



**Title:** *Near-term Risk Forecast: Evaluation of Performance and Recommendations Summary.*

**Primary Author(s):** *Chris Lautenberger, PhD (Reax Engineering) and David Saah, PhD (Spatial informatics Group, LLC).*

**Date:** *December 2023*

**Recommended Citation:**

*Lautenberger, C. and D. Saah. 2023. Near-term risk forecast: evaluation of performance and recommendations summary. CEC Agreement# EPC-18-026.*

## **Acknowledgements**

Funding for this study was provided by the California Energy Commission under contract agreement number EPC-18-026. The authors would like to thank the California Energy Commission for its support of this research and in particular Commission Agreement Managers David Stoms for their guidance.

## **About Pyregence**

The Pyregence Consortium is advancing scientific knowledge of wildfires and building next generation forecasting tools. Guided by an open-source philosophy, the Consortium is making the tools free and available to all, while also providing access to all underlying model inputs and datasets. Largely funded by a grant from the California Energy Commission, the Consortium is composed of leading researchers from 18 institutions across industry, academia, and government, as well as software developers and designers. Learn more about our work at [pyregence.org](http://pyregence.org).

### **DISCLAIMER**

This report was prepared as the result of work sponsored by the California Energy Commission. It does not necessarily represent the views of the Energy Commission, its employees or the State of California. The Energy Commission, the State of California, its employees, contractors, and subcontractors make no warranty, express or implied, and assume no legal liability for the information in this report; nor does any party represent that the uses of this information will not infringe upon privately owned rights. This report has not been approved or disapproved by the California Energy Commission nor has the California Energy Commission passed upon the accuracy or adequacy of the information in this report.

## Table of Contents

Introduction .....	4
Performance Evaluation – PyreCast Forecast System .....	5
Performance Evaluation – Underlying Spread Models .....	7
Recommendations .....	12

## Introduction

The PyreCast forecast tool developed as part of this project (see <https://pyrecast.org>) provides the first publicly available fire spread and fire risk forecasts in the United States. PyreCast has been operational since 2020 – spanning four “fire seasons”, and a great deal has been learned over this time. The tool has evolved and undergone many enhancements from 2020 to its current state.

This report, which focuses on the active fire forecasting part of the PyreCast system, provides a high-level overview of the strengths and weaknesses of PyreCast, makes recommendations for future development, and demonstrates some of the methodologies that have been developed to evaluate the underlying fire spread models (ELMFIRE and GRIDFIRE, in particular).

## Performance Evaluation – PyreCast Forecast System

### Active Fire Initialization

Active fire perimeters in PyreCast, which serve as the initial condition for a fire spread forecast, are created from satellite-based hot-spot data, infrared perimeters typically obtained from fixed wing aircraft, and FireGuard polygons. In 2023, several challenges were encountered with erroneous fire initializations due to data quality issues with satellite-based fire hotspot data, FireGuard polygons, and occasionally fire polygon data. Spurious heat detections are particularly problematic in that they cause the fire to spread in the model from a point on the perimeter that is already contained.

To address this issue, it is recommended to modify the PyreCast fire initialization submodels to be able ignore specific fire initialization data on a fire-by-fire basis so that a single piece of erroneous fire initialization GIS data did not render all future forecasts for that fire inaccurate. However, since this must be done manually and is extremely time consuming, it is recommended that real-time Geostationary satellites (GOES-16, etc.) be used to develop an improved fire initialization submodel.

### Rekindled fire after precipitation events

In 2023, large-scale precipitation fell on several large fires in the Western US that were being modeled in PyreCast, temporarily halting their spread but not extinguishing them completely. This led to the issue of “how long does the heat last” which for actively spreading fires has been assumed to be ~2 days. This meant that fires that had been rained on and stopped spreading for a 2 or more days were completely extinguished in PyreCast and in some cases removed from the system. This was problematic when these fires started spreading again because no forecasts were available in PyreCast.

