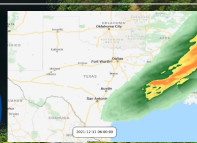
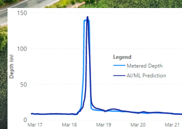
BLACK & VEATCH

# Perspectives on Machine Learning in the Real World for Flood Forecasting

Chris Ranck, PE, BCEE, BC.WRE, ENV SP, Black & Veatch

March 26, 2025








## Agenda

Machine Learning: What, How, Where?
Case Study Example #1: Citizens Energy Group
Case Study Example #2: Garcitas Creek
Case Study Example #3: Indian Creek
How to Get Started
Discussion

BLACK & VEATCH


## Machine Learning: What, How, Where?

What is Machine Learning?

- A Data-Driven computational process to develop patterns and relationships
- A calculation without an equation or model
- Can provide tremendous and instantaneous predictive power
- Computational engine behind a Digital Twin


BLACK & VEATCH

## Machine Learning: What, How, Where?

What Machine Learning *IS NOT*?

1. Something that will take away jobs
2. Software that costs tons of money
3. A black box and you have no idea what it's doing
4. So difficult that only a select handful of academics and technologists can do it
5. Inaccessible to the average water or wastewater practitioner
6. Neural networks eventually become sentient and create hostile armies of robots
7. The ML feeds all your confidential data to someone else's central AI

Image courtesy of Marvel Studios

 4

## Machine Learning: What, How, Where?

WEF Fact Sheet December 2023:

https://www.accesswater.org/?id=-10100500

Water Environment Federation
the water quality people

AI / ML in the Water and Wastewater Systems Sector:
GENERATIVE AI (GenAI)

INTRODUCTION

Generative AI (GenAI) is defined by the US government as "the class of AI models that emulate the structure and characteristics of input data in order to generate derived synthetic content." (E.O. 14110, 88 FR 75191). As a subset of AI ML Deep Learning, GenAI uses complex algorithms and statistical models that are trained using large amounts of data, which is often open source. Large Language Models (LLMs), for example, make use of training data to learn patterns that GenAI can then emulate.

before deployment. Yet only 24% of their [GenAI] projects will include a cybersecurity component within the next six months—and 69% say innovation takes precedence over cybersecurity for [GenAI]." (IBM, p.5) Utilities will need to balance cybersecurity concerns against innovation opportunities.

OPPORTUNITIES

GenAI is evolving rapidly and can deliver creative and innovative solutions for utilities. Common use cases include research and document preparation, image

 5

## Machine Learning: What, How, Where?

How Does it Work?

**Feed-Forward Neural Network**: Discrete Events, Asset Management

| Main ID | Material | Install Decade | Break ? | Soil Corrosivity | Break Year Days > 90 F | Break Year Days < 32 F | Age of Failure | ML Prediction |
|---|---|---|---|---|---|---|---|---|
| 9258 | Cast Iron | 1940s | No | None | 16 | 24 | N/A | 110 |
| 10603 | Cast Iron | 1930s | Yes | High | 17 | 46 | 82 | 84 |
| 20433 | Cast Iron | 1970s | Yes | High | 18 | 20 | 39 | 38 |
| 52160 | PVC | 1980s | Yes | High | 9 | 32 | 27 | 31 |
| 57359 | PVC | 1970s | No | Low | 16 | 24 | N/A | 70 |

 6

## Machine Learning: What, How, Where?

### How Does it Work?

**Long-Short Time Memory**: Time Series Forecasting

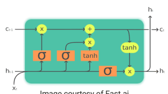
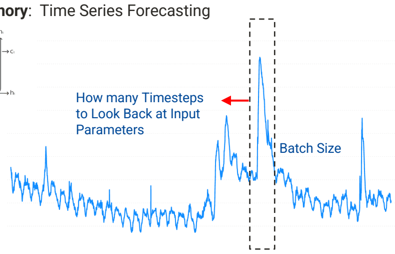


Image courtesy of Fast.ai








## Machine Learning: What, How, Where?

### Where to Apply Machine Learning?

- “I’d like more certainty in where to dispatch staff in big storms.”
- “I know there’s a pattern in the data but I can’t find it.”
- “I have data gaps to fill in.”
- “I need answers but I can’t collect data everywhere.”
- “I need answers but a traditional model would take years.”
- “I need answers in real time.”
- “Actually, I need answers a few hours ahead of real time.”
- “Did I say hours? I meant days.”

