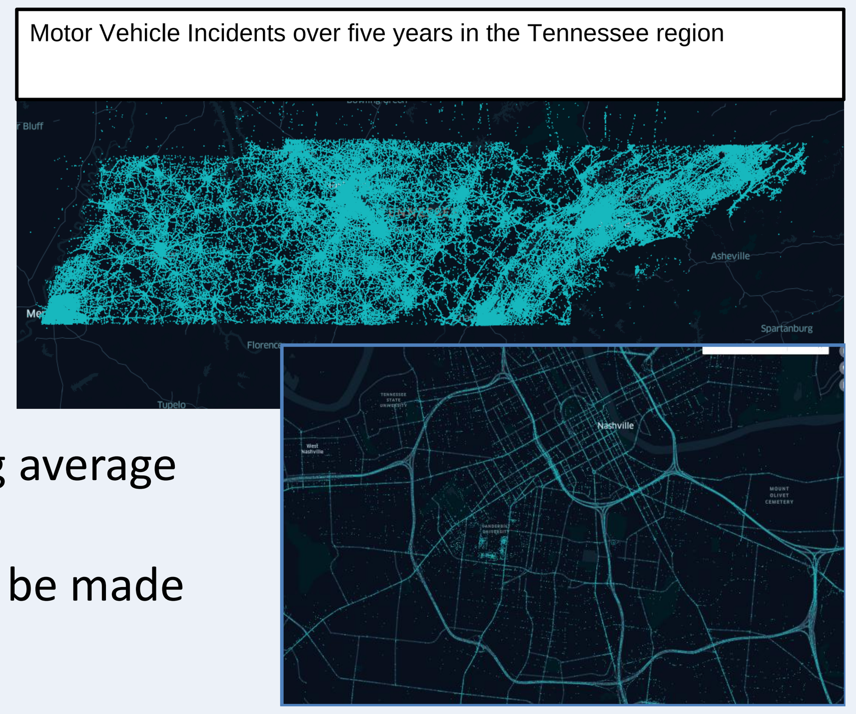
Integrated Safety Incident Forecasting and Analysis

Geoffrey Pettet[1] , Ayan Mukhopadhyay[3],Abhishek Dubey[1], Yevgeniy Vorobeychik[2]
Vanderbilt University[1], Washington University in St. Louis[2] Stanford University[3]

Project: Objective: Develop Principled Algorithmic Decision Procedures for Emergency Response

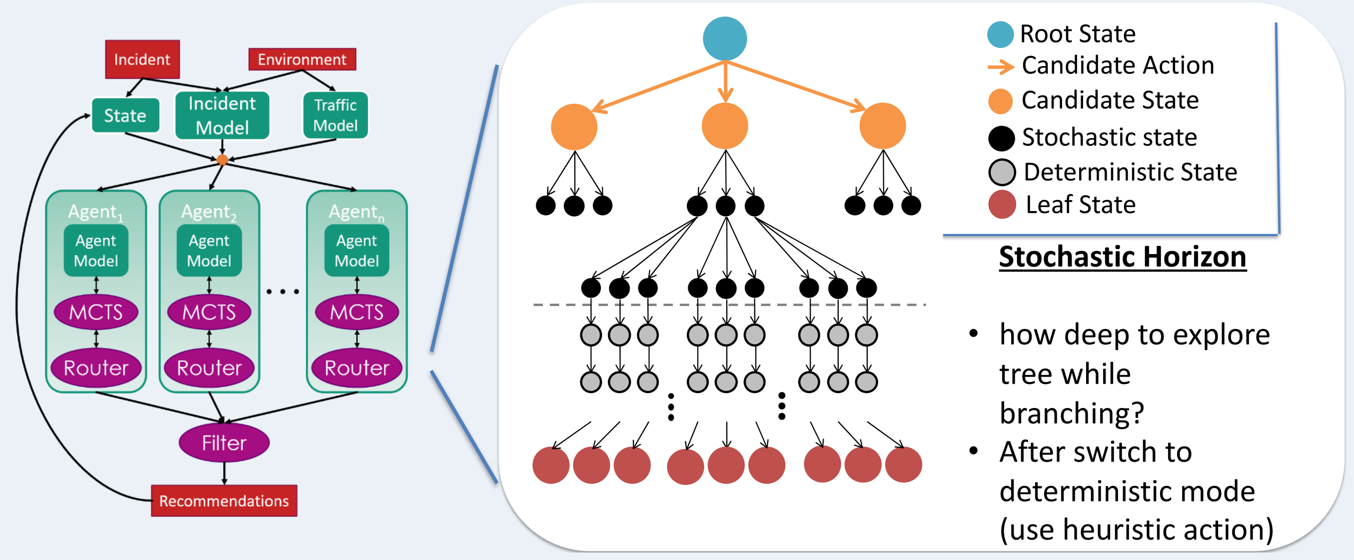
- There are limited emergency responder resources.
- How to assign resources to incidents while reducing average response time
- Decision must be made quickly.



