Anomaly Detection for ARM Radiometers using Machine Learning Algorithms

Laurie Gregory[a], Jeffery Mitchell[a], Richard Wagener[a], Lynn Ma[a] and Laura Riihimaki[b]

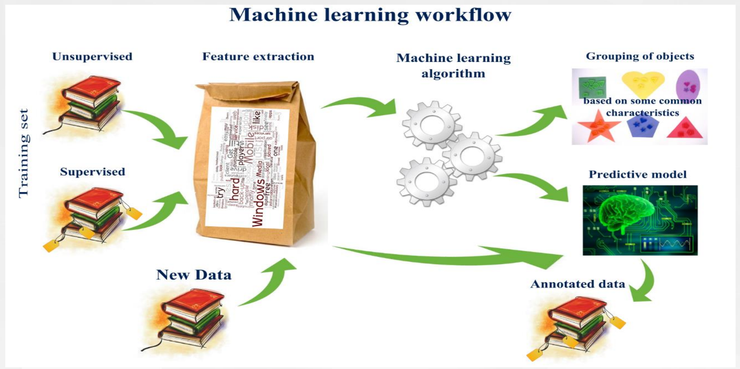
[a]Brookhaven National Laboratory; [b]Pacific Northwest National Laboratory

Machine Learning

Machine Learning
Machine learning is a data-based process. It works with data, often in multiple dimensions, to discover patterns that can be used to analyze data and make predictions. The use of machine learning applications has exploded over the past several years and is used in applications such as Movie/Purchase Recommendation Engines (Netflix, Amazon), Self-driving cars (Google, Uber), Financial Market Prediction and Natural Language Processing (Siri). Here we explore the use of machine learning for ARM instrument data quality applications.

ARM Applications
We examine the use of machine learning to aid ARM instrument mentors, whose main responsibilities are to ensure their instruments are producing high quality data. These include diagnosing and fixing instrument problems, checking data quality for both short term and long term data, and communicating data quality to users.

Machine Learning can be helpful to detect instrument data quality issues since it can:
- Discover trends
- Quickly identify multiple characteristics of a problem
- Identify problems that cannot be seen by eye
- Find problems across multiple datasets and data types
- Help to diagnose problems for instruments in the field
- Help to identify problems in sensor networks (could be used across multiple sites, models)
- Identify trends even when there is much variation in the dataset due to weather and function of the instrument



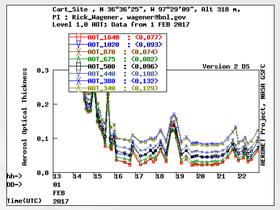
Step 1: Identify Patterns

Step 1: Identify patterns in the data.

Here we looked for patterns indicating potential problems with the instrument. We train the model to distinguish between data showing normal operation and data indicating problems.

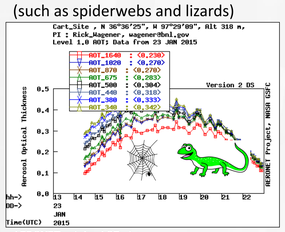

L. Gregory::CSPHOT at BNL, 2011

Normal Operation



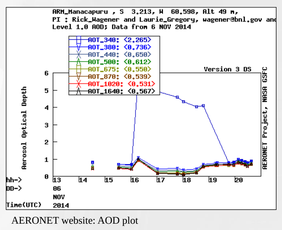
AERONET website: AOD plot

Tube Obstruction
(such as spiderwebs and lizards)



AERONET website: AOD plot

Filter Degradation

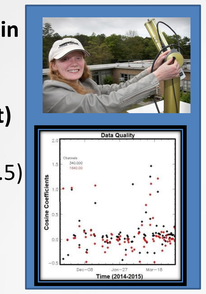


AERONET website: AOD plot

Step 2: Identify Features

Define patterns in the data in mathematical terms

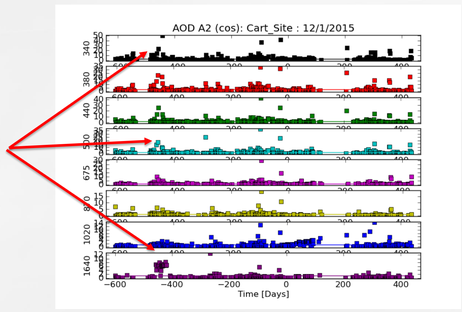
Example: Identify features that describe the cosine curve in the data (indication an obstruction).



The coefficent plot above shows the work of our summer student, Brooke Adams (top picture)

1) Fit to the curve using multiple-linear regression: AOD(t) = $A_0 + A_1*t + A_2*\cos(sza(t))$
where t is the fraction of the day (midnight = 0.0, noon = 0.5)
2) Identify features
For example AOD(noon) and the coefficients A1 and A2
3) Make cuts to account for normal variances in the data.
Sample cuts: Triplet var (for clouds)
4) Multiple features can be correlated to increase model sensitivity

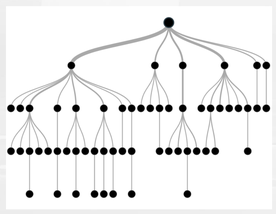
Days where spider web was obstructing the instrument.



SGP site (Oklahoma) over a 34 month time period from 4/1/2014 to 2/12/2017.

