SAND2018-4644PE

Pecos and Canary Webinar

Katherine Klise, David Hart

Sandia National Laboratories, Albuquerque, NM

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

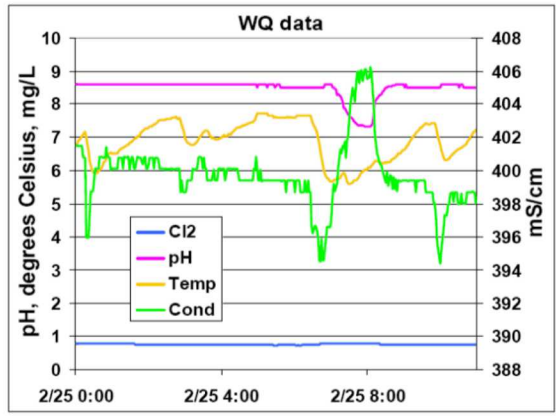
Canary and Pecos

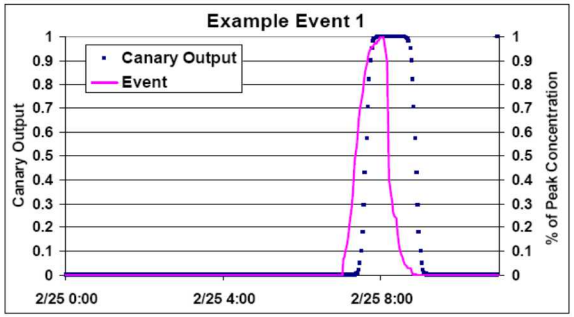
Sandia National Laboratories

- Motivation
  - Analyze large amounts of data that are collected from different types of sensors across multiple sites
  - Run quality control tests to distinguish between normal and anomalous conditions
  - Alert system operators and the public when conditions have changed
  - Generate reports and graphics
  - Identify issues quickly
  - Increase data integrity
  - Enhance understanding
- General purpose techniques have been applied to a wide range of applications
  - Water quality monitoring, treated and source water
  - Photovoltaics system performance
  - Weather station monitoring
  - Marine hydrokinetics performance
  - Computing performance

Canary

- Developed to monitor water quality data
  - Sensors collecting pH, turbidity, chlorine, conductivity, etc.
- Applications include Greater Cincinnati Water Works (17 sites since 2007) and Singapore Public Utility Board (70 sites since 2009)
- Canary functionality
  - Gathers data from a central database (generally every 1 to 5 minutes) using a user defined history window
  - Sequentially analyzes data using statistical algorithms including increment, linear filter, and MVNN
  - Integrates pattern matching and external data sources (operational controls, rain events)
  - Real time results are stored in a database or files
  - Generates graphics
  - Analysis can be run on an automated schedule based on connections to the database





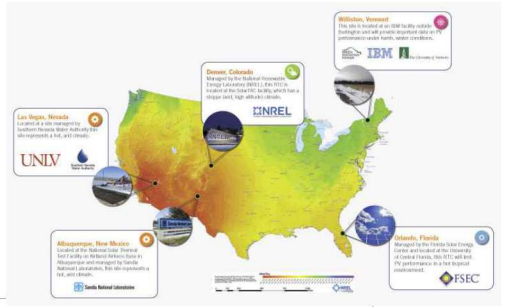
Canary Framework

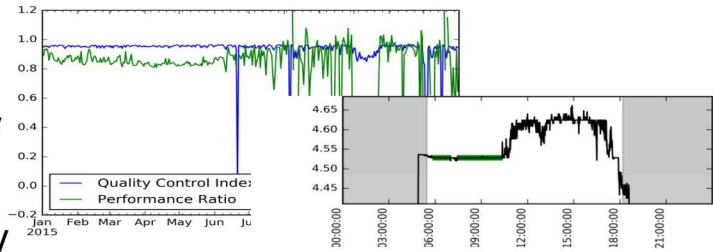
- Open source software, Matlab and Java versions
- YAML configure file stores
  - CANARY controls
  - Timing options
  - Datasources definitions
  - Signals definitions
  - Algorithms definitions
  - Monitoring stations definitions

```yaml
--- # This is a complete description of the YAML items
canary:
  run mode : "ENUM"
  control type: "ENUM"
  control messenger: "unique ID"
  driver files:
    - "jar file"

timing options:
  dynamic start-stop: "<bool> yes or no"
  date-time format: "format string"
  date-time start:  "formatted date and time"
  date-time stop:  "formatted date and time"
  data interval:  "formatted time"
  message interval:  "formatted time"

datasources:
- id:  "unique datasource ID"
  type:  "ENUM"
  location:  "URL"
  enabled:  "<bool> yes or no"
  time-step options:
    time-step field:  "field name"
    conversion function:  "function call up to string"
    conversion format:  "format specific to function"
  database options:
    time drift:  "<float> partial day"
    JDBC2 class name:  "full class name"
    input table:  "table name"
    input format:  "ENUM"
    input fields:
      time-step:  "field name (DateTime)"
      parameter tag:  "field name (Char*)"
      parameter value:  "field name (Real)"
      parameter quality:  "field name (Char*)"
    output table:  "table name"
    output format:  "ENUM"
    output fields:
      write conditions:  "ENUM"
      time-step:  "field name (DateTime)"
      instance id:  "field name (Char*)"
```

Pecos

- Developed to monitor data collected at DOE Photovoltaic Regional Test Centers
  - Weather stations and photovoltaic systems collecting irradiance, wind speed, temperature, voltage, current, power, etc. at 1 second to 1 minute resolution
  - 1.9 million data points per day (5 locations, 29 systems)
- Pecos functionality
  - Gathers data from a from a wide range of formats, including databases, file, from the web, directly from sensors. User defines timeframe.
  - Runs built in or custom quality control tests to identify anomalous conditions
  - Integrates custom analysis or external data into quality control tests
  - Computes general and custom performance metrics
  - Generates HTML formatted dashboards, reports, and interactive graphics
  - Analysis can be run on an automated schedule using OS scheduler





