Northeastern
Civil and Environmental Engineering

Machine Learning:
How It Can Support Innovation In WWT/WRR?

Can It Be Trusted???

NEWEA Annual Meeting 2023

Amy Mueller, Northeastern University

a.mueller@northeastern.edu

Agenda for the talk

- Machine learning – what is it & what is it good for?
- Proof of concept – ML applied to a WRR challenge
- How can we work together as a community to take advantage of ML?

"Machine Learning" – all the rage!

- Machine learning – simplest definition
  - Extracting information on patterns **from data**
  - Contrasts with physics-based modeling approaches (known equations)
  - ≠ artificial intelligence (AI) – but can be used together

- Compelling capabilities
  - Able to learn complex non-linear relationships
  - Even
    (i) when we don't know the equation for the actual physical relationship
    (ii) when the equations are known but too complex to viably model
  - Can simultaneously learn to predict multiple target parameters

Great! What do we need to get started?

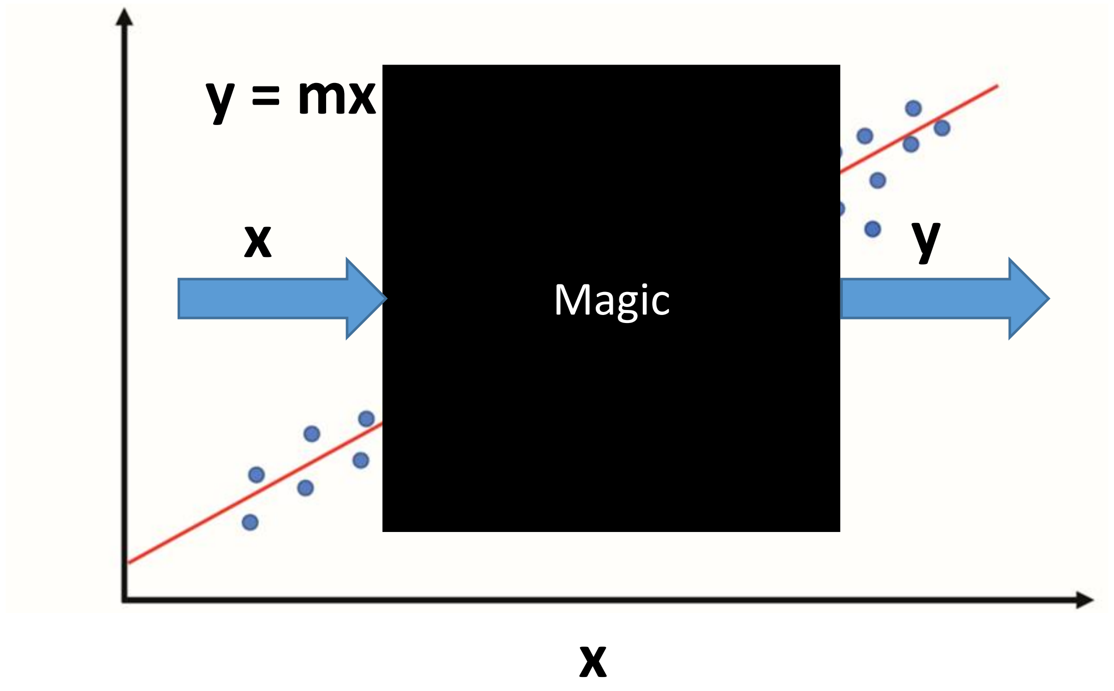
- A clear statement of the goal
  - Monitoring vs. controls?
  - Online vs. retrospective?

i.e., "What information can we use as model inputs? Do we need model inputs **in real time** (sensors)?"

- Some idea about the relationship between different signals

- Data – usually lots of data
  - For full range of conditions you need to model
  - **INCLUDING** (ideally) conditions "outside target norm"

Contrast to "traditional" models



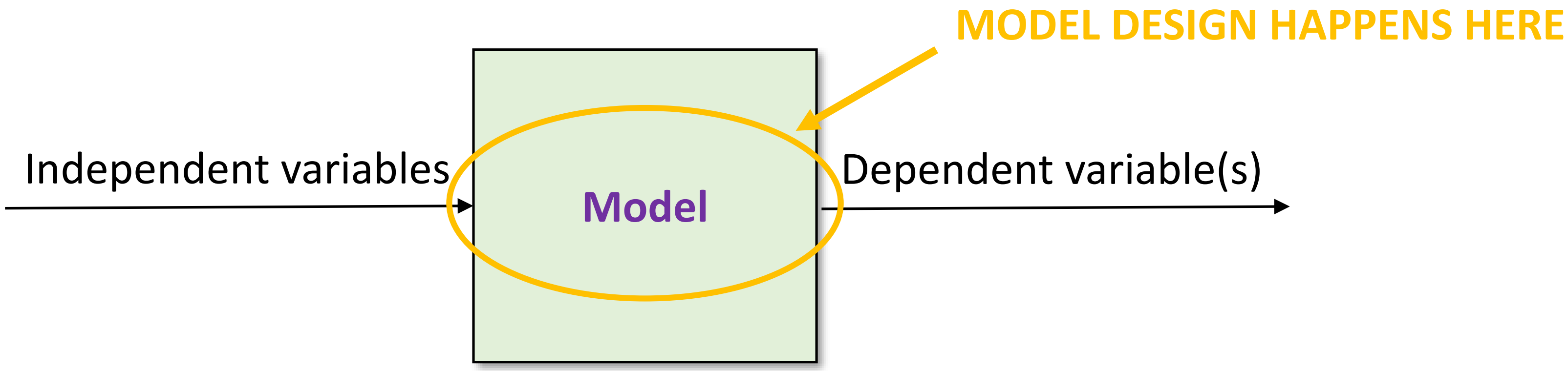
In ML system

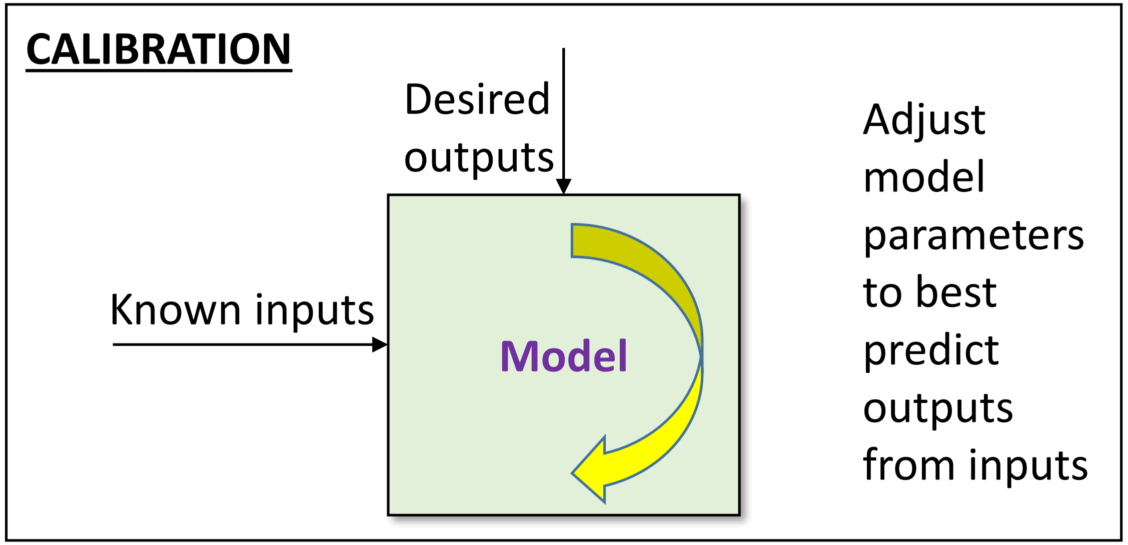
- NOT obvious where we shift from *interpolation* to *extrapolation*

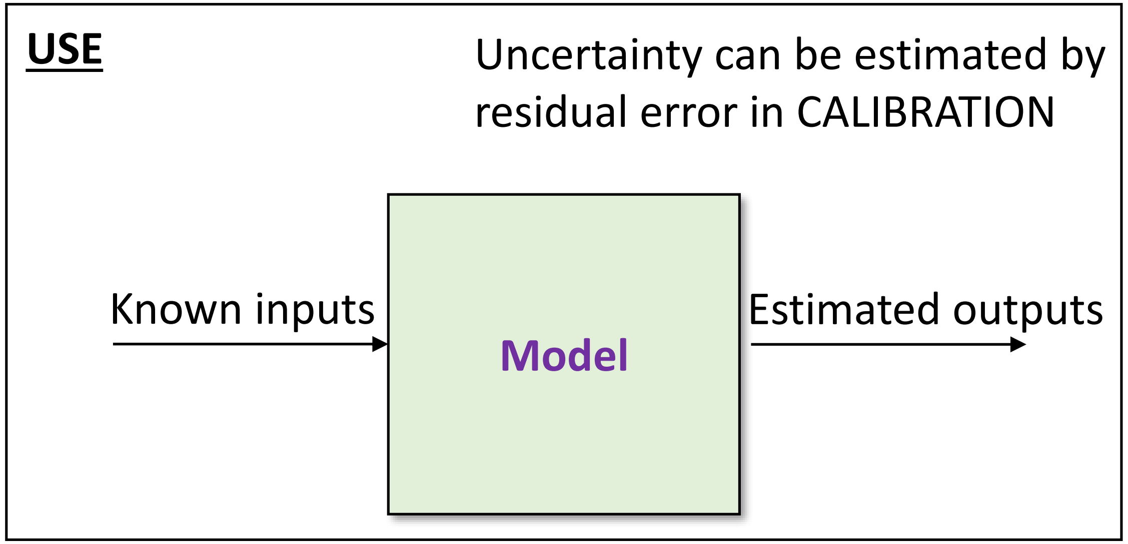
And in WW systems

- We usually don't want to push anything to the breaking point

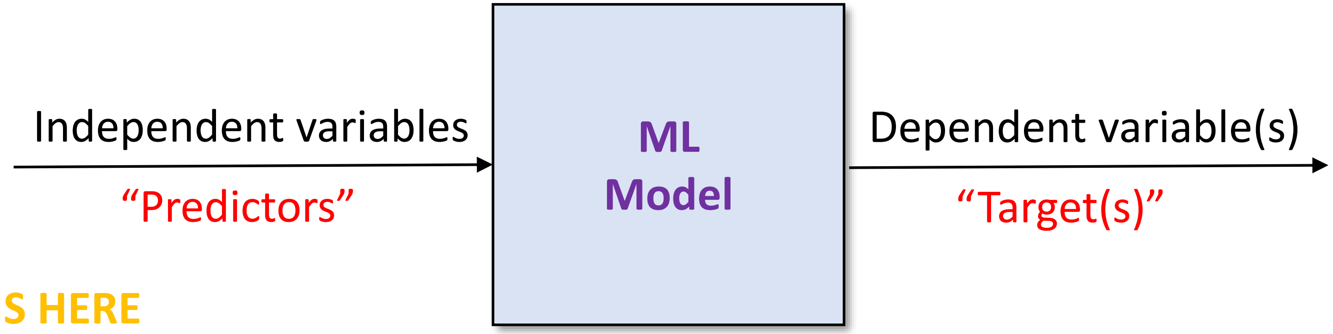
Vocab: what is a "model"?