It is recommended that heat signatures from past fires (MODIS/VIIRS/GOES) be analyzed concurrently with fuel characteristics and weather reanalysis data to develop improved estimates of “how long does the heat last” and “when will a rained-on fire start spreading again”. This could be accomplished by developing simple rules of thumb, physically inspired models, or by training a machine learning model on multiple fires.

### Scalability and computational resources

In late August / early September 2023, the total number of fires in PyreCast approached 70 fires due to lightning-caused fires in the Western US. During that time, PyreCast was issuing between 200 and 250 forecasts per day as somewhere between 2 and 4 are typically generated for each fire. With 2-week forecasts, 100 ensemble members, and each forecast modeled as suppressed and unsuppressed, the limits of PyreCast’s current on-premise computational resources were

reached. Active fire forecasts often remained queued for several hours until computational resources became available, delaying the timely issuance of forecasts.

To improve the timeliness with which forecasts are issued, it is recommended that a hybrid computational approach be developed to leverage on-demand cloud computing (GCP) when PyreCast’s on-premises compute resources become stressed. Although this will increase costs, it has the important benefit of improving the timeliness with which forecasts are issued.

### Forecast spatial domain

PyreCast (including active fires, risk, and weather) is currently limited to the Continental US (notably beyond the project’s scope of California). However, Alaska experienced a devastating fire season in 2022, and in 2023 fires in Hawaii destroyed over 2,000 structures and killed approximately 100 people. It is recommended that PyreCast’s forecast domain be extended to Alaska and Hawaii, not just for active fires but also for fire risk and fire weather.

### Forecast accuracy

Assessing overall forecast accuracy is extremely challenging because the accuracy of a fire spread forecast depends not only on the underlying fire spread model, but also on the accuracy of inputs (fuel characteristics, fuel moisture values, initial fire perimeter, and wind speed/direction). Additionally, with few exceptions such as fires burning in wilderness areas, most fires in the Continental US are suppressed. However, we currently have limited ability to model fire suppression and we often are unable to predict *a priori* if or when suppression activities will be undertaken on a fire.

A retrospective assessment of 214 separate 7-day forecasts made during the 2022 fire season was conducted. The results of this assessment are summarized in Table 1. Most forecasts were acceptable or better, with a small percentage of forecasts rated as poor or fair. This performance was found to be comparable to other models used by Federal agencies. PyreCast’s forecast skill is much better for wind-driven fires than for plume-dominated fires. This is a feature common to all wildfire spread models that are not coupled to the atmosphere.

*Table 1. Qualitative Forecast Skill for 214 Forecasts Issued During 2022.*

<b>Performance</b>	<b>Count</b>
Poor	2
Fair	7
Acceptable	100
Good	69
Excellent	36

Conducting a retrospective forecast assessment is extremely time consuming. Therefore, it is recommended to develop and implement a real-time forecast assessment system for PyreCast.

In this paradigm, PyreCast forecasts would be compared to fire progression data as soon as that data becomes available. This would provide quantitative real-time feedback as to forecast accuracy. For example, one week after a PyreCast forecast is issued, modeled progression for that forecast would be automatically compared to observed daily fire perimeters and the level of agreement quantified. This would be a significant improvement over the current approach where qualitative comparisons between modeled and observed spread are made manually for select fires.

## Performance Evaluation – Underlying Spread Models

A series of command line tools and API's have been developed to assess the performance of PyreCast's underlying spread models (ELMFIRE and GRIDFIRE) when simulating historical fires. A brief overview of these tools, including their application to forecast validation, is presented below. It is recommended that these tools be used to periodically quantify forecast skill for PyreCast's underlying models, although such assessments are extremely difficult for the reasons mentioned in the previous section.

