Office of Satellite and Product Operations
Environmental Satellite Processing Center

Leaf Area Index System Maintenance Manual

Version 1.2
April 18, 2025

U.S. Department of Commerce
National Oceanic and Atmospheric Administration
National Environmental Satellite, Data, and Information Service
Office of Satellite and Product Operations

Authors

Susan Haworth (OCS)

Updated by Jeffrey Augenbaum (OMS, ERT)

Approval Page

Environmental Satellite Processing Center
Leaf Area Index System Maintenance Manual

GROUP: OMS Date: 04/22/2025 Jeffrey Augenbaum, Lead Programmer	GROUP: OSPO Date: 05/16/2025 Hanjun Ding, Product Area Lead
GROUP: OMS Date: 05/15/2025 Clay Davenport, Products Manager	

Changes/Revisions Record

This system maintenance manual is changed as required to reflect system, operational, or organizational changes. Modifications made to this document are recorded in the Changes/Revisions Record below. This record will be maintained throughout the life of the document.

Version Number	Date	Description of Change/Revision	Section/Pages Affected	Changes Made by Name/Title/ Organization
1.0	03/2025	Initial version	All	Susan Haworth
1.1	04/2025	Updated for NCCF; Added links to NCCF SMM; Added links to Product monitoring and visualization; Added data flow diagrams	All; 1.3, 2.1, 2.2, 4.2.2, 4.2.3, 4.4.1, 5.2-5.3, 6.1-6.3; 5.1.1; 7.1	Jeff Augenbaum
1.2	04/18/2025	Transfer contents to latest template, perform minor copy editing and formatting	All	Hannah Bowie, Technical Writer, ERT
1.2	04/22/2025	Quality Assurance	All	Clint Sherwood, Quality Assurance Manager, ERT

Preface

This document comprises the National Oceanic and Atmospheric Administration (NOAA) National Environmental Satellite, Data, and Information Service (NESDIS), Office of Satellite and Product Operations (OSPO), publication of this Leaf Area Index (LAI) System Maintenance Manual (SMM). This document reflects current operations for the DOC/NOAA/NESDIS Environmental Satellite Processing Center (ESPC) (NOAA5045) information technology systems. This document describes the established ESPC procedures for LAI system maintenance in accordance with Federal, DOC, NOAA, NESDIS and OSPO requirements.

Future updates and revisions to this document will be produced and controlled by DOC/NOAA/NESDIS for ESPC information technology systems.

The published version of this document can be found at the OSPO SharePoint Products Library.

Table of Contents

Executive Summary 1

1. Introduction 3

1.1. Product Overview 3

1.2. Algorithm Overview 3

1.3. Interfaces Overview 4

2. Hardware 4

2.1. Hardware Description 4

2.2. Operating System 4

2.3. System Requirements 4

2.3.1. Storage Requirements 4

2.3.2. Computer Resource Requirements 5

2.3.3. Communication Needs 5

3. Software 5

3.1. Software Description 5

3.2. Directory Description 6

3.3. Source Code Description 7

4. Normal Operations 8

4.1. System Control 8

4.1.1. System Control Files 8

4.1.2. Processing Controls 10

4.2. Installation 10

4.2.1. Installation Items 10

4.2.2. Compilation Procedures 10

4.2.3. Installation Procedures 11

4.3. Configuration Procedures 11

4.3.1. Production Rules 11

4.4. Operations Procedures 12

4.4.1. Normal Operations 12

4.4.2. Data Preparation 12

4.5. Distribution 12
4.5.1. Data Transfer/Communications 12
4.5.2. Distribution Restrictions 12
4.5.3. Product Retention Requirements 12
4.5.4. External Product Tools 12
5. Monitoring and Maintenance 12
5.1. Job Monitoring 12
5.1.1. Product Monitoring and Visualization 12
5.2. Data Signal Monitoring 13
5.3. Product Monitoring 13
5.3.1. Unit Test Plans 13
5.3.2. Internal Product Tools 13
5.3.3. Performance Statistics 13
5.3.4. Product Monitoring 13
5.3.5. Product Criticality 13
5.4. Maintenance 13
5.4.1. Monitoring and Maintenance 13
5.4.1.1. Ingest Monitoring 13
5.4.2. Science Maintenance 13
5.4.3. Library Maintenance 13
5.4.4. Special Maintenance Procedures 13
5.4.5. Maintenance Utilities 13
5.5. Program Backup Procedures 14
6. Troubleshooting 14
6.1. Program Diagnosis and Recovery 14
6.1.1. Quality Control Output 14
6.1.2. Error Correction 14
6.1.3. Problem Diagnosis and Recovery Procedures 14
6.1.4. Data Recovery Procedures 14
6.1.5. Program Recovery Procedures 14
6.2. Application Shutdown and Restart 14

6.2.1. Application Shutdown Procedures 14
6.2.2. Application Restart Procedures 14
6.3. System Shutdown and Restart 14
6.3.1. System Shutdown Procedures 14
6.3.2. System Restart Procedures 15
6.3.3. System Reboot Procedures 15
7. Appendix 16
7.1. Data Flow 16
7.2. Input Data Files 19
7.3. Ancillary Data Files 20
7.3.1. Dynamic Ancillary Data 20
7.3.2. Static Ancillary Data 20
7.4. Look Up Tables 20
7.5. Intermediate Data Set Description 20
7.6. Output Data Set Description 21
7.7. Archive Data Description 25
8. Acronyms 26

List of Tables

Table 0-1 - Product Team Members 1
Table 2-1 - Storage Requirements 4
Table 3-1 - LAI Software Elements 5
Table 3-2 - Top Level Directory Tree Description 7
Table 4-1 - Application YAML File Arguments 8
Table 4-2 - Docker YAML File Arguments 10
Table 7-1 - Primary Input File Naming Conventions 19
Table 7-2 - Intermediate File Naming Conventions for Daily Unit 20
Table 7-3 - Output File Naming Conventions 21
Table 7-4 - LAI NetCDF4 Output File Description 22
Table 7-5 - LAI NetCDF4 Output File Metadata 22

List of Figures

Figure 7-1 - LAI Processing Architecture 17
Figure 7-2 - LAI Product Precedence 18
Figure 7-3 - IT Architecture & Network 18

Executive Summary

This System Maintenance Manual (SMM) describes the Leaf Area Index (LAI) algorithm package.

Leaf Area Index (LAI) is defined as one half the total green leaf area per unit horizontal ground surface area. It is an essential climate variable driving water fluxes, carbon fluxes, and energy exchanges, playing an important role in the models of the climate, hydrology, and ecology. As a fundamental attribute of vegetation, LAI is an essential climate variable.