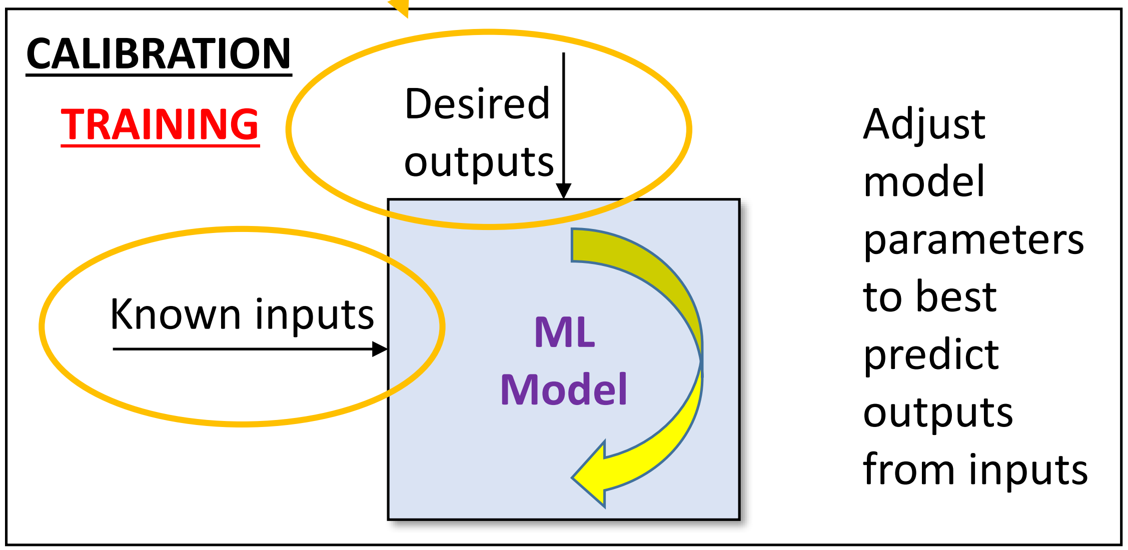


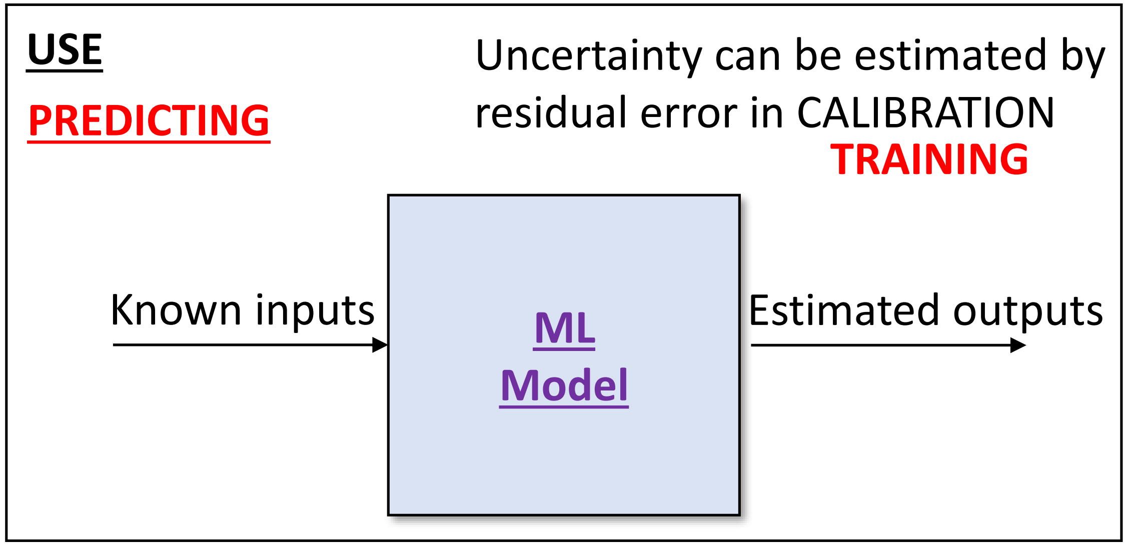


Vocab: what is an “ML model”?



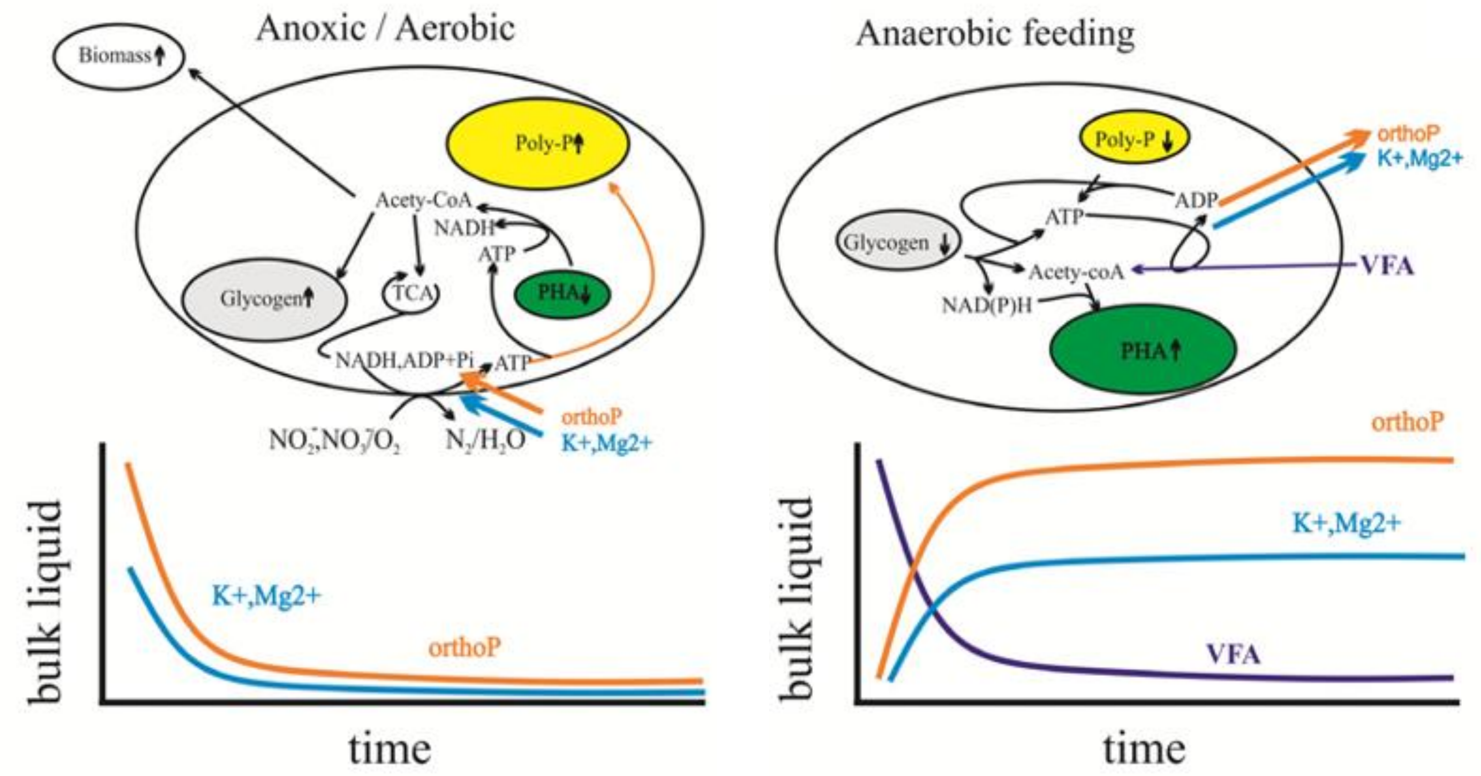
MODEL DESIGN HAPPENS HERE





Proof of concept… ML for EBPR controls

EBPR Recap



Co-transport relationship in PAOs

Microorganisms (**PAOs**) for **E**nhanced **B**iological **P R**emoval (**EBPR**) process:

