963), pp. 255-258.

Bunn, D. W. (1975), “A Bayesian approach to the linear combination of forecasts”, Operational Research Quarterly 26, 325-329.

Elliott, G., (2011), “Averaging and the Optimal Combination of Forecasts”, University of California, San Diego, (September 27, 2011).

Granger, C. W. J., and Ramanathan, R. (1984), “Improved methods of forecasting”, Journal of Forecasting 3, pp. 197-204.

Lilian M. de Menezes, Goldsmiths College, University of London and Derek W. Bunn and James W. Taylor, London Business School, “Review of Guidelines for the Use of Combined Forecasts”, European Journal of Operational Research, 2000, Vol. 120, pp. 190-204.

Ye Ren, P.N. Suganthan, and N. Srikanth, Ensemble methods for wind and solar power forecasting – A state-of-the-art review, Renewable and Sustainable Energy Reviews 50 (2015) pp. 82-91

APPENDIX A. BARNARD DATA

In his paper, Barnard used two alternative forecast models: an Adaptive Forecasting Model (a form of an exponential smoothing model) and a Box-Jenkins (ARIMA) model to forecast the monthly series of miles flown. In both cases, the one-step ahead forecasts are based only on historical passenger miles flown data. The data and forecasts published by Barnard, which formed the basis of the analysis performed by Bates and Granger, are presented in the Appendix. Passenger Miles Flown, Adaptive Forecast, and Box-Jenkins Forecast – G. A. Barnard. The original data published by G. A. Barnard are presented below.

Year	Month	Passenger Miles Flown (000s)	Adaptive Forecasting Forecast	Box-Jenkins Forecast
1951	1	145	136.0	134.0
1951	2	150	135.0	158.0
1951	3	178	157.0	167.0
1951	4	163	158.0	165.0
1951	5	172	162.0	158.0
1951	6	178	190.0	189.0
1951	7	199	213.0	206.0
1951	8	199	213.0	194.0
1951	9	184	185.0	187.0
1951	10	162	164.0	163.0
1951	11	146	145.0	141.0
1951	12	166	165.0	171.0
1952	1	171	171.0	174.0
1952	2	180	166.0	174.0
1952	3	193	193.0	207.0
1952	4	181	188.0	185.0
1952	5	183	190.0	186.0
1952	6	218	214.0	194.0
1952	7	230	242.0	231.0
1952	8	242	242.0	238.0
1952	9	209	212.0	217.0
1952	10	191	190.0	191.3
1952	11	172	165.0	171.6
1952	12	194	188.0	194.5
1953	1	196	195.0	199.0
1953	2	196	190.0	206.0
1953	3	236	218.0	212.0
1953	4	235	217.0	213.0
1953	5	229	226.0	238.0
1953	6	243	260.0	265.0
1953	7	264	288.0	254.0
1953	8	272	288.0	270.0
1953	9	237	249.0	248.0
1953	10	211	220.0	221.0
1953	11	180	192.0	192.0
1953	12	201	214.0	208.0
1954	1	204	218.0	207.0
1954	2	188	209.0	210.0
1954	3	235	236.0	236.0
1954	4	227	227.0	235.0
1954	5	234	227.0	231.0
1954	6	264	255.0	248.0
1954	7	302	285.0	284.0
1954	8	293	288.0	304.0
1954	9	259	249.0	255.0
1954	10	229	221.0	226.0
1954	11	203	194.0	201.0
1954	12	229	220.0	222.0
1955	1	242	229.0	229.0
1955	2	233	226.0	222.0
1955	3	267	265.0	275.0
1955	4	269	261.0	256.0
1955	5	370	369.0	369.0
1955	6	315	306.0	303.0
1955	7	364	346.0	345.0
1955	8	347	350.0	350.0
1955	9	312	306.0	309.0
1955	10	274	273.0	275.0
1955	11	237	241.0	249.0
1955	12	278	273.0	265.0

Year	Month	Passenger Miles Flown (000s)	Adaptive Forecasting Forecast	Box-Jenkins Forecast
1956	1	284	283.0	287.0
1956	2	277	277.0	282.0
1956	3	317	321.0	308.0
1956	4	313	316.0	317.0
1956	5	318	323.0	318.0
1956	6	374	368.0	359.0
1956	7	413	413.0	421.0
1956	8	405	398.0	399.0
1956	9	355	359.0	363.0
1956	10	306	317.0	322.0
1956	11	271	277.0	270.0
1956	12	306	313.0	313.0
1957	1	315	321.0	318.0
1957	2	301	312.0	308.0
1957	3	356	358.0	368.0
1957	4	348	350.0	349.0
1957	5	355	357.0	356.0
1957	6	422	405.0	408.0
1957	7	465	452.0	458.0
1957	8	467	453.0	457.0
1957	9	404	395.0	412.0
1957	10	347	351.0	356.0
1957	11	305	308.0	311.0
1957	12	336	349.0	343.0
1958	1	340	358.0	349.0
1958	2	318	347.0	329.0
1958	3	362	397.0	377.0
1958	4	348	383.0	360.0
1958	5	363	384.0	360.0
1958	6	435	430.0	434.0
1958	7	491	475.0	482.0
1958	8	505	471.0	490.0
1958	9	404	407.0	439.0
1958	10	359	355.0	354.0
1958	11	310	310.0	307.0
1958	12	337	347.0	353.0
1959	1	360	355.0	343.0
1959	2	342	345.0	332.0
1959	3	406	399.0	388.0
1959	4	396	393.0	384.0
1959	5	420	405.0	407.0
1959	6	472	464.0	486.0
1959	7	548	517.0	530.0
1959	8	559	519.0	553.0
1959	9	463	451.0	462.0
1959	10	407	402.0	414.0
1959	11	362	354.0	358.0
1959	12	405	402.0	387.0
1960	1	417	419.0	426.0
1960	2	391	410.0	402.0
1960	3	419	474.0	452.0
1960	4	461	457.0	418.0
1960	5	472	470.0	474.0
1960	6	535	533.0	537.0
1960	7	622	596.0	601.0
1960	8	606	598.0	628.0
1960	9	508	514.0	518.0
1960	10	461	454.0	448.0
1960	11	390	399.0	417.0
1960	12	432	448.0	444.0

Join us in creating a more **resourceful world**.
To learn more visit **itron.com**

CORPORATE HQ

2111 North Molter Road
Liberty Lake, WA 99019 USA

Phone: 1.800.635.5461
Fax: 1.509.891.3355