AMERICAN MADE SOLAR PRIZE

U.S. DEPARTMENT OF ENERGY

Net Load Forecasting Technical Webinar



This webinar will be recorded and posted on HeroX

Housekeeping

Two Options for Audio (select audio mode):

1. Listen through your computer:
   Click the ‘up arrow’ next to the “mute” button in the bottom left corner.
   Under “Select a Speaker,” click “Same as System.”
2. Listen by telephone:
   Click the ‘up arrow’ next to the “mute” button in the bottom left corner.
   Click “Switch to Phone Audio.”

To Ask a Question:

- Select the ‘Q&A’ button at the bottom of your screen and type in your question.
- Questions may be answered live, but all questions will be answered in writing on HeroX.

Having Trouble with the Webinar?

- Technical difficulties: Chat the webinar host for additional support.
- A video/audio recording of this webinar and the slide deck will be made available

Topics

1. Net Load Forecasting Prize Overview
2. EPRI
3. Next Steps
4. Q&A



Net Load Forecasting Prize Overview



What is the Net Load Forecasting Prize?

About the Prize:

- Seeking **probabilistic models that predict future amounts of net-load** a day in advance of the forecast, while promoting the adoption of probabilistic forecasts and evaluation tools for such forecasts.

- This prize offers a total of up to **$600,000 in cash prizes**, with three anticipated winners and three anticipated runners-up.

- Every day for four weeks (28 consecutive days), competitors will submit probabilistic net-load forecasts for each of the predetermined substations through the Solar Forecast Arbiter platform.

Prize Goals

The Net Load Forecasting Prize aims to:

(1) Increase stakeholder awareness of the state of the art in net load forecasting,

(2) Demonstrate the feasibility of fair and high-quality evaluations of probabilistic net load forecasts using a publicly available, open-source platform, and

(3) Promote the use of probabilistic forecast models and an industry-common evaluation platform with transparent metrics and specifications for probabilistic net load forecasts.



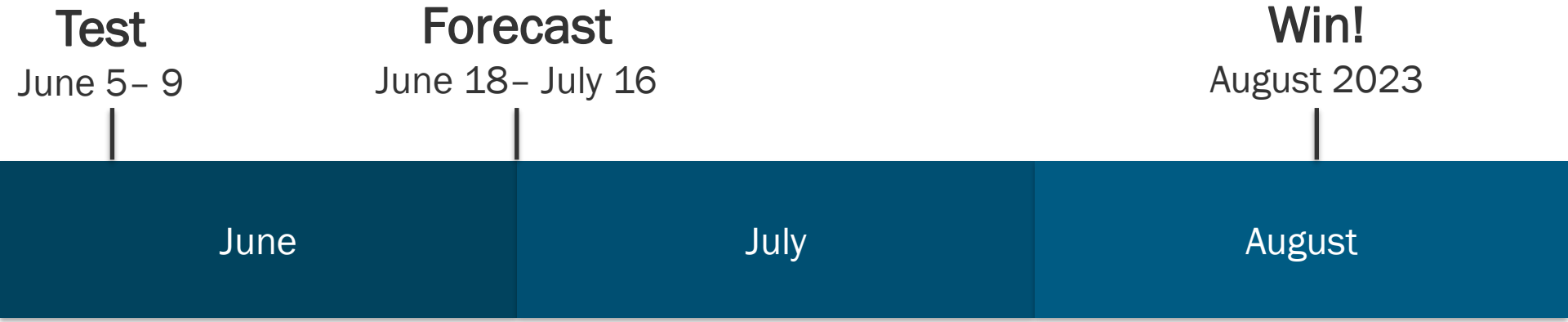
Prizes to Win

The Net Load Forecasting Prize offers a total prize pool of $600,000 in cash.

	Number of Prizes Awarded	Prizes
Winners	Up to three anticipated cash prizes	First: $200,000 Second: $150,000 Third: $100,000
Runners-up	Up to three anticipated cash prizes	Fourth, fifth, and sixth each receive $50,000

