ADVANCES IN USING MACHINE-LEARNING TOWARDS DREDGING-BEHAVIOR DETECTION

WEDA PACIFIC CHAPTER MEETING 2021

Seth Travis – RPS Group

29-Oct-2021

National Dredging Quality Management Program (USACE DQM)

- DQM's mission is to provide the USACE dredging manager with a nationally standardized, remote monitoring and documentation system.
- This system provides the Corps with timely data access, multiple reporting formats, full technical support including dredge certifications, data quality control, database management, and support for the DQM operating system.
- DQM maintains a web-based viewer and underlying database of operational data reported by dredge plants under their monitoring program
  ►Hoppers
  ►Scows
  ►Pipeline

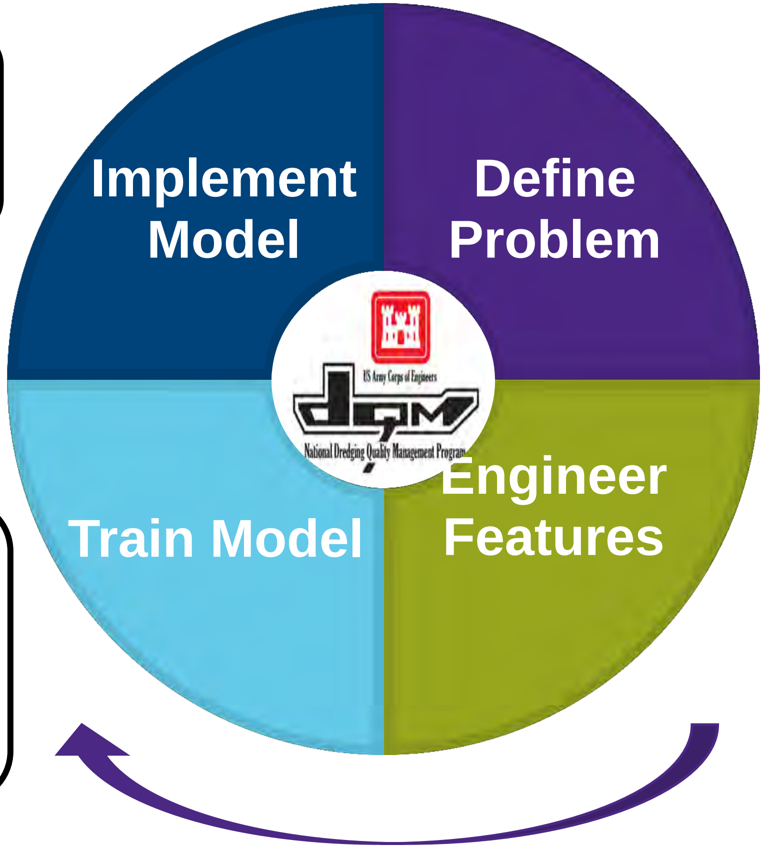
Advances in using machine-learning towards dredging-behavior detection

- National Artificial Intelligence Initiative Act of 2020 encourages implementation of AI and machine learning (ML) across federal agencies
- Dredging-activity of pipeline plants provided an excellent opportunity for DQM to implement ML methods
- Cutterhead and dustpan type pipelines have distinct behaviors, and need to be modeled separately

Advances in using machine-learning towards dredging-behavior detection

- Deploy the chosen model
- Evaluate model quality

- What does the solution look like?
- What metrics will be used to evaluate results?
- What data is available as inputs?



- Match problem with appropriate ML algorithm
- Validate model on unseen training sets
- Use metrics to evaluate model
- Optimize and debug

- Good data makes good results
- Generate labels / training sets
- Preprocess data
- Transform data into features

Adapted from: Google ML guides: https://developers.google.com/machine-learning/guides

Advances in using machine-learning towards dredging-behavior detection

Define Problem
10%

Engineer Features
70%

- **Construct Training Sets**
  - Collect raw data
  - Identify label and feature sources
  - Select sampling strategy
- **Transform Data**
  - Explore, clean, and preprocess data
  - Develop features to train model

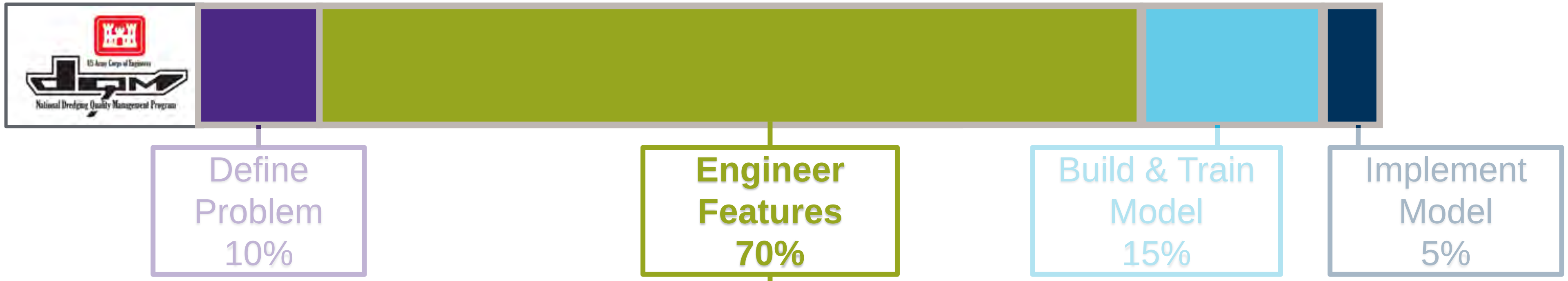
Build & Train Model
15%

- Choose ML algorithm
- Build model
- Train model
- Test and revise

Implement Model
5%

Adapted from: Google ML guides: https://developers.google.com/machine-learning/guides

Advances in using machine-learning towards dredging-behavior detection



Building Training Sets – Collect raw data:

- DQM began receiving pipeline data in 2019
- RPS assembled datasets and assigned idealized “dredging” activity
  - Cutterhead Data - 23 Plants, ~123 Days of Data
  - Dustpan Data - 4 Plants, ~45 Days of Data

