Emergen | Sim

Data Analysis and Simulation Modeling for Optimizing Urban Emergency Response Systems

By Surya Chand Rayala

The Dataset

overview of the project

- This project focuses on creating a data driven Emergency Response Environment, an advanced simulation framework designed for enhancing urban emergency response tactics. By leveraging a comprehensive dataset, I have developed a dynamic 30x30 grid simulator. This tool not only incorporates data analysis and probabilistic modeling to realistically simulate incidents but also tracks various response strategies and assigns rewards based on their effectiveness.

The Dataset

- The dataset given consists of two columns — 'Grid Cell' and 'Timestamp'.
- This dataset represents incidents recorded in different grid cells, with each entry comprising a grid cell identifier and a timestamp.

	Grid Cell	Timestamp
0	172	2019-03-27 10:40:12.529245
1	226	2019-03-27 10:16:42.028122
2	228	2019-03-27 10:55:01.115513
3	233	2019-03-27 10:20:15.934103
4	242	2019-03-27 08:50:32.356951

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 203215 entries, 0 to 203214
Data columns (total 2 columns):
 #   Column     Non-Null Count   Dtype 
---  ------     --------------   ----- 
 0   Grid Cell  203215 non-null  int64 
 1   Timestamp  203215 non-null  object
dtypes: int64(1), object(1)
memory usage: 3.1+ MB

(None,
             Grid Cell
 count  203215.000000
 mean      393.636336
 std       122.901614
 min        23.000000
 25%       345.000000
 50%       405.000000
 75%       406.000000
 max       887.000000,
    Grid Cell                   Timestamp
 0        172  2019-03-27 10:40:12.529245
 1        226  2019-03-27 10:16:42.028122
 2        228  2019-03-27 10:55:01.115513
 3        233  2019-03-27 10:20:15.934103
 4        242  2019-03-27 08:50:32.356951)
```

Transforming Data into Insights (Pre-Processing)

- Convert the 'Timestamp' column to datetime format and extract features like `DayOfWeek`, `Hour`, `Date`. Then Defined and applied a function `categorize_time_of_day` to classify hours into 'Morning', 'Afternoon', 'Evening', or 'Night'..

	Grid Cell	Timestamp	DayOfWeek	TimeOfDay	Hour	Date
0	172	2019-03-27 10:40:12.529245	Wednesday	Morning	10	2019-03-27
1	226	2019-03-27 10:16:42.028122	Wednesday	Morning	10	2019-03-27
2	228	2019-03-27 10:55:01.115513	Wednesday	Morning	10	2019-03-27
3	233	2019-03-27 10:20:15.934103	Wednesday	Morning	10	2019-03-27
4	242	2019-03-27 08:50:32.356951	Wednesday	Morning	8	2019-03-27

Calculating Incident Probabilities

- For every grid cell the probability for an incident to occur is calculated by using the Formula:

$$P(\text{Specific Incident}|\text{Total dataset}) = \frac{\text{Incidents in specific Grid Cell, DayOfWeek, and TimeOfDay}}{\text{Total incidents in all Grid Cells}}$$

Completed Dataframe(Ready for the Environment)

- Calculated incident probabilities for each grid cell across all times and days, and interpolated grid cells with no events. The result is a concise Incident Probability DataFrame, highlighting incident likelihoods across the grid.

Grid Cell	Friday Morning	Friday Night	Monday Afternoon	Monday Evening	Monday Morning	Monday Night	Saturday Afternoon	Saturday Morning	Sunday Afternoon	Sunday Evening	...
0	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	...
1	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	...
2	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	...
...	...	...	...	...	...	...	...	...	...	...	...
23	0.049180	0.049180	0.016393	0.032787	0.065574	0.065574	0.049180	0.032787	0.016393	0.032787	...
24	0.044118	0.088235	0.044118	0.058824	0.029412	0.073529	0.058824	0.014706	0.029412	0.029412	...
...	...	...	...	...	...	...	...	...	...	...	...
898	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	...
899	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	...

900 rows × 28 columns

The Environment

State and Action Space

- **State Space:**
  - Represented as a vector mapping the grid.
  - Each element denotes the severity level of incidents at specific locations.

- **Action Space:**
  - Defined by the grid cell numbers for each available resource.
  - Decisions involve choosing which grid cell each resource should be deployed to.

Resource Characteristics

- **Realistic Resource Constraints:**
    - The simulation incorporates realistic limitations for each type of emergency resource (e.g., ’A’, ’B’).
    - These constraints include operational ranges (effective distances from their stations) and specialization in handling specific severity levels of incidents.

Initializing the Environment

The simulation is influenced by the following key inputs:

1. The Incident probabilities dataset in the aforementioned format.
2. time chunk (e.g., morning, afternoon) and day of the week.
3. configurations of emergency resources, each with unique types and initial stationing.

Process of Incident Simulation

1. **Decision Cycle Initiation:**
    a. Each simulation Episode starts with a decision cycle to generate incidents.

2. **Random Sampling for Incidents:**
    a. Random values are sampled for each grid cell.

3. **Comparing Against Probabilities:**
    a. These values are compared with predefined probabilities, reflecting historical incident patterns and temporal variations.

4. **Incident Generation:**
    a. If a sampled value is below the grid cell's threshold probability, an incident is generated in that cell.

5. **Severity Level Determination:**
    a. The severity of each incident is assigned randomly, based on predefined likelihoods of different severity levels.

6. **Outcome:**
    a. This process results in a realistic and varied set of incidents, simulating the unpredictability of real-world emergency situations.

Resource Movement and Incident Response

- Resources are moved to the grid cells as per the actions taken. For each resource, the code checks if it's moved to a cell with an active incident.

- If a resource encounters an incident, the code evaluates whether the resource type is capable of addressing the incident based on its type and severity.

- The simulator also calculate the travel time for each resource to reach its designated grid cell. This as of now is based on Manhattan distance.

Reward Function

$$\text{Reward} = -\sum(\text{Severity of Unaddressed Incidents}) + \sum_{r \in \text{Responded}} (\text{Severity}_r \times (1 - \frac{\text{TravelTime}_r}{\text{Max Permissible Response Time}}))$$

Outputs of Sample Simulation Run

In the given simulation configuration, ten ambulances, categorized as either type 'A' or 'B' and assigned specific station coordinates, are integrated into the environment.

