NYU

Exploring Filtering Methods to Denoise Flood Depth Signals

Tanvi Bansal, Lake Wang, Jason Moon

Probabilistic Time Series Analysis Final Project

12.10.24

FloodNet NYC

Background

- Flooding events in urban areas are particularly difficult for emergency responders due to the variable behavior of flood profiles in these areas.
- Specifically, in urban areas flood depth, duration, severity, etc. vary significantly based on hyperlocal conditions.
- To address this, the FloodNet NYC team deployed a fleet of depth sensors across NYC in order to establish a real-time, hyperlocal flood monitoring system.

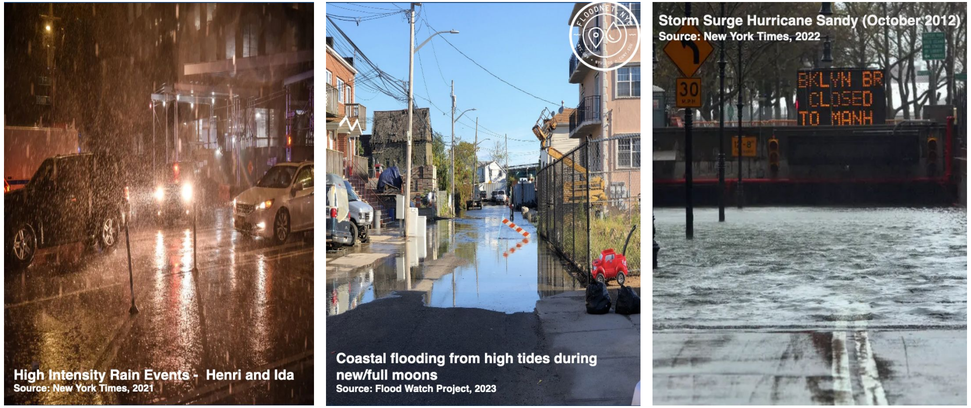

Figure 1. Various Flooding Conditions around NYC

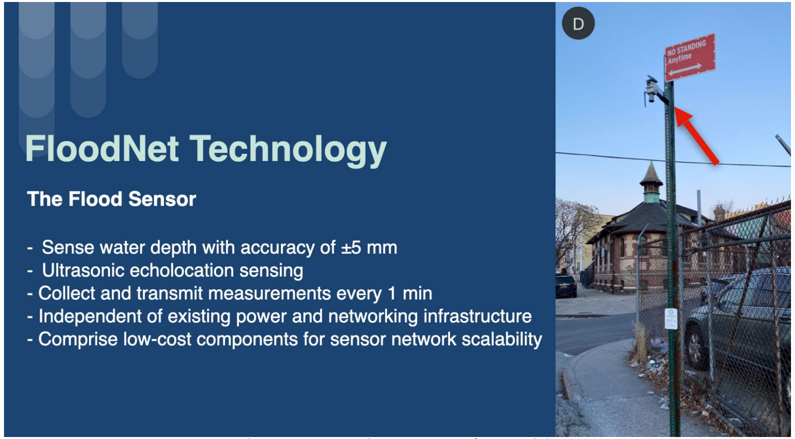

Figure 2. Depth Sensor Infographic

NYU

FloodNet NYC

Current Data & Methodology

- Sensors measure ground depth every minute, and an automated alerting system notifies stakeholders when flood events occur
- Current system suffers from high false alarm rate due to 2 key challenges:
  - (1) sensors capturing non-flood events
  - (2) measurement noise from environmental uncertainties
- The current filtering method (heuristic filter) reduced false-alarm rate from 95.2% to 87%