	New Mexico	Florida	Vermont	Nevada
Weather	Irradiance 1.00 Wind 1.00 Air Pressure 1.00 Humidity 1.00 Rainfall 1.00 Datalogger 1.00 Detailed Report Interactive Plot	Irradiance 1.00 Wind 1.00 Air Pressure 1.00 Humidity 1.00 Rainfall 1.00 Datalogger 1.00 Detailed Report Interactive Plot	Irradiance 0.57 Wind 1.00 Air Pressure 1.00 Humidity 1.00 Rainfall 1.00 Datalogger 1.00 Detailed Report Interactive Plot	Irradiance 1.00 Wind 1.00 Air Pressure 1.00 Humidity 1.00 Rainfall 1.00 Datalogger 1.00 Detailed Report Interactive Plot
Baseline	Irradiance 1.00 Temperature 1.00 Current 1.00 Voltage 1.00 Power 0.64 Detailed Report Interactive Plot	Irradiance 1.00 Temperature 0.19 Current 1.00 Voltage 1.00 Power 0.92 Detailed Report Interactive Plot	Irradiance 1.00 Temperature 1.00 Current 0.27 Voltage 1.00 Power 0.24 Detailed Report Interactive Plot	Irradiance 1.00 Temperature 1.00 Current 1.00 Voltage 1.00 Power 1.00 Detailed Report Interactive Plot

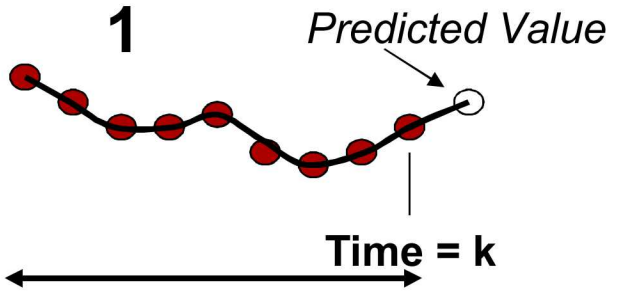
Pecos Framework

- Open source Python package for automated quality control and analysis of time series data
- Designed to be used for a wide range of applications
- Dependence on Numpy, Pandas, Matplotlib, Plotly, and Jinja facilitates a wide range of analysis and reporting capabilities
- Analysis is defined in the Python scripting environment (instead of in a configuration file)

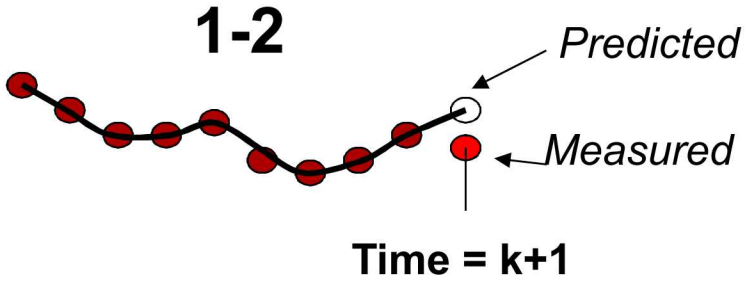

```
import pecos
import pandas as pd

pm = pecos.monitoring.PerformanceMonitoring()
...
df = pd.read_sql(sql_table_names,con=engine)
pm.add_dataframe(df)
pm.check_timestamp(900)
pecos.io.write_monitoring_report('report.html', pm)
pecos.graphics.plot_interactive_timeseries(pm.df, filename='plotly.html')
```

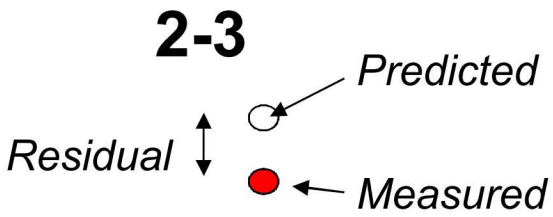
Key difference between Pecos and Canary

- Canary uses a history window to predict conditions at the next time step
- Outliers are automatically eliminated from future analysis
- Probability distribution is used to determine event probability



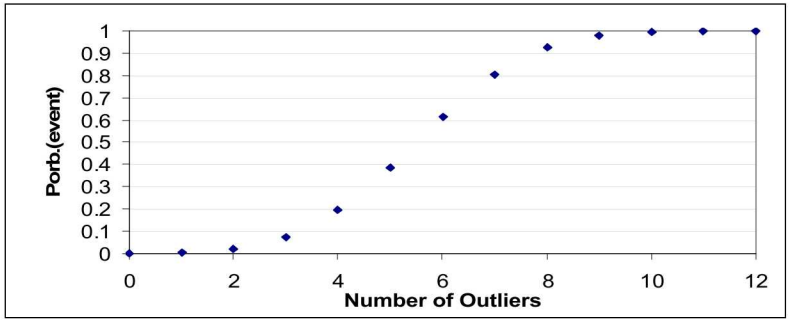
Window Length, (P = 9)





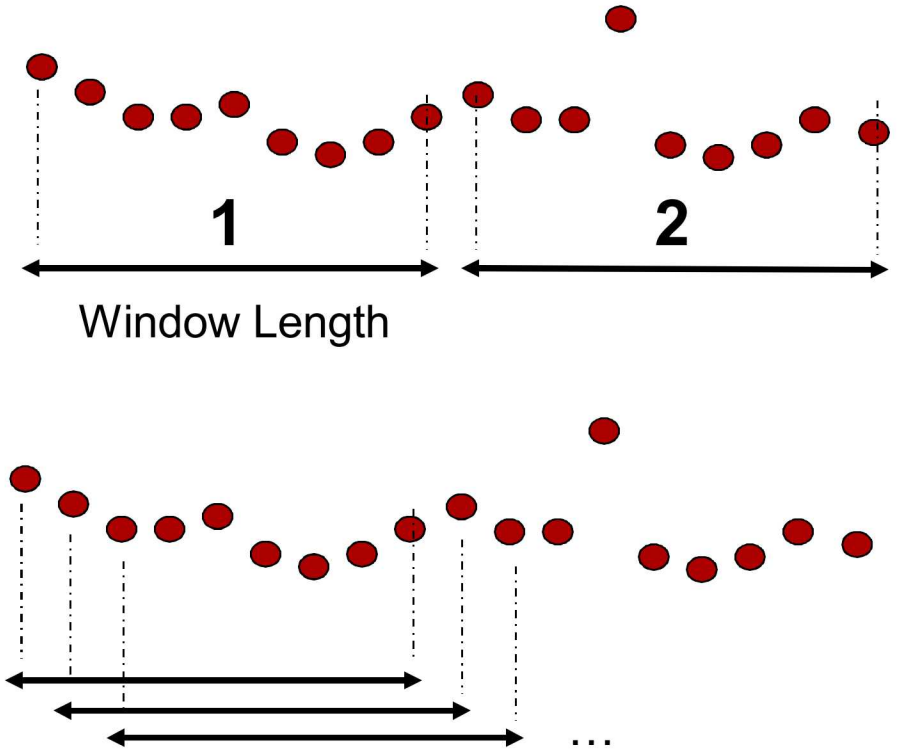
Residual at each time step compared to a threshold. Those that exceed the threshold are "outliers"