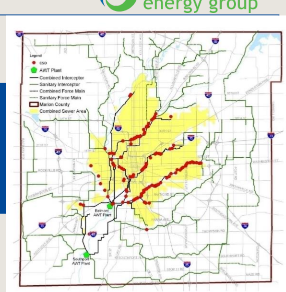




BLACK & VEATCH

citizens energy group

## Example 1: Citizens Energy Group



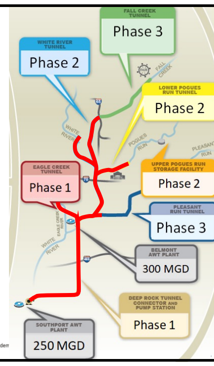
## Example #1: Citizens Energy Group

Citizens Energy Consent Decree Projects

- Phase 1: DigIndy Tunnel
  ~ 9 Miles, 80 MG
- Phase 2: DigIndy Tunnel
  ~ 17 Miles, 170 MG

Can we predict inflow ***before*** the event happens?

- Machine Learning to predict Inflow
- Link to 72-hour NOAA forecast
- Present Forecasts in Power BI



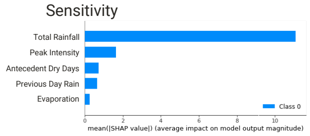


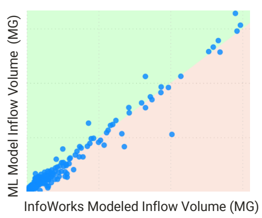

## Example #1: Citizens Energy Group

ML Tool Development: Dig Indy Tunnel Phase 2

- 2016 – 2019 Modeled Dataset
- Rainfall, Evaporation, Inflow
- Total Volume within 1%

Automated Parameter Sensitivity



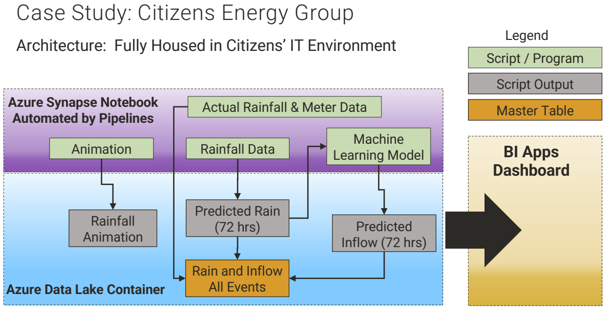







## Case Study: Citizens Energy Group

Architecture: Fully Housed in Citizens' IT Environment

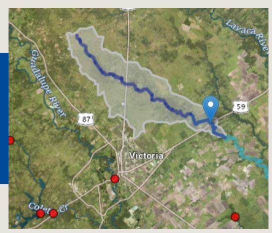


## Example #1: Citizens Energy Group


| Dates | Forecast Rain (in) | Gauged Rain (in) | Forecast Inflow (MG) | Metered Inflow (MG) |
|---|---|---|---|---|
| 8/9/2022 | 1.48 | 1.54 | 75 | 95 |
| 2/8/2023 – 2/9/2023 | 1.81 | 1.25 | 55 | 56 |
| 3/3/2023 | 2.92 | 2.24 | 120 | 188 |
| 1/23/2024 – 1/28/2024 | 3.04 | 2.30 | 57 | 57 |
| 3/31/2024 – 4/3/2024 | 4.27 | 3.87 | 114 | 117 |