```python
# Create an instance of the environment
resource_types = ['A', 'B', 'A', 'B', 'A', 'B', 'A', 'B', 'A', 'B']
stations = [(5, 5), (10, 10), (15, 15), (20, 20), (25, 25), (5, 15), (10, 25), (15, 5), (20, 15), (25, 25)]
env = EmergencyResponseEnv(interpolated_data, resource_types, stations)

# Number of steps for the simulation
num_steps = 1  # or more, as needed

# Run the simulation
env.reset(time='Afternoon', day='Saturday')
```

```
----- Step 1 -----
Simulated Incidents Incidents with Severity Levels:
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0
 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 2 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 2 0 0 0]
 Action taken:
[378, 285, 446, 556, 443, 393, 64, 363, 764, 157]
Incidents with Severity Levels after action:
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0
 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 2 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 2 0 0 0]
Reward: -27.95
Metrics after simulation:
Total Incidents responded: 2
Total Cumulative Time for responses: 41
Total Incidents Recorded: 21
Average Response Time: 20.5
Resolution Rate: 0.09523809523809523
```

Applying Reinforcement Learning

Configured Environment to work with PPO (Proximal Policy Optimization)

PPO Learning

```
------------------------------------------
| rollout/                |              |
|    ep_len_mean          | 1            |
|    ep_rew_mean          | -30.8        |
| time/                   |              |
|    fps                  | 98           |
|    iterations           | 2            |
|    time_elapsed         | 41           |
|    total_timesteps      | 4096         |
| train/                  |              |
|    approx_kl            | 0.0009378684 |
|    clip_fraction        | 0            |
|    clip_range           | 0.2          |
|    entropy_loss         | -68          |
|    explained_variance   | 0            |
|    learning_rate        | 0.0003       |
|    loss                 | 208          |
|    n_updates            | 10           |
|    policy_gradient_loss | -0.0211      |
|    value_loss           | 737          |
------------------------------------------
```

```
------------------------------------------
| rollout/                |              |
|    ep_len_mean          | 1            |
|    ep_rew_mean          | -29.5        |
| time/                   |              |
|    fps                  | 94           |
|    iterations           | 3            |
|    time_elapsed         | 64           |
|    total_timesteps      | 6144         |
| train/                  |              |
|    approx_kl            | 0.0016615719 |
|    clip_fraction        | 0            |
|    clip_range           | 0.2          |
|    entropy_loss         | -68          |
|    explained_variance   | 0            |
|    learning_rate        | 0.0003       |
|    loss                 | 130          |
|    n_updates            | 20           |
|    policy_gradient_loss | -0.0225      |
|    value_loss           | 494          |
------------------------------------------
```

```
------------------------------------------
| rollout/                |              |
|    ep_len_mean          | 1            |
|    ep_rew_mean          | -29          |
| time/                   |              |
|    fps                  | 92           |
|    iterations           | 4            |
|    time_elapsed         | 88           |
|    total_timesteps      | 8192         |
| train/                  |              |
|    approx_kl            | 0.0049760696 |
|    clip_fraction        | 0.00474      |
|    clip_range           | 0.2          |
|    entropy_loss         | -68          |
|    explained_variance   | 0            |
|    learning_rate        | 0.0003       |
|    loss                 | 235          |
|    n_updates            | 30           |
|    policy_gradient_loss | -0.0324      |
|    value_loss           | 645          |
------------------------------------------
```

```
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 1           |
|    ep_rew_mean          | -28.7       |
| time/                   |             |
|    fps                  | 91          |
|    iterations           | 5           |
|    time_elapsed         | 111         |
|    total_timesteps      | 10240       |
| train/                  |             |
|    approx_kl            | 0.019258928 |
|    clip_fraction        | 0.165       |
|    clip_range           | 0.2         |
|    entropy_loss         | -68         |
|    explained_variance   | 0           |
|    learning_rate        | 0.0003      |
|    loss                 | 222         |
|    n_updates            | 40          |
|    policy_gradient_loss | -0.0789     |
|    value_loss           | 674         |
-----------------------------------------
```

```
Out[56]: <stable_baselines3.ppo.ppo.PPO at 0x2a5beb250>
```

Configuration of the environment

```
In [17]: # Create an instance of the environment
resource_types = ['A', 'B', 'A']
stations = [(5, 5), (10, 10), (15, 15)]
env = EmergencyResponseEnv(interpolated_data, resource_types, stations)
env.reset(time='Afternoon', day='Saturday')
```

Predicting From Saved PPO Model

```
# Assuming 'obs' is your current environment observation
obs = env.reset()
action, _states = model.predict(obs, deterministic=True)
print(f"Optimal action for the given observation: {action}")
```

```
Optimal action for the given observation: [[182 313 774]]
```

Future Work

- Attempt to enhance the learning process by increasing the number of steps per episode and allocating more resources. Concentrate on a subset of grid cells instead of all 900 and assess the model's performance.

- If, despite these adjustments, the improvement at each time step is still not satisfactory, it will be necessary to revisit and modify the reward function and other constraints that were initially defined.