4 Probability distribution used to determine the event probability from the number of outliers over a given number of time steps



Key difference between Pecos and Canary

- Pecos uses the entire data set to identify anomalous conditions using simple to complex methods (some methods make use of moving windows)
- Outliers are not automatically eliminated from future analysis (though they could be)
- Probability distribution is not used to determine event probability (this could also be added)



Data can be preprocessed (filters, normalized)

Quality control tests include
- Missing, duplicate, non-monotonic timestamp
- Missing data
- Corrupt data
- Data outside expected range
- Stagnant, or abrupt change
- Outlier

For “data outside expected range”, data can be defined using the residual between measurement and model (simple model, system model, machine learning, or canary algorithms)

Pecos Overview

- Time-series data
- Pre-processing filters
- Composite signals
- Quality control tests
- Test results
- Reports and dashboards

Time series data

- Time series data loaded into Pecos as a Pandas DataFrame
  - Powerful time series analysis options
  - Datetime and timezone recognition
  - Merge multiple DataFrames in a single analysis
  - Data can be easily loaded from database, file, or web
- Data acquisition methods recently added to Pecos
  - Transfer data from sensors to an SQL database
- User defines the analysis timeframe (minute, hour, day, month, …)
- Data can be grouped and renamed according to type
- Repeat analysis automated using OS task scheduler (cron, tasks)

From database

```
sql_con= MySQLdb.connect(host=ip_address, port=…)
sql_query = "SELECT * FROM table…"
df = pandas.read_sql(sql_query, con=sql_con)
```

From file

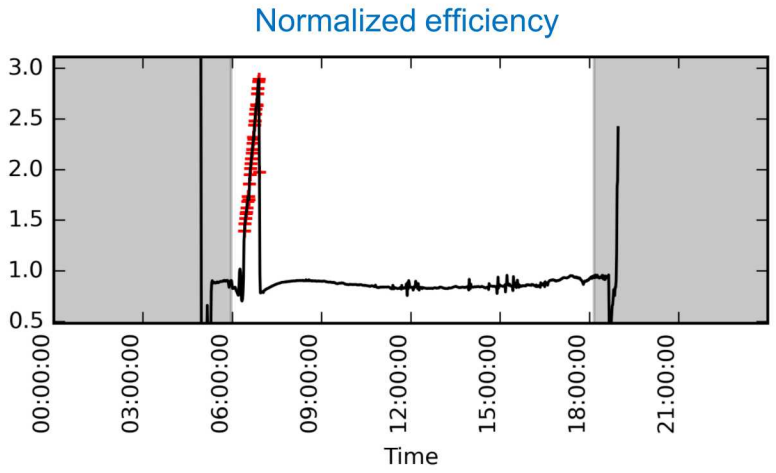
```
df = pandas.read_csv(filename)
```

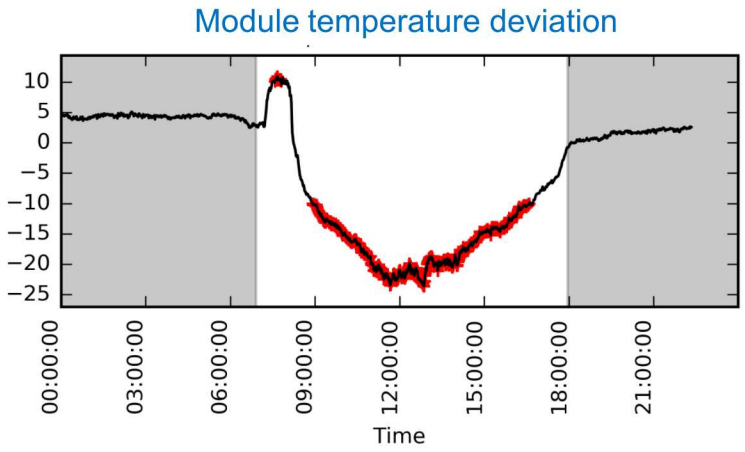
From the web

```
response = requests.get(url=http://developer.nrel.gov/pvdaq/api/…)
data = json.loads(response.text)
df = pandas.DataFrame(data=data['outputs']['data'])
```

Composite signals

- Composite signals are used to create new data from existing data or from a model
  - Compute relationships between data columns
  - Compare measured data to a model
    - System model
    - Machine learning
    - Clustering
- Examples
  - DC Power from current and voltage
  - Inverter efficiency from DC and AC power
  - Normalized efficiency from power and irradiance
  - Module temperature deviation
  - Relative error between model and data
- Composite signals can be used in the quality control tests

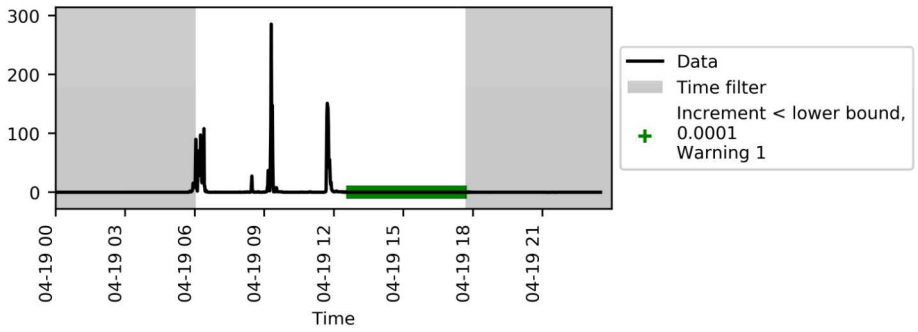




Quality Control tests

- Quality controls tests fall into five categories
  - Timestamp test
  - Missing data test
  - Corrupt data test
  - Range test
  - Dead sensor/abrupt change tests
  - Outlier test
- When a test fails, information is stored in a summary table which can be included in automated reports and saved to file/database. Graphics can be produced that pin point the data points that caused the test failure.

System Name	Variable Name	Start Date	End Date	Timesteps	Error Flag
PV System 1	Direct_Wm2	2017-04-19 12:36:00	2017-04-19 17:40:00	305	Increment < lower bound, 0.0001



Quality Control tests

- Timestamp test identifies duplicate, non-monotonic, and missing timestamps. Irregular timestamps can be preserved.
- Missing data test identifies column-time pairs that are missing.
- Corrupt data test screens for datalogger values that indicate corrupt data.