Advances in using machine-learning towards dredging-behavior detection

Define Problem 10%	**Engineer Features 70%**	Build & Train Model 15%	Implement Model 5%

Building Training Sets – Develop Features:

- Include variables that are most indicative of dredging activity
    - Raw data - Pump RPM, Slurry Density, Slurry Velocity, Cutterhead/Dustpan Depth, Vacuum and Outlet PSI, Vessel Heading and Speed
    - Calculations from raw data - running mean, standard deviation

Advances in using machine-learning towards dredging-behavior detection

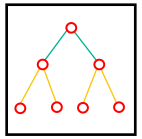
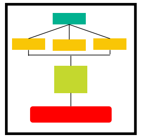
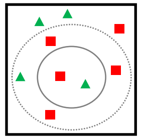
Define Problem 10%

Engineer Features 70%

Build & Train Model 15%

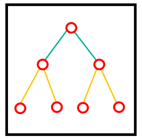
Implement Model 5%

Choose Machine Learning Algorithm

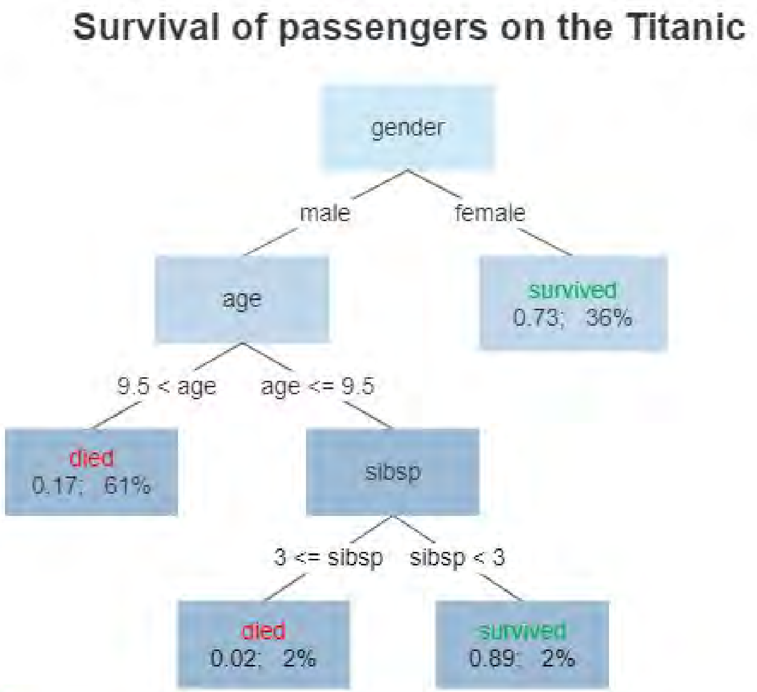
Decision Tree		Predictive model that splits data based on different classification rules
Ensemble Methods		Combines several base models and uses a voting logic in order to product one optimal model
K Nearest Neighbors		Classifies data points based on similarity measure and majority vote to its neighbors

Choose Machine Learning Algorithm

Decision Tree



Predictive model that splits data based on different classification rules



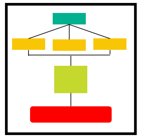
Decision Tree Methods:

- Checks each variable for optimal split point
- Decision is chosen as that with best split
  - Subsequent decisions are limited to data which falls into original decision
- Provides easy-to-understand decisions

Image taken from: https://en.wikipedia.org/wiki/Decision_tree_learning

Choose Machine Learning Algorithm

Ensemble Methods



Combines several base models and uses a voting logic in order to product one optimal model

Boosting Ensemble

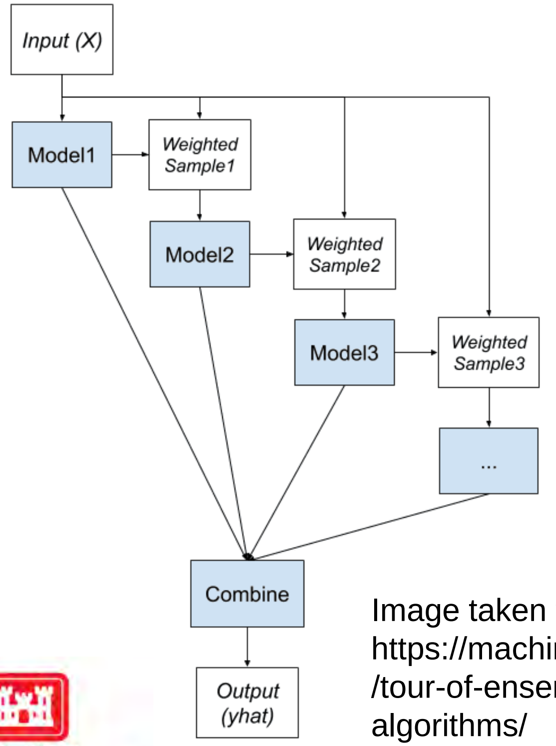


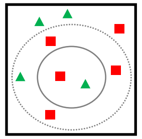
Image taken from: https://machinelearningmastery.com/tour-of-ensemble-learning-algorithms/

Ensemble Methods:

- Use multiple decision trees
  - Build an original decision tree
  - Weight new decision trees based upon what was misclassified
  - Ending model uses multiple weighted decision trees for final model build

Choose Machine Learning Algorithm

K Nearest Neighbors



Classifies data points based on similarity measure and majority vote to its neighbors

