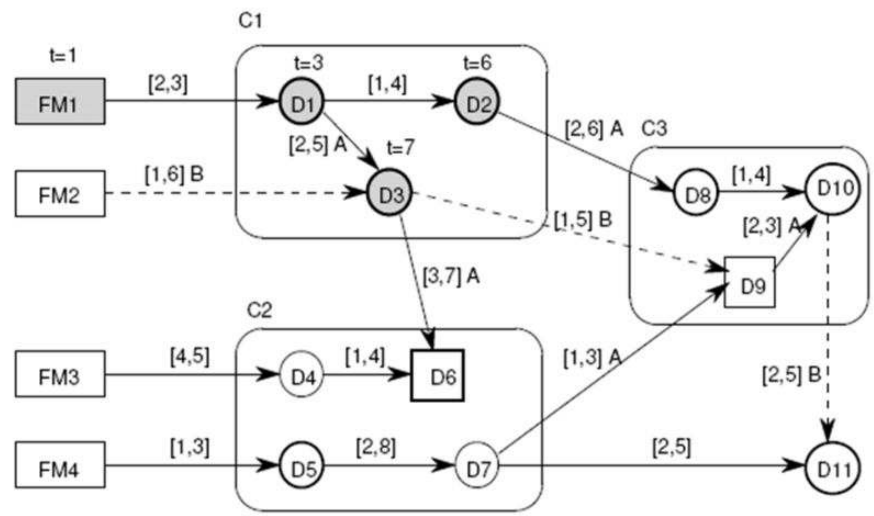
Incident Prediction and Dispatch Response

Abhishek Dubey
Vanderbilt University

Acknowledgement: National Science Foundation

The problem of incident prediction and mitigation response is not unique to Power systems

- Power system network and Transportation networks are graphical cyber-physical systems where
  - There are sources of potential
  - There are sinks of potential.
  - There are junctions that divide the potential effort.
  - The nodes of sources and sinks and efforts are connected through edges (transmission links in power networks) through which there is a flow of energy
  - The value of efforts and flows in different part of the network are correlated with their neighboring values.
  - A discrepancy in such values can be identified as anomaly.
  - An anomaly in any edge can cause cascading faults in other edges.



A logical failure timed propagation model of a graphical CPS

Domain	Effort	Flow	Edge	Junction
Electrical	Voltage	Current	Transmissi on feeder and lines	Relays and Breakers and Bus
Transporta tion	Congestion Factor	Volume rate flow of cars	Road	Intersectio ns
Water Network	Pressure	Volume rate flow	Pipes	Valves

Our approach

- Our approach for working with these graphical networks is to use a combination of data-driven and model-driven efforts to create classifiers to detecting anomalies, performing diagnosis and mitigate cascades.
- I am going to discuss one set of such algorithms from transportation networks today.

Handling Anomalies in Transportation Networks

- There are limited emergency responder resources.
- How to assign resources to incidents.
- Things to consider:
  - Reduce average response time
  - Decision must be made quickly
  - Legal constraints

Motor Vehicle Incidents over five years in the Tennessee region

There is one incident every 10 minutes on average in Nashville

Identifying Anomalies In Real-Time

Roadside Units (RSU): Decentralized traffic management

Definition: Low powered edge devices scattered throughout transportation network.

Provides various computation services for a collection of sensors, can communicate with central cloud and users.

Spare computation capacity on RSU nodes.

Zone: collection of sensors mapped to a single RSU

Each sensor provides real-time traffic reading every 30 second

Figure: Example Zone

Zone Level Anomaly Detection: Q-Ratio

Q-Ratio Definition

$$Q_t = \frac{HM_t}{AM_t}$$

HM_t: Harmonic Mean at time window t
AM_t: Arithmetic Mean at time window t
t: time window (ie 15, 30 minutes)