Original data

TIMESTAMP	Column A	Column B
1/1/2017 0:00	0	1
1/1/2017 1:00		2
1/1/2017 2:00	2	3
1/1/2017 3:00	3	4
1/1/2017 5:00	5	1
1/1/2017 6:00	6	2
1/1/2017 8:00	8	
1/1/2017 7:00	7	
1/1/2017 9:00	9	1
1/1/2017 9:00	9.5	2

Missing data

Missing timestamp

Non-monotonic timestamp

Duplicate timestamp

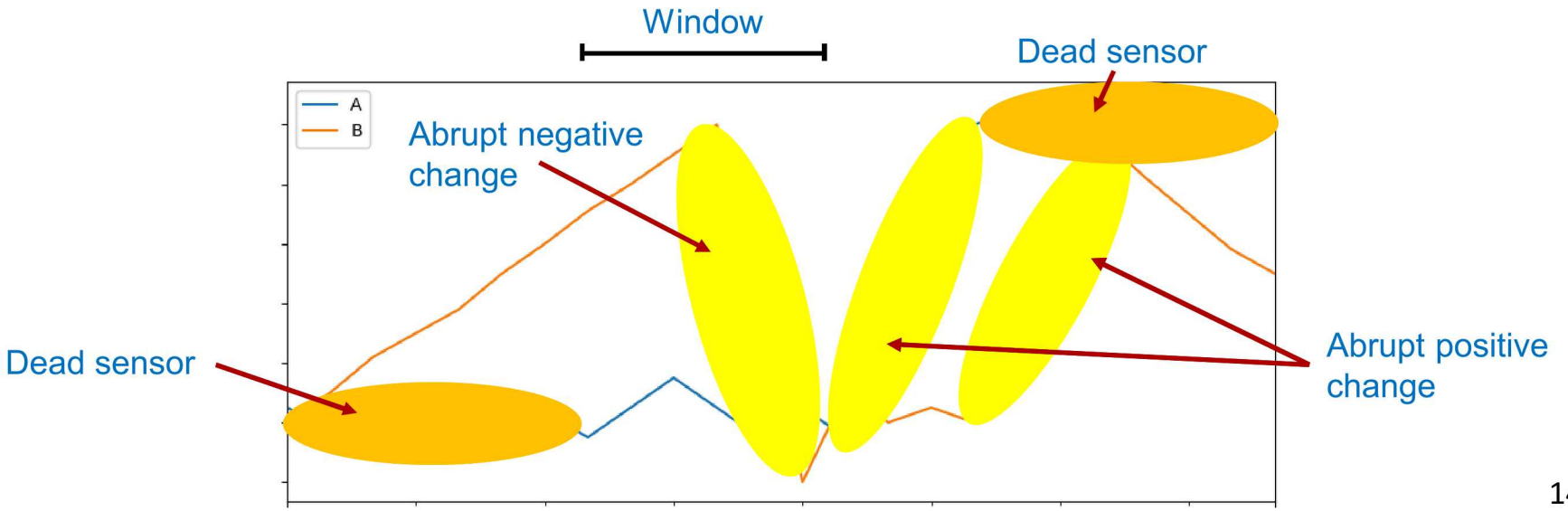
Corrupt data

Corrected data

TIMESTAMP	Column A	Column B
1/1/2017 0:00	0	1
1/1/2017 1:00	NaN	2
1/1/2017 2:00	2	3
1/1/2017 3:00	3	4
1/1/2017 4:00	NaN	NaN
1/1/2017 5:00	5	1
1/1/2017 6:00	6	2
1/1/2017 7:00	7	NaN
1/1/2017 8:00	8	NaN
1/1/2017 9:00	9	1

Quality Control tests

- Range tests checks if data is within expected bounds
  - Ambient temperature should be between -30 and 50 degrees C
  - Normalized efficiency (composite signal) should be between 0.5 and 1
- Dead sensor/abrupt change test checks if the difference between min and max is within expected bounds over a given time span
  - Voltage should not change by more than 80% rating within 15 minutes
  - The rain gauge should not increase by more than 2 inches in an hour
  - If the irradiance sensor changes by less than 0.0001 in 5 hours, it's probably dead
- Outlier tests checks if normalized data is within expected bounds (standard deviations) using a moving window



Test results, reports, and dashboards

- Quality control test results
  - System Name
  - Variable Name
  - Start time
  - End time
  - Number of timesteps
  - Error Flag
- Reports include
  - Custom Graphic
  - Performance Metric
  - Quality control test Results (table and graphics)
  - Notes include Pecos runtime errors and warnings
  - Configuration options used in the analysis
- Dashboard includes a cell for each system
  - Text
  - Graphics
  - Table
  - Links

	System Name	Variable Name	Start Date	End Date	Timesteps	Error Flag
1			2015-01-01 19:30:00	2015-01-01 19:30:00	1	Nonmonotonic timestamp
2			2015-01-01 17:00:00	2015-01-01 17:00:00	1	Duplicate timestamp
3			2015-01-01 05:00:00	2015-01-01 05:00:00	1	Missing timestamp
4		Wave Absolute Error C	2015-01-01 13:00:00	2015-01-01 14:45:00	8	Data > upper bound, 0.25
5	Simple	A	2015-01-01 12:15:00	2015-01-01 14:30:00	10	Increment < lower bound, 0.0001
6	Simple	B	2015-01-01 06:30:00	2015-01-01 06:30:00	1	Data < lower bound, 0
7	Simple	B	2015-01-01 15:30:00	2015-01-01 15:30:00	1	Data > upper bound, 1
8	Simple	C	2015-01-01 07:30:00	2015-01-01 09:30:00	9	Corrupt data

Pecos Dashboard

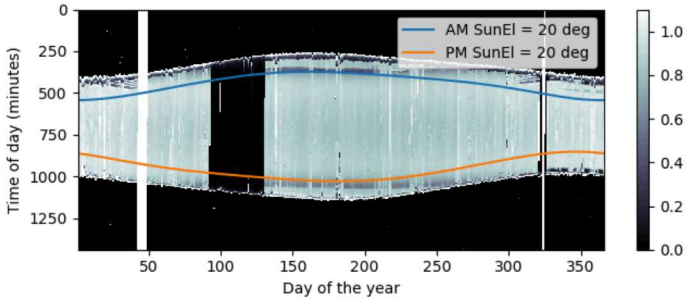
	System 1	System 2	System 3	System 4
Location 1	DA 1.00 QCI 1.00 EPI 0.99	DA 0.13 QCI 0.78 EPI nan	DA 1.00 QCI 1.00 EPI 0.98	DA 1.00 QCI 1.00 EPI 0.98
Location 2	DA 0.43 QCI 1.00 EPI 0.88	DA 1.00 QCI 0.93 EPI 0.01	DA 0.43 QCI 1.00 EPI 0.88	DA 1.00 QCI 0.93 EPI 0.98
Location 3	DA 1.00 QCI 0.93 EPI 0.98	DA 0.10 QCI 0.51 EPI nan	DA 0.57 QCI 0.84 EPI 0.63	DA 1.00 QCI 0.93 EPI 0.50