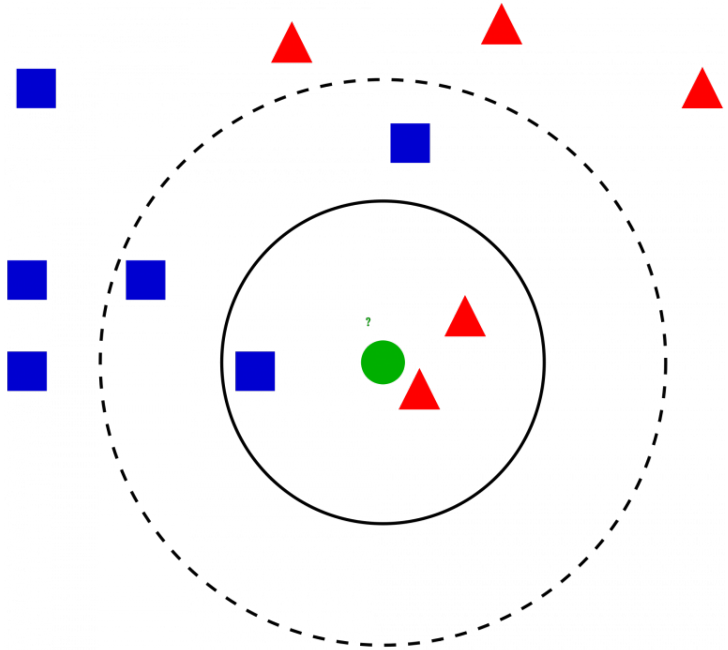


Image taken from: https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/

K-Nearest Neighbor Methods:

- At any given point, check what is the state of the “nearest” neighbors
- The most common state of the neighbors is assigned to the point
- K-points refers to the number (k) of neighbors to check
- Computationally intensive

Advances in using machine-learning towards dredging-behavior detection

Define Problem 10%	Engineer Features 70%	Build & Train Model 15%	Implement Model 5%

Choose Machine Learning Algorithm

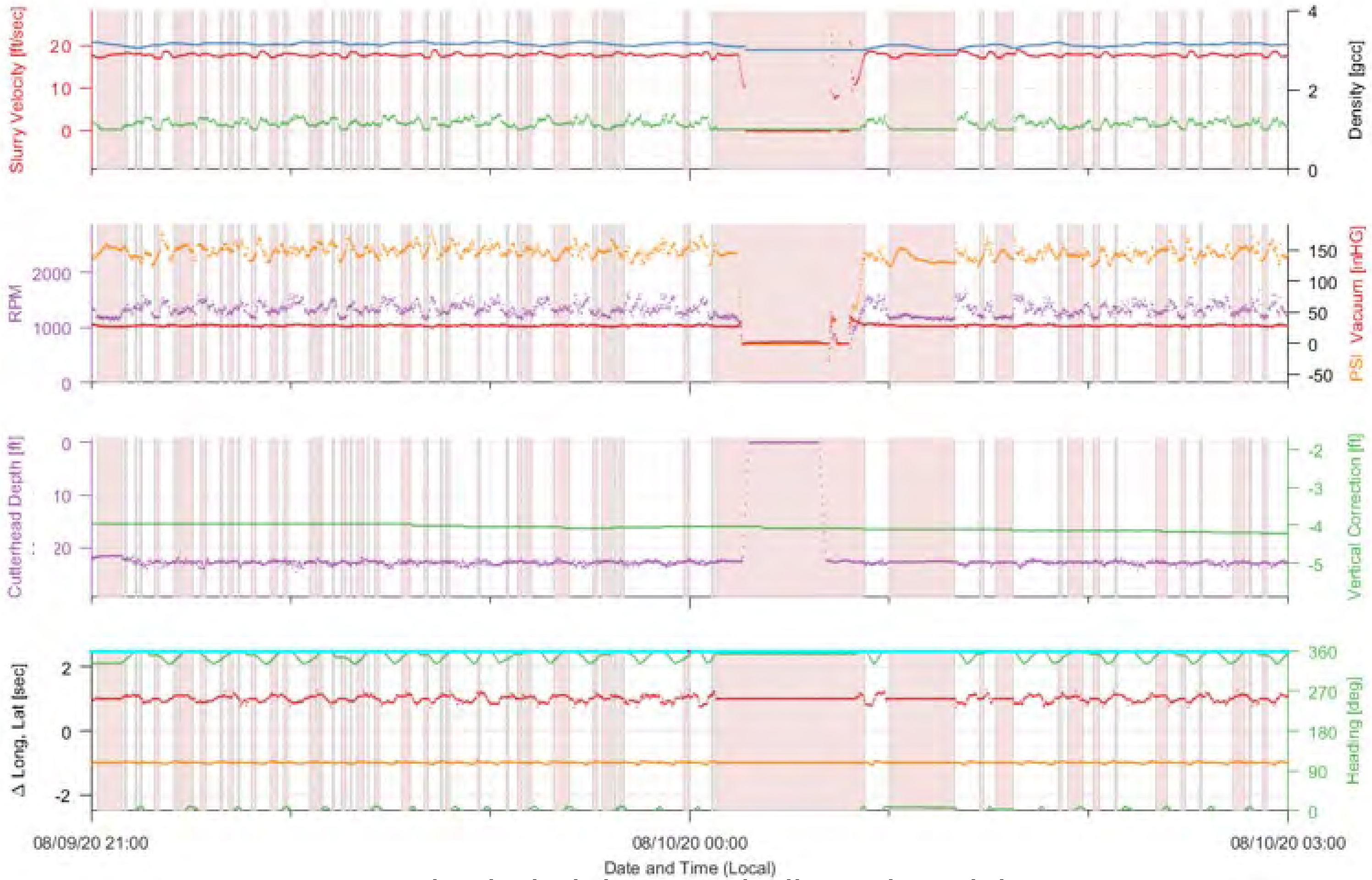
Decision Tree	Predictive model that splits data based on different classification rules
Ensemble Methods	Combines several base models and uses a voting logic in order to product one optimal model
K Nearest Neighbors	Classifies data points based on similarity measure and majority vote to its neighbors

For our use, Ensemble and KNN are most accurate

Ensemble is fastest to process (4-5x faster than KNN)