- The planning process should occur before incidents. It is difficult to justify sending anyone but the closest responder at the time of an incident's occurrence.
- Optimizing over responder distribution and response as a multi-objective optimization problem is typically computationally infeasible.
- Example: let the number of responders r=20, and the number of possible depot locations be d=30. Possible actions for dispatching is the number of responders -> **20**
- Possible actions for rebalancing is P(d, r) = 30!/10! = **7.31×10^{25}**.

Our Approach: Partially Decentralized Decision Process

- We focus on three problems (a) designing an accurate incident prediction model; (b) design approach for rebalancing the responders pre-incident and (c) designing an emergency response system that is equipped to deal with scenarios that require decentralized planning with very limited communication.



- Our approach is based on Multi-Agent Monte-Carlo Tree Search.
- Rather than building a monolithic, large search tree exploring all possible system states, each agent builds an individual tree focusing on the subset of actions relevant to them – i.e. their rebalancing action
- Reduces the number of states from P(d,r) to just the number of depots d for each agent.

__Reward function:__ The primary metric to consider is the response time for each incident
Secondly, the movement of responders needs to be controlled

$$r_s^a = \begin{cases} r_{s-1} - \alpha^{t_s}(t_s^a), & \text{if responding to an incident} \\ r_{s-1} - \alpha^{t_s}\psi\frac{\sum_{\lambda_k \in \Lambda}(d_a^k)}{|\Lambda|}, & \text{if balancing at } s \end{cases}$$

Online Incident Prediction

- ***Features:*** Weather, time, previous incidents, neighboring incidents
- Needs to react to dynamic incident occurrence
- ***Streaming survival analysis:***

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

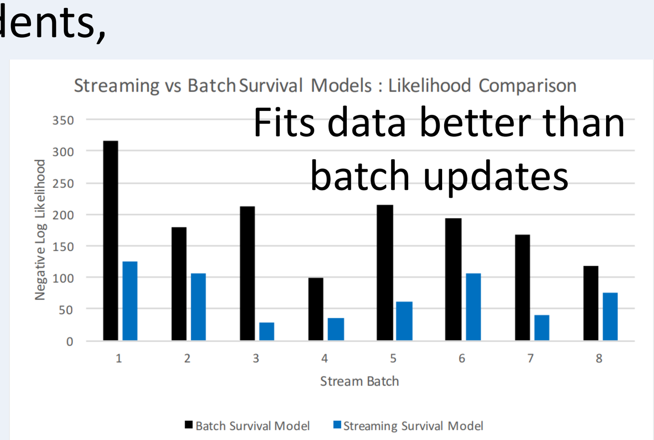
Probabilistic Model for Incident Prediction

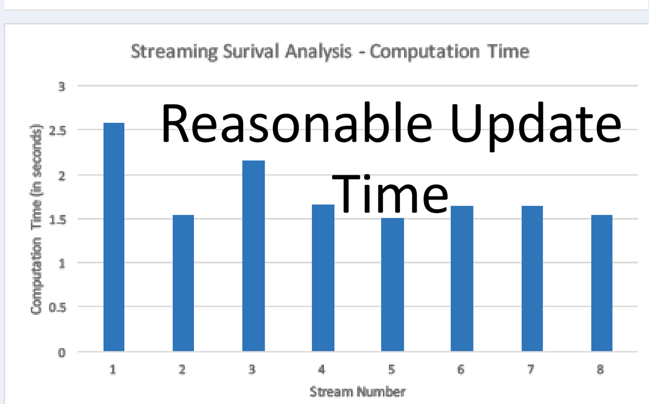
$$\beta^{p+1} = \beta^p + \alpha\nabla L(\beta^p, D')$$