DA = Data availability
QCI = Quality control index
EPI = Energy performance index

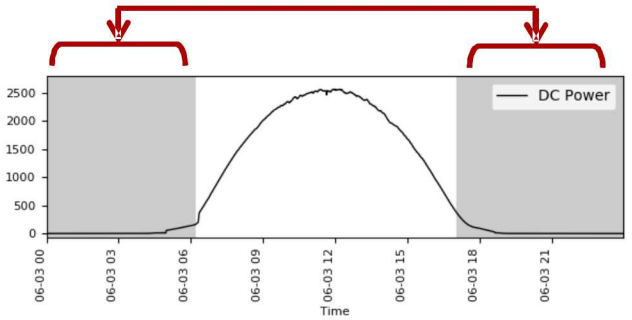
RTC Pecos Example

- Data inspection
  - Heatmaps used to identify systematic errors and trends, define filters, define quality control thresholds
- Basic quality control analysis is run daily, results emailed to stakeholders
  - Time filters based on sun position
  - Composite signals
  - System performance models using PVLIB
  - High level red-yellow-green status
  - Detailed level pinpointing quality control issues
- Summary reports are generated each year
  - Data availability
  - System availability
  - Quality control index
  - Energy performance index

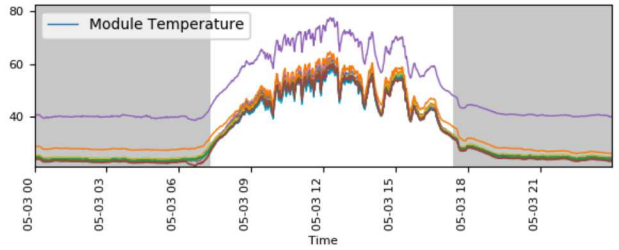
Normalized efficiency



Time filter based on sun position



Module temperature

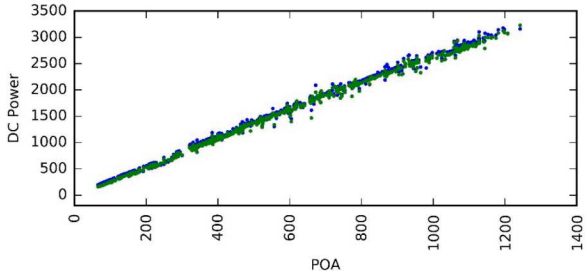


Daily Reports

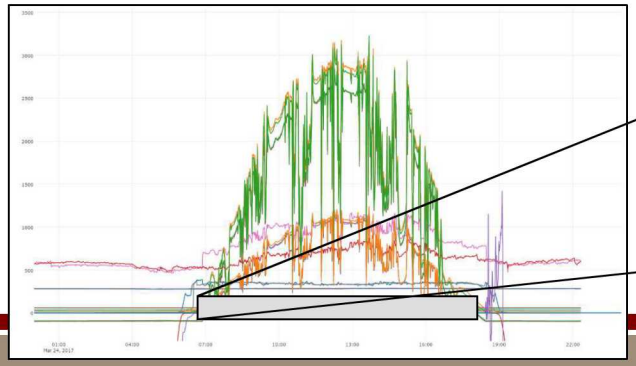
Red/yellow/green dashboard with links to details and interactive graphics

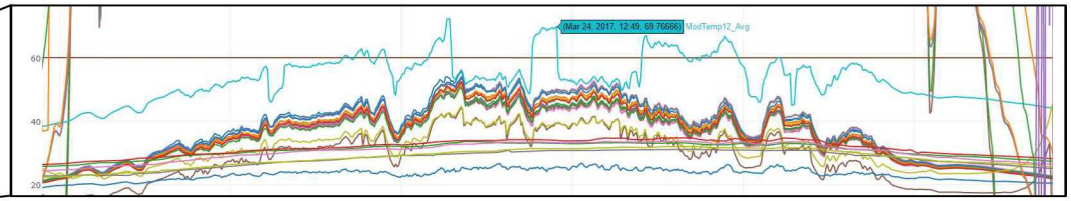
RTC Dashboard for 2017-03-24

	New Mexico	Florida	Vermont	Nevada
Weather	Irradiance 1.00 Wind 1.00 Air Pressure 1.00 Humidity 1.00 Rainfall 1.00 Datalogger 1.00 Detailed Report Interactive Plot	Irradiance 1.00 Wind 1.00 Air Pressure 1.00 Humidity 1.00 Rainfall 1.00 Datalogger 1.00 Detailed Report Interactive Plot	Irradiance 0.57 Wind 1.00 Air Pressure 1.00 Humidity 1.00 Rainfall 1.00 Datalogger 1.00 Detailed Report Interactive Plot	Irradiance 1.00 Wind 1.00 Air Pressure 1.00 Humidity 1.00 Rainfall 1.00 Datalogger 1.00 Detailed Report Interactive Plot
Baseline	Irradiance 1.00 Temperature 1.00 Current 1.00 Voltage 1.00 Power 0.64 Detailed Report Interactive Plot	Irradiance 1.00 Temperature 0.19 Current 1.00 Voltage 1.00 Power 0.93	Irradiance 1.00 Temperature 1.00 Current 0.27 Voltage 1.00 Power 0.24 Detailed Report Interactive Plot	Irradiance 1.00 Temperature 1.00 Current 1.00 Voltage 1.00 Power 1.00 Detailed Report Interactive Plot