Evaluate Results

- Initial ML Method
- Raw Input Data

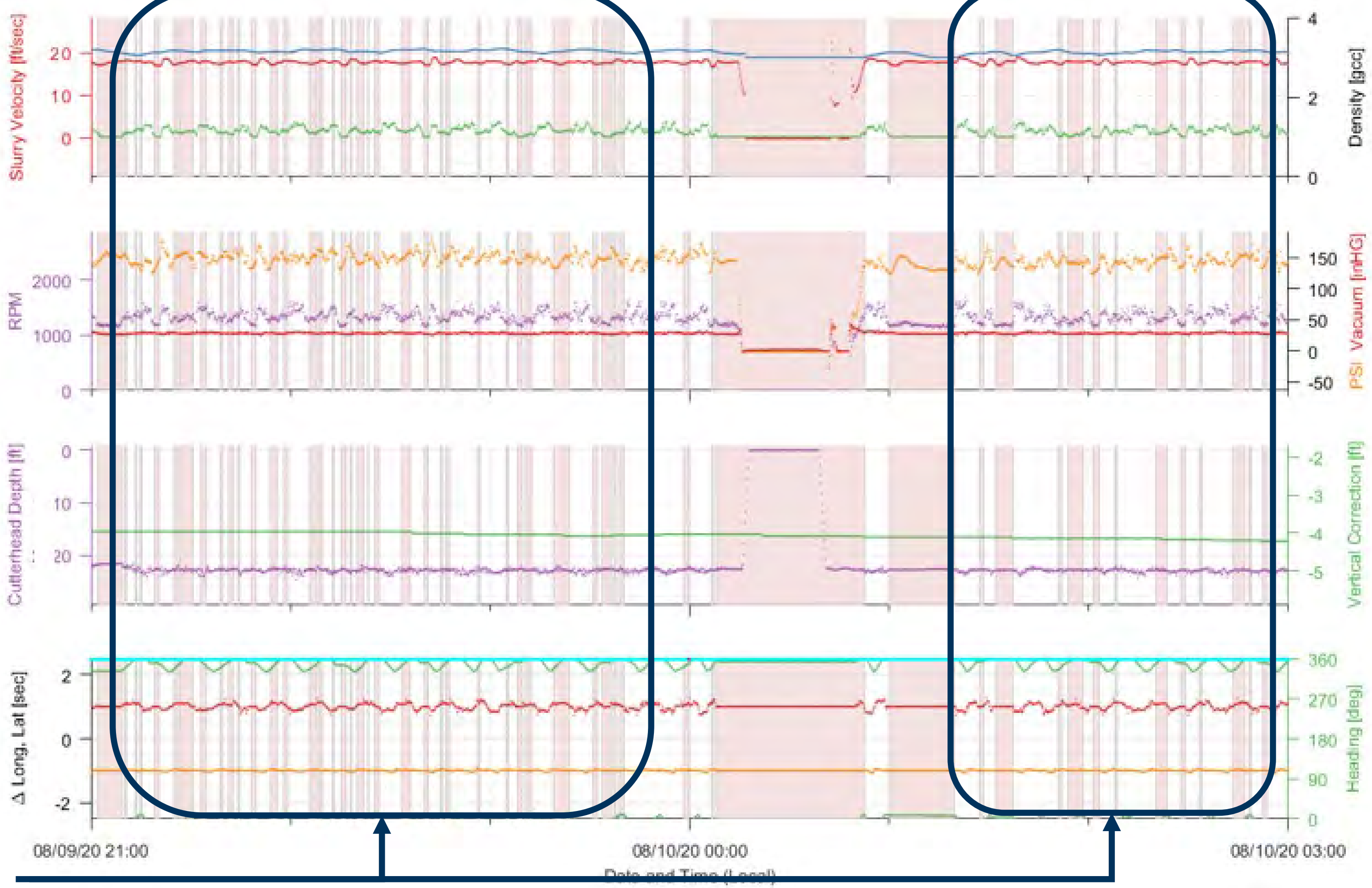


Note: Shaded pink areas indicate inactivity

Advances in using machine-learning towards dredging-behavior detection

Evaluate Results

- Initial ML Method
- Raw Input Data
- Identifies moderate activity, but also misses large periods



Likely
mis-classified

Note: Shaded pink areas indicate inactivity

US Army Corps of Engineers
National Dredging Quality Management Program

Advances in using machine-learning towards dredging-behavior detection

Define Problem 10%

Engineer Features 70%

Build & Train Model 15%

Implement Model 5%

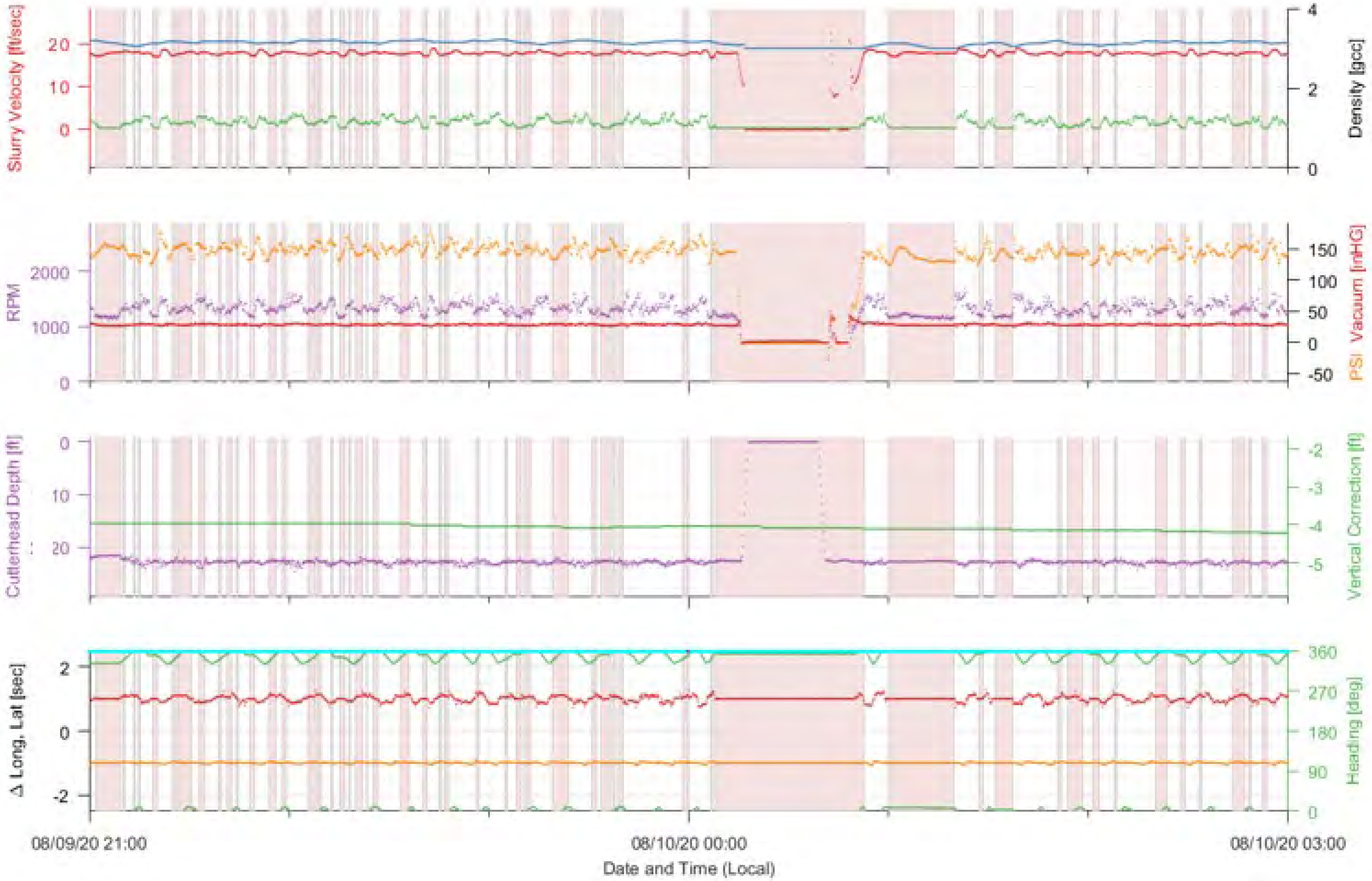
Evaluate and Revise – Develop Features:

- Expected behaviors and data limits vary by dredge plant
  - Example: max operating depth ranges from ~20 ft to ~60 ft
- By normalizing data by max operating value, we can use a single set of criteria to identify if signal is “on” or “off” across many plants
- New features are created by **normalizing or adjusting** important signals

Advances in using machine-learning towards dredging-behavior detection

Evaluate Results

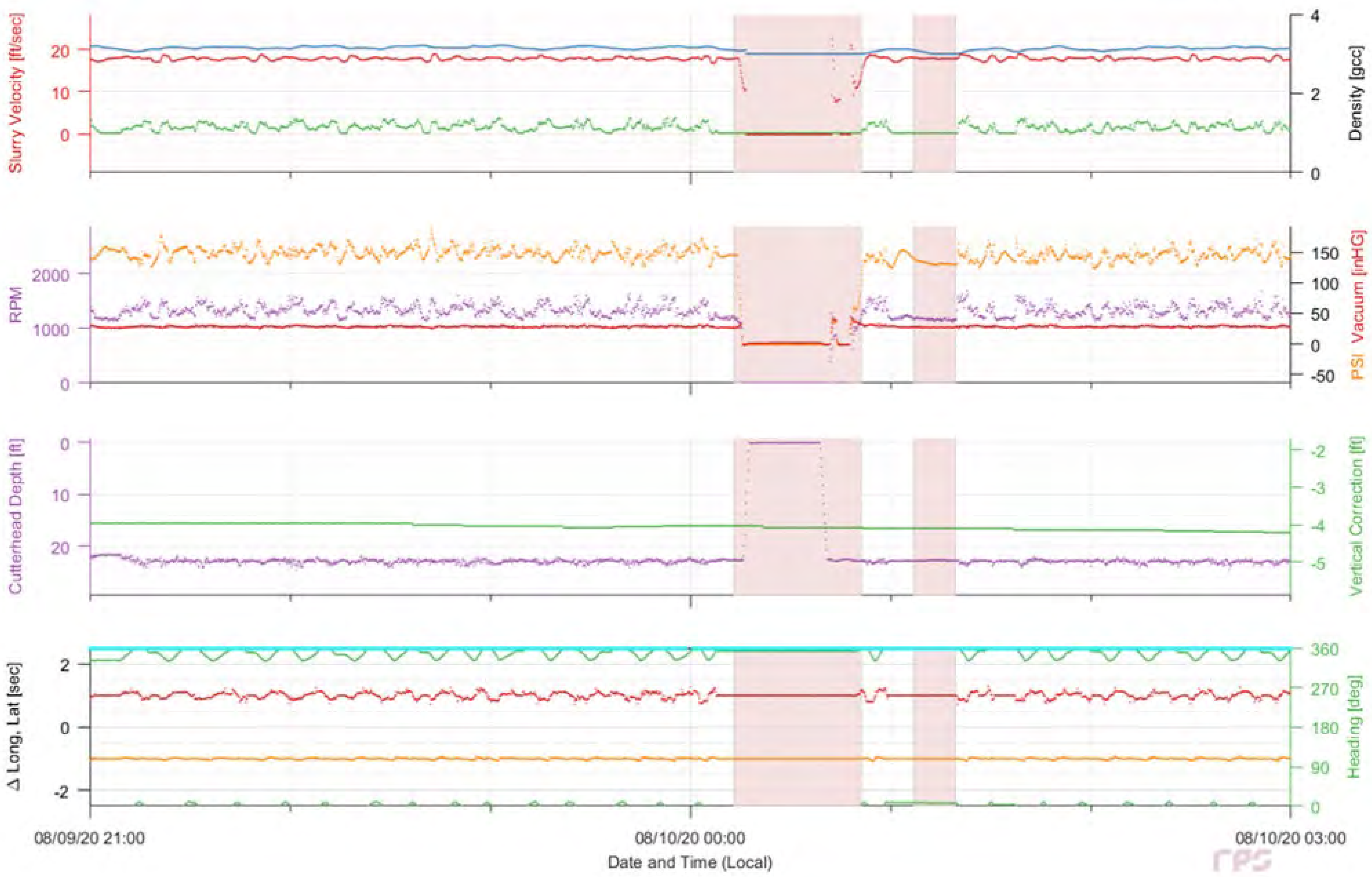
- Initial ML Method
- Raw Input Data



Note: Shaded pink areas indicate inactivity

Evaluate Results

- Updated ML Method
- Adjusted/Normalized Input Data
- Assignments are much improved