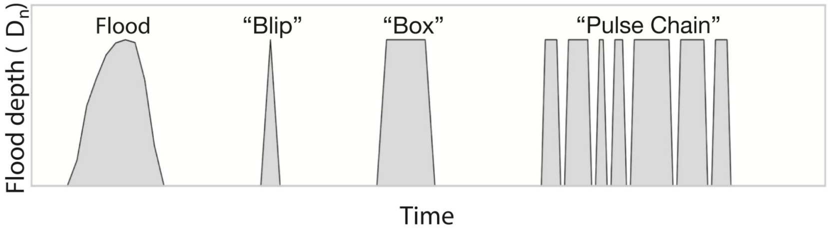

Figure 3. Sample Plot of Noisy Time Series Signal

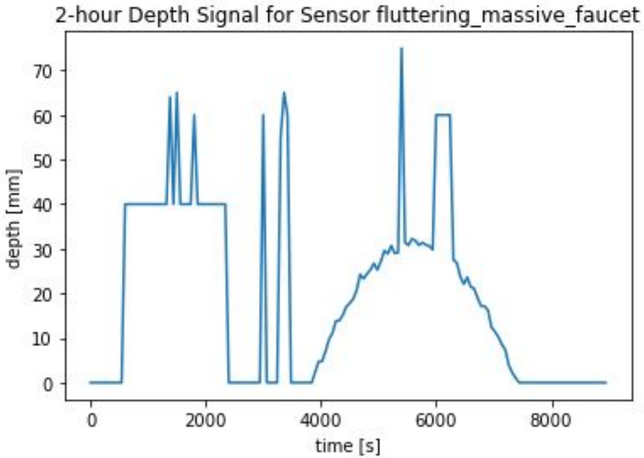

Figure 4. Plot of 2-hour Noisy Time Series Signal from Sensor in Queens

NYU

Motivating Question

How can we further isolate the true flood signal to improve system reliability?

Gaussian Process

Key Challenges
- Variable length of events, different shapes of floods

Method
- Transform flood data as continuous timeseries by normalizing
- Jointly fit gaussian process model with all flood data samples
    - Kernel: Matérn32 kernel (once differentiable RBF) + white-noise
- Calculate the number of values of non-flood events that are outside the 95% confidence interval

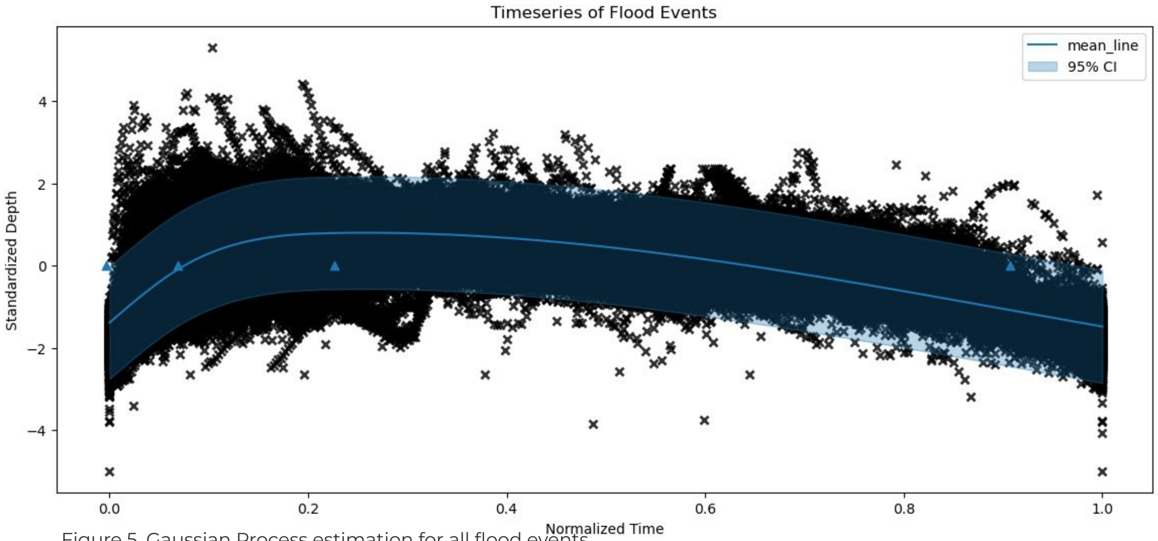

Figure 5. Gaussian Process estimation for all flood events

NYU

Kalman Filtering

Key Challenges

- Variable length of events, single set global parameters, continuous multi-modality readings for prediction

Obtaining Estimates

- Hidden dimension: 3 (depth, velocity, acceleration)
- dynamic dt in transition matrix fitting
- a single set of global parameters are iteratively updated with estimates from individual flood events
- Apply global parameters to filter all readings
- Global parameter estimates: Q ~ 0, R ~ 7

$$\textbf{Transition Matrix:} \quad \mathbf{A} = \begin{bmatrix} 0.99 & 0.95 & -0.14 \\ 0.00 & 0.99 & 1.84 \\ -0.00 & -0.00 & 0.97 \end{bmatrix}$$

$$\mathbf{A} = \mathbf{T}_{\text{numerator}} \cdot (\mathbf{T}_{\text{denominator}})^{\dagger}$$