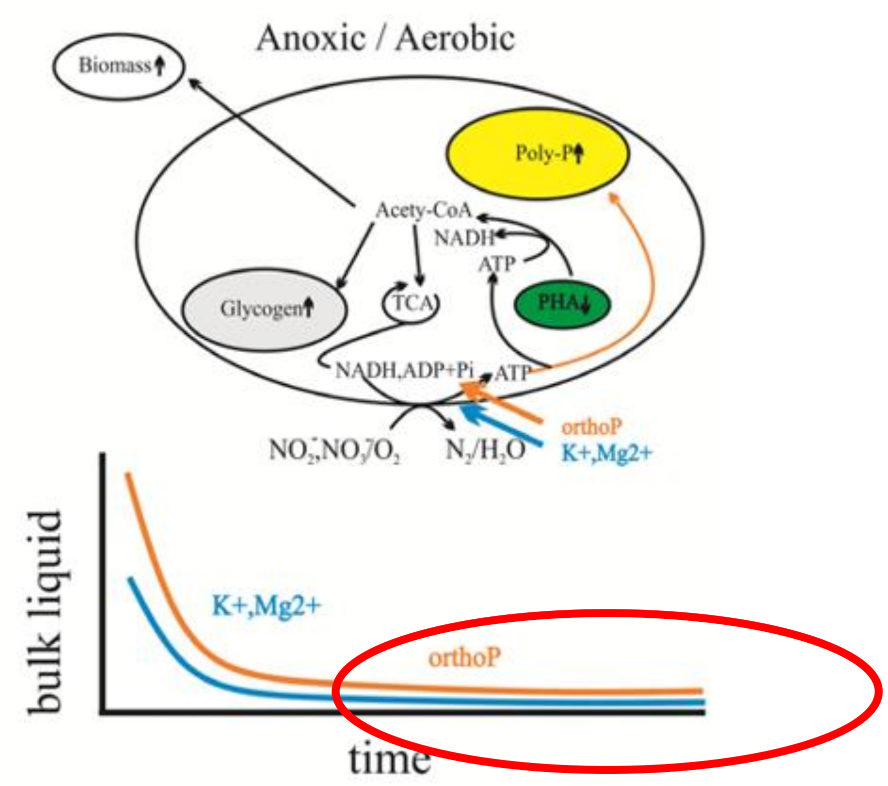
- anaerobic phase – release P, K, Mg
- aerobic phase – uptake P, K, Mg

Recovery target

Magnesium ammonium phosphorus ($MgNH_4PO_4$ = struvite)

Figure credit Winker lab

EBPR Challenge – Detect Removal Endpoint



In SBR

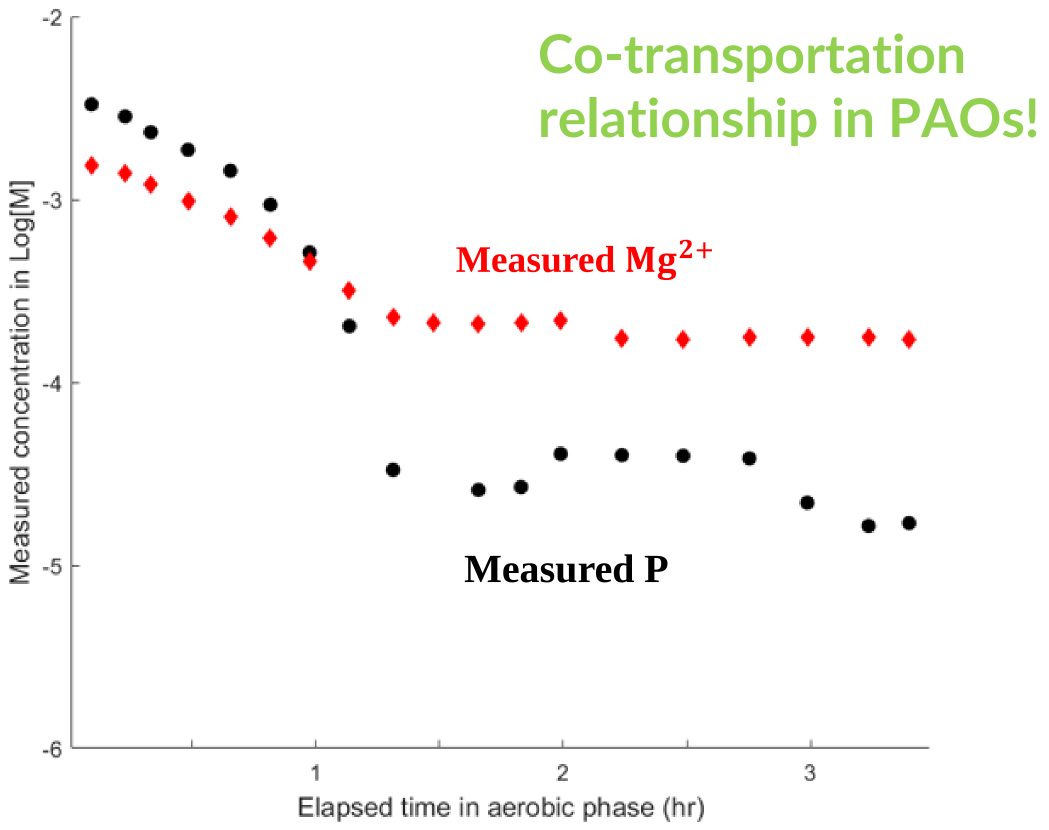
- Aerating (= $) to promote uptake of P by microbes
- Useful to know **in real time** when P concentrations are "sufficiently low" to start next batch

Challenge:

Lack of reasonable, affordable instrumentation for P!

Figure credit Winker lab

Proposal – Use what we know!



Sensors are available!

- K^+ and Mg^{2+} ISEs

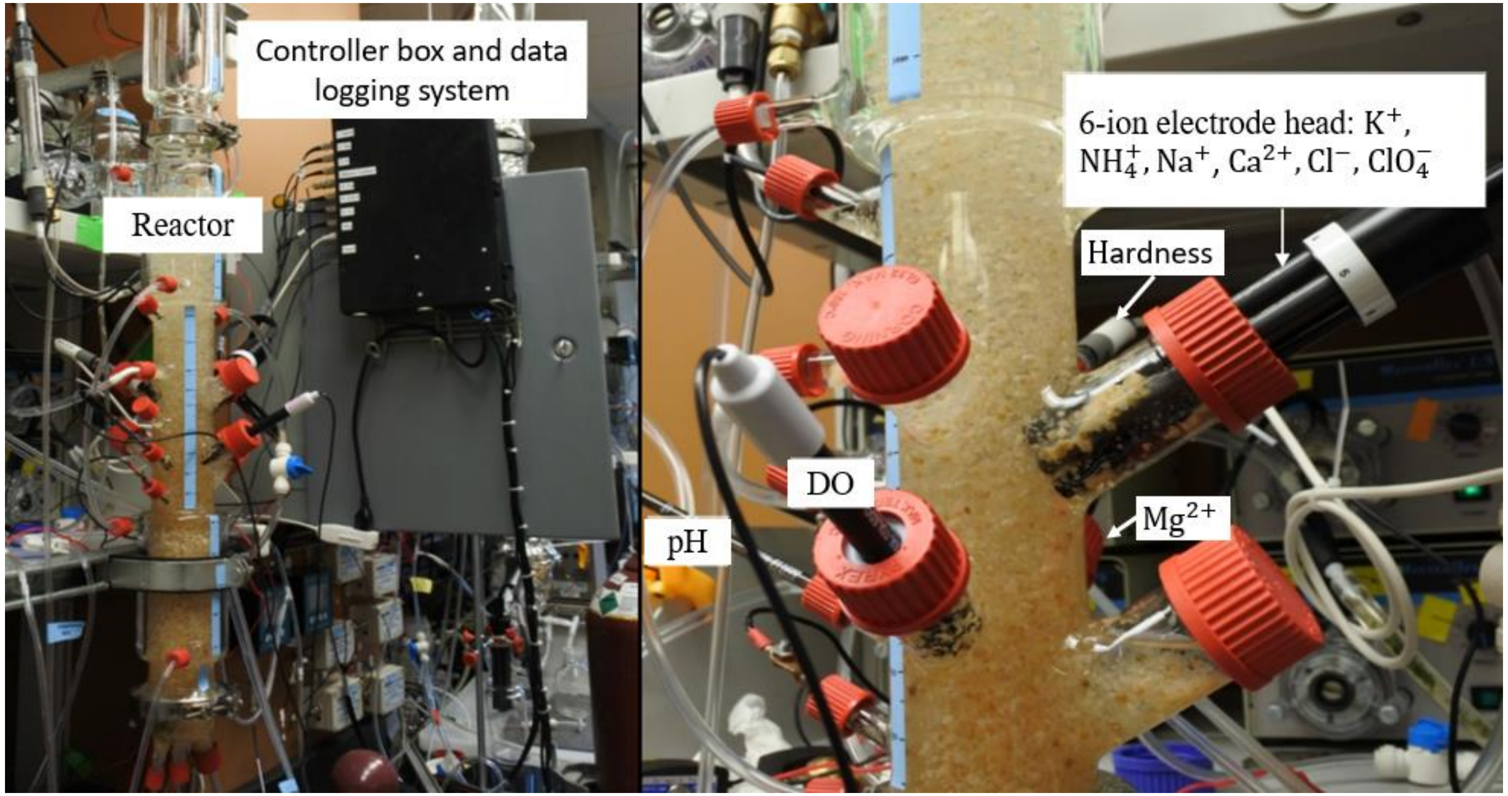
Because we know ISEs are imperfect

- Ca^{2+}, hardness ISEs (also sense Mg^{2+})
- Na^+, NH_4^+ ISEs (also sense K^+)

ML is a good option here because

- Co-transport relationship is complex & non-linear
- Physics of these sensors is complex & non-linear