Note: Shaded pink areas indicate inactivity

Advances in using machine-learning towards dredging-behavior detection

Define Problem 10%

Engineer Features 70%

Build & Train Model 15%

Implement Model 5%

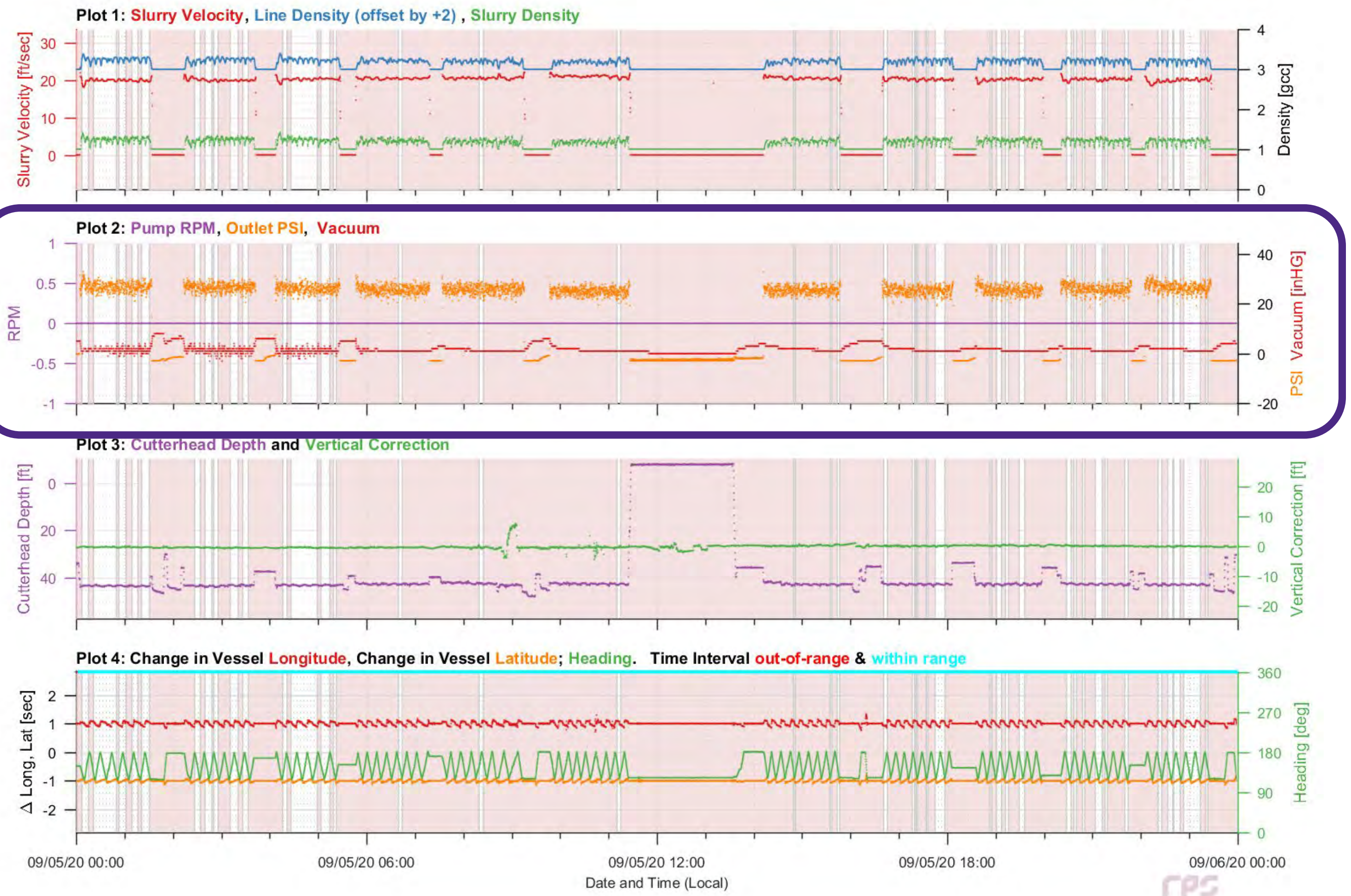
Evaluate and Revise – Develop Features:

- Bad sensor data can impact results
    - ML model assumes all input data is valid – bad inputs yield poor model outputs
- Some features are weighted higher in ML model (more significant) than others
- Develop **feature verification** process to reduce impact of bad data
    - Use same ML model but alter input value(s)
    - Only apply feature verification to most significant variables
    - Only apply feature verification if most other features are valid

Advances in using machine-learning towards dredging-behavior detection

Evaluate Results

- No Feature Verification
- Pump RPM reporting 0 throughout dataset
- Some areas identified, but majority of activity is missed