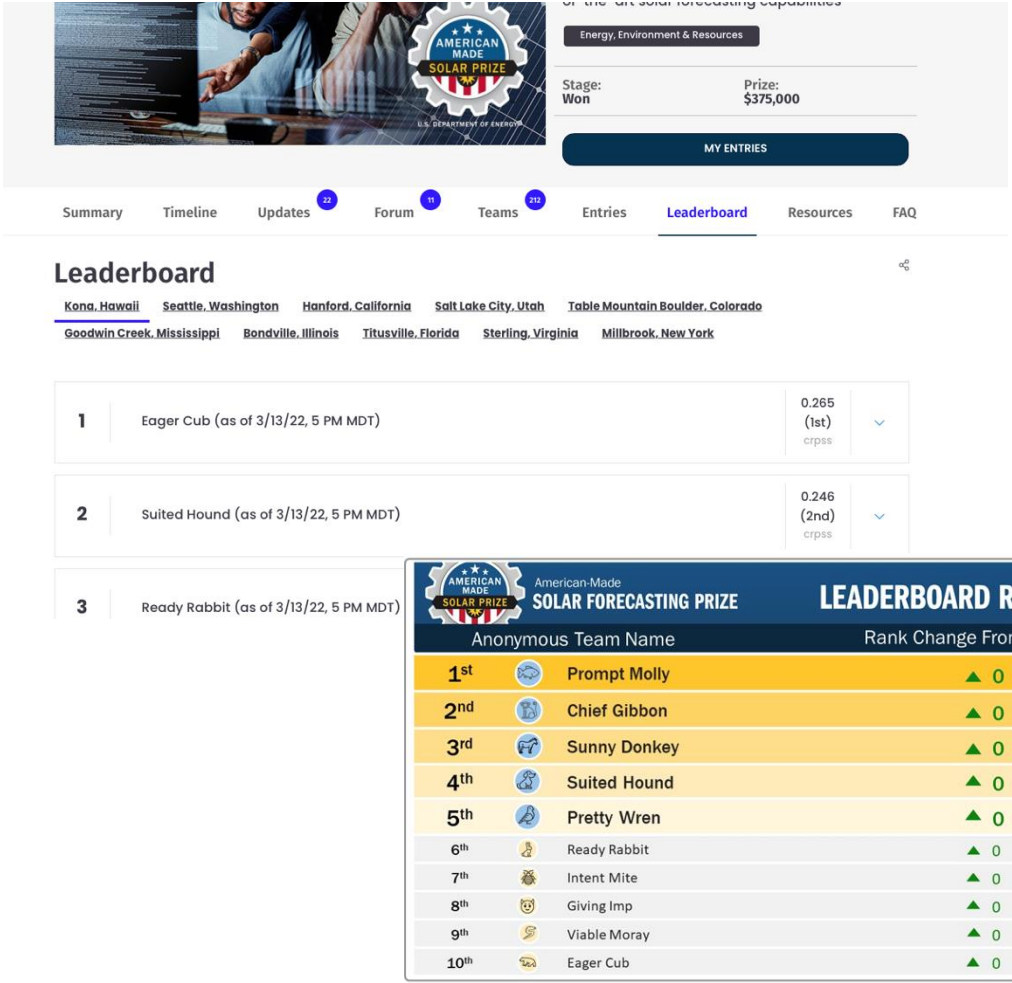
Important Dates

- **Solar Forecast Arbiter Test Week (via SFA platform):** June 5, 2023, through June 9, 2023
- **Forecast Submission Period (via SFA platform):** June 18, 2023, until July 16, 2023
- **Winners and Awards Announced:** August 2023 (anticipated)



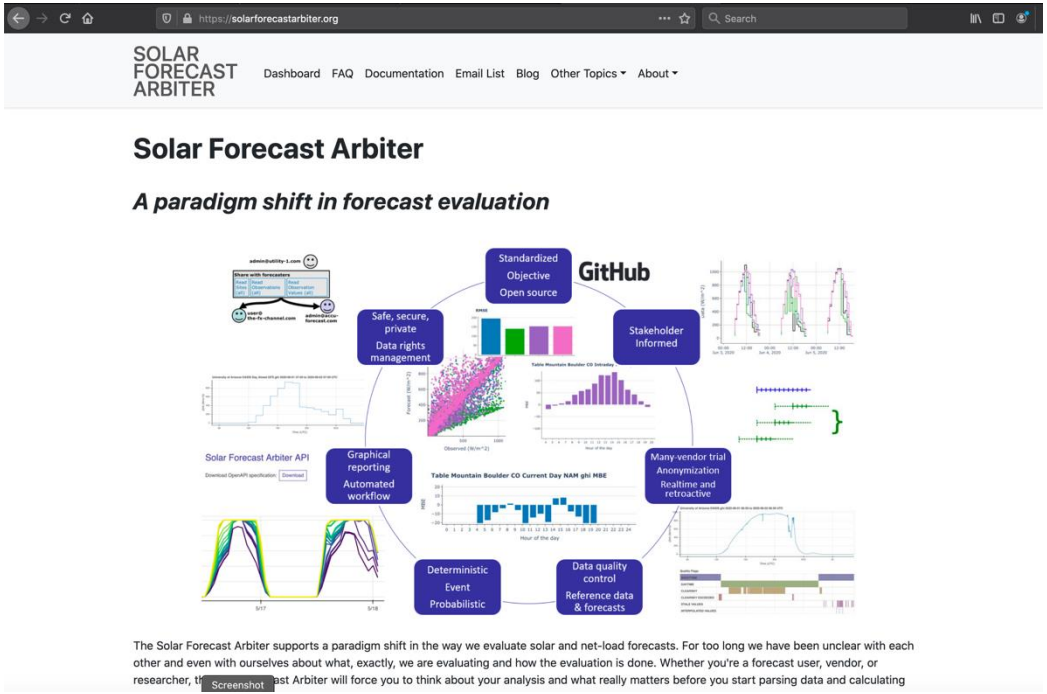
Leaderboard

- Throughout the competition, we expect to maintain a public, anonymized leaderboard.
- Teams will be able to see how they are performing against other teams.
- Following the competition, we will only reveal the names of winning teams.