Leaf Area Index (LAI) uses VIIRS Surface Reflectance as the primary input data and geometry data (GITCO) to produce two intermediate products:

- Daily green leaf coverage value (Daily LAI), and
- Weekly green leaf coverage value (Weekly LAI).

Both Daily and Weekly LAI are saved and used to create the final product: a global gap-free LAI end product at 1 km resolution with a step of 8-day.

Table 0-1 includes information about the LAI team member's organizations, roles, and contact information.

Table 0-1 - Product Team Members

Team Member	Organization	Role	Contact Info
Walter Wolf	OCS	OCS Product Management Division Chief	walter.wolf@noaa.gov
Yunyue Yu	NESDIS	Science Team Lead	yunyue.yu@noaa.gov
Yuan Zhou	NESDIS	Science Team	yuan.zhou@noaa.gov
Hanjun Ding	OSPO	OSPO PAL	hanjun.ding@noaa.gov
Isabel Jones	NESDIS	PPM	isabel.jones@noaa.gov
Curtiss Burnett	NESDIS	PIB	curtiss.burnett@noaa.gov
Priyanka Roy	OCS	ASSISTT Management Team	priyanka.roy@noaa.gov
Letitia (Tish) Soulliard	OCS	ASSISTT Integration Team Lead	letitia.soulliard@noaa.gov
Shukming Wu	OCS	ASSISTT Integration Team Lead	shukming.wu@noaa.gov
Mike Butler	OCS	ASSISTT Integration Team Lead	michael.butler@noaa.gov
Zhenpeng Li	OCS	ASSISTT Integration Team Lead	zhengpeng.li@noaa.gov
Dongsheng Wu	OCS	ASSISTT Integration Team	dongsheng.wu@noaa.gov
Brandon Laufer	OCS	ASSISTT Integration Team	brandon.laufer@noaa.gov
Yunhui Zhao	OCS	ASSISTT Configuration Management Team Lead	yunhui.zhao@noaa.gov
Susan Haworth	OCS	ASSISTT Technical Writer	susan.haworth@noaa.gov
Tracey Dorian	OCS	ASSISTT Lifecycle QA Team Lead	tracey.dorian@noaa.gov
Mingming Yao	OCS	ASSISTT Lifecycle QA Team	mingming.yao@noaa.gov
Hua Xie	OCS	ASSISTT Science QA Team Lead	hua.xie@noaa.gov

All low-level processing code in LAI is written in Fortran90, C, and C++. The low-level code performs all data processing, scientific computation, reading/writing, reformatting, and opening/closing of the files. All high-level code is written in Python. High-level tasks include file

management, system management, making system calls, and error trapping from low-level processing. The scripts act as a driver code for the lower-level processing. The driver scripts manage the LAI software and call any necessary unit scripts for the algorithm package's processing.

The NESDIS' Policy on Access and Distribution of Environmental Data and Products is provided at http://www.ospo.noaa.gov/Organization/About/access.html.

Users need to fill out the Data Access Request Form located on this site and submit to the PAL with a copy to nesdis.data.access@noaa.gov. This address provides the OSPO Data Access Team a copy of the correspondence. Once the request is approved by the OSPO management the data will be delivered by the Data Distribution System (DDSProd) currently distributing the ESPC data products and later by the Product Distribution and Access (PDA) system. ESPC User Services (SPSDuserservices@noaa.gov) should be contacted for any data accessibility and data distribution.

1. Introduction

1.1. Product Overview

Leaf Area Index (LAI) is defined as one half the total green leaf area per unit horizontal ground surface area. It is an essential climate variable driving water fluxes, carbon fluxes, and energy exchanges, playing an important role in the models of the climate, hydrology, and ecology. As a fundamental attribute of vegetation, LAI is an essential climate variable.

Leaf Area Index (LAI) uses VIIRS Surface Reflectance as the primary input data and geometry data (GITCO) to produce two intermediate products:

- Daily green leaf coverage value (Daily LAI), and
- Weekly green leaf coverage value (Weekly LAI).

Both Daily and Weekly LAI are saved and used to create the final product: a global gap-free LAI end product at 1 km resolution with a step of 8 days.

The LAI product is for daytime observations only.

LAI can benefit users in the following ways:

- Substantially improves the prediction accuracy of NCEP global and mesoscale models (GFS and NAM).
- Substantially improves the impact over land of satellite-measured leaf area index in surface-sensitive satellite channels in the data assimilation in NCEP global and regional data assimilation systems (GDAS and NDAS).
- Provides an important input for many ecological and hydrological models.

1.2. Algorithm Overview

Satellite LAI datasets, recorded over the past two decades, have been utilized extensively across various applications. Leveraging the legacy of established satellite products like the Moderate Resolution Imaging Spectroradiometer (MODIS), Global Land Surface Satellite (GLASS), and Geoland2/BioPar (GEOV2) LAI products, a data-driven methodology has been developed to obtain near-real-time LAI from VIIRS observations. Prior to implementation, a machine learning algorithm is tuned and trained based on a comprehensive suite of representative datasets.

The VIIRS LAI product is designed to be a temporally smoothed, global, gap-free dataset. The operational procedure is segmented into three phases. The first two steps are daily processing, with up to 8 days' data being sustained for the weekly processing, which will be run every 8 days.

1. **Daily Surface Reflectance Generation**: Utilizing the VIIRS gridding tool, granule data is mapped onto a global grid in a sinusoidal projection. The surface reflectance (SR) compositing process then identifies and selects the highest quality SR and corresponding angles for each grid cell.

2. **Daily LAI Retrieval**: A previously trained machine learning algorithm performs the clear-sky LAI retrieval, leveraging the daily SR together with auxiliary data.
3. **8-Day LAI Compositing and Post-Processing**: From the daily LAI outputs, the optimal quality LAI is chosen for each 8-day interval. Subsequently, a temporal smoothing and gap-filling (TSGF) procedure is applied to produce the final, gap-free product.

1.3. Interfaces Overview

Before reviewing this System Maintenance Manual (SMM), please request the live master NCCF SMM (refer to *System Maintenance Manual - NESDIS Common Cloud Framework (NCCF)*) from the OSPO PALs in Table 0-1. The NCCF system overview is described in the master NCCF SMM: NCCF Description and Overview (or Document Object: 4,5).

2. Hardware

2.1. Hardware Description

The hardware is described in the master NCCF SMM: NCCF Description and Overview (or Document Object: 4,5): Infrastructure.

2.2. Operating System

The NCCF operating system description can be found in the master NCCF SMM: Operating System (or Document Object: 64).

2.3. System Requirements

2.3.1. Storage Requirements

Table 2-1 contains information describing the approximate size of input files, output files, and system files associated with the LAI algorithm package. The approximate sizes listed represent the size of one file.