$$\mathbf{T}_{\text{numerator}} = \sum_{i=1}^{N} \sum_{t=2}^{T_i} \mathbf{x}_t^{(i)} \mathbf{x}_{t-1}^{(i)\top}, \quad \mathbf{T}_{\text{denominator}} = \sum_{i=1}^{N} \sum_{t=2}^{T_i} \mathbf{x}_{t-1}^{(i)} \mathbf{x}_{t-1}^{(i)\top}$$

$$\mathbf{Q} = \frac{\sum_{i=1}^{N} \mathbf{Q}_{\text{fit}}^{(i)}}{N \cdot (T_i - 1)}$$

$$\mathbf{R} = \frac{\sum_{i=1}^{N} \mathbf{R}_{\text{fit}}^{(i)}}{N \cdot T_i} + \epsilon \mathbf{I}$$

NYU

Particle Filtering

Key Challenges

- Large number of particles needed due to 3D latent state, complex state and likelihood definitions due to multimodality

Obtaining Estimates

- Assumptions
  - True state condition: z_c ∈ {rest, flood}, updates independent of past
  - True state position: z_p = [pos, vel, acc], updates dependent on past
- State update

$$p(z_{c,t}) = \begin{cases} 0.7, \mathrm{c} = \mathrm{rest} \\ 0.3, \mathrm{c} = \mathrm{flood} \end{cases}$$

$$z_{p,t} = \begin{cases} \{0,0,0\}, \mathrm{c} = \mathrm{rest} \\ A \cdot z_{p,t-1} + \mathcal{L}(\mu = 0, \sigma = .01), \mathrm{c} = \mathrm{flood} \end{cases}$$

$$dt = time[t] - time[t-1]$$

$$A = \begin{bmatrix} 1 & dt & 0.5 \cdot dt^2 \\ 0 & 1 & dt \\ 0 & 0 & 1 \end{bmatrix}$$

- Likelihood

$$p(x_t | z_{c_t, p_t}) = \begin{cases} Uniform(0, x_t), z_{c_t} = \mathrm{rest} \\ \Gamma(\gamma = 0.288, \mu = z_{p_t}, \sigma = 2), z_{c_t} = \mathrm{flood} \end{cases}$$

- Number of samples: 5000

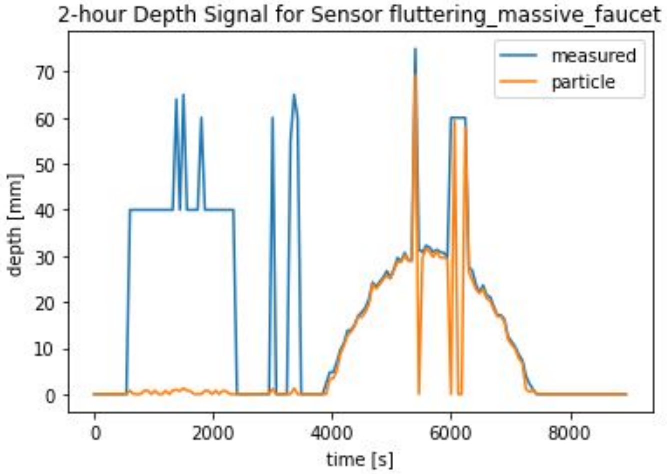

Figure 6. Sample Plot of Noisy Signal Before/After Particle Filtering

Results

Metrics:

$$\text{Sample Level Accuracy} = \frac{\sum_{t=1}^{N} 1_{\text{if data point t is filtered}}}{\sum_{t=1}^{N} 1}$$

$$\text{Event Level Accuracy} = \frac{\sum_{n=1}^{\text{Total Events}} 1_{\text{if data points is filtered > threshold}}}{\sum_{n=1}^{\text{Total Events}} 1}$$

Results:

	Heuristic (Baseline)	GP Regression	Kalman	Particle
Average Sample Level [%]	88.04	29.23	85.99	99.93
Average Event Level [%]	86.14	65.06	84.77	93.99