Solar Forecast Arbiter – Partnership with EPRI

- The Net Load Forecasting Prize will use the Solar Forecast Arbiter (SFA), an open-source, cloud-ready platform that facilitates the evaluation of forecasts for solar irradiance, solar power, and net load.
- As co-sponsor of the prize, EPRI will provide historical net load data for competitors to train forecasting models, as well as providing operational support for the forecast evaluation platform used during the evaluation phase of the prize.



https://forecastarbiter.epri.com

EPRI

David Larson, PhD



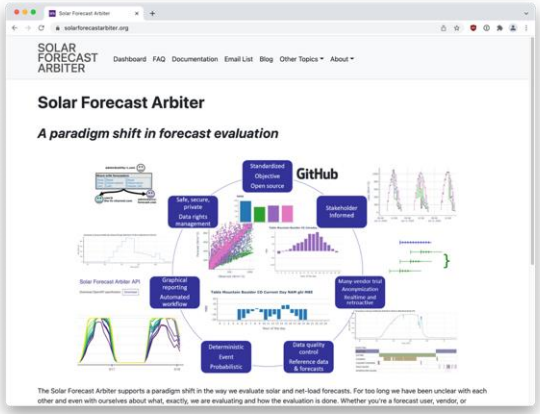
Today's tutorial will help you understand:

1. the process (start-to-finish) for submitting forecasts
2. where to find more info on the Arbiter platform
3. practical considerations and solutions to common issues

Note: there will be a test week to allow you to validate your forecast submission scripts

Arbiter: Components

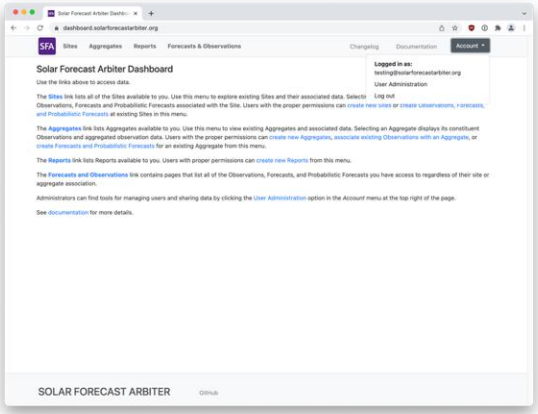
(1) Info site



Documentation, guides, etc.
(publicly available)

forecastarbiter.epri.com

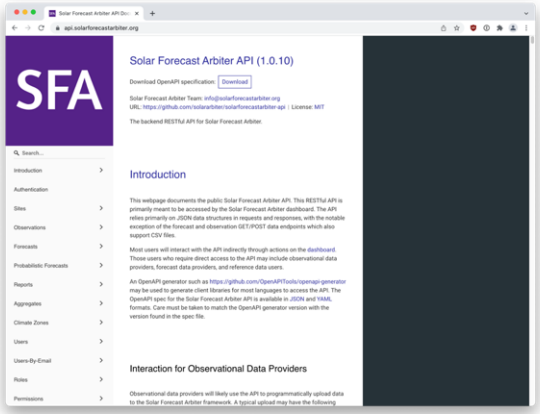
(2) Dashboard



Web interface
(secure access)

dashboard.forecastarbiter.com

(3) API



Programmatic interface
(secure access)

api.forecastarbiter.com

Arbiter User Access for the NLF Prize

- **each team is assigned an anonymized name in the Arbiter**
  - e.g., Team Diamond and Team Ruby instead of Company XYZ and University ABC

- **each team receives one set of credentials** specific to the NLF Prize
  - credentials are sent by EPRI after signing the NDA

- **authentication (username + password) is handled using Auth0**

Arbiter Data Model

- a **Site** describes a physical location (name, lat-lon coordinates, timezone, etc.)
  - example: the NOAA SOLRAD site in Seattle, WA

- an **Observation** contains time-series data (observations) and specify the data's associated variable, units, etc. and **Site**
  - example: GHI [W/m^2] observations at the Seattle SOLRAD site

- a **Forecast** contains time-series data (forecasts) and specify the data's associated variable, units, etc. and **Site**
  - example: day-ahead forecast of temperature [°C] at the Seattle SOLRAD site

Arbiter Data Model (cont'd)

- a Probabilistic Forecast constant value is like a Forecast but is associated with a probability
  - ex: Prob(Power ≤ X MW) = 50% ➔ predict MW value for p50

