Using deep learning to monitor the operational performance of an anaerobic wastewater treatment plant

Jose Bolorinos, Chris Pierce, CS230, Stanford

Motivation

Anaerobic reactors are new, low energy alternatives to conventional wastewater treatment that can harvest energy from wastewater as methane. However, they require careful operation as anaerobic organisms are very pH sensitive. Availability of low-cost sensors means that these systems can be monitored in real-time to identify operational issues. We explored the advantage of deep learning as an efficient way to predict a reactor's pH with a 24 hour time horizon (long enough ahead to prevent operational issue)

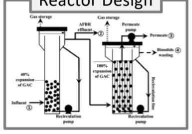
Data and Setting

Codiga Center, experimental, pilot scale, anaerobic membrane-based wastewater treatment facility on Stanford campus

Codiga Center

Data from real time sensors in facility's reactors:
- pH
- temperature
- conductivity
- Flowrates (water + biogas)

Reactor Design



Data aggregated into 12,768 hours and segmented into:
- Training set (8,000 hours)
- Dev set (1,000 hours)
- Test set (1,000 hours)

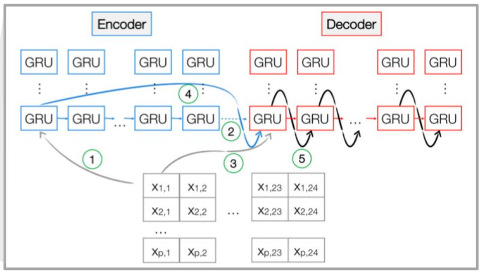
Conclusions & Next Steps

Takeaways
- Predictive performance on par with facebook prophet (could outperform with further hyperparameter tuning), easily outperforms ARIMA
- Error of +/- 0.1 24h ahead => could anticipate drop in pH in time to shield microorganisms from harmful drop in pH
- Co-prediction of related sensors generally worsens performance
  - Conductivity (salts + buffer) closely related to pH in influent (AT203)
  - pH in reactor (AT305) is mediated by microorganisms

Further work:
- Transfer learning: anticipate sensor drift and need for recalibration
- RNN model to predict biogas generation?

Network Architecture

Approaches tested:
- ResNeT CNN
- Inception CNN
- Simple RNN
- Standard RNN sequence model
- Encoder – Decoder RNN
- Encoder – Decoder RNN + teacher forcing mechnanism
- Encoder – Decoder RNN + teacher forcing + output alignment
- Encoder – Decoder RNN + teacher forcing + output alignment + attention mechanism

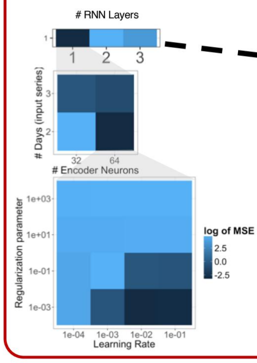


Process Description
1. Feed input timeseries data into encoder
2. Take encoder hidden state and feed into decoder as initial state
3. Feed input timeseries data into decoder
4. Feed encoder output into decoder
5. "Teacher Forcing" - feed output of previous decoder step into next decoder step

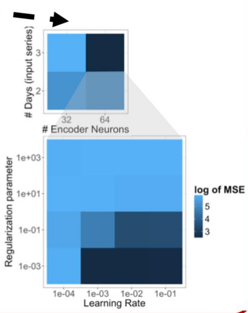
Hyperparameter search

Nested grid search of 384 models run with GRUs in Stanford's Sherlock cluster (validated with dev set)

1-channel model

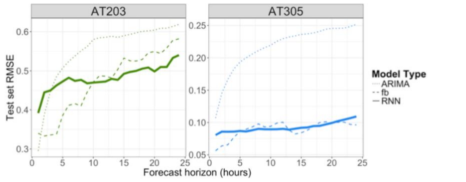


2-channel model



Predictive Performance

1-channel model



2-channel model