Table 1. Evaluation Results for Filtering Methods Applied

NYU

Results: Gaussian Process

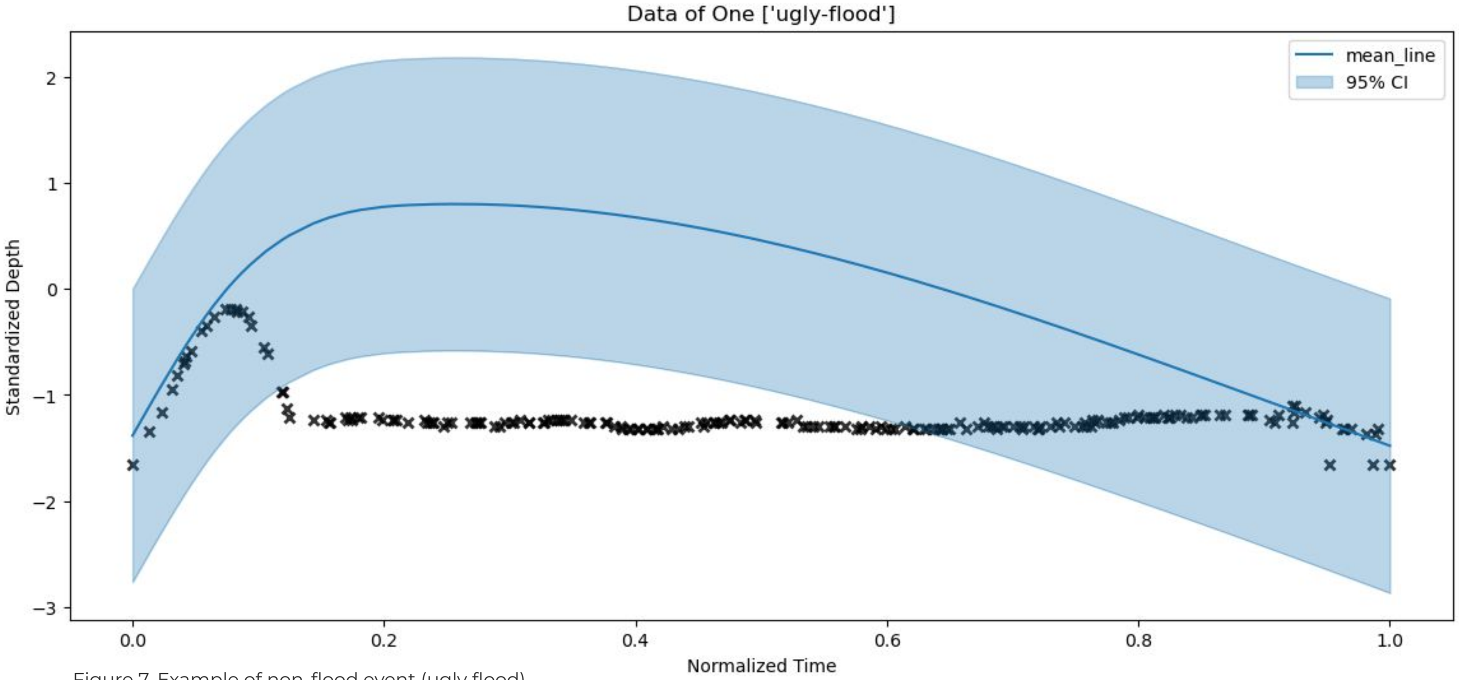

Figure 7. Example of non-flood event (ugly flood)

NYU

Key Learnings

Method Comparison

	Strengths	Weaknesses
Heuristic Filter	- Low computational complexity - Strong capture of known noise signal artifacts	- No flexibility for modeling autoregressive transitions - Poor capture of new/varying noise signal artifacts
Kalman Filter	- Dynamic dt in transition matrix help better estimates - Global parameters reconstruct Individual event signals well - Handles noise events within floods well	- Poor reconstruction of continuous readings due to multimodality of signal
Particle Filter	- Flexibility of observation likelihood definition helps overcome the multimodality limitation of the Kalman filter - Non-gaussian noise and flexible latent state update fits true signal well	- High computational expense; run time ~6x longer than kalman filter - Poor handling of noise events within floods - Computational expense limits further improvement through added complexity to likelihood/state definitions
GP Regression	- Easy to model using combinations of kernels - Accurate modeling of flood	- Difficult to model discrete time varying samples - High variance from modeling all floods together - Detection of non-flood events is challenging due to overlapping data and high variance

Table 2. Method comparison

Conclusion & Next Steps

Conclusion

- GP does not perform well given flood signals and non-flood signals share the similar pattern in the beginning and in the end of the timeseries.
- Kalman filtering works better than GP because it has greater flexibility for flood estimates, but fails to generalize towards multimodality
- Particle filtering works best bc it has the flexibility we need for this application, but likely infeasible for production use due to computational expense

Next Steps

- HMM/IMM to model signals through discrete state transitions
- EKF/UKF to model floods as non-linear state transitions
- Fourier transform to remove undesirable noise
- Time series clustering to find characteristic embedded features

Questions?

NYU