### Validation test case – 2018 County Fire

This validation case demonstrates how to run a hindcast of a single fire with ELMFIRE and compare modeled fire perimeters to observed fire perimeters. California's 2018 County Fire, which burned in Yolo and Napa Counties in June 2018, is used as a template. Additional fires can be simulated by modifying this template, and an example of how to do this is provided at the end of this validation case.

After ELMFIRE has been installed (see [https://elmfire.io/getting\\_started.html](https://elmfire.io/getting_started.html)) the single-fire validation case (2018 County Fire) can be viewed as follows:

```
$ cd $ELMFIRE_BASE_DIR/validation/01-single-fire
$ pico 01-ca-county.sh
```

This test case contains the following lines (truncated at the end of the inputs specification section):

```

#!/bin/bash

# Specify inputs below:

YEAR=2018
FIRENAME=ca-county
ACTIVE_FIRE_TIMESTAMP_NUM=1
ALREADY_BURNED_TIMESTAMP_NUM=null
WEST_BUFFER=30
SOUTH_BUFFER=30
EAST_BUFFER=30
NORTH_BUFFER=30
NUM_ENSEMBLE_MEMBERS=100
RUN_HOURS=48
FUEL_SOURCE=landfire
FUEL_VERSION=1.4.0
RUN_TEMPLATE=hindcast

export USE_SLURM=no
export CALC_FITNESS=yes

# No need to modify from here down:
...

```

An explanation of the inputs specification part of this script is lines is:

- YEAR=2018 and FIRENAME=ca-county: The script `get_polygons.py` needs the year and fire name to retrieve fire perimeter GIS data.
- ACTIVE\_FIRE\_TIMESTAMP\_NUM and ALREADY\_BURNED\_TIMESTAMP\_NUM: Rather than explicitly specifying fire perimeter timestamps to use in fire initialization, the timestamp number (of sequentially sorted available timestamps) is used to initialize the actively burning fire perimeter (ACTIVE\_FIRE\_TIMESTAMP\_NUM) and the already burned / black area (ALREADY\_BURNED\_TIMESTAMP\_NUM). The latter can be set to null as in this case if the entire fireline is active.
- WEST\_BUFFER, SOUTH\_BUFFER, EAST\_BUFFER, NORTH\_BUFFER: This is the size of the buffer, in units of km, that is added around the active fire polygon. The buffer should be large enough to prevent spreading fires from hitting the edge of the computational tile. Therefore, smaller buffers can be used for short-duration hindcasts or fires burning under moderate weather conditions than for long-duration hindcasts or rapidly spreading fires.
- NUM\_ENSEMBLE\_MEMBERS: For this particular validation case, ELMFIRE is run in ensemble mode where specific inputs are perturbed from their baseline values to address model uncertainty (see the User Guide for details on how ensembles are constructed). By default the ensemble consists of 100 members, but this can be increased or decreased as desired.
- RUN\_HOURS: The simulation run time defaults to 48-hours but longer or shorter runs can be used.

- FUEL\_SOURCE and FUEL\_VERSION: The script fuel\_wx\_ign.py needs to know which fuel source (here, LANDFIRE) and version (here, 1.4.0) should be used for creating fuel and topography inputs. Currently, all major LANDFIRE releases (1.4.0, 2.2.0, etc.) are available and the user should select a version that is appropriate for the era of the fire being simulated. Details can be found at the LANDFIRE Schedule / Versions page.
- RUN\_TEMPLATE: Different run templates can be used to setup different base runs. Specifically, RUN\_TEMPLATE specifies a subdirectory off of \$ELMFIRE\_BASE\_DIR/runs/hindcasts/templates that contains customized run scripts and the elmfire.data configuration file.
- export USE\_SLURM=no: ELMFIRE is run on several clusters that use the Slurm Workload Manager for job queueing and prioritization. With USE\_SLURM=no, ELMFIRE will run locally. However, if ELMFIRE is being run on a cluster where Slurm is used for job management, setting USE\_SLURM=yes will allow ELMFIRE to run on remote nodes.
- export CALC\_FITNESS=yes: By default, a fitness metric that quantifies how well modeled fire perimeters match observed perimeters will be calculated for each ensemble member. This causes a large number of intermediate files to be generated and adds considerably to the required postprocessing time. Setting export CALC\_FITNESS=no can be useful for initial testing/debugging to cut down on cycle time.