Table 2-1 - Storage Requirements

Catalog	Storage Item	Approximate Size	Number of Expected Files Per Run
Input Files	L1B VIIRS Geometry	100 MB	547
	L2 VIIRS Surface Reflectance	209 MB	547
Intermediate Output / Input Files	Daily LAI	190 KB	2,320
	Weekly LAI	400 KB – 8 MB	4,032
Final Output File	Weekly Global LAI	130 MB	1
System Files	LAI Log	1 MB	1

2.3.2. Computer Resource Requirements

The LAI algorithm package requires the following libraries and utilities to successfully complete a run of processing:

- Python 3.9
  - netCDF4=1.6.2
  - joblib=1.2.0
  - numpy=1.23.4
  - autologging=1.3.2
  - pyyaml=6.0

Refer to the LAI algorithm package for more information concerning the specific version numbers of these libraries and utilities.

2.3.3. Communication Needs

There are no special communication needs associated with the LAI algorithm package.

3. Software

3.1. Software Description

Table 3-1 provides a short summary of the two processing units that make up the LAI algorithm package.

Table 3-1 - LAI Software Elements

Processing Unit	Description
Daily	Runs every 24 hours when the surface reflectance and geometry data are available. Processing steps include: 1. Takes granule Lat/Lon data as input, sourced from VIIRS GITCO data, and generates mapping indices that store information for mapping granule data to a global sinusoidal grid at a resolution of 500m. 2. Takes the mapping indices, along with VIIRS L2 SR and GITCO data, and generates global gridded SR data, including VIIRS I1, I2, and I3 bands, solar/satellite angles, and quality flags. 3. Takes the global gridded SR data and produces the global daily LAI dataset.

Processing Unit	Description
Weekly	Runs daily after the Daily unit completes and, at a minimum, should be run every 8 days. Processing steps include: 1. Takes the last 8 days of Daily LAI as inputs and performs compositing to generate an 8-day LAI dataset with broader coverage and reduced noise. 2. Takes the dataset and generates a smoothed and gap-free 8-day LAI by utilizing information from the previous 14 weeks' 8-day data along with the LAI climatology. 3. Aggregates the smoothed and gap-free 8-day LAI tile data to form the final global product, which involves converting the resolution from 500m to 1km and reprojecting the original sinusoidal tile grids to the equal Lat/Lon grids. NOTE: The weekly unit creates output every 8 days starting on the eighth day of the year (January 8th). Since a year is not exactly divisible by 8-day periods, a few days into the next year is needed to complete the final 8-day cycle. If the Weekly unit is run every day, there is no output except on every 8-day cycle.

3.2. Directory Description

The CCAP consists of 3 gzip'd tar files:

- LAI_v1-1_CODE_20240628.tar.gz
- LAI_v1-1_DATA_20240628.tar.gz
- LAI_v1-1_DOCS_20240628.tar.gz

One way to unpack the algorithm package involves running the following command:

```
tar -xvzf <tarfile_name>
```

where `<tarfile_name>` is the name of the tarfile you wish to unpack.

The following is a top-level directory tree describing the LAI algorithm package. Note that this is only a top-level directory tree and reflects the directory structure of the algorithm package immediately after it is unpacked in the current working directory. The main software directory is fully expanded to provide additional software information to end users of both this document and the expected output product files.

```
CODE/
├── docker/
└── leaf-area-index/
      ├── ancillary_data/
      ├── bin/
      ├── config/
      ├── packages/
      ├── setEnvLib/
      ├── source/
      ├── test_data/
      └── wrapper/
```

Table 3-2 contains a brief description of each of the top-level directories.

Table 3-2 - Top Level Directory Tree Description

Directory	Description
CODE	Contains the science code supporting the algorithm package. Contains Python wrapper scripts and configuration YAML files to launch the Docker container, run the Docker container, and support the application.
DATA	Contains test data associated with the algorithm package. Includes input, output, and logs.
DOCS	Contains documentation associated with the algorithm package.

3.3. Source Code Description

Table 3-3 lists the major files present within the source code for the LAI algorithm package.

File Name	Description
leaf-area-index/source/daily_sr/	
daily_sr_composition_module.f90	Contains a subroutine to calculate the land surface reflectance for each tile.
daily_sr_output_module.f90	Writes the datasets and dataset attributes to the output NetCDF file.
get_config_module.f90	Contains a subroutine to set up initialization of static and dynamic configuration files.
init_all_granule_module.f90	Contains a subroutine to set up initialization of static and dynamic configuration files.
julday_module.f90	Contains a subroutine to calculate a Julian day from the input Z time.
main_vsdr_I3.f90	The main driver for the VIIRS Daily surface reflectance gridding algorithm.
nf90_handle_error_module.f90	Contains a subroutine to check for errors in performing any NetCDF calls and reports information to the user in the event of an error.
read_config_file_module.f90	Reads in the various configuration files, including: • gridded_sr_config.pcf (dynamic inputs from NDE) • gridded_sr_static.cfg (static inputs for writing metadata) • config.ini (static inputs for running code)
read_hdf5_module.f90	Contains a subroutine to read hdf5 format data.
set_metadata_module.f90	This module writes various metadata to the output NetCDF file.
type_kinds.f90	Contains the specific type kinds needed for all variable declarations to prevent bad assumptions.
vveg_para_mod.f90	Contains several variables that are initialized by external parameter files, along with a few parameters. Does not contain subroutines or functions.
leaf-area-index/source/gridding_img/	
defs.h	Defines several constants and prototypes used by the gridding tool.
gridding.c	The main driver for the gridding tool.
input.c	Writes the datasets and dataset attributes to the output NetCDF file.
mapping.c	Writes the datasets and dataset attributes to the output NetCDF file.
output.c	Contains the subroutines used for mapping tiles to their respective granules, and for mapping granules to their respective tiles.
leaf-area-index/source/lai_tsgf/	
defs.h	Defines several constants and prototypes used by the LAI smoothing and gap filling.

File Name	Description
error.cpp	Writes the datasets and dataset attributes to the output NetCDF file.
error.h	Prototype of the function Error which prints error message.
filter.cpp	Implements functions used in time series smoothing of LAI.
filter.h	Prototype of functions used in time series smoothing of LAI.
gapfilling.cpp	Implementation of C++ class gapfilling, which is used to read a water mask tile in the GVF grid system.
galfilling.h	Performs the LAI gapfilling using Climatology.
laidata.cpp	Defines the class of the Input/Output data for LAI algorithm.
laidata.h	Prototype of C++ class lai.
tsgf_main.cpp	The main driver for the temporal smoothing and gap filling for LAI.