Detection Threshold

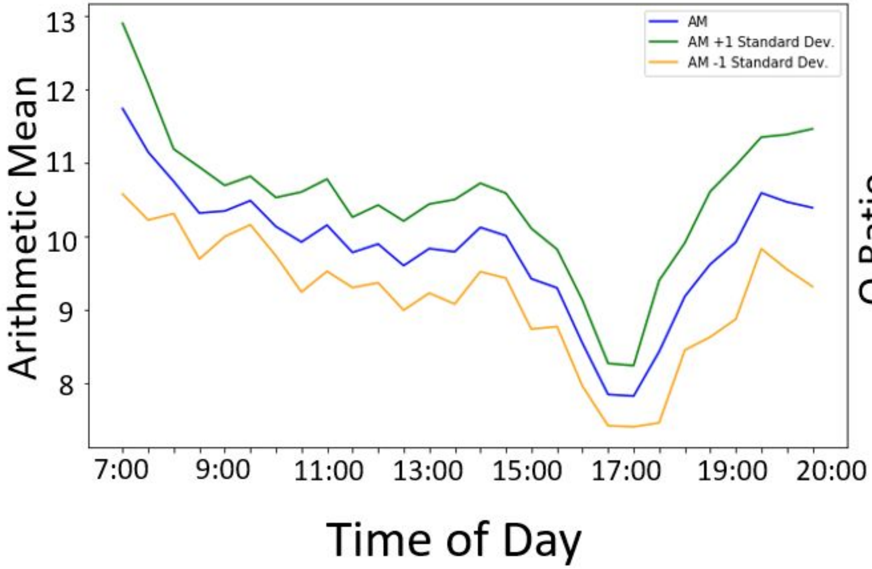
$$Q_t < Q_{h,t}^{mean} - \epsilon * Q_{h,t}^{std}$$

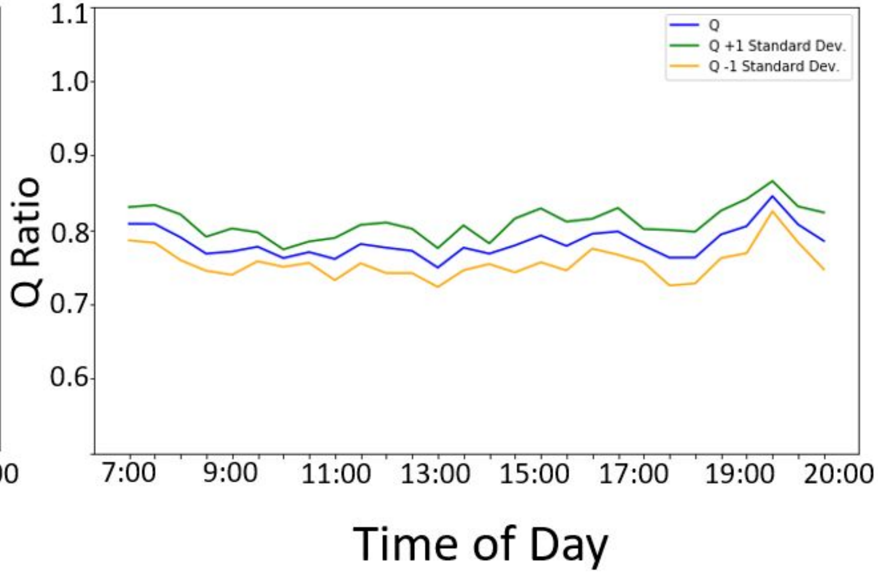
$$Q_t > Q_{h,t}^{mean} + \epsilon * Q_{h,t}^{std}$$

ϵ: Detection threshold

Why Q-Ratio?

- Stable over time
- Invariant to minimal changes in sensor data
- Responds quickly to data-integrity attacks