Need DATA – lab scale pilot system



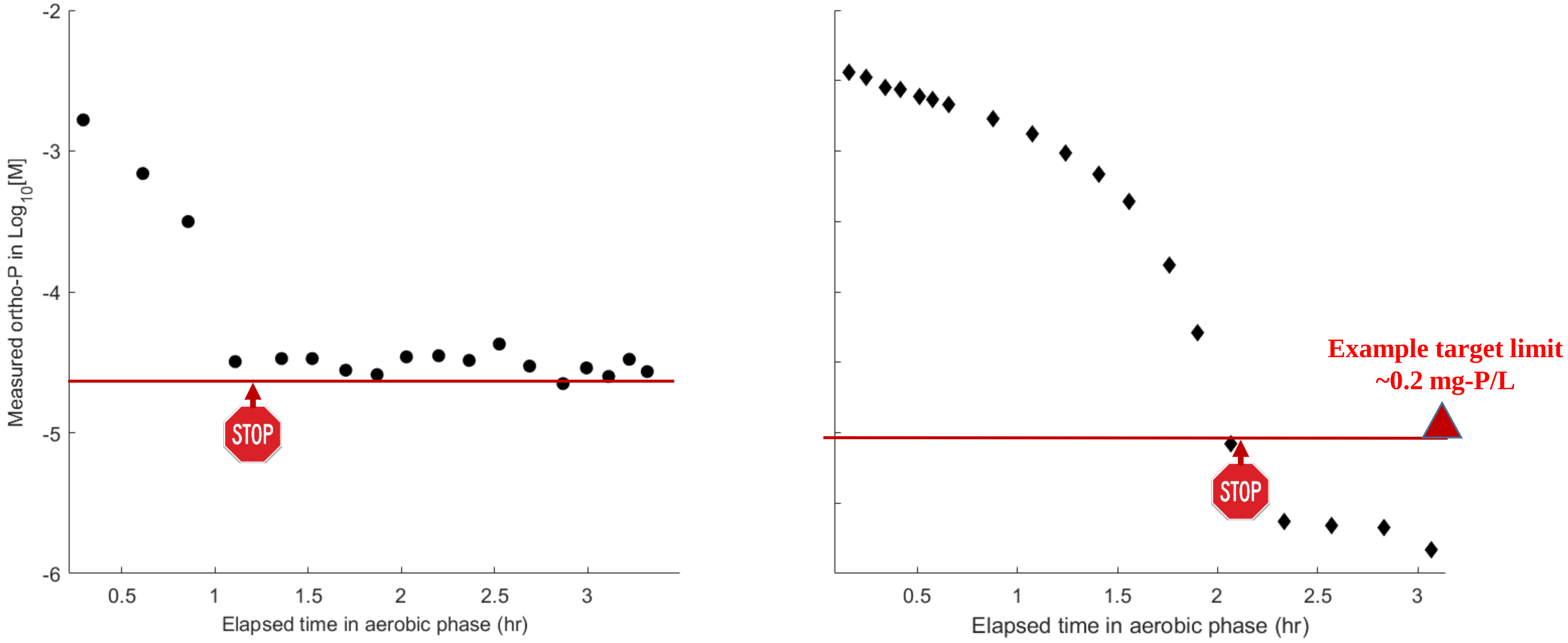
Lab reactor at UW (Pic. by: Amy Mueller)

- No sensor for P
- Some extra sensors included for evaluation

- Collected data for 10 "normal" uptake/release cycles and 10 "extreme" uptake/release cycles

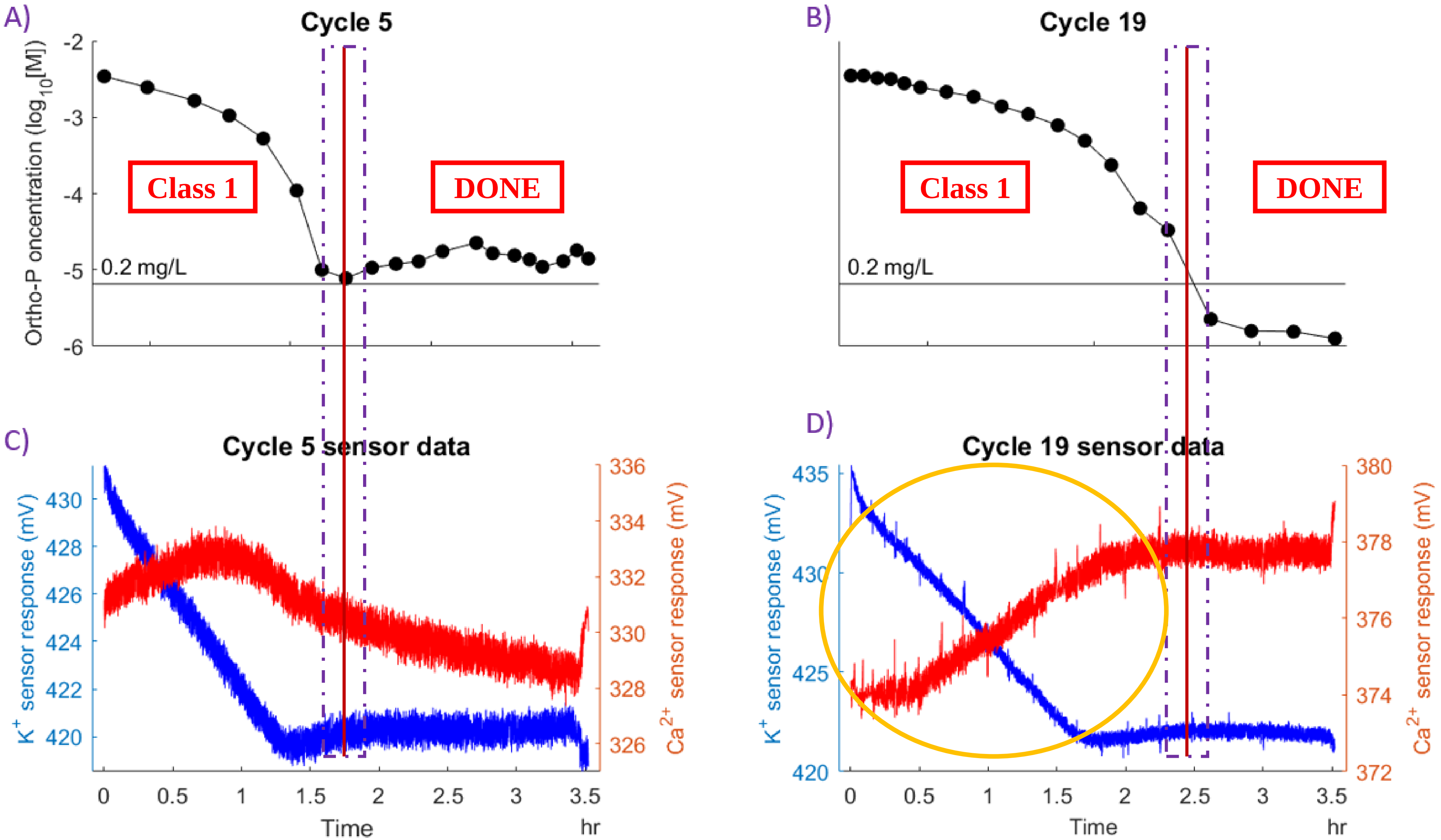
- "Extreme" cycles varied feed media in a way that would disrupt sensor accuracy but not biology

Observed two depletion patterns → Need controller to identify stop point in each



Real data!

“normal”



“extreme”

Machine learning – tested several methods

Support ve

Logistic re

Can we identify the “correct” stopping point?

What is the cheapest sensor hardware set we can recommend?

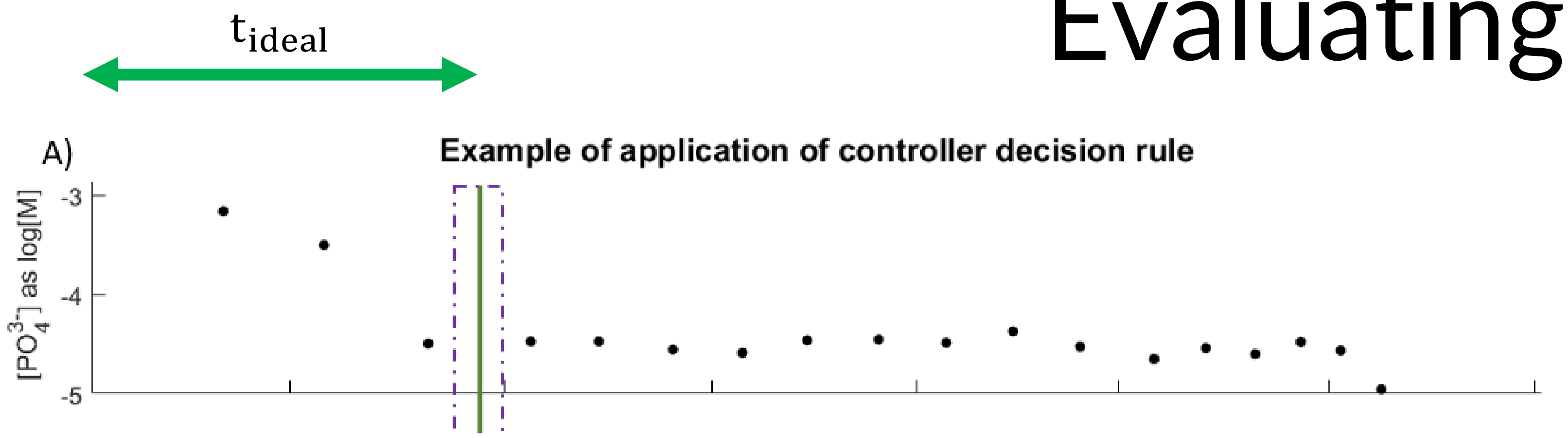
leveloped
ication methods

l for (relatively)
training datasets

ent model types

l each with
le combo/# of
sensor inputs (up to 7)