BLACK & VEATCH

## Example #2: Garcitas Creek

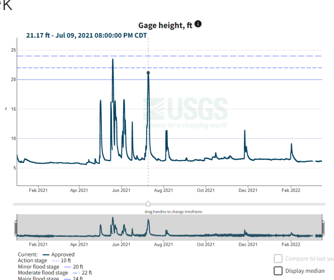





## Example #2: Garcitas Creek

Garcitas Creek near Inez
USGS 08164600

- Drainage Area: 91.7 Square Miles
- Flood stage:
  - Action Stage: 10 feet
  - Minor Flood: 20 feet
  - Major Flood: 24 feet
- Station collects Stage, Flow, ~~Rainfall~~

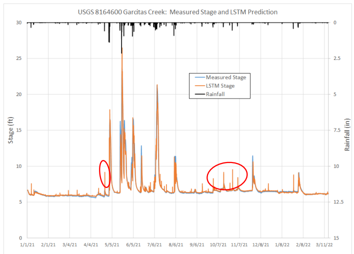



BLACK & VEATCH

## Example #2: Garcitas Creek

### LSTM Neural Network

- Flood Events Predicted
- Action Events Predicted
- No False Positives
- No False Negatives
- Several Near-false Positives
- Predicts ~ 16 hours into the future




BLACK & VEATCH

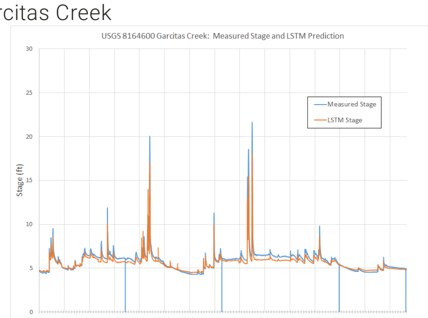


## Example #2: Garcitas Creek

### How does 2022-2025 Look?

For more information:

LinkedIn Video of the LSTM development and application

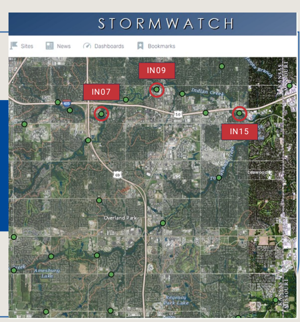



BLACK & VEATCH



BLACK & VEATCH

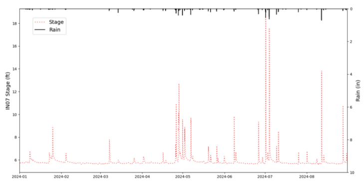
## Example #3: Indian Creek, Kansas



## Example #3: Indian Creek

Process

- Focus on peak events, events reaching Flood stage or Monitor Stage
- Data Sources:
  - Rainfall and Stage data from StormWatch
  - Daily temperature from NOAA station at MCI
  - Temperature data will be converted to evaporation




BLACK & VEATCH

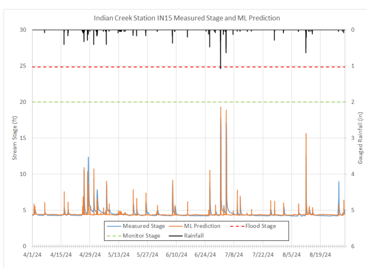




## Example #3: Indian Creek

Results:  Station IN15

- Parameters
  - Input:  Rain from IN07, IN09, IN15
  - Input:  Daily evaporation
  - Offset:  4 hours
  - Batch size:  72 hours
  - Nodes:  25
  - Dropout:  10%
- Peak Stage
  - No False Positives or Negatives
  - July 2024 Peak Stages ~ 1.5' High