- a Probabilistic Forecast is a collection of Probabilistic Forecast constant value objects that (together) described a cumulative distribution function (CDF)
  - ex: Prob(Power ≤ X_1 MW) = 10%, Prob(Power ≤ X_2 MW) = 20%, …
    ➔ forecast predicts X_1 [MW], X_2 [MW], …

Arbiter Data Model Impacts on NLF Prize

1. each **Site** has an **Observation**, whose associated time-series data will be used in evaluating forecast accuracy

2. you will submit forecasts per **Site** for each probability
   - one **Probabilistic Forecast** per **Site**
   - one **Probabilistic Forecast constant value** per probability in each **Probabilistic Forecast**
   - 4 sites * 11 percentiles = 44 forecast values per hour

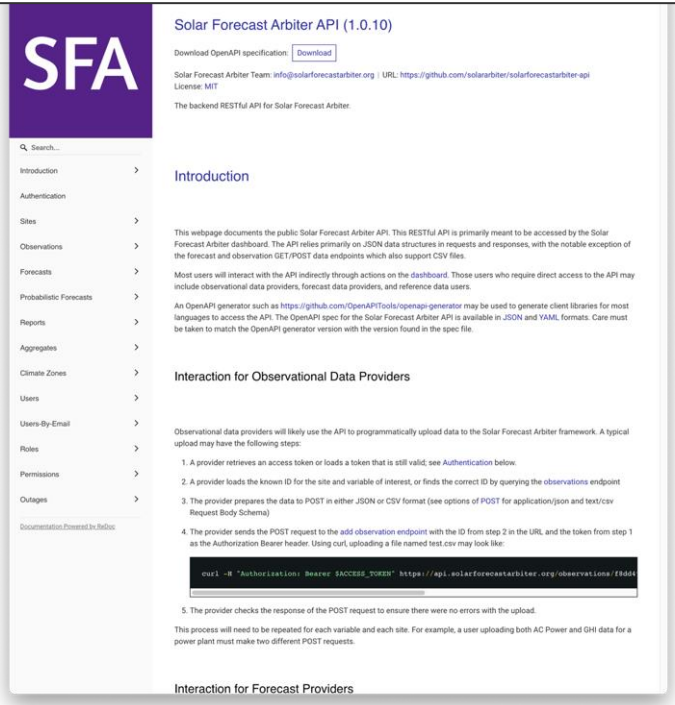
3. each **Site**, **Observation**, etc. has a unique ID (UUID), which is shown in the SFA Dashboard
   - the UUID values are required for many API requests
   - example UUID: af1e53ca-7e3d-11e9-ab45-52540015d5ce

Application Programming Interface (API)

- Arbiter has a RESTful API:
  - allows interacting with the Arbiter programmatically
  - essential for automated forecast submission

- Arbiter provides a Python wrapper for the API:
  - part of the solarforecastarbiter package*
  - `solarforecastarbiter.io.api`

https://api.forecastarbiter.com



* https://solarforecastarbiter-core.readthedocs.io/

“Barebones” Example: Single Prob. Power Forecast

```python
import pandas as pd
from solarforecastarbiter.io import api

# get API token
token = api.request_cli_access_token(
    "foo@bar.com",  # username
    "hunter2"       # password
)

# init an API session
session = api.APISession(token, base_url="https://api.forecastarbiter.com")

# forecast data to be submitted
forecast_id = "983691e4-080a-11ea-8a5f-0a580a8003a5"   # ID of the forecast
forecast_series = pd.Series(
    index=[pd.Timestamp("2022-01-20 15:00")],   # timestamp
    data=[10.321]                               # forecast value [MW]
)

# upload the forecast data
session.post_probabilistic_forecast_constant_value_values(
    forecast_id, forecast_series
)
```

1. get API token

2. initialize API session

3. organize data

4. upload data

“Barebones” Example: Single Prob. Power Forecast

```python
import pandas as pd
from solarforecastarbiter.io import api

# get API token
token = api.request_cli_access_token(
    "foo@bar.com",  # username
    "hunter2"       # password
)

# init an API session
session = api.APISession(token, base_url="https://api.forecastarbiter.com")

# forecast data to be submitted
forecast_id = "983691e4-080a-11ea-8a5f-0a580a8003a5"   # ID of the forecast
forecast_series = pd.Series(
    index=[pd.Timestamp("2022-01-20 15:00")],   # timestamp
    data=[10.321]                               # forecast value [MW]
)

# upload the forecast data
session.post_probabilistic_forecast_constant_value_values(
    forecast_id, forecast_series
)
```