Evaluating the Controller



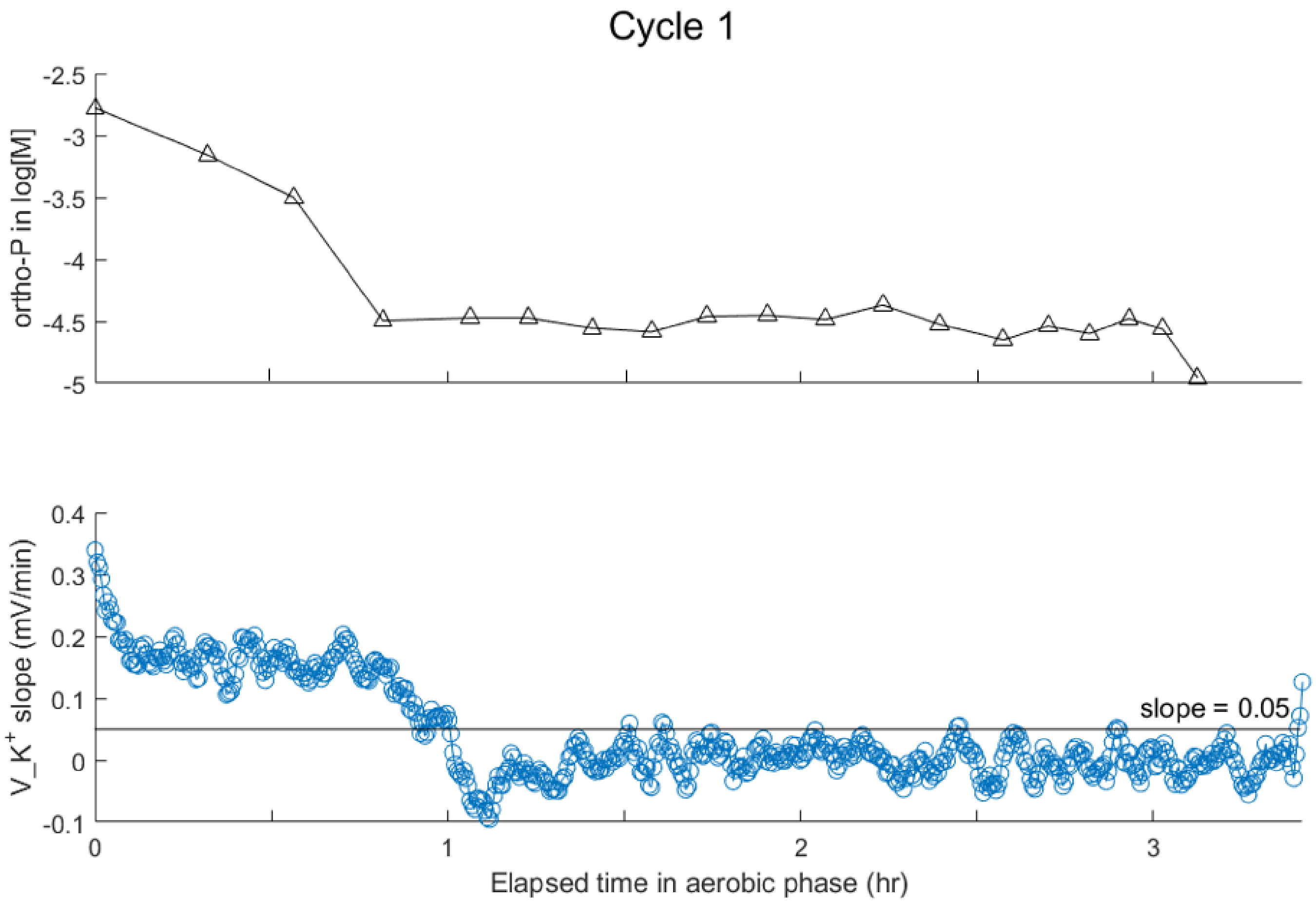
$t_{trigger}$: 5th consecutive **DONE** report

$$\text{Error} = \frac{t_{trigger} - t_{ideal}}{t_{ideal}}$$

Results

- The sensor combination that minimized error was… **ONLY A SINGLE K^+ SENSOR** !
- Using the “slope” configuration (sensor change per time) was far more robust to system variability & sensor noise
- This 1-sensor system was also **optimal for the “extreme” cases**
- Choice of ML model was not important (all 4 worked)

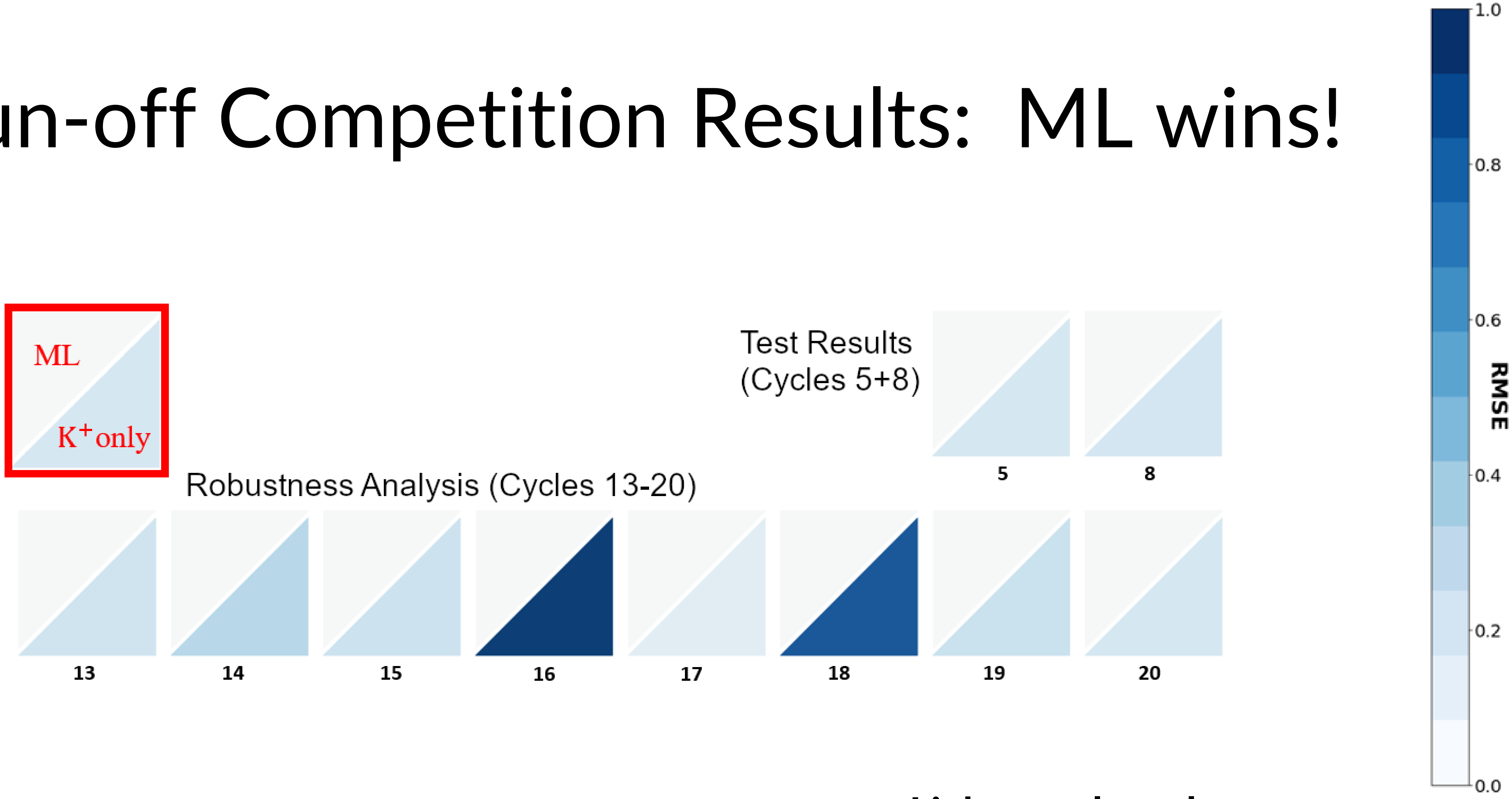
Wait – is ML even needed here??



Run one last competition – "simple" K+ threshold vs ML model

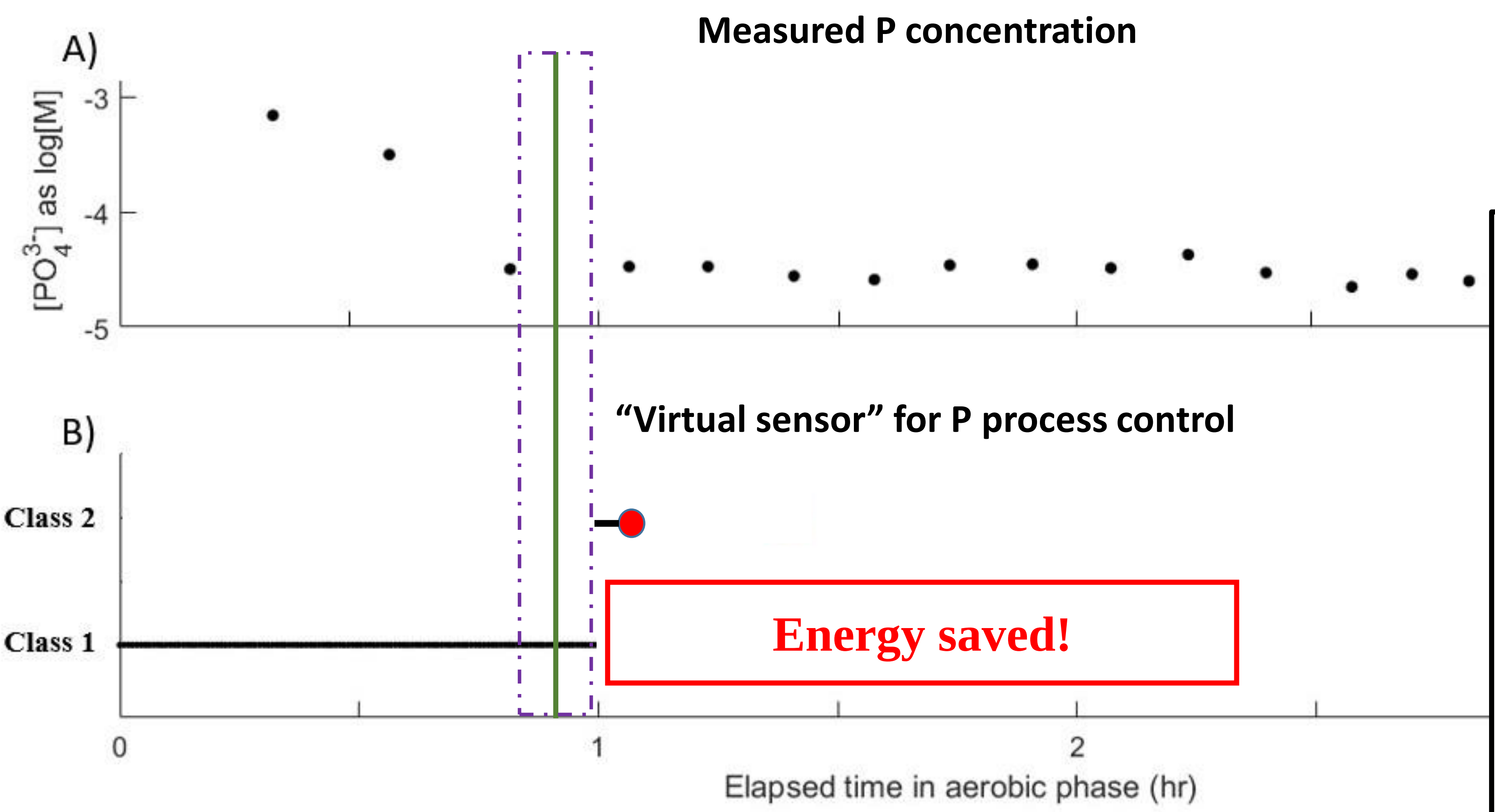
- Use K+ sensor slope data (since it was more robust)
- Simple threshold-based rule – choose slope cutoff based on training data