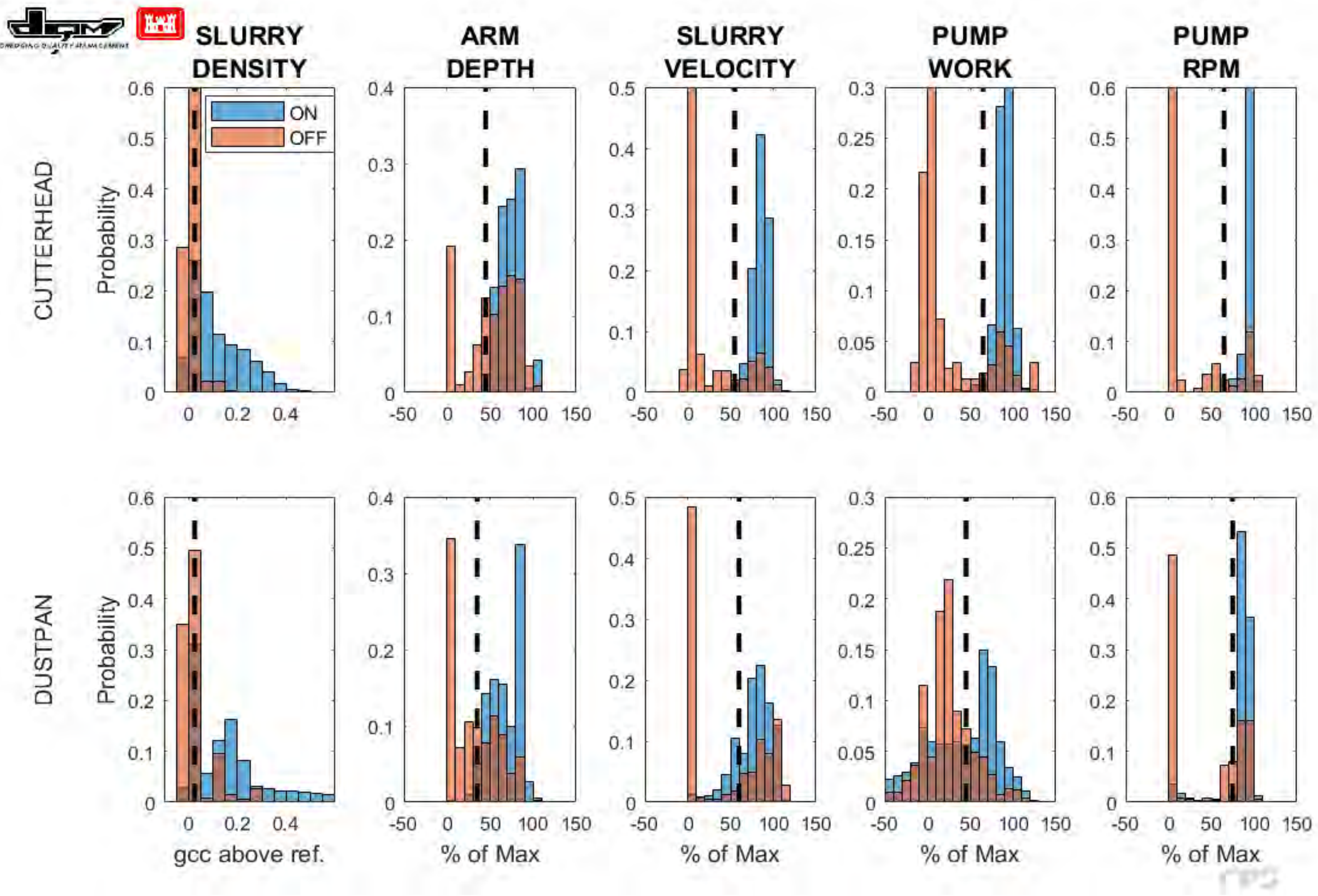


Note: Shaded pink areas indicate inactivity

Advances in using machine-learning towards dredging-behavior detection

Feature Verification

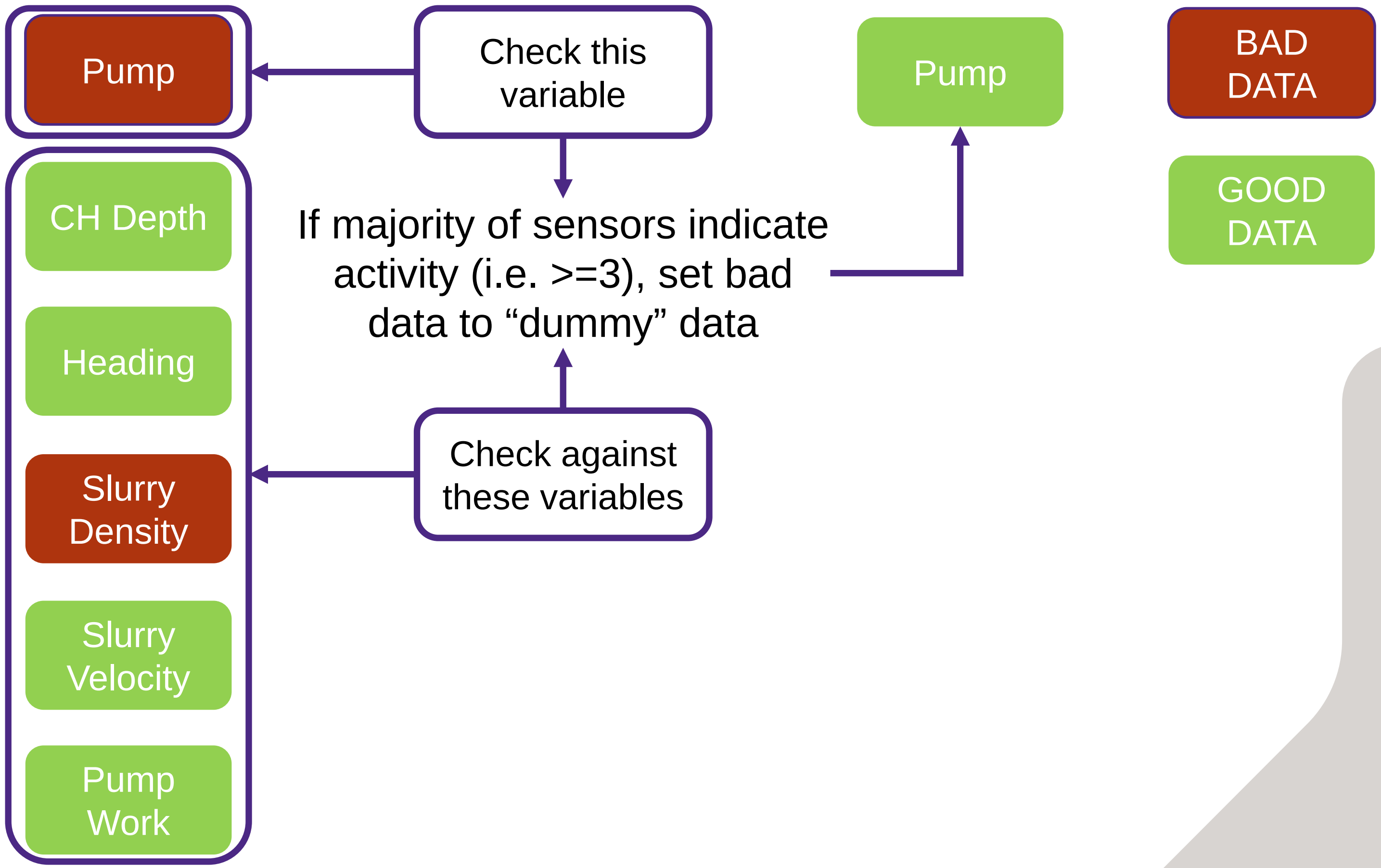
- Find optimal splits for each variable
- Use these splits as thresholds for determining if a variable is “likely good”