1. get API token

2. initialize API session

3. organize data

4. upload data

“Barebones” Example: Single Prob. Power Forecast

```python
import pandas as pd
from solarforecastarbiter.io import api

# get API token
token = api.request_cli_access_token(
    "foo@bar.com",  # username
    "hunter2"       # password
)

# init an API session
session = api.APISession(token, base_url="https://api.forecastarbiter.com")

# forecast data to be submitted
forecast_id = "983691e4-080a-11ea-8a5f-0a580a8003a5"   # ID of the forecast
forecast_series = pd.Series(
    index=[pd.Timestamp("2022-01-20 15:00")],   # timestamp
    data=[10.321]                               # forecast value [MW]
)

# upload the forecast data
session.post_probabilistic_forecast_constant_value_values(
    forecast_id, forecast_series
)
```

1. get API token

2. initialize API session

3. organize data

4. upload data

“Barebones” Example: Single Prob. Power Forecast

```python
import pandas as pd
from solarforecastarbiter.io import api

# get API token
token = api.request_cli_access_token(
    "foo@bar.com",  # username
    "hunter2"       # password
)

# init an API session
session = api.APISession(token, base_url="https://api.forecastarbiter.com")

# forecast data to be submitted
forecast_id = "983691e4-080a-11ea-8a5f-0a580a8003a5"   # ID of the forecast
forecast_series = pd.Series(
    index=[pd.Timestamp("2022-01-20 15:00")],   # timestamp
    data=[10.321]                               # forecast value [MW]
)

# upload the forecast data
session.post_probabilistic_forecast_constant_value_values(
    forecast_id, forecast_series
)
```

1. get API token

2. initialize API session

3. organize data

4. upload data

A More “Realistic” Example*

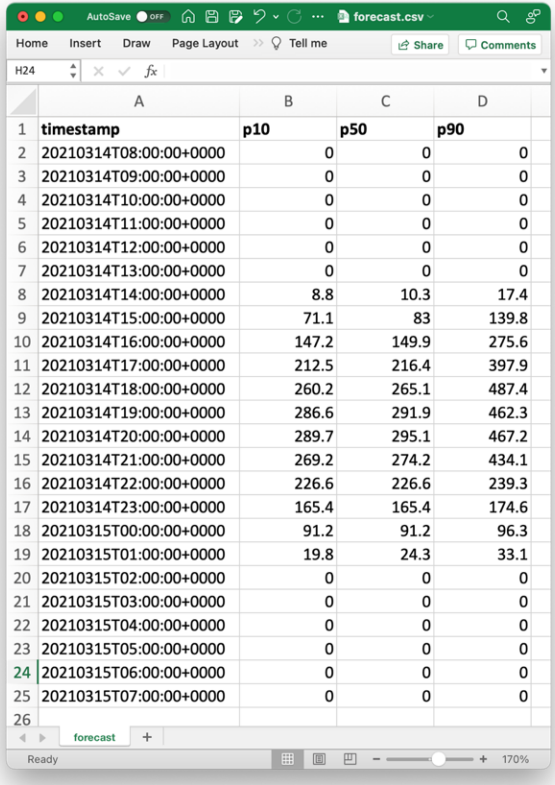
Assumptions:

- forecasts already generated
- forecast saved as CSV with one column per probability (e.g., p10 = 10%)

Steps to submit forecasts via API:

1. get API token and init API session
2. load forecast data for all probabilities
3. for each probability: upload data

*this is not the only way to submit forecasts, but is more realistic compared to the “barebones” example



	A	B	C	D
1	timestamp	p10	p50	p90
2	20210314T08:00:00+0000	0	0	0
3	20210314T09:00:00+0000	0	0	0
4	20210314T10:00:00+0000	0	0	0
5	20210314T11:00:00+0000	0	0	0
6	20210314T12:00:00+0000	0	0	0
7	20210314T13:00:00+0000	0	0	0
8	20210314T14:00:00+0000	8.8	10.3	17.4
9	20210314T15:00:00+0000	71.1	83	139.8
10	20210314T16:00:00+0000	147.2	149.9	275.6
11	20210314T17:00:00+0000	212.5	216.4	397.9
12	20210314T18:00:00+0000	260.2	265.1	487.4
13	20210314T19:00:00+0000	286.6	291.9	462.3
14	20210314T20:00:00+0000	289.7	295.1	467.2
15	20210314T21:00:00+0000	269.2	274.2	434.1
16	20210314T22:00:00+0000	226.6	226.6	239.3
17	20210314T23:00:00+0000	165.4	165.4	174.6
18	20210315T00:00:00+0000	91.2	91.2	96.3
19	20210315T01:00:00+0000	19.8	24.3	33.1
20	20210315T02:00:00+0000	0	0	0
21	20210315T03:00:00+0000	0	0	0
22	20210315T04:00:00+0000	0	0	0
23	20210315T05:00:00+0000	0	0	0
24	20210315T06:00:00+0000	0	0	0
25	20210315T07:00:00+0000	0	0	0
26				

Figure: Example CSV with p10, p50 and p90 forecasts for next 24 hours.