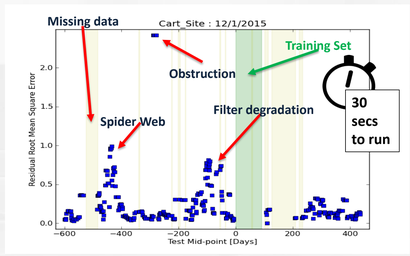
Step 3: Train and Run Model Results

Model



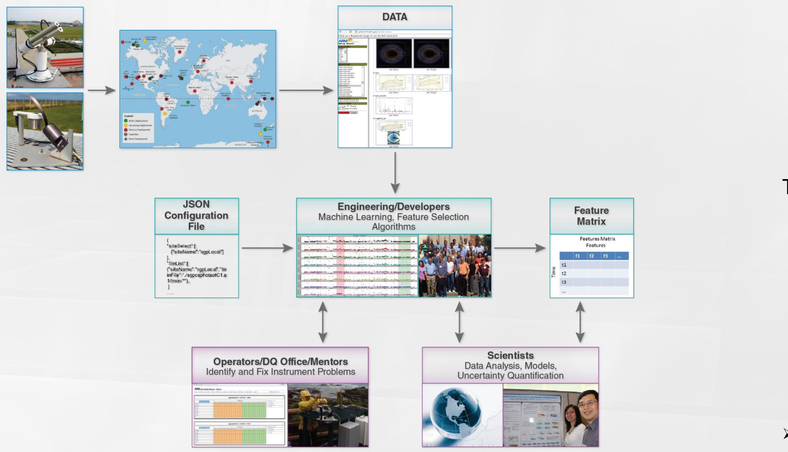
- Random forest regression model implemented
- Builds an ensemble of decision trees using a random sampling of a subset of both the training set and features
- Each decision tree is grown to minimize the residual sum of squares
- Final answer is the average of all decision trees

Results



- Model incorporates many features of instrument key measurements simultaneously
- "Trained" for periods when instrument operated normally
- Deviations against training fit indicate anomalies
- Validated using existing Data Quality Reports
- **Faster, more sensitive than human eyes, and automated**
- **One two-year run took only 30 seconds**
- Currently testing model in operational mode
- Model can be used for other sites

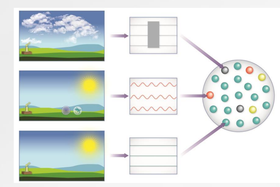
Applications/Future Work



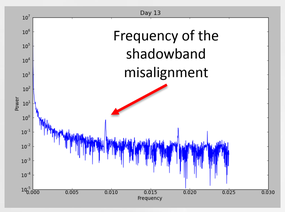
- Model is written in an Open Source Python Framework and used by mentors for Data Quality inspection
- Feature matrix is used by scientists for Uncertainty Quantification development
- Could be incorporated into ARM Development Environment (ADI)
- Scripts could be developed for DQ office

Other ARM applications for Machine Learning

Applying Machine Learning for data quality for multiple sites

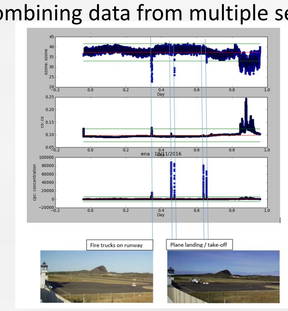


Testing on other instruments : MFRSR



- Currently testing use of machine learning on MFRSR Data
- So far, during the feature identification phase, a Fast Fourier Transform (FFT) was implemented to monitor the shadowband misalignment problem.
- The FFT was known to detect the shadowband problem (Alexandrov, 2007)
- This is the first time that the FFT has been automated.
- Over a 2.5 year period at the SGP site, all known days with problems were identified with only 2 isolated days with false positives.

Combining data from multiple sensors



We are exploring the use of combining image data from a camera with multiple AOS instruments. Check out and see our poster for more details: **"Identifying the Influence of Local Source Emissions on the Regional Representativeness of AOS Measurements Using Machine Learning"**

More Information:

Cimel (CSPHOT) Instrument Page:
http://www.arm.gov/instruments/csphot
Aeronet:
http://aeronet.gsfc.nasa.gov/
ARM eXternal Data Center (XDC):
http://www.xdc.arm.gov/, xdc_oper@arm.gov.
ARM Google:
http://google.arm.gov/ search for "Cimel OR CSPHOT OR CSPOT"

References:

Adams, B. , L. Gregory, R. Wagener, "Automatically detecting typical failure signatures in Cimel Sun-photometer data to improve data quality", Poster presented at New York Scientific Data Summit, NYU ,New York, August 2-5, 2015
Alexandrov, M.D, et al, "Optical depth measurements by shadow-band radiometers and their uncertainties", M.D. Alexandrov et al., APPLIED OPTICS _ Vol. 46, No. 33 _ 20 November 2007
Machine Learning workflow images: http://www.datascienceassn.org/content/machine-learning-workflow.
Instrument images: www.arm.gov
Applications images: www.arm.gov and www.gettyimages.com
Random Forest Image: https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python/

Acknowledgments

We would like to thank Yelena Belyvania for her help with graphics. We would also like to thank Connor Flynn for his help with the MFRSR data algorithms.