Advances in using machine-learning towards dredging-behavior detection

Feature Verification

- Example: Cutterhead plant with bad pump data
  - PUMP_RPM_NORM registers as 0 throughout dataset



Advances in using machine-learning towards dredging-behavior detection

Define Problem 10%

Engineer Features 70%

Build & Train Model 15%

Implement Model 5%

Evaluate and Revise – Feature Verification:

- Can have significant impact when there is questionable data
- By using high threshold criteria, we attempt to only “correct” the worst data
- Limitations:
  - Not intended to work with multiple bad inputs
  - Some variables still are more important to get right than others

Advances in using machine-learning towards dredging-behavior detection

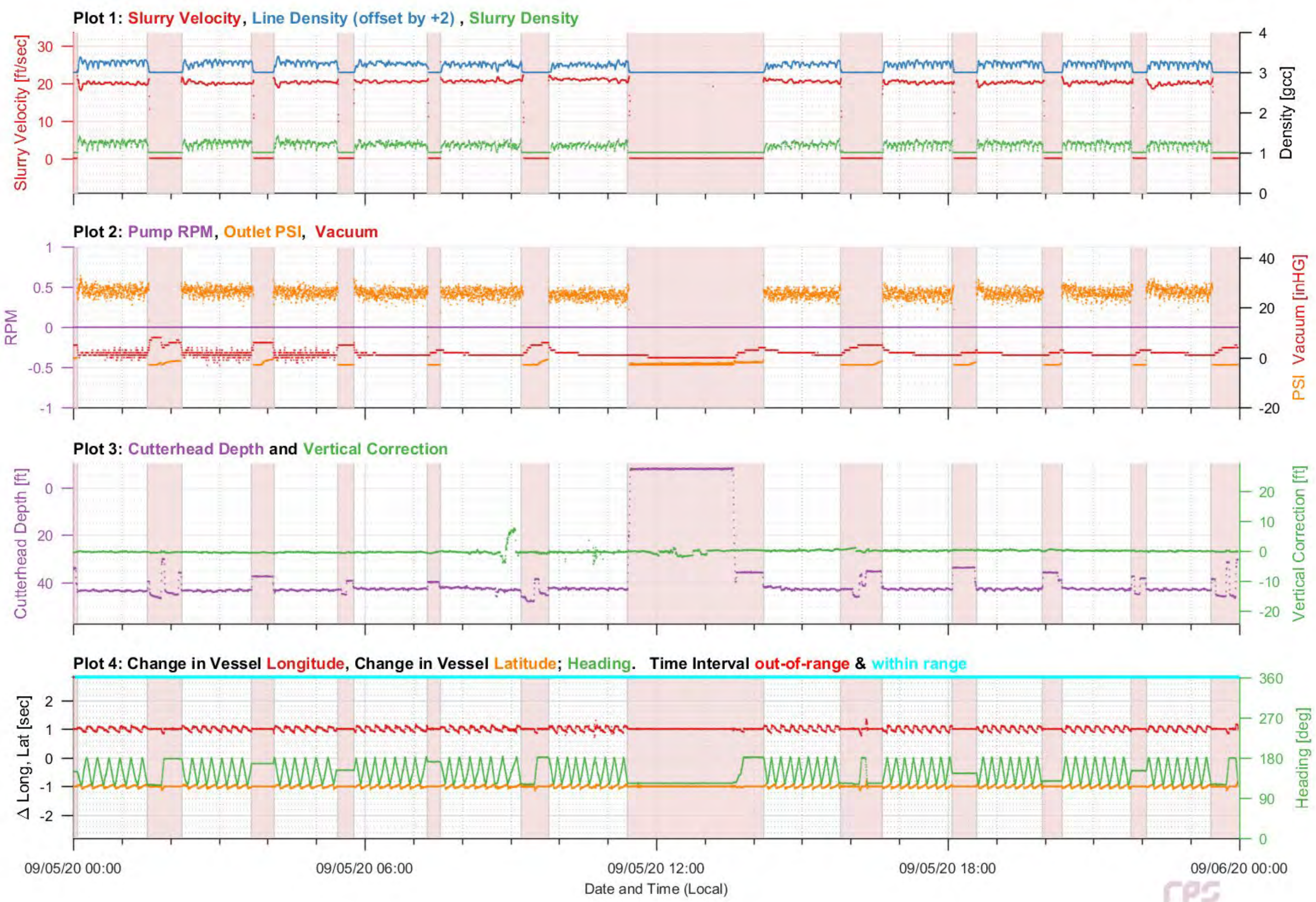
Evaluate Results

- No Feature Verification
- Pump RPM reporting 0 throughout dataset
- Some areas identified, but majority of activity is missed

Note: Shaded pink areas indicate inactivity

Evaluate Results

- With Feature Verification
- Pump RPM reporting 0 throughout dataset
- Majority of activity is identified



Note: Shaded pink areas indicate inactivity

Advances in using machine-learning towards dredging-behavior detection

Define Problem	Engineer Features	Build & Train Model	Implement Model
10%	70%	15%	5%

Summary

- Building training sets for data is the most significant effort in developing machine-learning methods
- Uniform data:
  - Normalizing data allows a single set of criteria to be used for many plants
- High-quality data:
  - Active adjustment of the worst quality data can significantly improve results

Advances in using machine-learning towards dredging-behavior detection

Define Problem 10%

Engineer Features 70%

Build & Train Model 15%

Implement Model 5%

Summary

- Machine learning methods have been more effective at assigning dredging/not dredging states than previous analytical methods
  - Previous methods accuracy:
    - 84.4% for cutterhead, 74.7% for dustpan
  - Machine learning method accuracy:
    - 98.7% for cutterhead, 98.0% for dustpan

Comments/Questions?

Seth Travis: seth.travis@rpsgroup.com