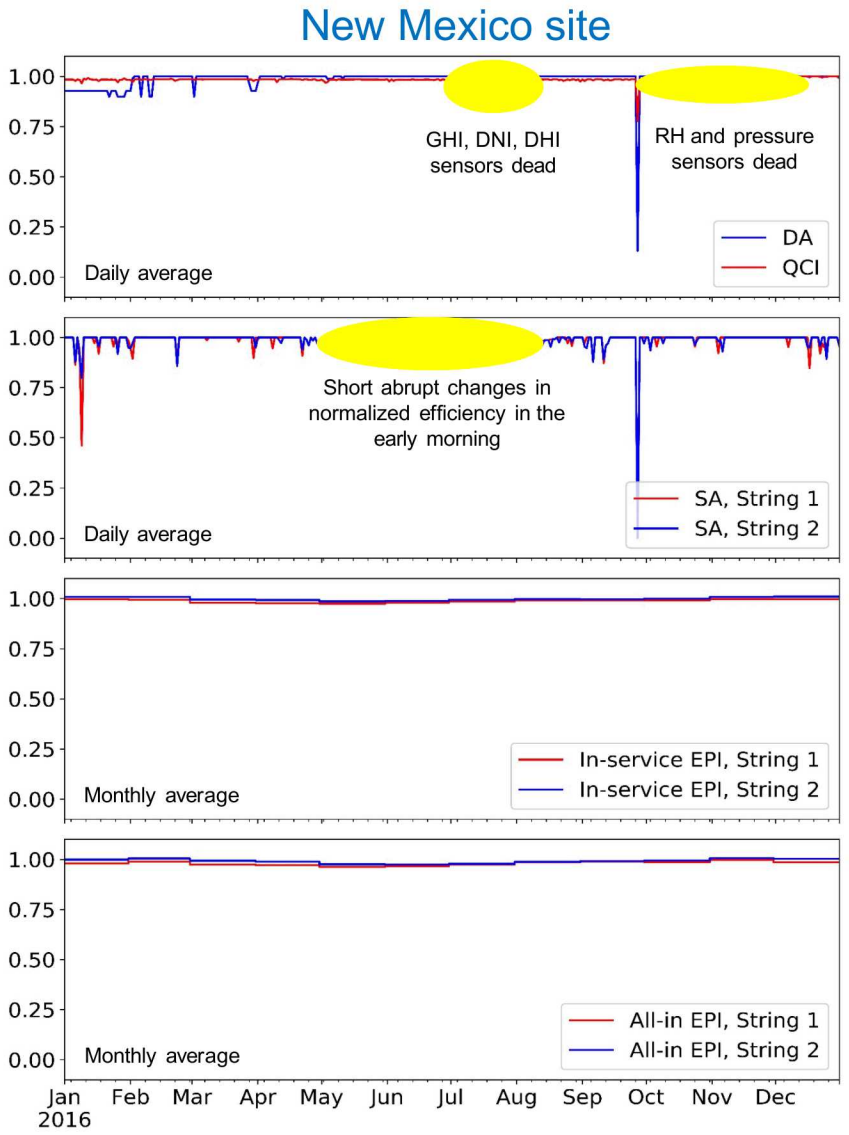
ModTemp12_Avg Temperature Deviation



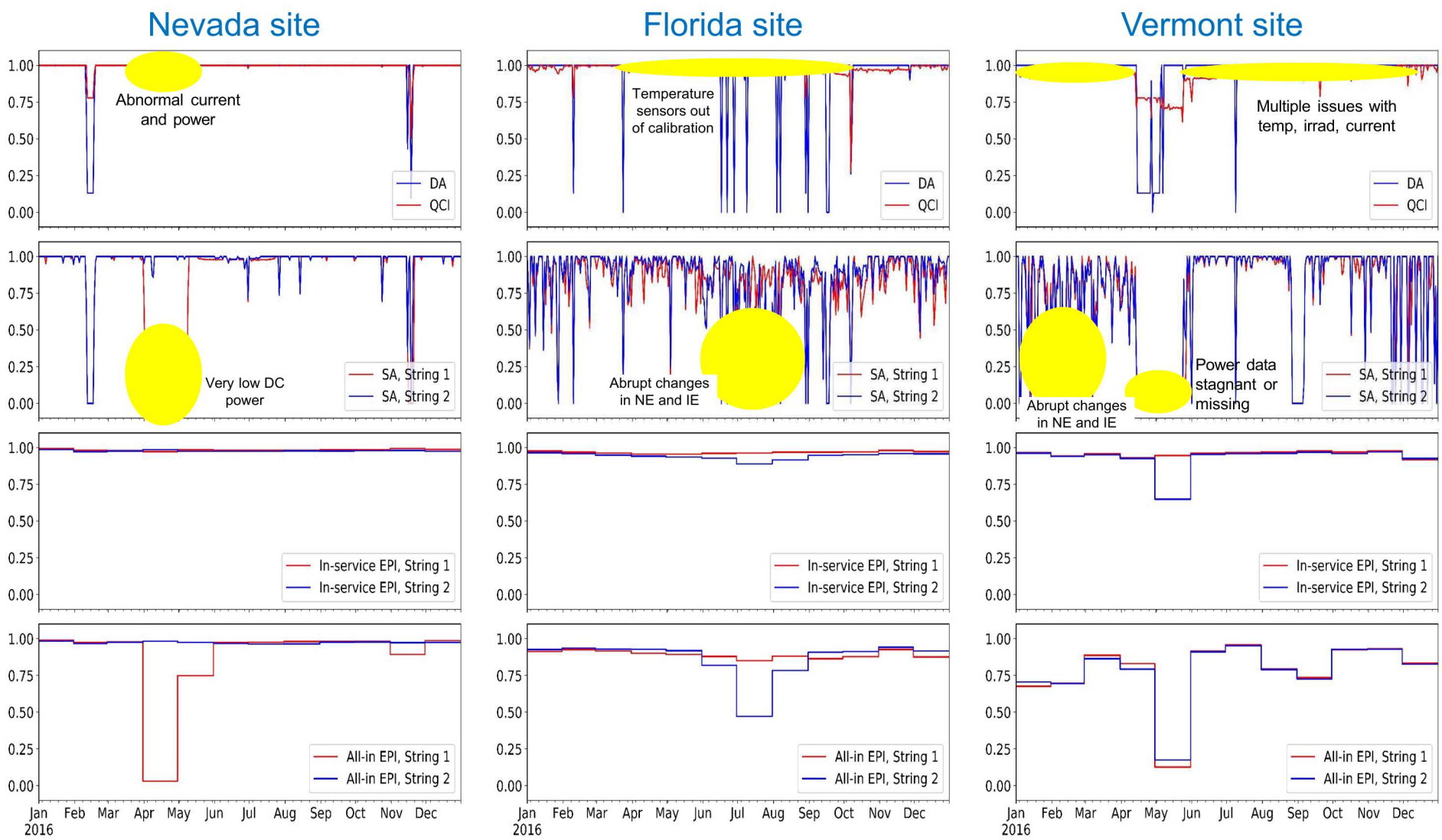


Metrics

- Data availability (DA): percent of expected data that was recorded
- Quality control index (QCI): percent of available data that passed all quality control tests
- System availability (SA): percent of data associated with power, inverter efficiency, normalized efficiency, and power performance index that passed all quality control tests
- Energy performance index (EPI): ratio of measured energy to expected energy (in-service and all-in)