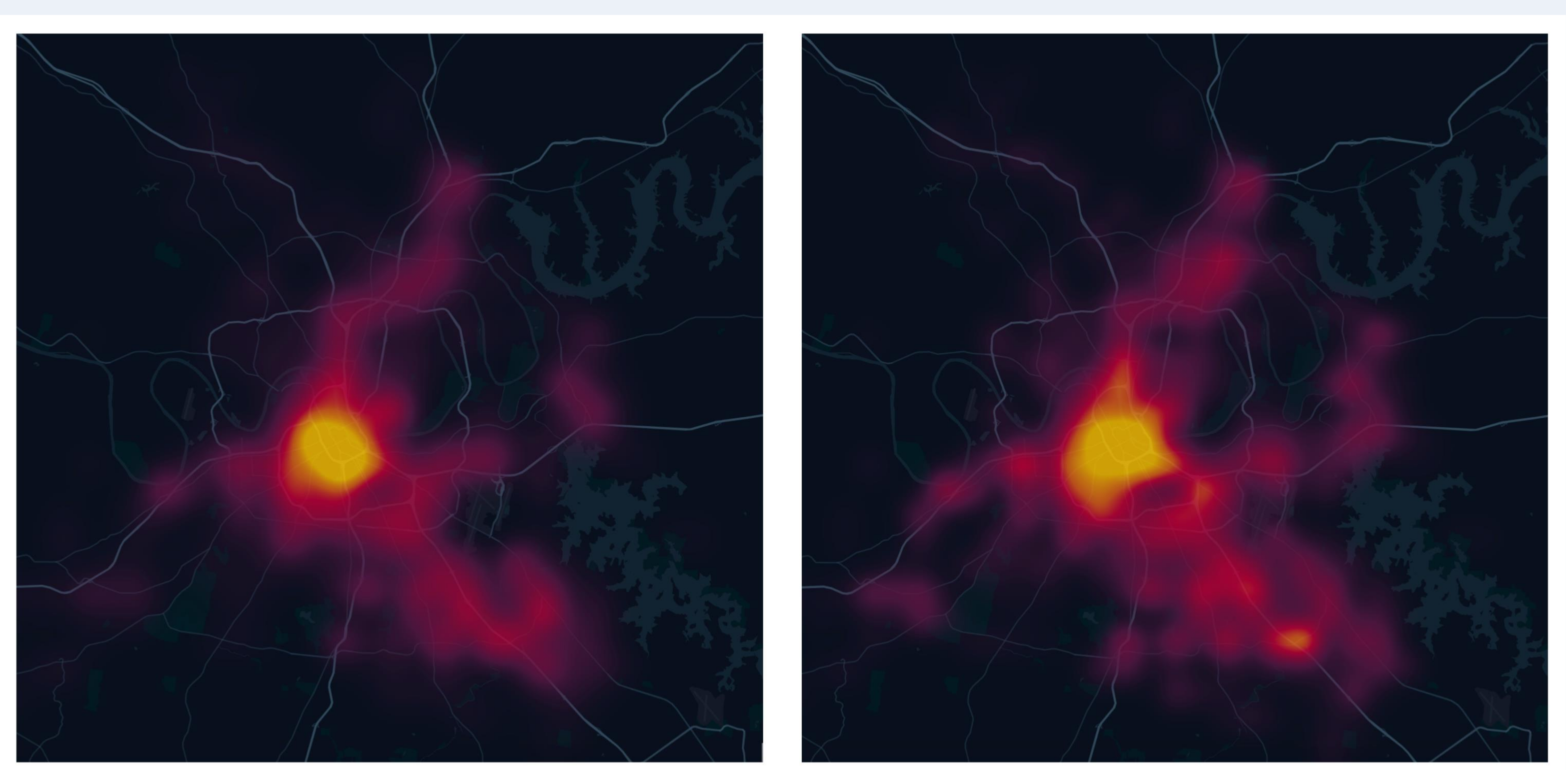
Online Update of Coefficients

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

Gradient Calculation







Comparison of (1) incidents predicted by model (left), and (2) real incident distribution (right) over January 2019

Preemptive Rate Based Rebalancing

- We use a multi-class queue model to enable the responders to anticipate the action of other responders.
- Multiple cells serviced by each depot and vice versa
- Must split request rate for cells between depots
- Since depots closer to a cell are more likely to service it, rates are split such that they are inversely proportional to the distance

$$\sum_{d \in D} v_g^d = v_g$$

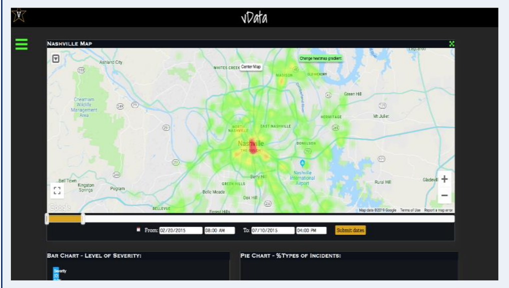
$$\text{dist}(d_1, g)v_g^{d_1} = \text{dist}(d_2, g)v_g^{d_2} \quad \forall d, d_2 \in D$$

$$\pi = \sum_{d \in D}\sum_{g \in G} responseTime(c, v_g^d, \mu) + travelTime(d, g)$$