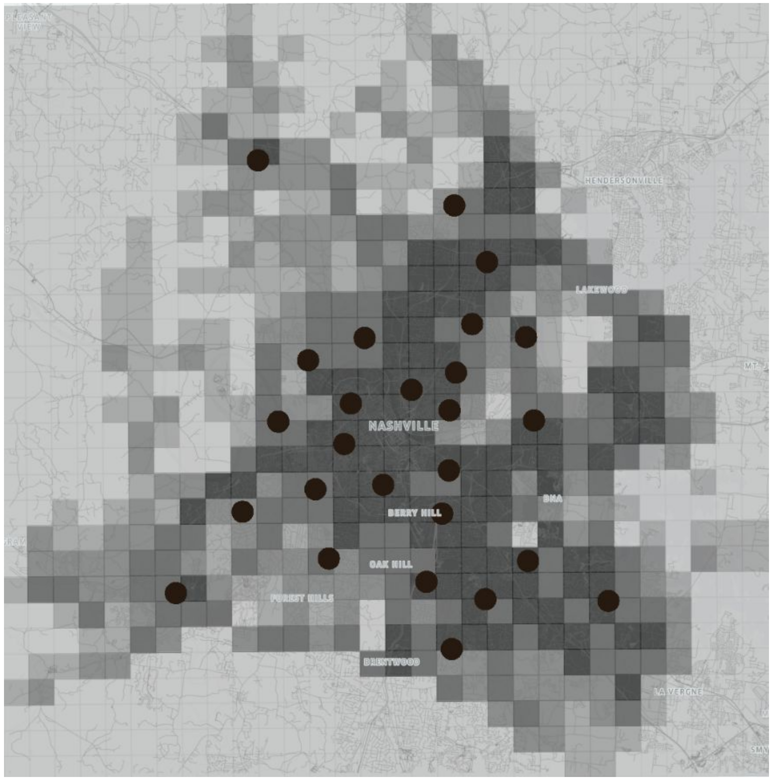




Estimating Future Anomalies

Estimating Anomalies in Transportation Networks

- Given a finite set of grids over a geographical region, and a dataset *D* of time-stamped incidents.
- D : $\{\{x_1, w_1\}, \{x_2, w_2\}, \ldots, \{x_n, w_n\}\}$ where *x* is time of occurrence
- w_i - set of features associated with the i^{th} incident
- $t_i = x_i - x_{i-1}$
- Past rate of incidents, weather condition in the area, speed limit etc.
- **GOAL** : Learn a probability distribution *f(t|w)*



Estimating Anomalies in Transportation Networks

- We use survival analysis - a class of methods to find *inter-arrival* times.

$$\log(t_i) = \sum_j \beta_j w_j + \epsilon$$

- We use **Maximum Likelihood Estimation** to estimate the parameters.

$$L = \prod_i h(\log(t_i) - \bar{\beta}W)$$

- However, accidents often cascade and the survival model has to be updated online.

- Solution
  - Stochastic Gradient Descent

$$h_K(k) = e^{k-e^k}$$

Online Estimation

- Let *D′* represent a stream of new incidents.

$$\beta^{p+1} = \beta^{p} + \alpha \nabla L(\beta^{p}, D^{'})$$

- Assume that β^p is already known.

$$L = \sum_{i=1}^{k} \log\ e^{(\log\tau_i - \beta^* w) - e^{(\log\tau_i - \beta^* w)}}$$

- Our goal is to update β^p to β^{p+1} without re-learning the entire model.

- We take gradient steps for each parameter based on *D′*

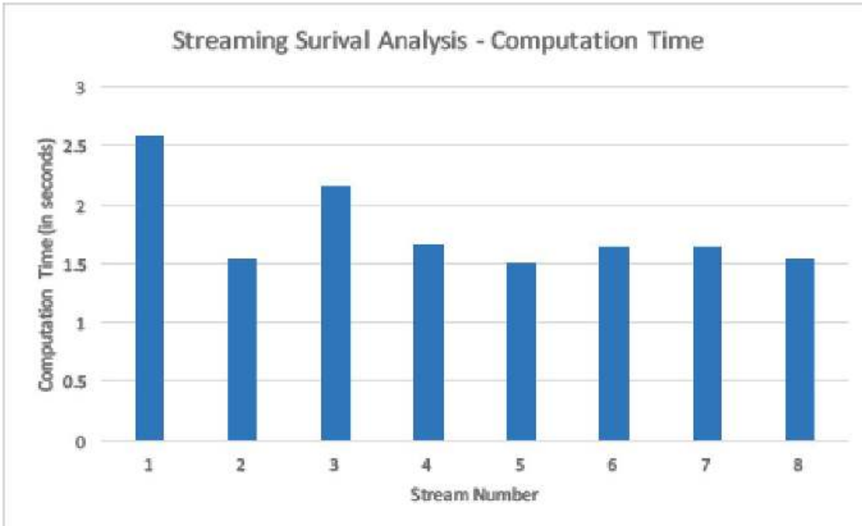
$$\frac{\partial L}{\partial \beta_j} = \sum_{i=1}^{k} -w_{ij} + w_{ij}\{e^{(\log\tau_i - \beta^* w_i)}\}$$