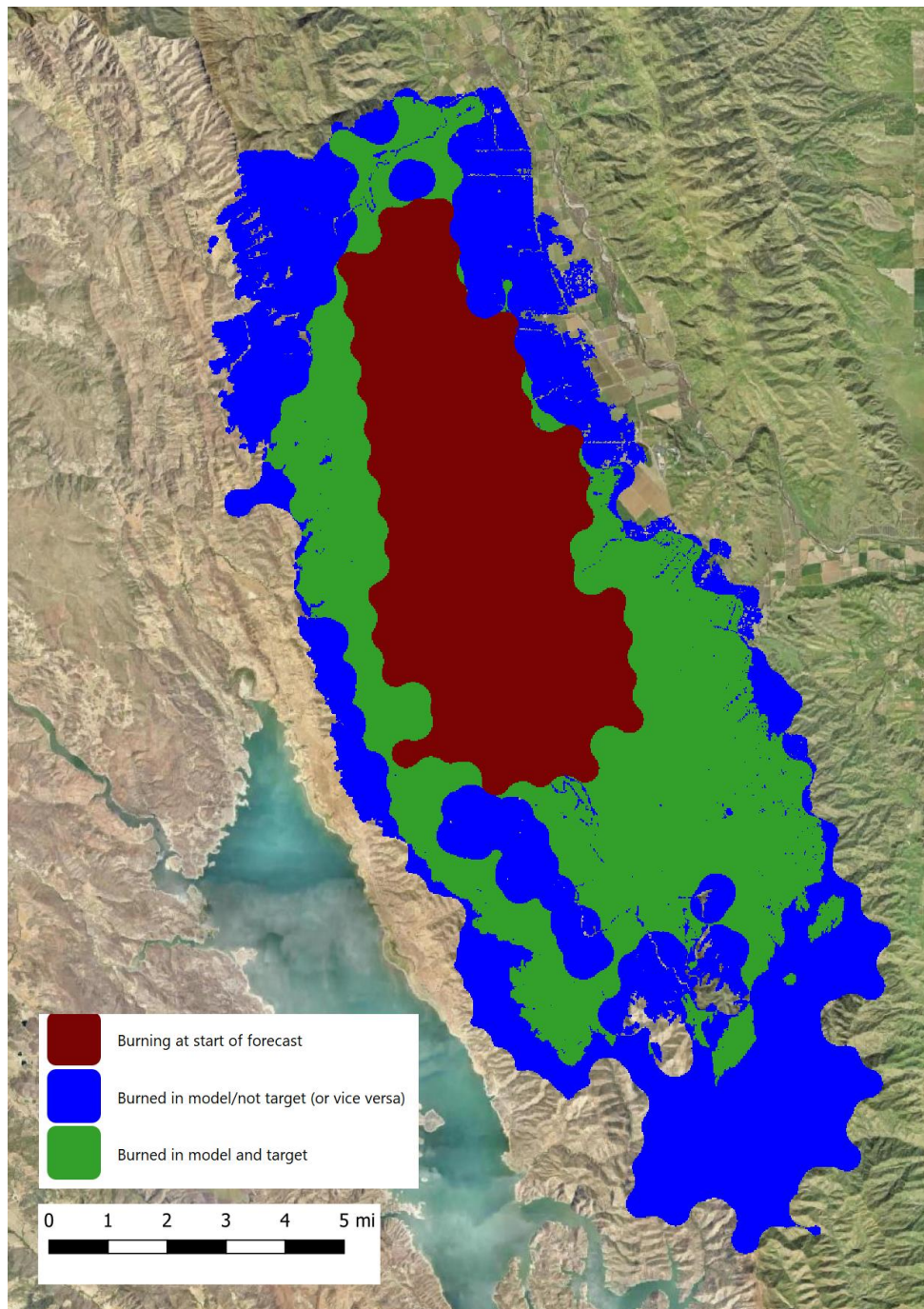
After executing the run script (./01-ca-county.sh), inputs are created, ELMFIRE is executed, and post-processing is conducted. If the run is successful, the directory ca-county\_20180701\_062200 containing model inputs and outputs is created. Some of the more important files in this directory include:

- fuel and wx directories: These folders contain the fuel/topo and weather inputs that were generated by fuel\_wx\_ign.py.
- elmfire.out: ELMFIRE's log file written to disk at run time. If outputs are missing, this file can be useful for diagnosing what went wrong.
- toa directory: Time of arrival (in seconds) rasters for each ensemble member. Naming convention is time\_of\_arrival\_XXXXXX\_YYYYYYY.tif where XXXXXX is the ensemble member number and YYYYYYY is the simulation end time in seconds since only a single time of arrival raster is dumped for each ensemble member.
- targ directory: This directory contains "target" fire perimeter rasters against which modeled fire perimeters are compared when calculating goodness of fit. Naming convention is target\_SECONDS.tif where SECONDS is the number of elapsed seconds from the simulation start time.
- model directory: This directory contains modeled fire perimeter rasters for each ensemble member at each of the times for which target fire perimeter data are available. These are used for calculating the correlation between modeled and observed perimeters (next bullet).
- corr directory: Correlation rasters. Naming convention is similar to the time of arrival rasters, except the time field in the filename corresponds to a target time. For example, corr\_0000031\_44220.tif is the correlation raster for ensemble member 31 at 44,220 seconds after ignition. Pixels are coded as 0 (not burned in target, not burned in model), +1 (burned in model and burned in target), or -1 (burned in model and not burned in target or vice versa).

- `fire_size_stats.csv`: Summary statistics / diagnostics such as modeled fire area, run time, etc. for each ensemble member.
- `coeffs.csv`: Companion file to `fire_size_stats.csv`. Provides values of adjustable model coefficients for each ensemble member. The rightmost column is an overall fitness metric that was calculated by summing over the correlation rasters as described above. This file is particularly useful for model calibration where an optimization algorithm or machine learning is used to improve agreement between modeled and observed perimeters. Additional details can be found in the Monte Carlo Simulation section of the User's Guide.

Figure 1 below shows one of the correlation rasters from a single ensemble member. The ELMFIRE simulation was initialized with the red area. The blue and green area show the correlation between modeled and observed fire perimeter after 36 hours of spread (from the red area). Green areas were burned in both the modeled and observed perimeter, and blue areas were burned in one but not the other.

Additional fires can be modeled by editing a few lines in the `01-single-fire.sh` run script. This typically involves running queries with `get_polygons.py` to find available fires and perimeters, setting the `YEAR` and `FIRENAME` inputs accordingly, and adjusting other inputs as desired.



**Figure 1. Comparison of modeled and observed fire progression for County Fire.**

## Recommendations

Under this CEC EPIC project, excellent progress has been made in developing PyreCast and disseminating the models and data on which it is built. Although this EPIC project is nearing its end, the project team recommends several areas where the utility and accuracy of PyreCast could be improved with future development efforts, including:

- Improve forecast initialization to eliminate spurious heat detections.
- Develop prediction techniques to forecast when/where a fire rekindles after rain.
- Implement a hybrid computational approach where cloud resources are leveraged when on-premises compute resources are stressed.
- Extending PyreCast spatial domain to include Alaska and Hawaii.
- Implement real-time automated forecast validation metrics.
- Conduct periodic assessments of the underlying forecast models (ELMFIRE and GRIDFIRE) using the API's described in the previous section.