4. Normal Operations

4.1. System Control

4.1.1. System Control Files

There must be at least one YAML file present in the algorithm package responsible for guiding the inner script as it chooses what processing or preprocessing steps will occur, which granule will be processed, and which span of time will be observed. For convenience, all items in the file that have a possibility of variation or are system specific will be located at the top of the YAML file as anchors. Each of these arguments are listed in Table 4-1.

Table 4-1 - Application YAML File Arguments

Argument Name	Description of Value
logging_level	Sets the logging level for Python's logging module. Options are: DEBUG, INFO, WARNING, ERROR, or CRITICAL.
Unit	The unit being run: daily or weekly.
spec/production/site	Value for the **production_site** metadata.
spec/production/environment	Value for the **production_environment** metadata.
spec/parameters/date	Date of the files being run.
spec/parameters/satname	Name of the satellite being run (j01, j02, or npp).
spec/directory/bin	Location of the executables directory inside the Docker container (should not need to be altered).
spec/directory/ancillary_data	Location of the ancillary directory inside the Docker container (should not need to be altered).
spec/directory/input	Location of the input directory inside the Docker container (should not need to be altered).
spec/directory/working	Location of the working directory inside the Docker container (should not need to be altered).
spec/directory/log_dir	Location of the log directory inside the Docker container (should not need to be altered).
spec/directory/output	Location of the output directory inside the Docker container (should not need to be altered).

File Placement for Daily Unit

To execute the Daily processing unit, ensure the input files are placed into the correct directories.

These directories must be set up as follows:

input/

```
└── <satellite>/
    └── <date>/
        ├── GITCO/
        └── SurfRefl/
```

where:

- <satellite> can be JPSS1, JPSS2, or SNPP.
- <date> is in the format YYYYMMDD.

File Placement for Weekly Unit

To execute the Weekly processing unit, ensure the input files are placed into the correct directories. These directories must be set up as follows:

input/

```
└── LAI/
    ├── 8Day/
    │    └── <satellite>/
    │        └── <year>/
    │           ├── <doy>/ (previous data if available and hereafter)
    │           ├── <doy>/
    │           ├── <doy>/
    │           ├── <doy>/
    │           ├── <doy>/
    │           ├── <doy>/
    │           ├── <doy>/
    │           ├── <doy>/
    │           ├── <doy>/
    │           ├── <doy>/
    │           ├── <doy>/
    │           ├── <doy>/
    │           └── <doy>/
    └── Daily/
        └── <satellite>/
            └── <year>/
                └── <month>/
                    ├── <day>/ (previous data if available and
hereafter)
                    ├── <day>/
                    ├── <day>/
                    ├── <day>/
```

```
├── <day>/
├── <day>/
├── <day>/
└── <day>/
```

where:

- `<satellite>` can be JPSS1, JPSS2, SNPP.
- `<year>` is a four-digit year.
- `<doy>` is a three-digit Julian day of the year; should be 14 weeks of data.
- `<month>` is a two-digit month.
- `<day>` is a two-digit day; should be 8 days of data.

Execution Commands

To execute the Daily or Weekly unit, run the following commands:

```
python $home/CODE/leaf-area-index/wrapper/launch_lai.py
$home/CODE/leaf-area-index/config/docker_config.yaml
```

4.1.2. Processing Controls

There must be at least one YAML file present in the algorithm package that contains information concerning the setup of the Docker run command. For convenience, all items in the file that have a possibility of variation or are system specific will be located at the top of the YAML file as anchors. Each of these arguments are listed in Table 4-2.

Table 4-2 - Docker YAML File Arguments

Argument Name	Description of Value
&wrapper	Path to the directory containing the python wrapper scripts.
&app_yaml	Path to the application YAML file.
&packages	Path to the surface reflectance algorithm python files.
&bin_dir	Path to the directory containing the executables.
&ancillary_dir	Path to the ancillary files.
&input	Path to the input files.
&output_dir	Path to the output location.
&log_dir	Path to the log output.
&sat	Satellite ID for docker container naming (j01, j02, or npp).
&caseid	Case ID for the run.

4.2. Installation

4.2.1. Installation Items

For more information concerning the installation items associated with the LAI algorithm package, refer to Section **Error! Reference source not found. - Error! Reference source not found.**.

4.2.2. Compilation Procedures

A statically compiled executable has been provided in **$home/CODE/leaf-area-index/bin**. If an executable need to be compiled, the instructions are as follows:

1. Change to the **$home/CODE/leaf-area-index** directory.
2. Run `./build_alg clean` to make sure no executable exists.
3. Run `./build_alg`.
4. In the **$home/CODE/leaf-area-index/bin** directory, the executables gridding, main_vdsr_l3, and tgsf_main should have been created.

4.2.3. Installation Procedures

4.3. Configuration Procedures

Please refer to the README document included with this delivery package.

See the master NCCF SMM: Installation Procedures (or Document Object: 77).

4.3.1. Production Rules

Daily Unit

The Daily unit should be run every 24 hours when the surface reflectance and geometry data are available.

The Daily unit takes granule Lat/Lon data as input, sourced from VIIRS GITCO data, and generates mapping indices that store information for mapping granule data to a global sinusoidal grid at a resolution of 500m. The unit then takes these mapping indices, along with VIIRS L2 SR and GITCO data, and generates global gridded SR data, including VIIRS I1, I2, and I3 bands, solar/satellite angles, and quality flags. Finally, the unit takes the global gridded SR data and produces the global daily LAI dataset.

Weekly Unit

The Weekly unit can be run daily after the Daily unit completes. At a minimum, it should be run every 8 days.

NOTE: The Weekly unit creates output every 8 days starting on the eighth day of the year (January 8th). Since a year is not exactly divisible by 8-day periods, a few days into the next year are needed to complete the final 8-day cycle. If the unit is run every day, there will be no output except on every 8-day cycle.

The Weekly unit takes the last 8 days of Daily LAI as inputs and performs compositing to generate an 8-day LAI dataset with broader coverage and reduced noise. It then takes this dataset and generates a smoothed and gap-free 8-day LAI by utilizing information from the previous 14 weeks' 8-day data along with the LAI climatology. Finally, the unit aggregates the smoothed and gap-free 8-day LAI tile data to form the final global product, which involves converting the resolution from 500m to 1km and reprojecting the original sinusoidal tile grids to the equal Lat/Lon grids.

4.4. Operations Procedures

4.4.1. Normal Operations

Please refer to the master NCCF SMM: Procedures for Normal Operations (or Document Object 10).

4.4.2. Data Preparation

4.5. Distribution

4.5.1. Data Transfer/Communications

Please refer to the master NCCF SMM: Data Transfer/Communications and Data Preparation (or Document Object 73, 81).

4.5.2. Distribution Restrictions

There are no restrictions on the distribution of LAI products.

4.5.3. Product Retention Requirements

No specific requirement for this product.

