Analyzing Storm Events Impact on Rivers

Project Members: Dylan O'Keeffe, Derek Riccardi, and Kevin Rhoades
Project Advisor: Laura Dietz
dto1002@wildcats.unh.edu, dar1025@wildcats.unh.edu, klr1016@wildcats.unh.edu

Forecasting Storms

Introduction

Storms cause drastic changes in flow rate in rivers and the water runoff can bring large amounts of contaminants with them, which can be harmful. Our goal is to provide researchers with a command line interface (CLI) to help them isolate storm events for more targeted analysis.

Methodology

Storm event is defined as an increase in flow rate due to precipitation.

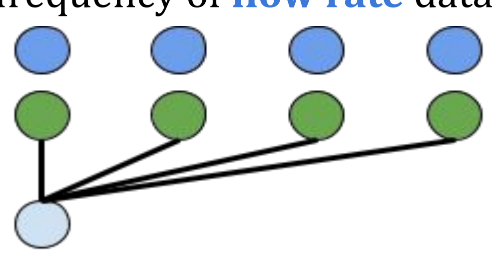
Automatically generate storm event labels using precipitation data

Apply machine learning models to that data

Classification of flow vs solute vs time graphs

Data Processing

Data Merging -
Upsample **precipitation data** to match frequency of **flow rate** data.



Data Balancing & Downsampling -
Class imbalance favoring negative labels.
Solution: downsample, selecting balanced regions encompassing storm events.

Storm Classification

Classification Models

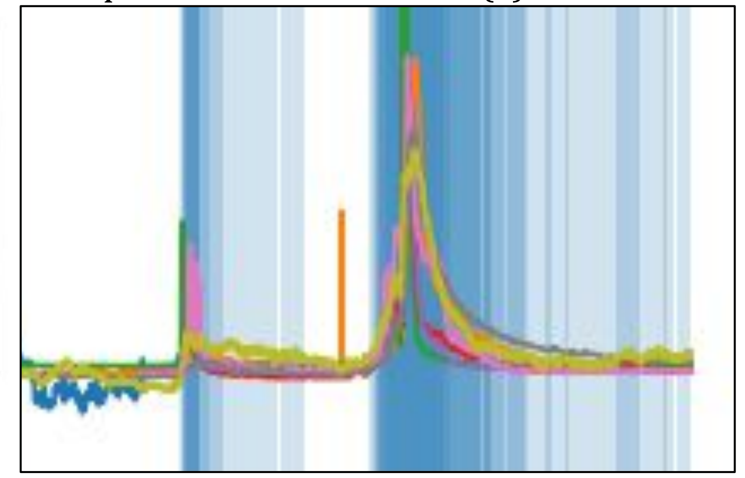
Logistic Regression

- Observation at time t -> Classification at time t

Convolutional Neural Network

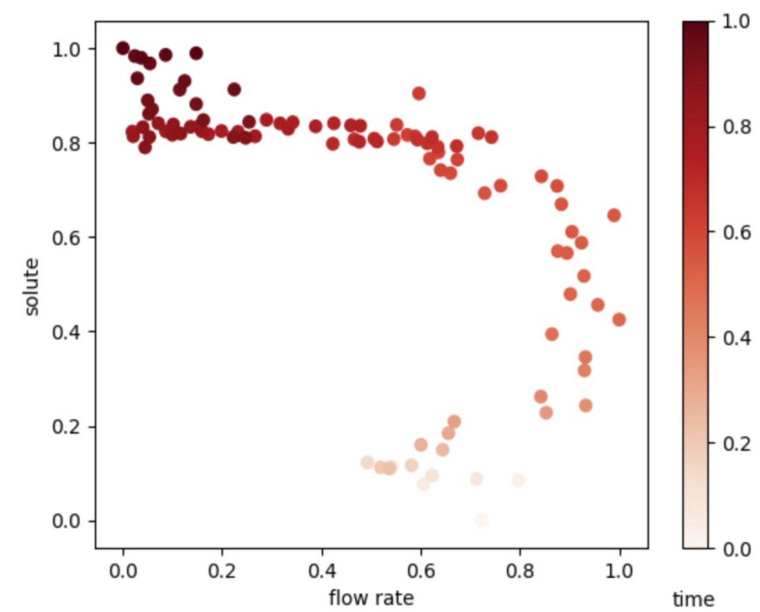
- Just like Logistic Regression, but also uses some number of previous time steps.