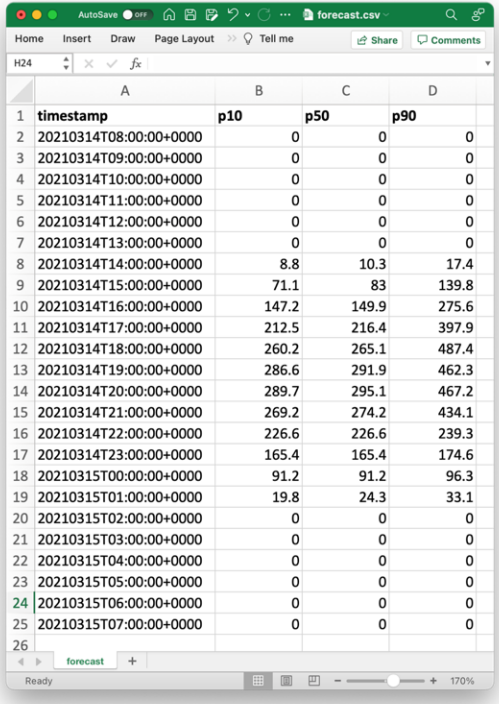
A More “Realistic” Example (cont’d)

```python
import pandas as pd
from solarforecastarbiter.io import api

# get API token and init API
token = api.request_cli_access_token("foo@bar.com", "hunter2")
session = api.APISession(token, base_url="https://api.forecastarbiter.com")

# load probabilistic forecasts as pandas.DataFrame
df = pd.read_csv(
    "forecast.csv", parse_dates=[0], index_col=0
)

# submit forecast for each probability one-by-one
forecast_ids = [
    # column name in DataFrame, UUID in SFA
    ["p10", "2fbf2068-7010-11ec-92f9-0a580a80025d"],  # Prob(GHI < X W/m^2) = 10%
    ["p50", "2fc0dd3a-7010-11ec-a1ac-0a580a80025d"],  # Prob(GHI < X W/m^2) = 50%
    ["p90", "2fc2882e-7010-11ec-886c-0a580a80025d"],  # Prob(GHI < X W/m^2) = 90%
]
for col, forecast_id in forecast_ids:
    forecast_series = df[col]  # get data as pandas.Series
    session.post_probabilistic_forecast_constant_value_values(
        forecast_id, forecast_series
    )
```



	A	B	C	D
1	timestamp	p10	p50	p90
2	20210314T08:00:00+0000	0	0	0
3	20210314T09:00:00+0000	0	0	0
4	20210314T10:00:00+0000	0	0	0
5	20210314T11:00:00+0000	0	0	0
6	20210314T12:00:00+0000	0	0	0
7	20210314T13:00:00+0000	0	0	0
8	20210314T14:00:00+0000	8.8	10.3	17.4
9	20210314T15:00:00+0000	71.1	83	139.8
10	20210314T16:00:00+0000	147.2	149.9	275.6
11	20210314T17:00:00+0000	212.5	216.4	397.9
12	20210314T18:00:00+0000	260.2	265.1	487.4
13	20210314T19:00:00+0000	286.6	291.9	462.3
14	20210314T20:00:00+0000	289.7	295.1	467.2
15	20210314T21:00:00+0000	269.2	274.2	434.1
16	20210314T22:00:00+0000	226.6	226.6	239.3
17	20210314T23:00:00+0000	165.4	165.4	174.6
18	20210315T00:00:00+0000	91.2	91.2	96.3
19	20210315T01:00:00+0000	19.8	24.3	33.1
20	20210315T02:00:00+0000	0	0	0
21	20210315T03:00:00+0000	0	0	0
22	20210315T04:00:00+0000	0	0	0
23	20210315T05:00:00+0000	0	0	0
24	20210315T06:00:00+0000	0	0	0
25	20210315T07:00:00+0000	0	0	0
26				