4.5.4. External Product Tools

There are no external product tools associated with the LAI algorithm package. Since the final output files of the algorithm package are NetCDF4 files, external users can choose their own preferred tools to display and analyze these output files.

5. Monitoring and Maintenance

5.1. Job Monitoring

5.1.1. Product Monitoring and Visualization

Product quality is monitored using the NCCF Product Monitoring Tool athttps://nccf-prod-dashboard.nccf.nesdis.noaa.gov/mtool/index.html.

Users can use this page to monitor summaries of the LAI quality based on parameter thresholds determined by the PAL.

The NCCF Products Visualization Page is located at https://www.ospo.noaa.gov/products/land/vegetation/lai/.

LAI products are generated weekly.

5.2. Data Signal Monitoring

5.3. Product Monitoring

5.3.1. Unit Test Plans

Testing of the LAI algorithm package's products occurs with each update to the algorithm package. The science teams, who develop the products, test them for accuracy and validation. The STAR group tests the algorithm and scripts to ensure that requirements are met. Then, Operations must test the products to make sure that they run successfully on their systems. If there are problems in any one of the testing procedures, then the relevant groups must work together to correct any issues.

5.3.2. Internal Product Tools

There are no internal product tools provided with the current LAI algorithm package.

5.3.3. Performance Statistics

5.3.4. Product Monitoring

5.3.5. Product Criticality

5.4. Maintenance

5.4.1. Monitoring and Maintenance

See the master NCCF SMM: Maintenance Utilities (or Document Object: 84).

5.4.1.1. Ingest Monitoring

See the master NCCF SMM: Data Transfer/Communications and Data Preparation (or Document Object: 73).

5.4.2. Science Maintenance

If applicable, product quality monitoring is performed by the OSPO product quality monitoring system and the STAR developers. STAR and OSPO personnel should communicate regularly to discuss potential data quality issues along with formulating and scheduling updates to the LAI science code.

5.4.3. Library Maintenance

See the master NCCF SMM: Library Maintenance (or Document Object: 71).

5.4.4. Special Maintenance Procedures

At the end of each year, the following steps need to be performed:

- Reset January as the first day of the first week.
- The 46th weekly always goes 3 days into a new year (2 days for a leap year).

5.4.5. Maintenance Utilities

5.5. Program Backup Procedures

See the master NCCF SMM: Data Recovery Procedures and Program Recovery Procedures (or Document Object: 89, 90).

6. Troubleshooting

6.1. Program Diagnosis and Recovery

See the master NCCF SMM: Problem Diagnosis and Recovery Procedures (or Document Object: 82).

6.1.1. Quality Control Output

Details about program execution can be found in the log files produced by the algorithm package. Each run produces multiple logs that can be used to determine if the run was successful or if there were errors.

Details about product quality can be found in the **quality_information** variable contained within the output files (see Table 7-4: LAI NetCDF4 Output File Description).

6.1.2. Error Correction

See the master NCCF SMM: Error Correction – Warnings and Messages for Systems and Error Codes, Menus and Navigation (or Document Object: 43,44,45).

6.1.3. Problem Diagnosis and Recovery Procedures

See the master NCCF SMM: Problem Diagnosis and Recovery Procedures (or Document Object: 82).

6.1.4. Data Recovery Procedures

See the master NCCF SMM: Data Recovery Procedures (or Document Object: 89).

6.1.5. Program Recovery Procedures

6.2. Application Shutdown and Restart

See the master NCCF SMM: Program Recovery Procedures (or Document Object: 90).

6.2.1. Application Shutdown Procedures

See the master NCCF SMM: Application Shutdown Procedures (or Document Object: 94).

6.2.2. Application Restart Procedures

See the master NCCF SMM: Application Restart Procedures (or Document Object: 92).

6.3. System Shutdown and Restart

See the master NCCF SMM: Reboot Procedures, Restart Procedures and Shutdown Procedures (or Document Object: 83, 93, 95).

6.3.1. System Shutdown Procedures

6.3.2. System Restart Procedures

6.3.3. System Reboot Procedures

7. Appendix

7.1. Data Flow

The VIIRS LAI product is designed to be a temporally smoothed, global, gap-free dataset. The operational procedure is segmented into three phases, as depicted in Figure 7-1. The first two steps are daily processing, with up to 8 days' data being sustained for the weekly processing, which will be run every 8 days.

1. **Daily Surface Reflectance Generation**: Utilizing the VIIRS gridding tool, granule data is mapped onto a global grid in a sinusoidal projection. The surface reflectance (SR) compositing process then identifies and selects the highest quality SR and corresponding angles for each grid cell.
2. **Daily LAI Retrieval**: A previously trained machine learning algorithm performs the clear-sky LAI retrieval, leveraging the daily SR together with auxiliary data.
3. **8-Day LAI Compositing and Post-Processing**: From the daily LAI outputs, the optimal quality LAI is chosen for each 8-day interval. Subsequently, a temporal smoothing and gap-filling (TSGF) procedure is applied to produce the final, gap-free product.