Yearly Energy Evaluation

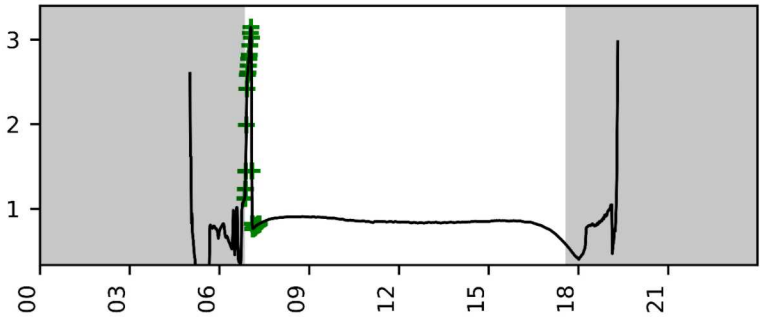


Detailed and Summary Information

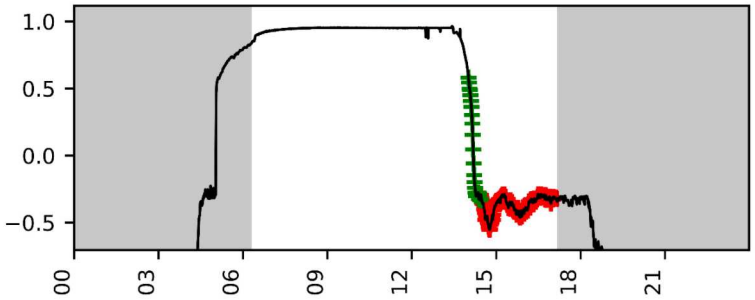
- Details on quality control issues
  - Datalogger/file transfer issues
  - Calibration issues
  - Unexpected shading
  - Dead sensors
  - Sensors that change erratically
  - Sensor drift
  - Underperforming inverters
- Summary metrics

	NM	NV	FL	VT
DA	99%	98%	96%	95%
QCI	98%	99%	98%	92%
SA, String 1	98%	86%	83%	72%
SA, String 2	98%	97%	84%	72%
In-service measured energy (kWh)	11517	10476	8693	5528
In-service expected energy (kWh)	11608	10679	9104	5802
All-in expected energy (kWh)	11696	11390	9911	7305
In-service EPI	99%	98%	95%	95%
All-in EPI	98%	92%	88%	76%

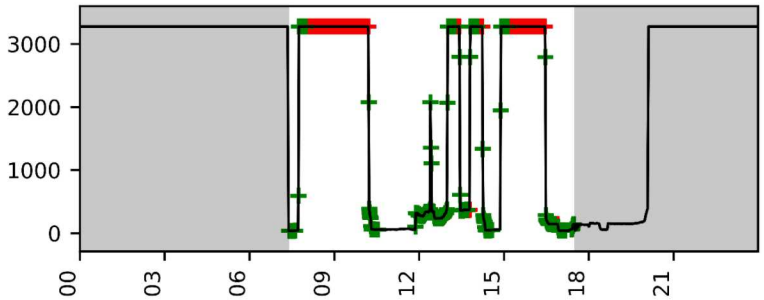
Normalized Efficiency, New Mexico site



Inverter Efficiency, Nevada site



Module Temperature, Florida site



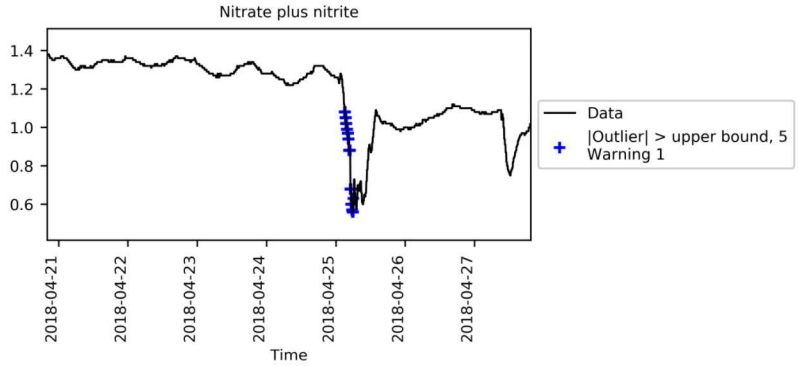
Pecos example for Village Blue

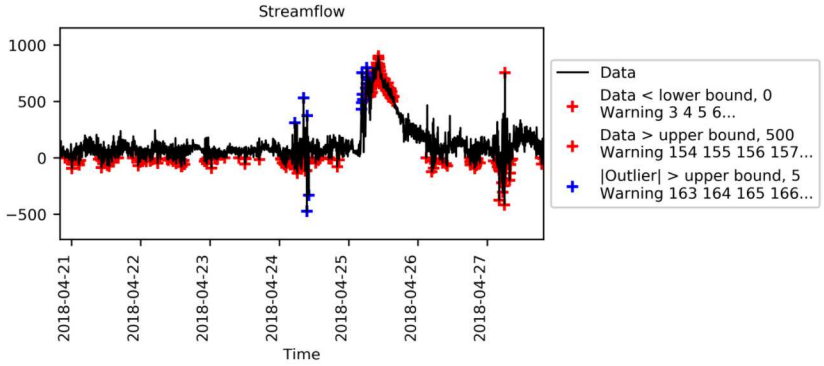
- Example uses previous 7 days of data at the Jones Falls site (Baltimore). Data retrieved from https://waterservices.usgs.gov
- Simple quality control tests include:
  - Check to make sure data is collected at least every 15 minutes
  - Check for missing data
  - Check for corrupt data (each variable is given a noDataValue in Currents)
  - Check to see if data is with the expected ranges →
  - Check for outliers (>5 standard deviations using a 2 day moving window)
- Results were gathered in an HTML report
- An interactive plot of the raw data was created,
- This analysis could be run on an automated schedule.