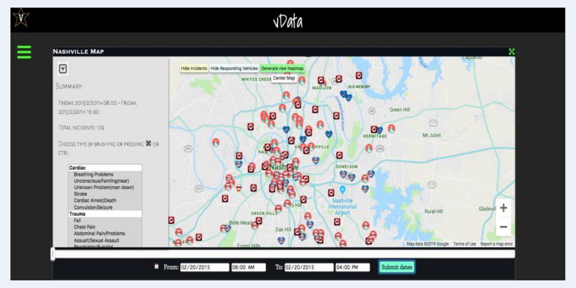
Where *c* is number of responders at depot, *v* is split incident rate, and *μ* is the service rate. ResponseTime() is M/M/c queue response time, travelTime() is the time to travel from depot *d* to the grid in question *g*

- To score a particular assignment of responders to depots, all response times (computed from multi class queue formulation) are summed using the split rates.
- To determine the best placement, an iterative greedy search is used to select the best depots one at a time using the above score

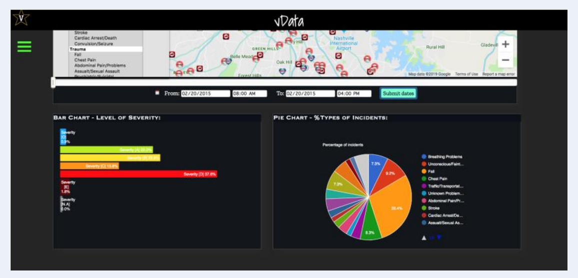
Integrated Dashboard



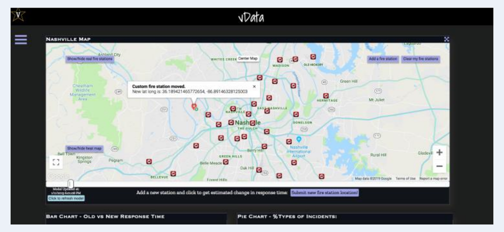
Incident heatmap



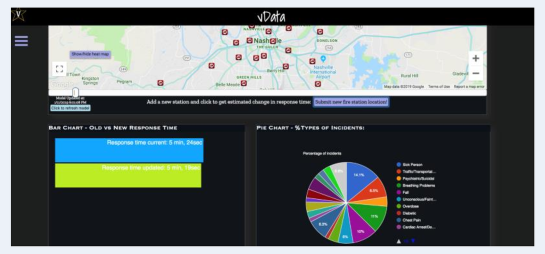
Incident Queries



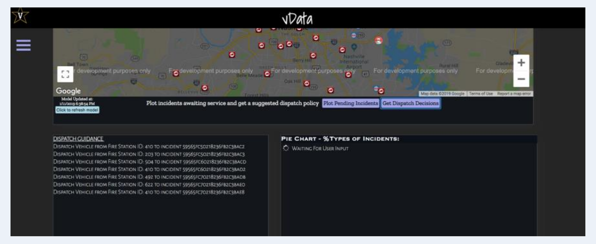
Incident Statistics



What if analysis? Visualize and relocate Fire stations



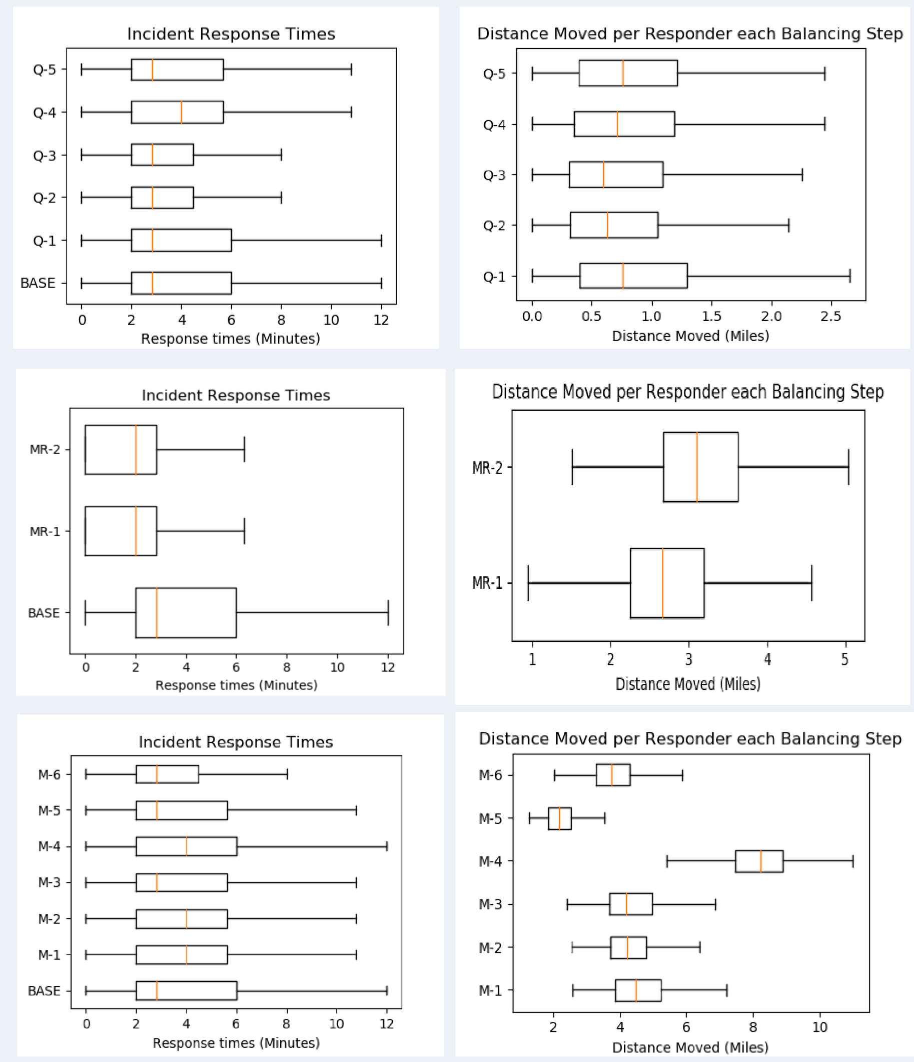
Expected Response Time Change Due to Relocated or New Depots



Suggested Dispatch Decisions. Future Work (Integrate Rebalancing)

Results

Identifier	Description	Hyper-Parameter Choices
BASE	Greedy Baseline Without Rebalancing	N/A
Q-1	Queue Based Rebalancing Policy with RoI of 1	RoI = 1
Q-2	Queue Based Rebalancing Policy with RoI of 2	RoI = 2
Q-3	Queue Based Rebalancing Policy with RoI of 3	RoI = 3
Q-4	Queue Based Rebalancing Policy with RoI of 4	RoI = 4
Q-5	Queue Based Rebalancing Policy with RoI of 5	RoI = 5
MR-1	MMCTS - using an oracle for future incidents and a Static Agent Policy	*Same as MMCTS Baseline M-1*
MR-2	MMCTS - using an oracle for future incidents and a Queue Rebalancing Policy	*Same as MMCTS Baseline M-1*