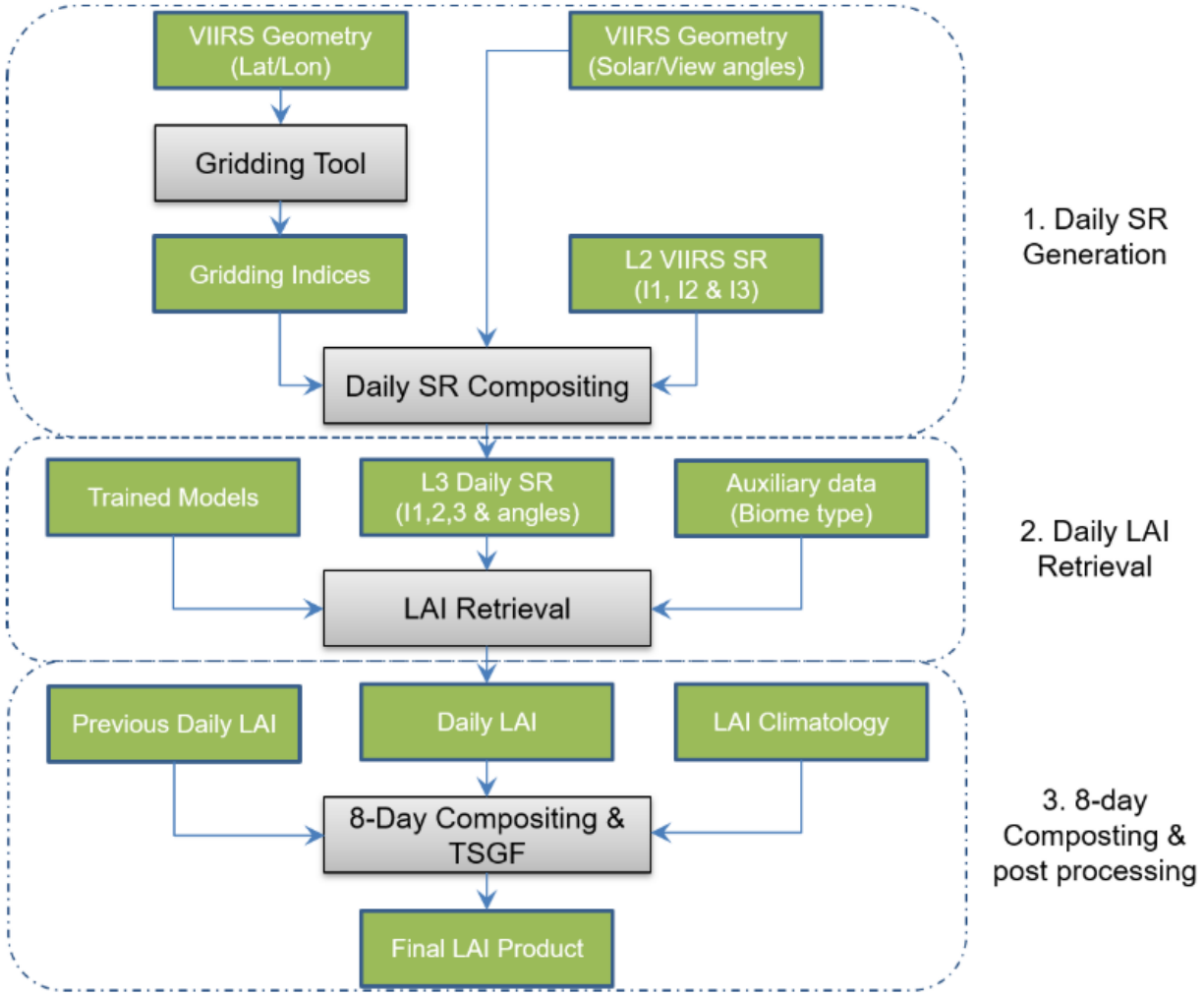

Figure 7-1 - LAI Processing Architecture

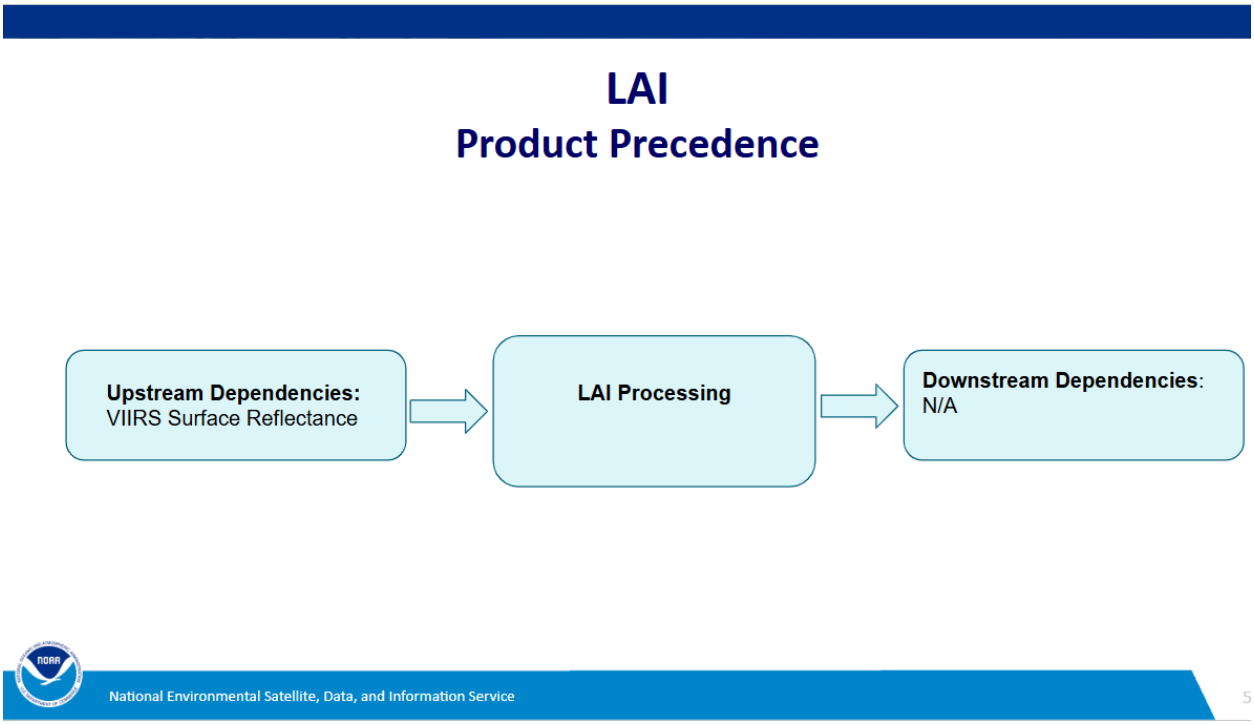

Figure 7-2 - LAI Product Precedence

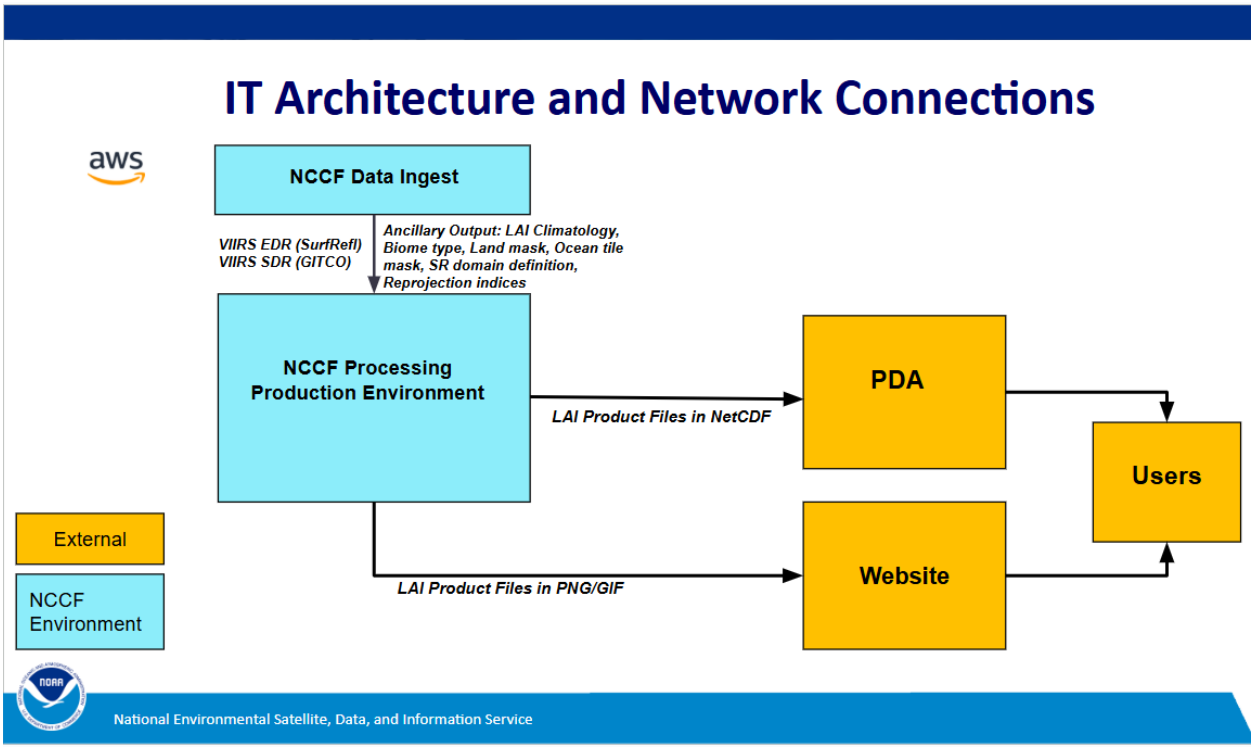

Figure 7-3 - IT Architecture & Network

7.2. Input Data Files

Table 7-1 lists information for the LAI input files.

Table 7-1 - Primary Input File Naming Conventions

Description of File	Type of File	Naming Convention
L1B VIIRS Geometry	Primary Input	GITCO_<sat>_d<YYYYmmdd>_t<HHMMSSf>_ e<HHMMSSf>_b<orbit>_c<YYYYmmddHHMMSSssssss>_<source>.h5
L2 VIIRS Surface Reflectance	Primary Input	SurfRefl_v<x>r<y>_<sat>_s<YYYYmmddHHMMSSf>_ e<YYYYmmddHHMMSSf>_c<YYYYmmddHHMMSSf>.nc

Where:

<sat>	→	The satellite source: npp, n20, or n21.
d<YYYYmmdd>	→	The date of the start of the granule in 4-digit year, 2-digit month, and 2-digit day format.
t<HHMMSSf>	→	The start time of the granule in 2-digit hour, 2-digit minute, 2-digit second, and 1-digit microseconds format.
e<HHMMSSf>	→	The end time of the granule in 2-digit hour, 2-digit minute, 2-digit second, and 1-digit microseconds format.
<orbit>	→	The satellite orbit number.
c<YYYYmmddHHMMSSssssss>	→	The creation timestamp for the granule in 4-digit year, 2-digit month, 2-digit day, 2-digit hour, 2-digit minute, 2-digit second, and 6-digit microseconds format.
<source>	→	The source of the file, including, but not limited to, noac_ops and oeac_ops.

<x>	→	The version number of the Surface Reflectance file.
<y>	→	The revision number of the Surface Reflectance file.
s<YYYYmmddHHMMSSf>	→	The start timestamp for the granule in 4-digit year, 2-digit month, 2-digit day, 2-digit hour, 2-digit minute, 2-digit second, and 1-digit microseconds format.
e<YYYYmmddHHMMSSf	→	The end timestamp for the granule in 4-digit year, 2-digit month, 2-digit day, 2-digit hour, 2-digit minute, 2-digit second, and 1-digit microseconds format.

7.3. Ancillary Data Files

7.3.1. Dynamic Ancillary Data

There is no dynamic ancillary data used for the Daily or Weekly unit.

7.3.2. Static Ancillary Data

Static ancillary data needed to generate the LAI Daily and Weekly products are located in the following directory:

- $home/CODE/leaf-area-index/ancillary_data

7.4. Look Up Tables

