

---

# Apollo

Aug 12, 2019



<b>1</b>	<b>Documentation</b>	<b>1</b>
<b>2</b>	<b>Thesis</b>	<b>9</b>
<b>3</b>	<b>Acknowledgements</b>	<b>13</b>



### 1.1 Overview: The *What* and *Why* of Apollo

Apollo is an open source system for machine learning from numerical weather forecasts. Its purpose is to facilitate data collection, model training and evaluation, and the serving of predictions.

- *Use Cases*
  - *Irradiance Forecasting*
  - *Model Evaluation*
- *System Architecture*
  - *Data Ingestion Subsystem*
  - *Modeling Subsystem*
  - *Storage Subsystem*
  - *Service Subsystem*

#### 1.1.1 Use Cases

##### Irradiance Forecasting

---

**Todo:** TODO

---

##### Model Evaluation

## Apollo

---

**Todo:** TODO

---

### 1.1.2 System Architecture

Fig. 1: System Architecture of Apollo

#### Data Ingestion Subsystem

---

**Todo:** TODO

---

#### Modeling Subsystem

---

**Todo:** TODO

---

#### Storage Subsystem

---

**Todo:** TODO

---

#### Service Subsystem

---

**Todo:** TODO

---

## 1.2 Quickstart: Running Apollo with Docker

```
git clone https://github.com/cbarrick/apollo.git
docker build apollo -t apollo
docker run -it apollo
```

## 1.3 Installing Apollo Locally

## 1.4 Python API Reference

### 1.4.1 Modeling Framework

Apollo's core API is the modeling framework.

#### Model Classes

---

```
apollo.models.Model
apollo.models.IrradianceModel
apollo.models.NamModel
```

---

#### Utility Functions

---

```
apollo.models.list_models
apollo.models.list_templates
apollo.models.load_model
apollo.models.load_model_at
apollo.models.load_model_from
apollo.models.make_estimator
apollo.models.make_model
apollo.models.make_model_from
```

---

### 1.4.2 NAM Forecast Data

Apollo uses the North American Mesoscale (NAM) forecast system, a numerical weather simulation produced by the National Oceanic and Atmospheric Administration (NOAA). Apollo can be configured to collect NAM forecasts as training data for machine learning models.

#### Data Access

---

```
apollo.nam.download
apollo.nam.open
apollo.nam.open_range
apollo.nam.CacheMiss
```

---

#### Geographic Coordinates

---

```
apollo.nam.NAM218
apollo.nam.proj_coords
apollo.nam.slice_geo
```

---

#### Useful Constants

---

```
apollo.nam.ATHENS_LATLON
apollo.nam.PLANAR_FEATURES
```

---

## Apollo

---

### See also:

[NAM Home Page](#) Detailed documentation of NAM.

[National Centers for Environmental Information](#) Access to raw NAM forecast data.

[Inventory of File nam.t00z.awphys00.tm00.grib2](#) Catalog of variables included in NAM forecasts. (Apollo does not support every variable.)

### 1.4.3 Feature Extraction

---

`apollo.time_of_day`

---

`apollo.time_of_year`

---

`apollo.is_daylight`

---

### 1.4.4 Time Series Related

Timestamps in Apollo adhere to the following conventions:

- Timestamps are always UTC.
- Timezone-naive inputs are interpreted as UTC.
- Timezone-aware inputs in a different timezone are converted to UTC.

Apollo extends common Pandas utilities to support these conventions.

---

`apollo.Timestamp`

---

`apollo.DatetimeIndex`

---

`apollo.date_range`

---

### 1.4.5 Metrics

---

`apollo.metrics.all`

---

`apollo.metrics.mae`

---

`apollo.metrics.r2`

---

`apollo.metrics.rmse`

---

`apollo.metrics.stdae`

---

### 1.4.6 Visualizations

Apollo includes several visualization routines.

---

`apollo.date_heatmap`

---

`apollo.date_heatmap_figure`

---

`apollo.nam_figure`

---

### 1.4.7 Data Access

Apollo stores models and datasets in the *Apollo database*. The database is a regular directory specified by the `APOLLO_DATA` environment variable, defaulting to `/var/lib/apollo`. In the Apollo Docker image, the database



---

is a volume mounted to `/apollo-data`.

---

`apollo.path`

---

## 1.5 REST API Reference

---

**Todo:** Document

---

## 1.6 Command Line Reference

Apollo provides a command line toolbox for managing forecast data, developing forecast models, and administering the server. All Apollo commands sport a `--help` option with detailed descriptions. This documentation provides an overview of the commands.

- *apollo*
- *apollo ls*
- *apollo predict*
- *apollo train*
- *apollo score*
- *apollo nam download*

### 1.6.1 apollo

#### Summary

The Apollo CLI toolbox.

#### Usage

```
apollo [-h] [--quiet | --debug | --log LEVEL] COMMAND ...
```

#### Description

The `apollo` command provides a *toolbox* style CLI, like `git`. The root `apollo` command takes a single required argument, `COMMAND`, giving the subcommand to execute. Optional arguments that come before the subcommand are applicable to all subcommands, while arguments that come after are specific to the subcommand.