forecast.csv

A More “Realistic” Example (cont’d)

```python
import pandas as pd
from solarforecastarbiter.io import api

# get API token and init API
token = api.request_cli_access_token("foo@bar.com", "hunter2")
session = api.APISession(token, base_url="https://api.forecastarbiter.com")

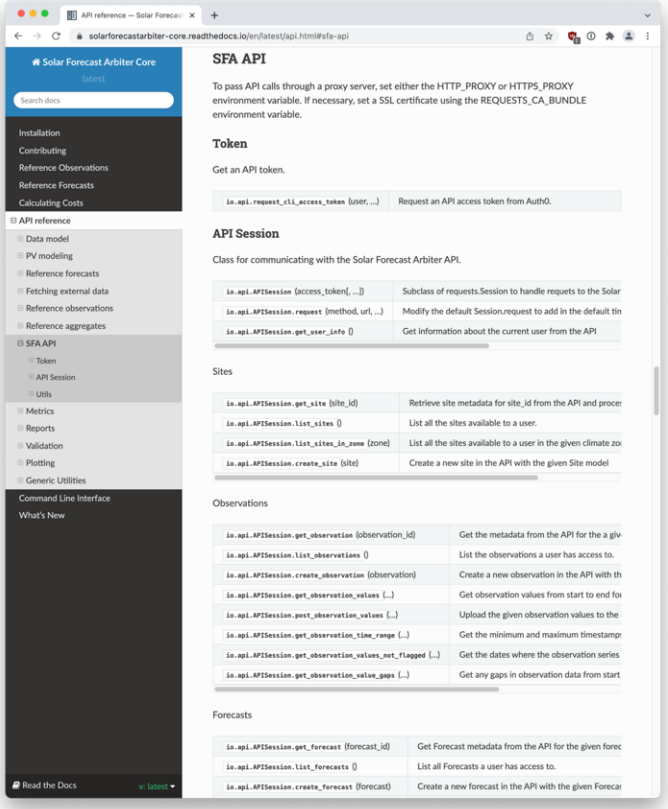
# load probabilistic forecasts as pandas.DataFrame
df = pd.read_csv(
    "forecast.csv", parse_dates=[0], index_col=0
)

# submit forecast for each probability one-by-one
forecast_ids = [
    # column name in DataFrame, UUID in SFA
    ["p10", "2fbf2068-7010-11ec-92f9-0a580a80025d"],  # Prob(GHI < X W/m^2) = 10%
    ["p50", "2fc0dd3a-7010-11ec-a1ac-0a580a80025d"],  # Prob(GHI < X W/m^2) = 50%
    ["p90", "2fc2882e-7010-11ec-886c-0a580a80025d"],  # Prob(GHI < X W/m^2) = 90%
]
for col, forecast_id in forecast_ids:
    forecast_series = df[col]  # get data as pandas.Series
    session.post_probabilistic_forecast_constant_value_values(
        forecast_id, forecast_series
    )
```

1. init API session
2. load data from CSV
3. iterate forecasts
4. upload data

Other Uses of the API

- download historical data for a site
- get metadata about a Site (e.g., lat-lon coordinates), Observation (e.g., interval length), Forecast (e.g., issue time), etc.
- get a list of all sites, observations, etc. that your user has access to
- get a list of UUIDs for all forecasts your user has access to
- get a list of probabilities (“constant values”) for a given Probabilistic Forecast



Note on API Authentication

- Arbiter uses Auth0 with JSON Web Tokens (JWT)
  - tokens last ~3-hours
  - Auth0 rate limits token requests

➔ we **strongly** recommend reusing tokens:

Good: reuse token ✓

```
token = api.request_cli_access_token()
for site in ["X", "Y", "Z"]:
    for forecast_id in forecast_ids:
        submit_forecast()
```

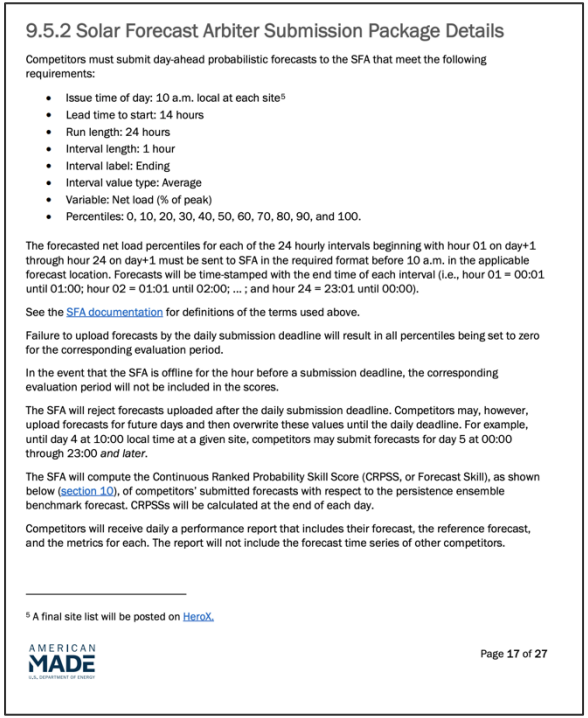
Bad: multiple tokens *X*

```
for site in ["X", "Y", "Z"]:
    for forecast_id in forecast_ids:
        token = api.request_cli_access_token()
        submit_forecast()
```

Note on Forecast Schedules

- EPRI will configure the **ProbabilisticForecast** objects in the Arbiter to match the NLF Prize rules
  - each team will only have access to their own **ProbabilisticForecast** and will not be able to see other teams' forecast data

- during the Test Week and Submission Period, **late forecast submissions will be rejected**
  - the Arbiter will return an error if a forecast is late



9.5.2 Solar Forecast Arbiter Submission Package Details

Competitors must submit day-ahead probabilistic forecasts to the SFA that meet the following requirements:

- Issue time of day: 10 a.m. local at each site[5]
- Lead time to start: 14 hours
- Run length: 24 hours
- Interval length: 1 hour
- Interval label: Ending
- Interval value type: Average
- Variable: Net load (% of peak)
- Percentiles: 0, 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100.