All look up tables associated with the LAI algorithm package are included with the static ancillary data described in Section 7.3.

7.5. Intermediate Data Set Description

Table 7-2 lists information for the LAI intermediate output files for the Daily and Weekly units. The Daily LAI and Weekly LAI files are also input files for the final product.

Table 7-2 - Intermediate File Naming Conventions for Daily Unit

Description of File	Type of File	Naming Convention
Daily LAI	NetCDF4	DLY-LAI_v1r0_<sat>_h<xx>v<yy>_s<YYYYmmddHHMMSSf>_ e<YYYYmmddHHMMSSf>_c<YYYYmmddHHMMSSf>.nc
Weekly LAI	NetCDF4	WKL-LAI_v1r0_<sat>_h<xx>v<yy>_s<YYYYmmddHHMMSSf>_ e<YYYYmmddHHMMSSf>_c<YYYYmmddHHMMSSf>.nc

Where:

<sat>	→	The satellite source: npp, n20, or n21.
h<xx>	→	The tile name from 00 to 71 from left to right.
v<yy>	→	The tile name from 00 to 71 from north to south.
s<YYYYmmddHHMMSSf>	→	Start time in 4-digit year, 2-digit month, 2-digit day, 2-digit hour, 2-digit minute, 2-digit second, and 1-digit microseconds format.
e<YYYYmmddHHMMSSf>	→	End time in 4-digit year, 2-digit month, 2-digit day, 2-digit hour, 2-digit minute, 2-digit second, and 1-digit microseconds format.
c<YYYYmmddHHMMSSf>	→	Creation time in 4-digit year, 2-digit month, 2-digit day, 2-digit hour, 2-digit minute, 2-digit second, and 1-digit microseconds format.

7.6. Output Data Set Description

Table 7-3 lists information for the LAI output files.

Table 7-3 - Output File Naming Conventions

Description of File	Type of File	Naming Convention
Weekly Global LAI	NetCDF4	WKL-LAI-GLB_v1r0_<sat>_s<YYYYmmddHHMMSSf>_ e<YYYYmmddHHMMSSf>_c<YYYYmmddHHMMSSf>.nc
LAI Log	Log	LAI_<sat>_<type>_s<YYYYmmddHHMMSSf>_ e<YYYYmmddHHMMSSf>_c<YYYYmmddHHMMSSf>.log

Where:

<sat>	→	The satellite source: npp, n20, or n21.

<type>	→	Daily or Weekly.
s< YYYYmmddHHMMSSf >	→	The start timestamp for the granule in 4-digit year, 2-digit month, 2-digit day, 2-digit hour, 2-digit minute, 2-digit second, and 1-digit microseconds format.
e< YYYYmmddHHMMSSf >		The end timestamp for the granule in 4-digit year, 2-digit month, 2-digit day, 2-digit hour, 2-digit minute, 2-digit second, and 1-digit microseconds format.
c< YYYYmmddHHMMSSf >		The creation timestamp for the granule in 4-digit year, 2-digit month, 2-digit day, 2-digit hour, 2-digit minute, 2-digit second, and 1-digit microseconds format.