## 1.6.2 apollo ls

### Summary

List items within the Apollo database.

### Usage

```
apollo ls [-h] [COMPONENT]
```

### Description

The `apollo ls` command is for listing different items stored in the Apollo database. You can optionally specify a component to list only those items.

Components include:

- `models`: The trained models.
- `templates`: Templates for training new models.
- `nam`: Available NAM forecasts.

### Examples

List everything in the database:

```
$ apollo ls
models/linear-nam-uga
models/xgboost-nam-uga
...
templates/linear-nam
templates/xgboost-nam
...
nam/2017-01-01T00Z
nam/2017-01-01T06Z
nam/2017-01-01T12Z
nam/2017-01-01T18Z
nam/2017-01-02T00Z
nam/2017-01-02T06Z
nam/2017-01-02T12Z
nam/2017-01-02T18Z
...
```

List only NAM forecasts:

```
$ apollo ls nam
2017-01-01T00Z
2017-01-01T06Z
2017-01-01T12Z
2017-01-01T18Z
2017-01-02T00Z
2017-01-02T06Z
2017-01-02T12Z
```

(continues on next page)

(continued from previous page)

```
2017-01-02T18Z
```

```
...
```

### 1.6.3 apollo predict

#### Summary

Execute an Apollo model

---

**Todo:** Document

---

### 1.6.4 apollo train

#### Summary

Train a new model

---

**Todo:** Document

---

### 1.6.5 apollo score

#### Summary

Compute metrics for model output

---

**Todo:** Document

---

### 1.6.6 apollo nam download

#### Summary

Download and process a NAM forecast

---

**Todo:** Document

---



## CHAPTER 2

---

Thesis

---

---

**Note:** These are my thesis chapters and won't appear in the final docs.

---

### 2.1 Introduction

---

**Todo:** TODO

---

### 2.2 System Design

#### 2.2.1 Use Cases

---

**Todo:** Copy from *Overview: The What and Why of Apollo*.

---

#### 2.2.2 System Architecture

---

**Todo:** Copy from *Overview: The What and Why of Apollo*.

---

#### 2.2.3 Deployment

## Apollo

---

### Microservice architectures

---

**Todo:** TODO

---

### Containers

---

**Todo:**

- What are containers?
  - Docker and the Dockerfile language
  - Kubernetes and Docker Swarm
- 

### REST / HTTP

---

**Todo:** TODO

---

### uWSGI & Nginx

---

**Todo:** TODO

---

## 2.3 The NAM-UGA Dataset

### 2.3.1 NAM

---

**Todo:**

- What is the NAM forecast model?
  - NOAA
  - GRIB
- 

### 2.3.2 WRF-NMM

---

**Todo:** NAM is an instance of the Weather Research and Forecasting (WRF) nonhydrostatic mesoscale model (NMM)

---

---

### 2.3.3 NAM-UGA

---

**Todo:**

- Feature subset
  - Geographic subset
  - Data collection period
  - netCDF
- 

## 2.4 Experiments

### 2.4.1 Description of Train and Test Data

---

**Todo:**

- Simple test-train split.
  - Train data from 2017.
  - Test data from 2018.
  - Calendar heatmap graphic.
- 

### 2.4.2 Evaluation Metrics

---

**Todo:**

- MAE vs. MSE
  - MAPE vs. sMAPE
  - $R^2$
  - day-night vs day-only
- 

### 2.4.3 Preprocess & Feature Selection

---

**Todo:**

- What features are useful?
  - Computed time-of-day and time-of-year features.
  - Training with day-night vs day-only.
-

## 2.4.4 Empirical Learning Curve

---

**Todo:**

- Error vs amount of training data used.
  - Linear regression vs. Random Forest vs. GBTs.
- 

## 2.4.5 Reference Time vs Forecast Time

---

**Todo:**

- Error for each (reftime, forecast time) pair.
  - Heatmap.
- 

## 2.5 Conclusions

---

**Todo:** TODO

---



---

## Acknowledgements

---

Apollo was developed at the [UGA Institute for Artificial Intelligence](#) in Athens, GA and funded by the Georgia Power Company.

The primary authors of Apollo are:

- Chris Barrick
- Zach Jones
- Aashish Yadavally
- Dr. Fred Maier

A more complete listing of code contributors can be found [here](#).

The research and development of Apollo was directed by:

- Dr. Fred Maier (UGA)
- Dr. Khaled Rasheed (UGA)
- Will Hobbs (GA Power)

Apollo's logo was designed by [Freepik](#) and is used under the terms of the [Flaticon basic license](#). The logo was obtained from [Flaticon](#).

The Apollo wordmark was added to the logo and uses the [Fira Sans](#) font. Fira Sans was designed by [bBox Type](#) and is used under the terms of the [SIL Open Font License \(OFL\)](#).