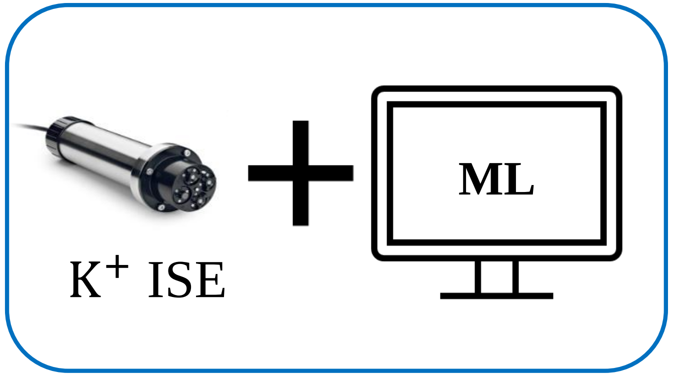
Run-off Competition Results: ML wins!



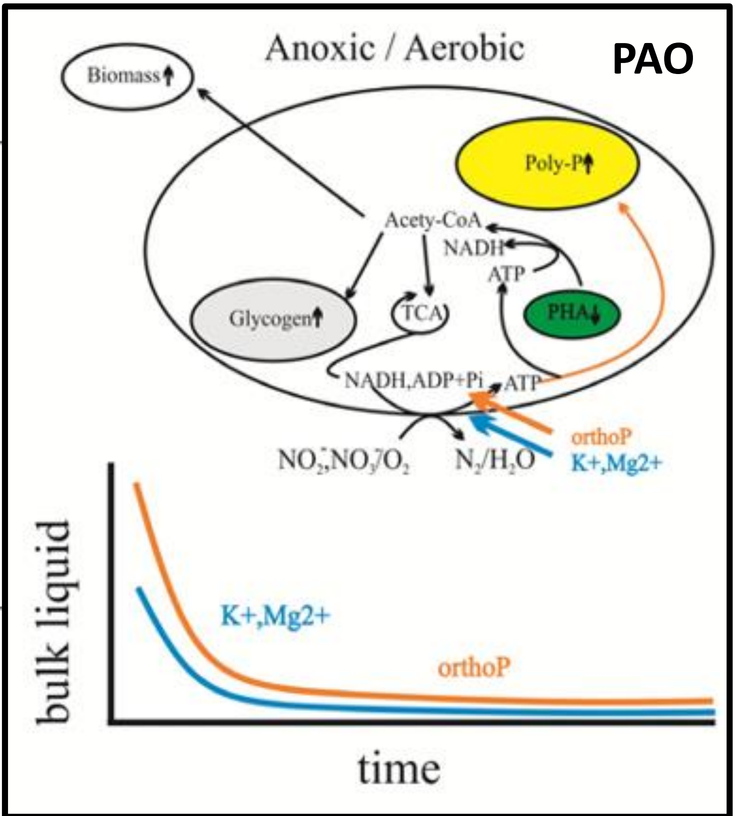
Lighter color = lower error

WW: P removal process control





100% accuracy!



So! CAN we operationalize ML for WW??

1. **Are the right predictor signals available?** For real-time monitoring or controls, need reasonable sensor data.
2. **Training data are critical:** do we have a pilot system we can "crash" and not worry too much (or can we simulate it)?
3. **Formatting the data** to best "teach" the algorithms is often more important than the choice of ML algorithm (within limits)
4. **Metric of success** needs to be in "WW framework" (NOT "ML framework")
5. **Implementing ML on SCADA?** While training is a lot of work, these algorithms run fast once trained & are easily ported to ops

SHOULD we operationalize ML for WW??

1. **Can we use a physics-based model?** If we already know the equations & it is computationally tractable, stick with that.
2. **Correlation vs causation.** Do we know which signals are trustworthy as predictors?
3. **Can we define and characterize failure modes?** (Even in a related pilot?) If not, there can be high risk in the edge cases.
4. **Are we generating an actionable insight?** Finding patterns can be satisfying, but how does it improve operations?

Thought provoking… how to move forward?

Collaboration, collaboration, collaboration!

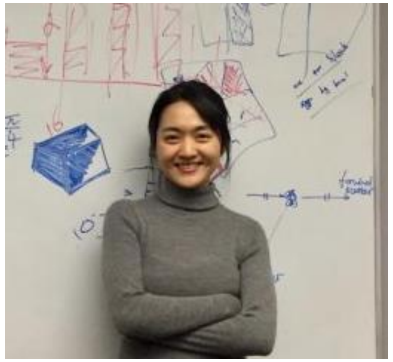
Operations + Consulting + Academia

- Defining the problem: how to promote optimized plant ops
- Pilot scale systems for robustness assurance
- “Computer science types” to streamline algorithm dev
- Metrics/results *in the context of improvement to ops*

Between multiple plants to test transferability and share learning

Acknowledgements – Questions ?

- Research group
  - Wenjin Zhang



- Collaborators:
  - Mari Winkler, University of Washington
- Funding
  - Northeastern University Faculty Funds

Contact:

Amy Mueller

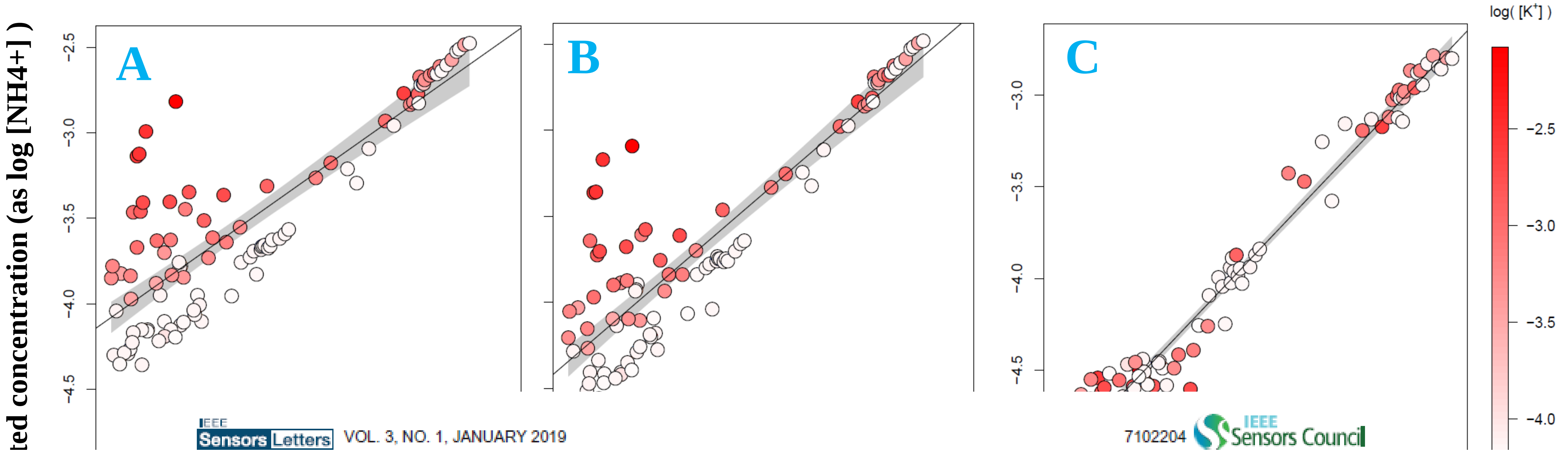
a.mueller@northeastern.edu

Additional Slides

Core steps/issues

- Interpolation vs. extrapolation
- Balanced datasets (to model anomalies when needed)
- Data normalization – minimize size of training dataset needed
- Defining metric of success – needs to be in context of plant ops. Cost decrease, removal efficiency increase, expanded set of conditions we can manage

WW: low concentration NH_4^+



IEEE Sensors Letters VOL. 3, NO. 1, JANUARY 2019
7102204 IEEE Sensors Council

Sensor data fusion

Data Fusion for Environmental Process Control: Maximizing Useful Information Recovery under Data Limited Constraints

Andrew M. Snauffer[1*], Umang Chauhan[1], Kathryn Cogert[2], Mari K. H. Winkler[2], and Amy V. Mueller[1]

[1]*Department of Civil and Environmental Engineering, Northeastern University, Boston, MA 02115 USA*
[2]*Department of Civil and Environmental Engineering, University of Washington, Seattle, WA 98195 USA*