The contents of the NetCDF output file are described in Table 7-4.

Table 7-4 - LAI NetCDF4 Output File Description

Variable	Type	Description	Dim	Units	Range
LAI	16-bit integer	8-Day Global Leaf Area Index	20000 x 40000	m2/m2	N/A
quality_information	String	Total number of retrievals, percentage of optimal retrievals, percentage of sub optimal retrievals, percentage of bad retrievals	1	N/A	N/A

The metadata for the NetCDF output file is described in Table 7-5.

Table 7-5 - LAI NetCDF4 Output File Metadata

Attribute	Description	Type	Array Size
Conventions	A text string identifying the netCDF conventions followed.	String	Scalar
_NCProperties	NetCDF and HDF version numbers (automatically generated).	String	Scalar
cdm_data_type	The data type, as derived from Unidata's Common Data Model Scientific Data types and understood by THREDDS.	String	Scalar
creator_email	The email address of the person (or other creator type specified by the creator_type attribute) principally responsible for creating this data.	String	Scalar

Attribute	Description	Type	Array Size
creator_name	The name of the person (or other creator type, such as a RDAC, specified by the creator_type attribute) principally responsible for creating this data.	String	Scalar
creator_url	The URL of the of the person (or other creator type specified by the creator_type attribute) principally responsible for creating this data.	String	Scalar
date_created	The date on which this version of the data was created.	String	Scalar
day_night_data_flag	Describes sunlight conditions for observation: day, night, or both.	String	Scalar
geospatial_lat_max	Describes a simple upper latitude limit; may be part of a 2- or 3-dimensional bounding region. Specifies the northernmost latitude covered by the dataset.	64-bit floating-point	1
geospatial_lat_min	Describes a simple lower latitude limit; may be part of a 2- or 3-dimensional bounding region. Specifies the southernmost latitude covered by the dataset.	64-bit floating-point	1
geospatial_lat_resolution	Information about the targeted spacing of points in latitude.	String	Scalar
geospatial_lat_units	Units for the latitude axis described in geospatial_lat_min and geospatial_lat_max attributes.	String	Scalar
geospatial_lon_max	Describes a simple longitude limit; may be part of a 2- or 3-dimensional bounding region. Specifies the easternmost longitude covered by the dataset.	64-bit floating-point	1
geospatial_lon_min	Describes a simple longitude limit; may be part of a 2- or 3-dimensional bounding region. Specifies the westernmost longitude covered by the dataset.	64-bit floating-point	1
geospatial_lon_resolution	Information about the targeted spacing of points in longitude.	String	Scalar
geospatial_lon_units	Units for the longitude axis described in geospatial_lon_min and geospatial_lon_max attributes.	String	Scalar
history	Provides an audit trail for modifications to the original data.	String	Scalar
id	An identifier for the data set, provided by and unique within its naming authority.	String	Scalar
institution	The name of the institution principally responsible for originating this data.	String	Scalar
instrument	Name of the contributing instrument(s) or sensor(s) used to create this data set or product. I	String	Scalar

Attribute	Description	Type	Array Size
keywords	A comma-separated list of key words and/or phrases. Keywords may be common words or phrases, terms from a controlled vocabulary (GCMD is often used), or URIs for terms from a controlled vocabulary.	String	Scalar
metadata_link	A URL that gives the location of more complete metadata.	String	Scalar
naming_authority	The organization that provides the initial id for the dataset.	String	Scalar
platform	Name of the platform(s) that supported the sensor data used to create this data set or product. Platforms can be of any type, including satellite, ship, station, aircraft or other.	String	Scalar
processing_level	A textual description of the processing (or quality control) level of the data. Options are: L2P, L3U, L3C, L3S, L4 and GMPE.	String	Scalar
production_environment	Processing string responsible for generating the product.	String	Scalar
production_site	Processing site for the product.	String	Scalar
project	The name of the project(s) principally responsible for originating this data.	String	Scalar
publisher_email	The email address of the person (or other entity specified by the publisher_type attribute) responsible for publishing the data file or product to users, with its current metadata and format.	String	Scalar
publisher_name	The name of the person (or other entity specified by the publisher_type attribute) responsible for publishing the data file or product to users, with its current metadata and format.	String	Scalar
publisher_url	The URL of the person (or other entity specified by the publisher_type attribute) responsible for publishing the data file or product to users, with its current metadata and format.	String	Scalar
source	The method of production of the original data.	String	Scalar
standard_name_vocabulary	The name and version of the controlled vocabulary from which variable standard names are taken.	String	Scalar
summary	A paragraph describing the dataset, analogous to an abstract for a paper.	String	Scalar
time_coverage_end	Describes the time of the last data point in the data set.	String	Scalar
time_coverage_start	Describes the time of the first data point in the data set.	String	Scalar

Attribute	Description	Type	Array Size
title	A short phrase or sentence describing the dataset.	String	Scalar

7.7. Archive Data Description

The final LAI product, which is a temporal smoothing 8-days global product, will be archived at NCEI.

END OF DOCUMENT

8. Acronyms

Acronym	Definition
ASSISTT	Algorithm Scientific Software Integration and System Transition Team
ATBD	Algorithm Theoretical Basis Document
CCAP	Cloud Containerized Algorithm Package
DDS	Data Distribution System
DOC	Department of Commerce
ERT	Earth Resources Technology, Inc.
ESPC	Environmental Satellite Processing Center
EUM	External Users' Manual
GDAS	Global Data Assimilation System
GEOV2	Geoland2
GFS	Global Forecast System
GLASS	Global Land Surface Satellite
JPSS	Joint Polar Satellite System
LAI	Leaf Area Index
MODIS	Moderate-resolution Imaging Spectroradiometer
NCCF	NESDIS Common Cloud Framework
NCEP	National Centers for Environmental Prediction
NDAS	North American Model Data Assimilation System
NESDIS	National Environmental Satellite, Data, and Information Service
NetCDF	Network Common Data Form
NOAA	National Oceanic and Atmospheric Administration
OCS	Office of Common Services
OMS	Operations, Maintenance, and Sustainment
OSPO	Office of Satellite and Product Operations
PAL	Product Area Lead
PDA	Product Distribution and Access
PIB	Product Implementation Branch
PPM	Project Portfolio Management
QA	Quality Assurance
SMM	System Maintenance Manual
S-NPP	Suomi National Polar-orbiting Partnership
SR	Surface Reflectance
STAR	Center for Satellite Applications and Research
TSGF	Temporal Smoothing and Gap-Filling
VIIRS	Visible Infrared Imaging Radiometer Suite