The forecasted net load percentiles for each of the 24 hourly intervals beginning with hour 01 on day+1 through hour 24 on day+1 must be sent to SFA in the required format before 10 a.m. in the applicable forecast location. Forecasts will be time-stamped with the end time of each interval (i.e., hour 01 = 00:01 until 01:00; hour 02 = 01:01 until 02:00; … ; and hour 24 = 23:01 until 00:00).

See the SFA documentation for definitions of the terms used above.

Failure to upload forecasts by the daily submission deadline will result in all percentiles being set to zero for the corresponding evaluation period.

In the event that the SFA is offline for the hour before a submission deadline, the corresponding evaluation period will not be included in the scores.

The SFA will reject forecasts uploaded after the daily submission deadline. Competitors may, however, upload forecasts for future days and then overwrite these values until the daily deadline. For example, until day 4 at 10:00 local time at a given site, competitors may submit forecasts for day 5 at 00:00 through 23:00 *and later*.

The SFA will compute the Continuous Ranked Probability Skill Score (CRPSS, or Forecast Skill), as shown below (section 10), of competitors' submitted forecasts with respect to the persistence ensemble benchmark forecast. CRPSSs will be calculated at the end of each day.

Competitors will receive daily a performance report that includes their forecast, the reference forecast, and the metrics for each. The report will not include the forecast time series of other competitors.

[5] A final site list will be posted on HeroX.

AMERICAN MADE

Page 17 of 27

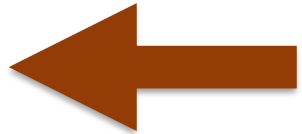
Figure: Screenshot of the NLF rules document.

Other Things to Be Aware Of

- **timestamps and timezones:** the Arbiter interprets timestamps without timezone info as UTC
  - “2023-05-23 10:00:00” = 10am UTC = 3am PDT (UTC-7)
  - “2023-05-23 10:00:00-07:00” = 10am PDT

- **API rate limiting:** the Arbiter has rate limiting to help prevent abuse
  - current limits are 100/min or 3,000/hour (per user)

- **API URL is different:** new URL is https://api.forecastarbiter.com
  - requests to the previous URL will fail



Frequently Asked Questions

- **Q:** What happens if I submit a forecast and then submit a (new) forecast for the same timestamp?
- **A:** The newer (most recently submitted) forecast will overwrite the older forecast.

- **Q:** What happens if I don't submit a forecast in time? Or my forecast submission has missing values?
- **A:** Any missing forecast values will be replaced with zeros.

- **Q:** What if I don't use Python/I want to use another programming language to submit forecasts?
- **A:** The Arbiter's API is a RESTful API and therefore language agnostic. You can submit forecasts via HTTP requests using Java, Julia, R, Ruby, etc.

Frequently Asked Questions (cont'd)

- **Q:** Why can't I upload forecasts via the Arbiter Dashboard (instead of using the API)?
- **A:** The Dashboard does support uploading time-series data (observations and forecasts) through your web browser. However, we recommend automating your uploads using the API as it 1) reflects real-world conditions, 2) helps minimize "human errors" and 3) is more efficient, scalable, etc. than manually uploading (which would uploading one CSV file per site per probability.

- **Q:** Where do I find the forecast ID (UUID) values?
- **A:** UUID values are listed on the Dashboard for each Site, Observation, etc. and are also available via the API.

Frequently Asked Questions (cont'd)

- **Q:** Can I use the Arbiter API to download data?
- **A:** Yes, the API can be used to download data, including the historical/training data for each site. The result is the same as using the Dashboard.

- **Q:** How often is observation data updated in the Arbiter?
- **A:** No guarantees are made re: timing of data availability, but the aim is to update each day, though there may be a lag of ~24–48 hours.

- **Q:** What to do if I get an error with the API (e.g., uploading forecasts fail)?
- **A:** There is not a singular answer but start by reading the error messages (and associated documentation), and then check for common issues (e.g., incorrect user credentials, typo in your code) and confirm whether the error is reproduceable (e.g., test on another computer and/or network connection). If you have exhausted things to check yourself, then contact <forecastarbiter@epri.com>

Next Steps



Next Steps

NDA

- Everyone has been sent the NDA
- Once the NDA is signed, you will receive credentials and can gain access to the utility data

Solar Arbiter

- Log into the Forecast Arbiter to ensure your credentials work
- Once logged in, you should see 4 Sites listed under the "2023 Net Load Forecasting Prize" (the 4 substations)
- Your team name is anonymized using a randomly assigned gemstone, e.g., "Team Diamond" and "Team Ruby" instead of "Company XYZ" and "University ABC"

Office Hours

- Have more questions or need additional assistance?
- Join us for office hours on June 7th

Q&A

- Email Solar.Prize@nrel.gov with any questions.