EBPR cycles

~60 min anaerobic phase

~210 min aerobic phase

~10 min feeding

Reactor operation

~8 min settling

20 cycles measured

Cycle 1 – 10 cycles are <u>“normal”</u>

25 samples / cycle

The reactor influent* is similar to real operating EBPR reactors

Cycle 11 – 20 cycles are <u>chemically-varied</u>

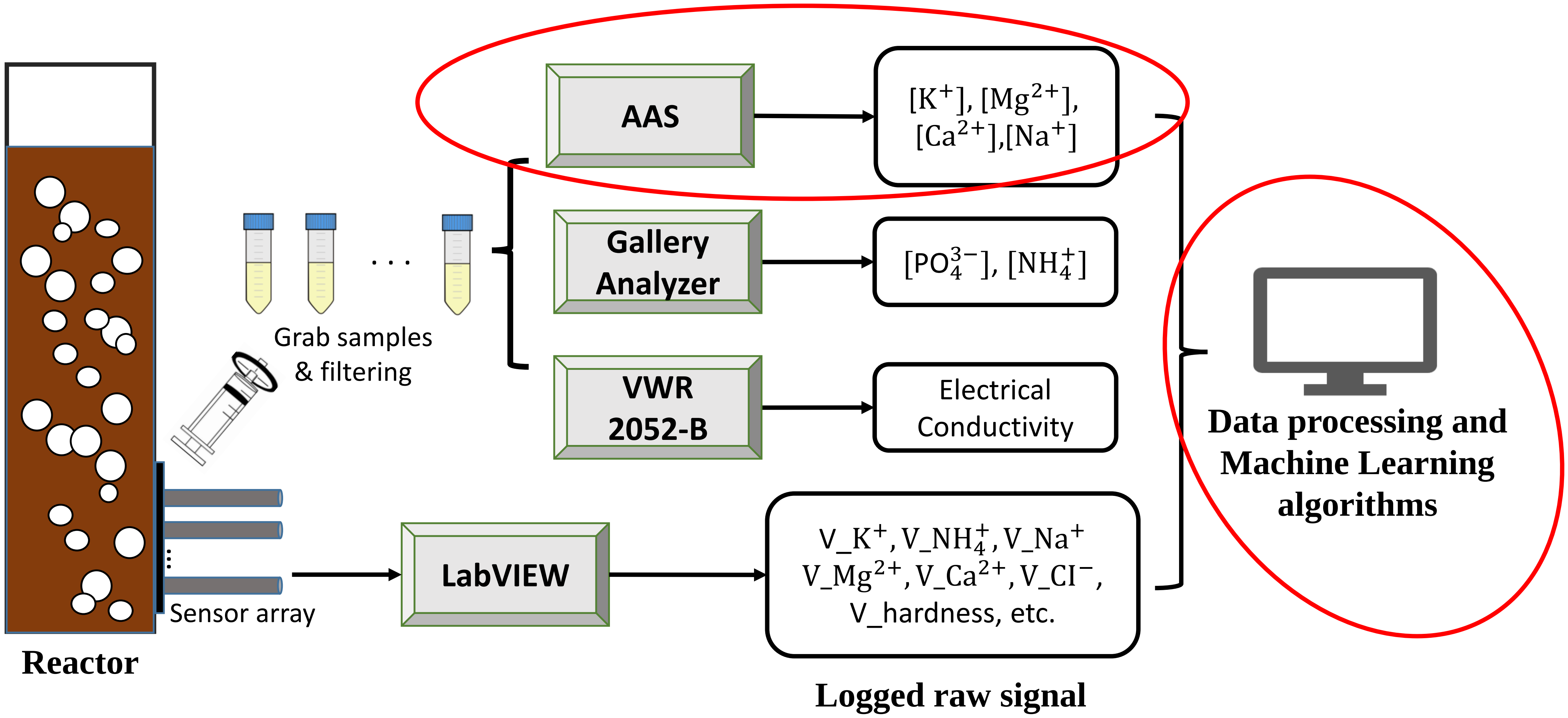
25 samples / cycle

The Mg^{2+}/Ca^{2+} ratio was changed in influent recipe by increasing Ca^{2+} concentration

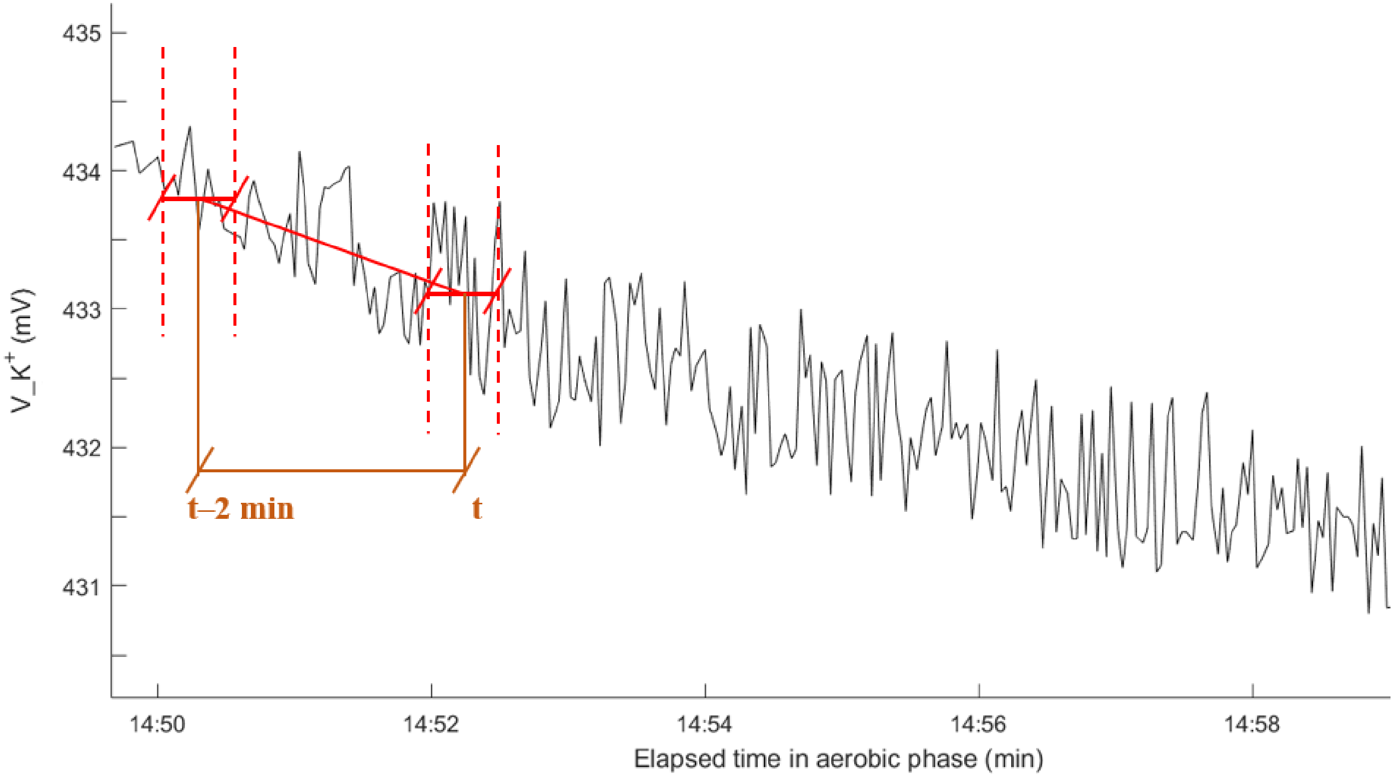
*Influent = synthetic wastewater

Reactor operation from Wei, Stephany P., et al (2021)

Project overview



Predictors – input data extraction



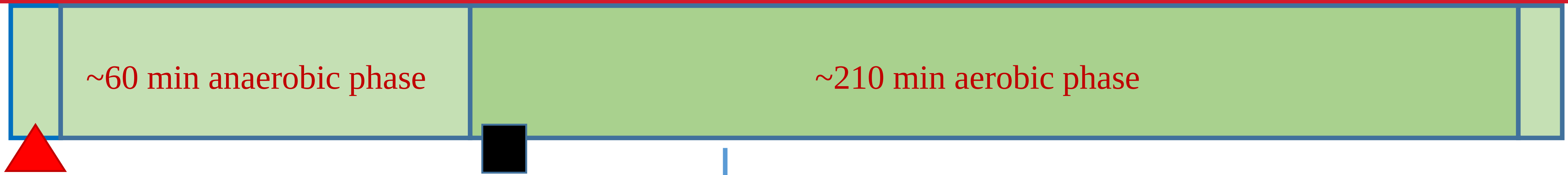
Raw predictor extraction:

- Mean value of a 35-sec window

Slope-based predictor extraction:

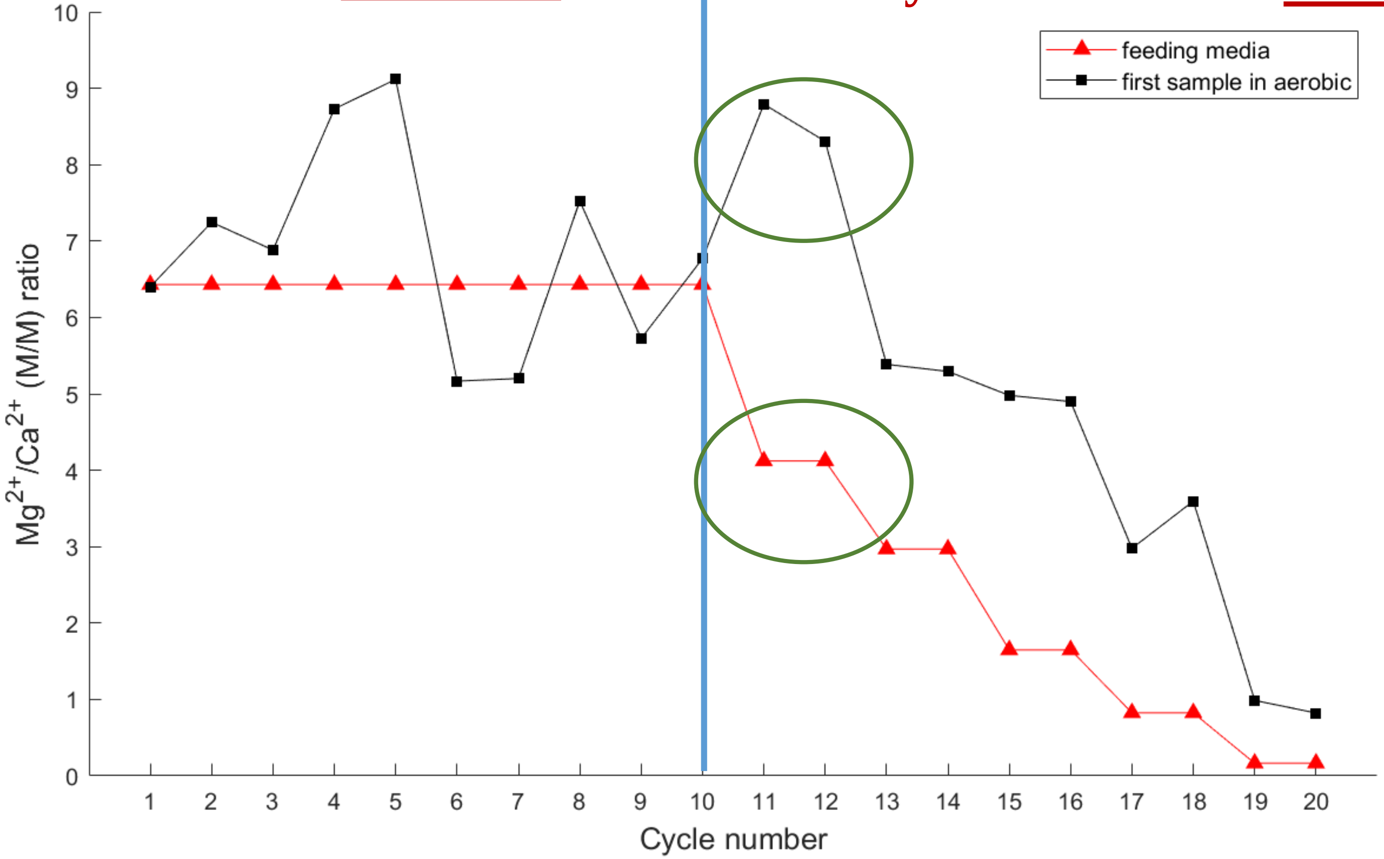
- Two sensor readings separated by a 2-min time window