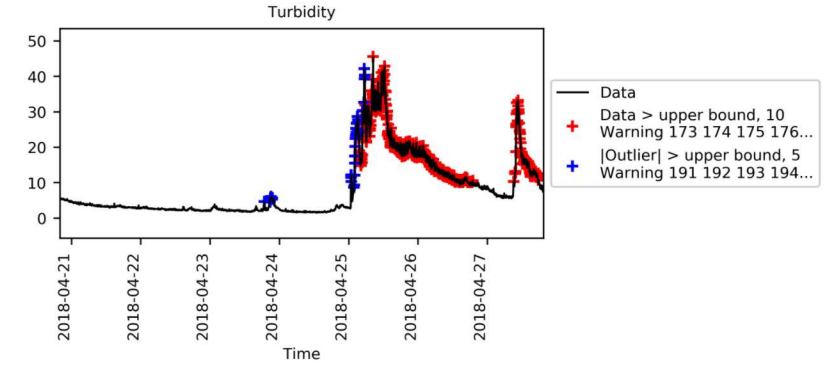
Dissolved oxygen (5 - 50 mg/L)
Gage height (-2 - 2 ft above reference)
Nitrate plus nitrite (0 - 10 mg/L)
Specific conductance (0 - 5000 microsiemens per cm at 25C)
Streamflow (0 - 500 ft3/s)
Temperature (0 - 21 degrees C)
Turbidity (0 - 10 FNU)
pH (6.5 - 9)

Pecos example for Village Blue

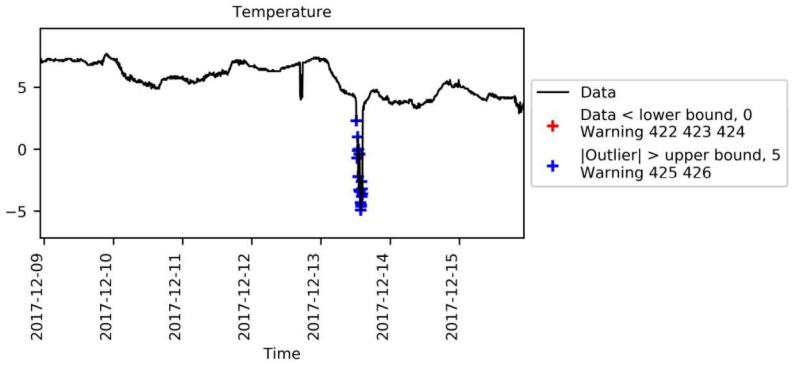
From analysis run April 27

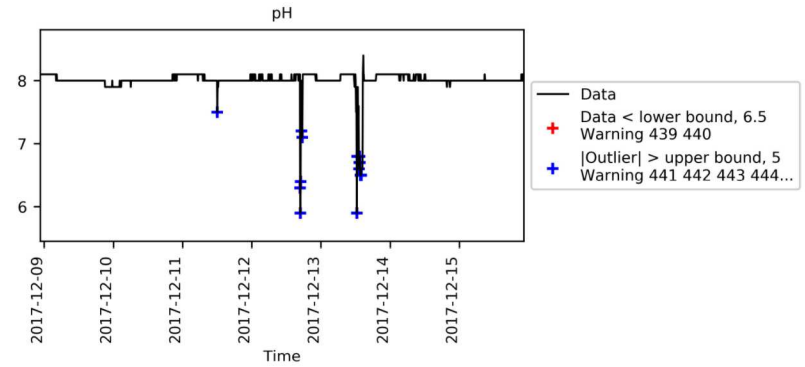






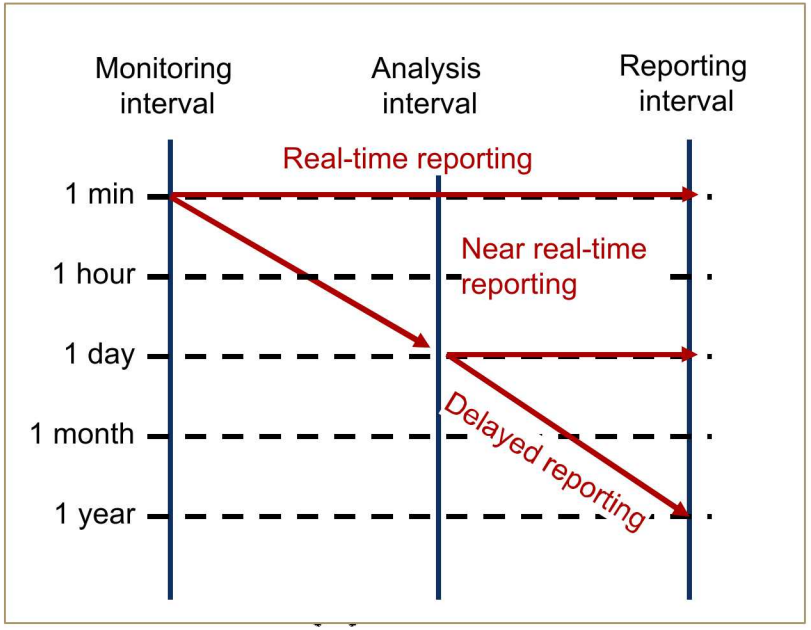
From analysis run December 15

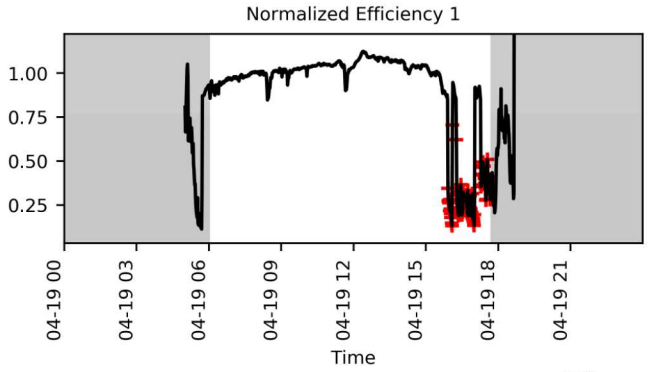




Considerations

- How much data are you dealing with?
- How often should you retrieve data for analysis?
- How often should you report results?
- What type of information do you want to gain from running quality control tests?
- How do you want to display and store results?
  - Simple to complex
  - Red/yellow/green approach
  - Time series or interactive graphics
  - Performance history
  - Dashboards hosted on the web
  - Email alerts





Next steps for Pecos

- Integrate Canary-like behavior
  - History window, remove outliers, event probability
  - Increment, linear filter, and multivariate nearest neighbor algorithms
- Interactive dashboards using Dash
  - Integrate real-time data visualization with results from quality control analysis, connected directly to the database
- Parallel analytics using Dask