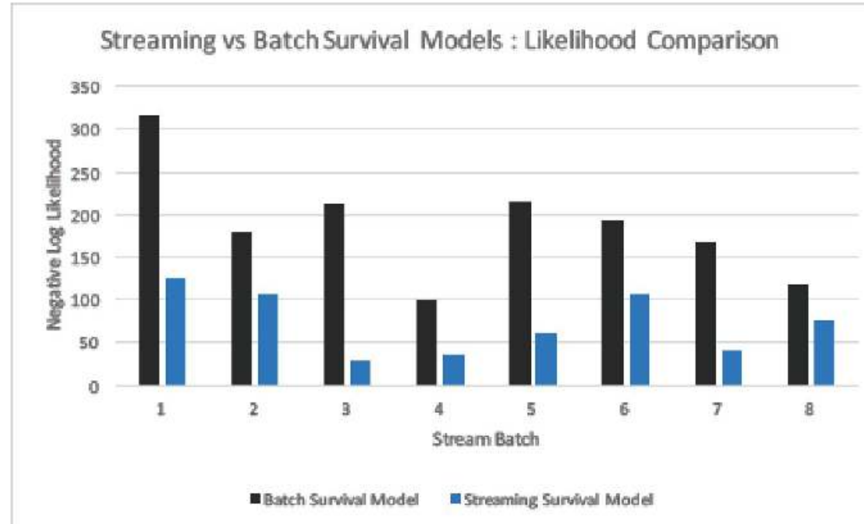
Online Estimation

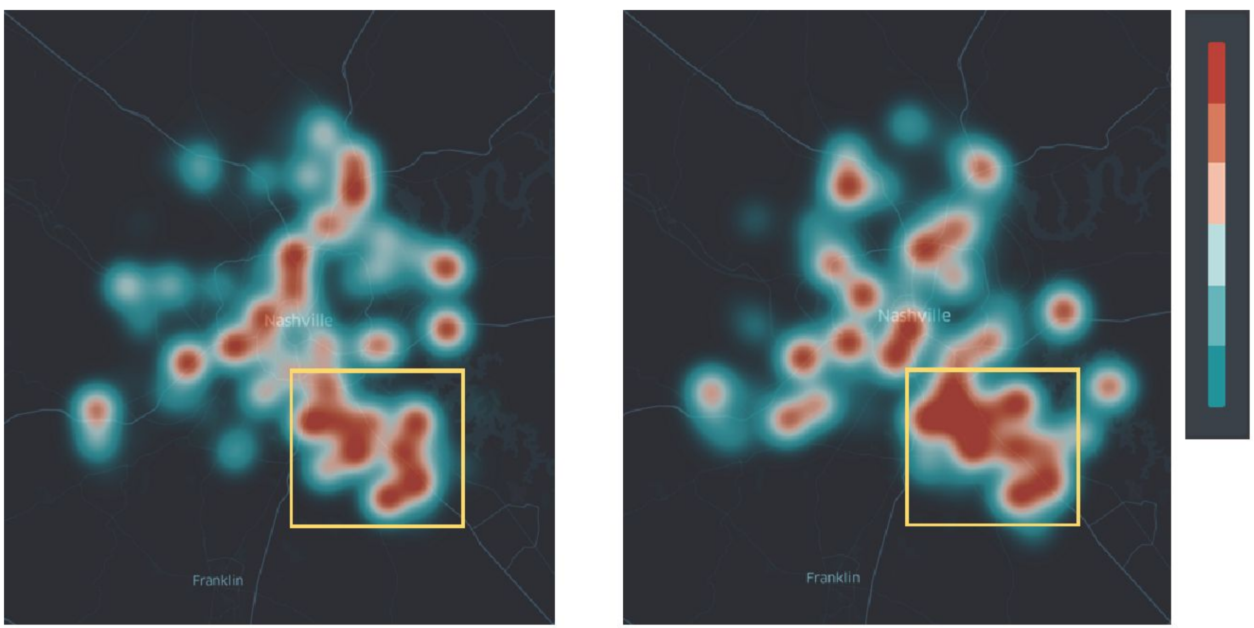
- Let *D′* represent a stream of new incidents.
- Assume that β^p is already known.
- Our goal is to update β^p to β^{p+1} without re-learning the entire model.
- We take gradient steps for each parameter based on *D′*