Example Classification: Time (x) vs Flow Rate (y)

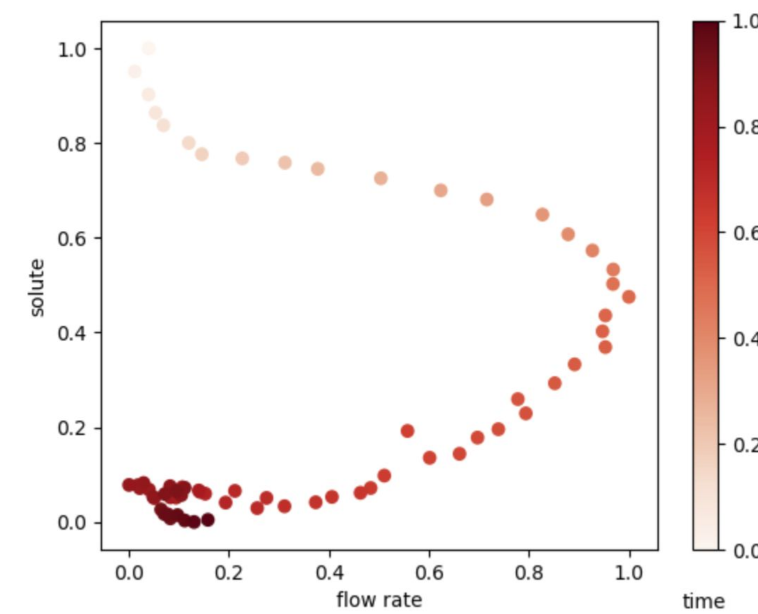


Blue regions indicate positive classification label

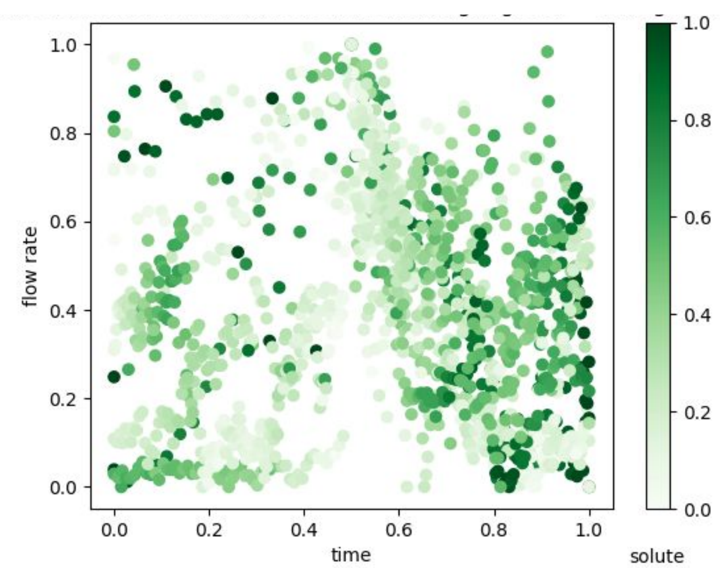
Solute Based Storm Analysis



Example of a counterclockwise movement storm



Example of a clockwise movement storm



Shows how solute can vary across different storms

Storm Analysis

- Different storms fall into 2 main categories of graphs: clockwise and counterclockwise movement

- We are able to autonomously classify these graphs as counter clockwise/clockwise by analyzing how the data moves over time

Our Command Line Interface Tool

Commands:

- *Train:* Trains and outputs a classification model on the provided data.
- *Test*: Evaluates a classification model using Accuracy, Precision, and Recall metrics
- *Predict*: Saves a classification model's predictions on a set of observations.
- *Visualize*: Creates plots shown in our analysis section
- *Classify*: Classifies the solute vs flow rate plots into counterclockwise vs clockwise plots

Inputs are CSV files containing flow rate (Observations) & precipitation data (Labels).

Results

Using precipitation data was effective at identifying storm regions to analyze.

Classification of the solute vs flow rate plots into clockwise vs counterclockwise movement works well for most storms.

The machine learning models we've produced so far have performed poorly, due to biases.

Conclusions

Our CLI can be used in conjunction with precipitation data for clockwise vs counter clockwise analysis. It's machine learning capabilities are not ready for use.

References

River Data: https://ddc.unh.edu/
Precipitation Data: https://www.ncei.noaa.gov/
Python Libraries: Pandas, Matplotlib, TensorFlow, NumPy, SciKit-Learn