Demonstration of the Data Fusion Labeler (dFL) for Multiyear Weather Data

Sophelio LLC / dFL Project Team

Abstract

This document presents a comprehensive demonstration of the *Data Fusion Labeler (dFL)* applied to multiyear meteorological datasets. The example illustrates how simple weather data—temperature, pressure, and precipitation—can be harmonized, visualized, and labeled through dFL's modular data fusion and autolabeling architecture. The workflow exemplifies the broader paradigm of using harmonization frameworks for time-series modeling, anomaly detection, and AI-assisted scientific research.

1 Introduction

In this demonstration, the *Data Fusion Labeler (dFL)* is used to load and harmonize approximately ten years of daily weather data for Boston, Massachusetts. The data are retrieved automatically using the `weather_data_fetcher.py` utility, which leverages the open-source Meteostat API [1]. The dataset includes three key meteorological signals:

- **Precipitation (mm/day)**
- **Temperature (°C)**
- **Pressure (hPa)**

Each year of data is treated as a distinct record, with one record per file. The workflow demonstrates how even simple environmental data can be used to explore labeling, automation, and process reproducibility within dFL.

2 Data Loading and Harmonization

To begin, the user opens dFL and loads the data source through:

Settings → Open File → `weather_data_provider.py`

Once loaded, the provider automatically detects all available records (2014–2023) and infers the proper signals. dFL applies its core harmonization operations—trimming, normalization, and smoothing—through the `weather_utilities.py` backend module.

Normalization reveals long-term correlations between temperature, precipitation, and pressure. The timeline browser shows each record as a distinct entry, while the data explorer confirms all signals are temporally aligned. These harmonization operations ensure that subsequent analyses (visualization, labeling, or forecasting) are based on physically consistent data.

3 Interactive Visualization: Precipitation Threshold Plot

Using dFL's prompt-driven customization interface, a new visualization is introduced through natural language:

> *"Create an interactive precipitation plot per record that highlights time periods exceeding a user-defined threshold."*

The underlying implementation, detailed in `weather_utilities.py` and `weather_data_provider.py`, defines a custom grapher called `Precipitation Threshold Plot`. As described in [2], this interactive Plotly-based graph displays precipitation (or any chosen signal) and dynamically highlights regions above a user-specified threshold.

The graph parameters include:

- **Threshold Value:** numeric (default = 5.0)
- **Signal to Plot:** selection among Temperature, Pressure, or Precipitation

As the threshold slider moves, dFL dynamically updates highlighted regions where precipitation surpasses the limit, producing an immediate visualization of storm events or anomalous weather periods. The shaded regions, hover tooltips, and theme-aware design demonstrate dFL's integrated customization capabilities.

4 Integrating a Custom AutoLabeler

To convert threshold visualization into actionable metadata, a custom autolabeler—`perform_precipitation_thres`
added via natural language prompting. The prompt:

> *"Add a custom autolabeler called 'Precipitation Threshold' that labels all intervals in which precipitation exceeds a user-specified threshold."*

This creates a new entry in the `autolabeling_dictionary` within `auto_labeling.py`. The implementation, described in [3], detects all continuous time intervals above a precipitation threshold and merges nearby events into single labels. The algorithm executes in linear time ($O(n)$) and handles missing data, nonuniform sampling, and invalid parameters robustly.

The autolabeler is parameterized by:

- **Precipitation Threshold (mm):** Default = 5.0
- **Merge Time Window (hours):** Default = 1.0
- **Label Type:** Peak, Dip, or Anomaly

These labels are created persistently within dFL's metadata layer and can be merged, filtered, or exported for downstream analysis—such as training predictive models of precipitation events from temperature and pressure trends.

5 Exporting Labels and Next Steps

Labeled datasets are exported directly through dFL:

$$\text{Current Labels} \rightarrow \text{Download Labels from ALL Records}$$

The exported CSV files contain start and end times (T_1, T_2) for each labeled precipitation event. Because dFL records every transformation and labeling operation in a version-controlled data lineage graph, all results are fully reproducible.

This labeled data can be used to train models that forecast rainfall probability from historical pressure and temperature patterns or to validate climate trend models across multiple decades.

6 Conclusion and Future Directions

This demonstration highlights the extensibility of the Data Fusion Labeler across scientific domains. While this example focuses on meteorological data, the same architecture can be applied to climate studies, environmental monitoring, and energy systems.

The workflow illustrates:

- Ingestion and harmonization of time-series weather data.
- Creation of custom interactive threshold visualizations.
- Automated labeling of event intervals for downstream ML training.
- Reproducibility through dFL's integrated provenance system.

Future work may include:

- Multi-site correlation studies (e.g., precipitation synchrony across cities).
- Seasonal anomaly detection through ensemble threshold labeling.
- Integration of predictive transformers for atmospheric pattern analysis.

The weather demonstration thus serves as a minimal, generalizable template for applying dFL to any temporally resolved dataset—illustrating the same harmonization, visualization, and labeling philosophy used in complex fusion-energy applications [4].

Acknowledgements

The authors acknowledge Meteostat for providing the open-access weather API and the dFL engineering team for implementing interactive graphers and autolabelers under the Sophelio harmonization framework.

References

[1] Benjamin Kaiser and contributors. Meteostat: Historical weather and climate data api, 2025.

[2] Sophelio dFL Team. *Precipitation Threshold Plot – User and Implementation Guide*. Sophelio LLC, 2025. Internal technical documentation.

[3] Sophelio dFL Team. *Precipitation Threshold Autolabeler – Implementation and User Guide*. Sophelio LLC, 2025. Internal technical documentation.

[4] Craig Michoski, Matthew Waller, Brian Sammuli, Zeyu Li, Tapan Ganatma Nakkina, Raffi Nazikian, Sterling Smith, David Orozco, Dongyang Kuang, Martin Foltin, Erik Olofsson, Mike Fredrickson, Jerry Louis-Jeune, David R. Hatch, Todd A. Oliver, Mitchell Clark, and Steph-Yves Louis. The data fusion labeler (dfl): Challenges and solutions to data harmonization, labeling, and provenance in fusion energy, 2025.