Feature	**Description**
Time of day	Each day was divided into 6 equal time zones with binary features for each.
Weekend	Binary features to consider whether crime took place on a weekend or not.
Season	Binary features for winter, spring, summer and fall seasons.
Mean Temperature	Mean Temperature in a day
Rainfall	Rainfall in a day
Past Incidents	Separate variables considered for each discrete crime grid representing the number of incidents in the last two days, past week and past month. We also looked at same incident measures for neighbors of a grid.

Evaluation of Incident Prediction Model







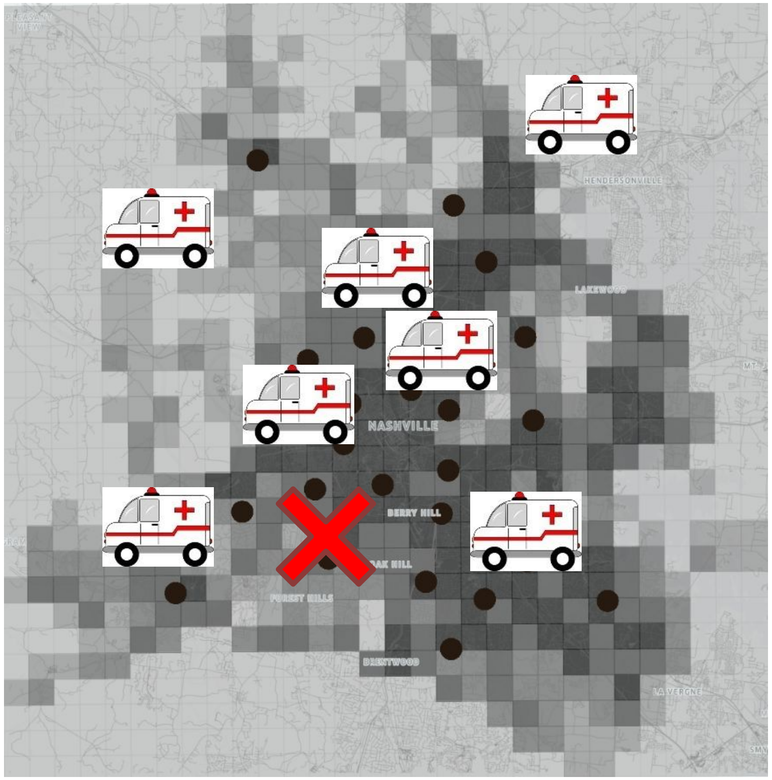
(a)

(b)

Deciding Response

Dispatching Problem

- Given:
  - The location of a pending incident
  - Sampled chains of incidents over time and space
  - A routing and travel time model
  - The status, current locations, and depot locations of all emergency service vehicles
- Find a near optimal resource assignment for the pending incident



State Action Tree: Example

Key

- Root Node
- Candidate Action
- Candidate State
- Stochastic state
- Deterministic State
- Leaf State

t_0

Canidate Actions $\in \rightarrow 3$

- Candidate Action Factor $\in$:

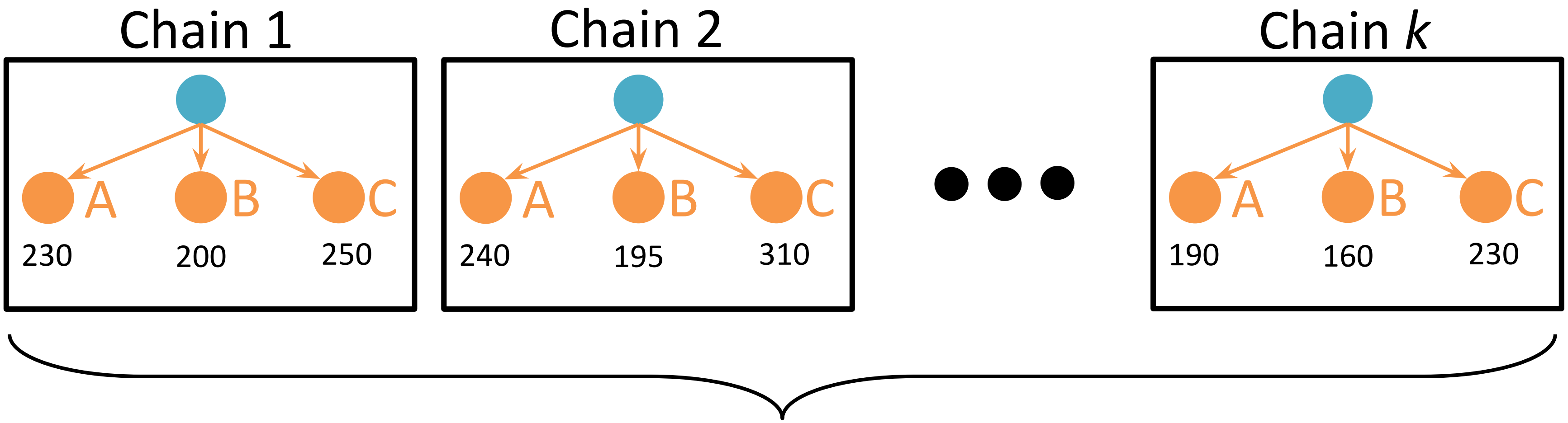
- Stochastic Horizon h^s
  - Controls how deep to explore tree while branching
  - Once reached, we switch into deterministic mode
    - We assume that future rewards are sufficiently discounted
    - Assume heuristic action (send closest responder)
  - In this example $h^s = 3$