BLACK & VEATCH

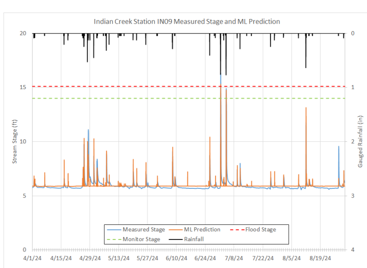




## Example #3: Indian Creek

Results:  Station IN09

- Parameters
  - Input:  Rain from IN07, IN09, IN15
  - Input:  Daily evaporation
  - Offset:  4 hours
  - Batch size:  60 hours
  - Nodes:  50
  - Dropout:  10%
- Peak Stage
  - No False Positives or Negatives
  - July 2024 Peak Stages ~ 1' Low

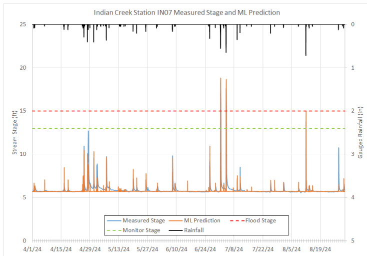



BLACK & VEATCH

## Example #3: Indian Creek

Results: Station IN07

- Parameters
  - Input: Rain from IN07, IN09, IN15
  - Input: Daily evaporation
  - Offset: 4 hours
  - Batch size: 72 hours
  - Nodes: 25
  - Dropout: 10%
- Peak Stage
  - No False Positives or Negatives
  - (Near false positive 8/11/2024)
  - July 2024 Peak Stages within 1'

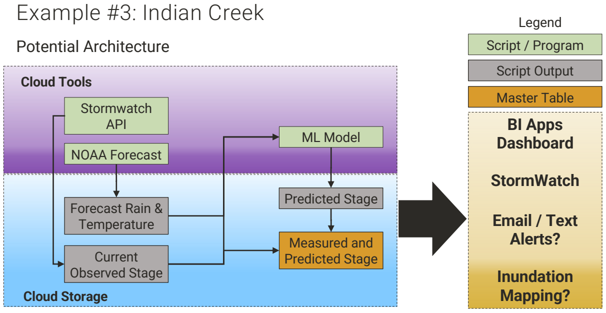







## Example #3: Indian Creek

Potential Architecture








BLACK & VEATCH

# How to Get Started

## How to Get Started

**Before Getting Started, Consider:**

1. Who needs to see this?
2. How will they use the predictions?
3. What do you want to predict?
4. What data can you get to generate the predictions?
5. How often do you need to refresh the predictions?

BLACK & VEATCH





## How to Get Started

- Commercial Software not necessary
- Available via Microsoft, Google, Amazon Environments
- Can Deploy at any scale and within Existing IT Infrastructure
- If all else fails, ask ChatGPT

RA

```
    y.append(data[i, 0])
X = np.array(X)
y = np.array(y)
X = np.reshape(X, (X.shape[0], X.shape[1], 1))

# Split the data into training and testing sets
train_size = int(len(X) * 0.8)
X_train, X_test = X[:train_size], X[train_size:]
y_train, y_test = y[:train_size], y[train_size:]

# Train the model
model.fit(X_train, y_train, epochs=100, batch_size=32)

# Evaluate the model on the testing set
scores = model.evaluate(X_test, y_test)
print("\n%s: %.2f" % (model.metrics_names[0], scores))

# Make predictions on new data
new_data = np.array([[0.5], [0.6], [0.7]])
new_data = (new_data - data.mean()) / data.std()
new_data = np.reshape(new_data, (1, timesteps, 1))
prediction = model.predict(new_data)
print("Predicted river stage: %.2f" % prediction[0, 0])
```

In this code, we first define the architecture of our LSTM model, which consists of one LSTM layer with 50 neurons and one output layer with 1 neuron.

We then compile the model using the `mean_squared_error` loss function and the `adam` optimizer.

BLACK & VEATCH



BLACK & VEATCH

# Discussion

+1 317 288 3985
ranckcj@bv.com