M-1	MMCTS - Baseline The foundation for the parameter search. Each parameter varies independently while other parameters retain these values. (All M-* experiments use generated incident chains and a Static Agent Policy)	MCTS Iteration Limit = 250 Lookahead Horizon = 120 min Reward Distance Weight ψ = 10 Reward Discount Factor = 0.99995 Rebalance Period = 60 min
M-2	MMCTS - Iteration Limit of 100	MCTS Iteration Limit = 100*
M-3	MMCTS - Iteration Limit of 500	MCTS Iteration Limit = 500*
M-4	MMCTS - Reward Distance Weight ψ of 0	Reward Distance Weight ψ = 0*
M-5	MMCTS - Reward Distance Weight ψ of 100	Reward Distance Weight ψ = 100*
M-6	MMCTS - Rebalance Period of 30 minutes; Lookahead Horizon of 30 minutes	Lookahead Horizon = 30 min Rebalance Period = 30min*



- Data: Nashville, TN incident data
- Training for predictive model: 1-1-2018 to 1-1-2019
- Testing: 1-1-2019 to 2-1-2019
- Utilized Regions of Interest (RoI) for queue model: Only depots within a cell's RoI are considered when splitting its rate
- Encourages even responder distribution
- Reduces computation time
- Explored several parameters for MMCTS, particularly the distance reward weight and iteration limit for MCTS
- Compared to the incumbent policy of greedy dispatch without rebalancing as a baseline
- An Oracle refers to an incident predictor with perfect knowledge about future incidents (best case scenario for MCTS)

__MMCTS performs better than greedy baseline with most parameters__
__Oracle solutions show the potential upside. It requires an even better incident forecasting model.__

This work is sponsored by The National Science Foundation under award number CNS1640624
In collaboration with Nashville Metro Information Technology Services and Nashville Fire Department
https://github.com/scope-lab-vu/DataDrivenEMSDispatch.