- For simplicity, assume 3 candidates for this example

Stochastic Horizon
$h^s = 3$

500 500 200 500 500 500 500 500 500

State Action Tree



A (230 + 240 + … + 190) / k = 223

B (200 + 195 + … + 160) / k = 191

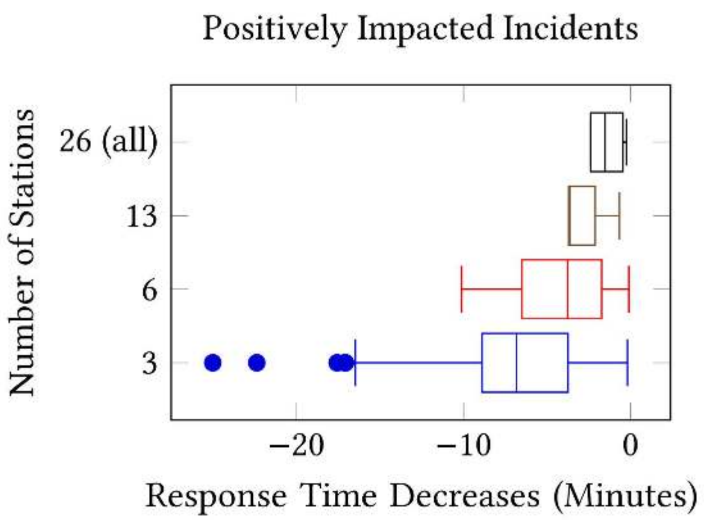
C (250 + 310 + … + 230) / k = 265

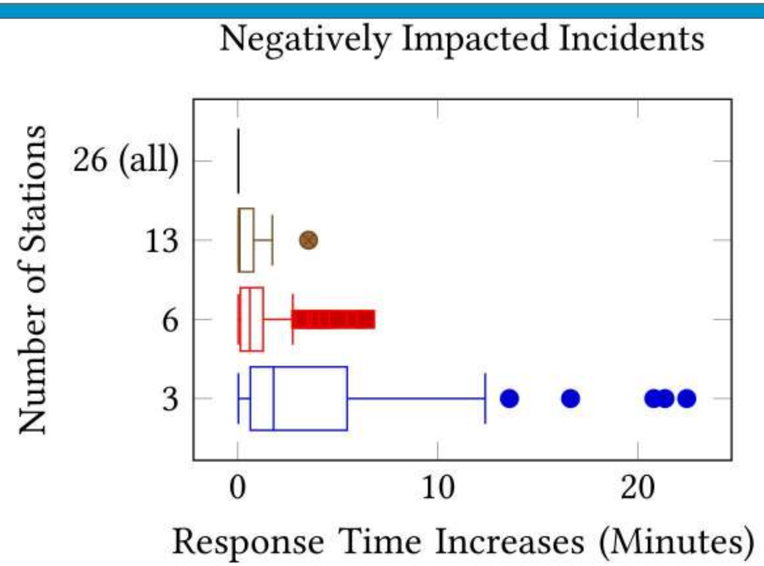
- Average costs for each candidate action across each simulation
- Choose the action with the minimum average cost

Tree Search Hyper Parameters

- Important: efficient exploration of the search space
  - Must run in (soft) real time
- Hyper parameter review:
  - Simulation budget: How many trees to build in parallel to combat on model variance
  - Candidate Action Factor: How many actions to explore at each state within the stochastic horizon
  - Stochastic horizon: how deep in the tree to explore before switching to heuristic actions
  - Discount factor: how much to discount future rewards; ensures rewards go to 0

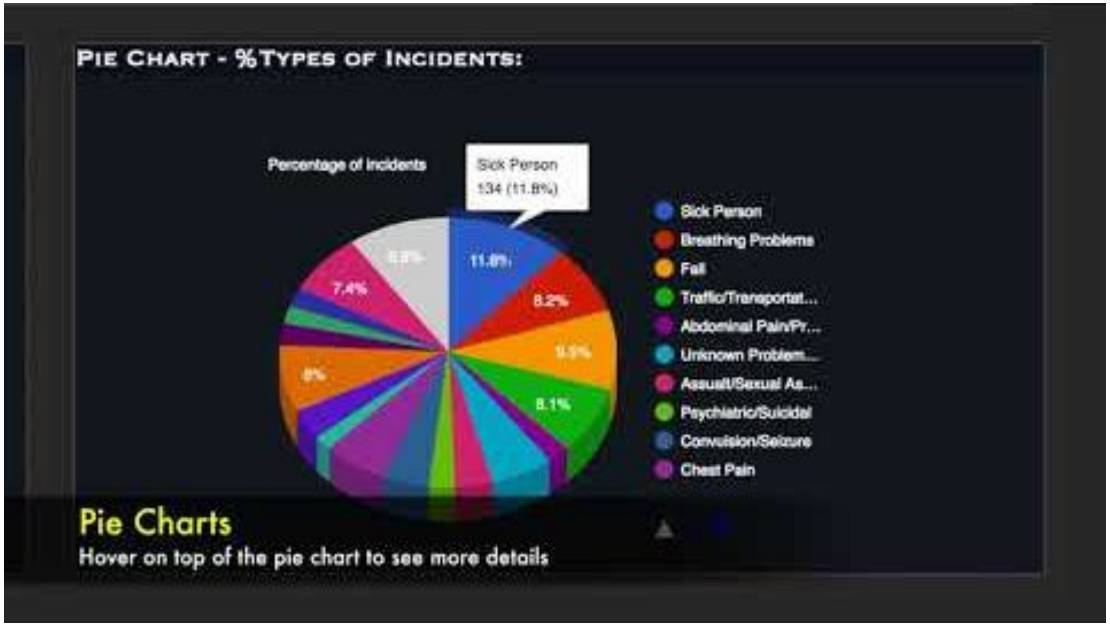
Concluding Remarks





- Traffic accident data from Nashville, TN
  - Training data: 12 months (9345 incidents)
  - Testing data: 2 months (1386 incidents)
- Tradeoff necessary to reduce average response times
  - Some incidents negatively impacted

Dashboard



Integrated Analytics Dashboard is available on github