EBPR cycles for training

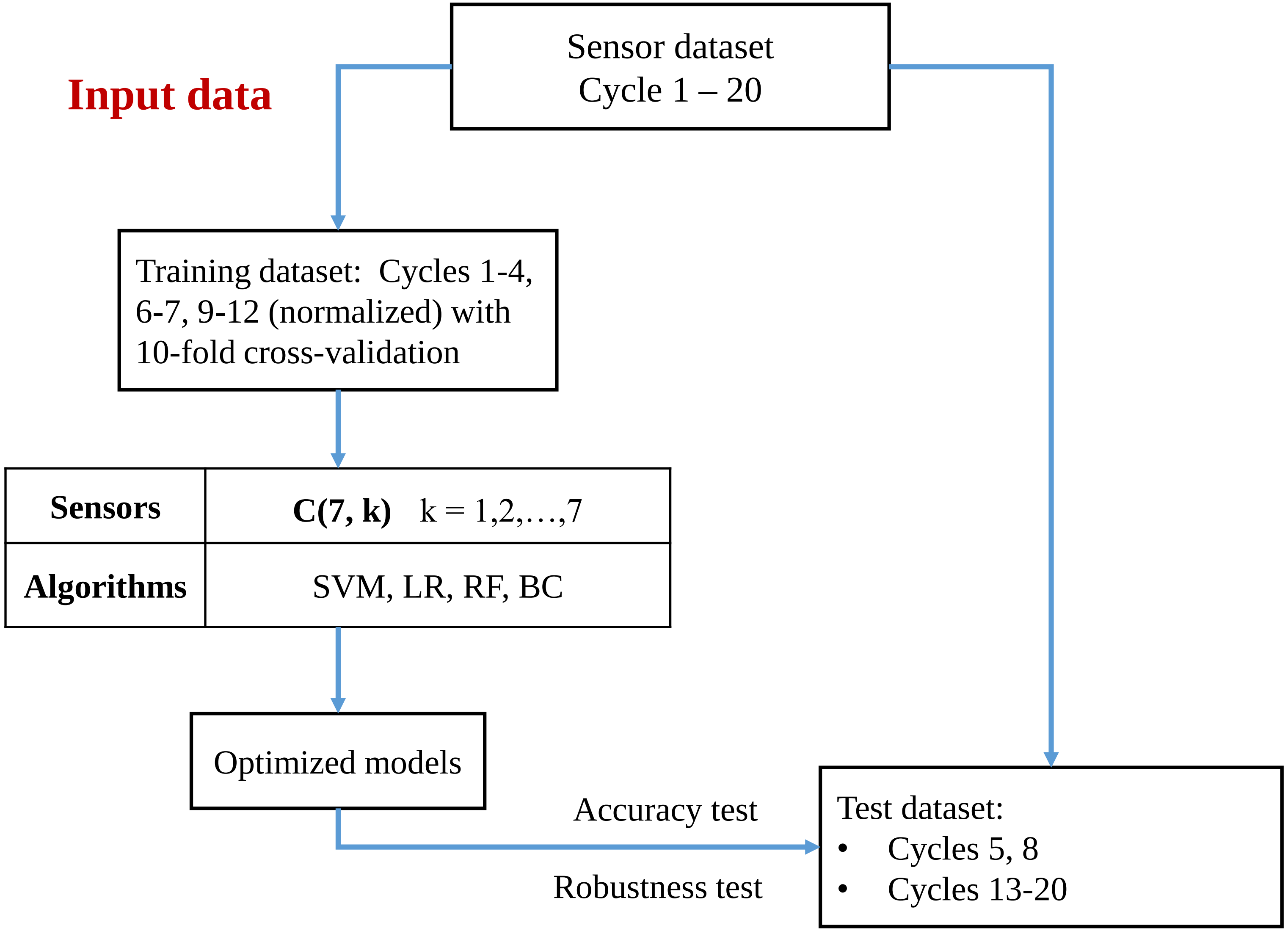


Cycles 1 – 10 are "normal"

Cycles 11 – 20 are chemically-varied



Model training

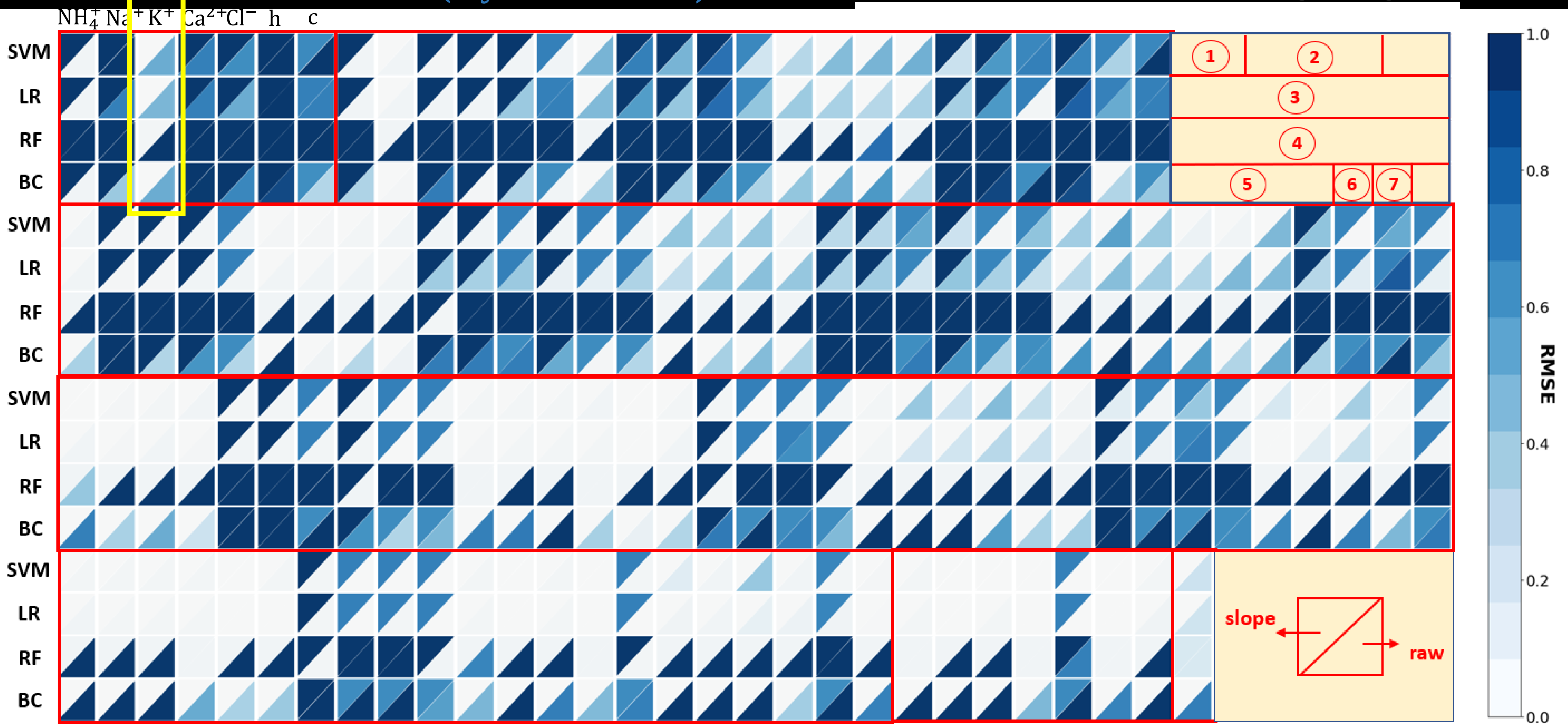


	Data size	
Training	3048	Class 1: 1175 Class 2: 1873
Test	741	Class 1: 319 Class 2: 422
“Extreme”	1824	Class 1: 1060 Class 2: 764

Optimized parameters

Model	Parameter search space	Tuning	Optimal setting
SVM	**Kernel choice:** linear, Gaussian, polynomial **Misclassification penalty factor (C):** $\log[10^{-3}, 10^{3}]$	Default	Linear log(298.38)
LR	**Regularization function:** Lasso, Ridge **Regularization strength (λ):** [0, 0.1]	Default Manual	Lasso 0.0035
RF	**Tree size:** [5,300]	Manual	12
BC	**Kernel:** Gaussian, triangular, Epanechnikov, uniform **Kernel smoothing window width:** $[10^{-2}, 1]$	Default Manual	Gaussian 0.1149

Results on test datasets (Cycles 5 and 8)



Results on sensitivity data (Cycle 19)