# Error Injection Robustness Classifier

Joshua Huang
joh009@ucsd.edu

Marlon Garay
mjgaray@ucsd.edu

Mentor: Babak Salimi
bsalimi@ucsd.edu

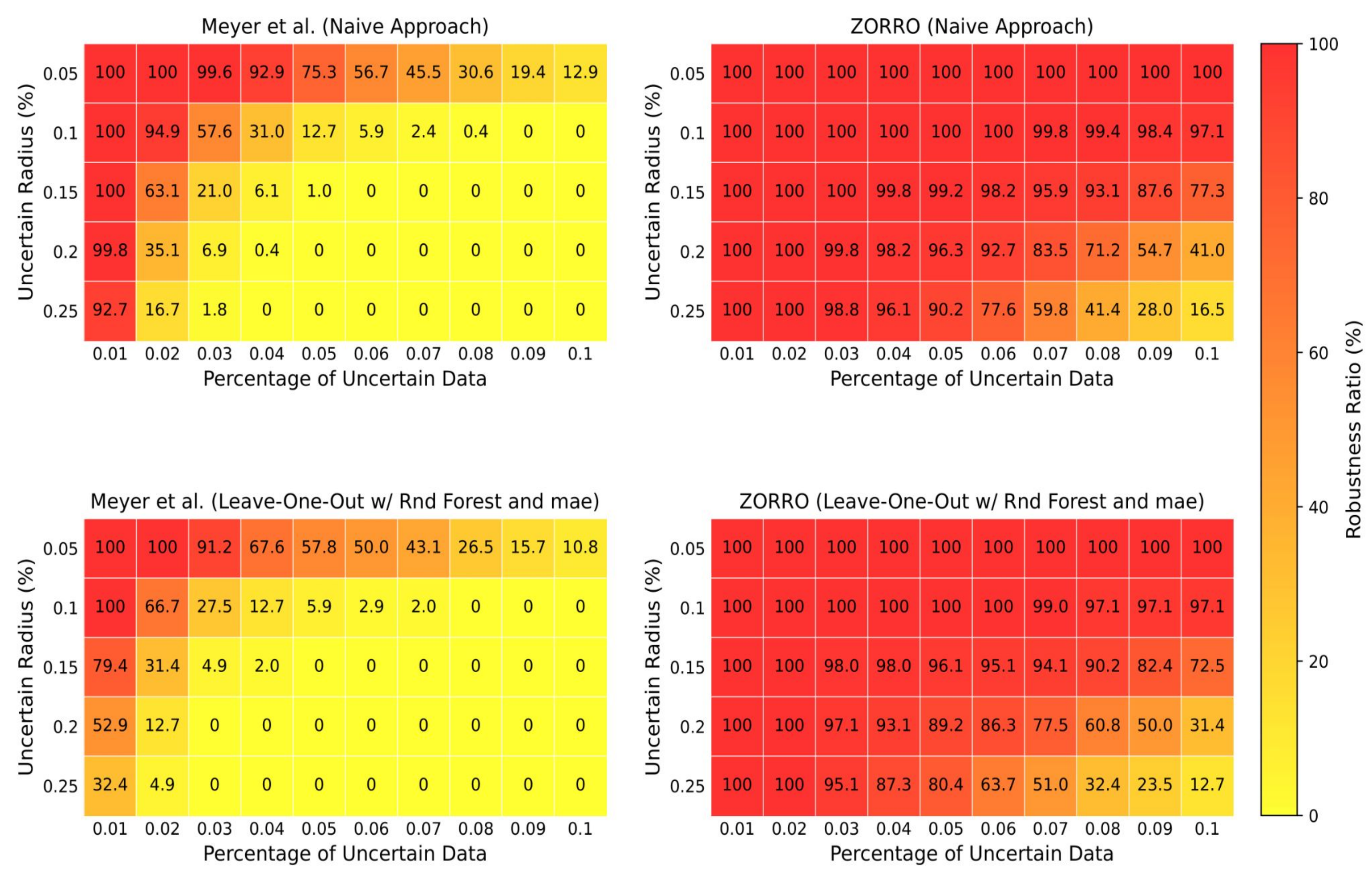
UC San Diego
HALICIOĞLU DATA SCIENCE INSTITUTE

## 1. Introduction

- ZORRO effectively preserves dataset robustness under random error injection attacks
- Gopher is a method that finds the best dataset representation out of all possible feature patterns
- We use Gopher to identify the worst-case error injection, then apply ZORRO to measure dataset robustness under these conditions.
- This returns value that can represent the dataset's worst case robustness, allowing us to identify how reliable it is!

## 2. Leave One Out

- We attempted different approaches to capturing the best "important" indices to a dataset since we theorized they'd be most useful for the worst case error injection
    - The first method to find the best set of target indices was the leave one out approach
- Naive Approach is Random Sampling Error
- Below LOO method utilizes a model of RandomForestRegressor and a metric of Mean Absolute Error



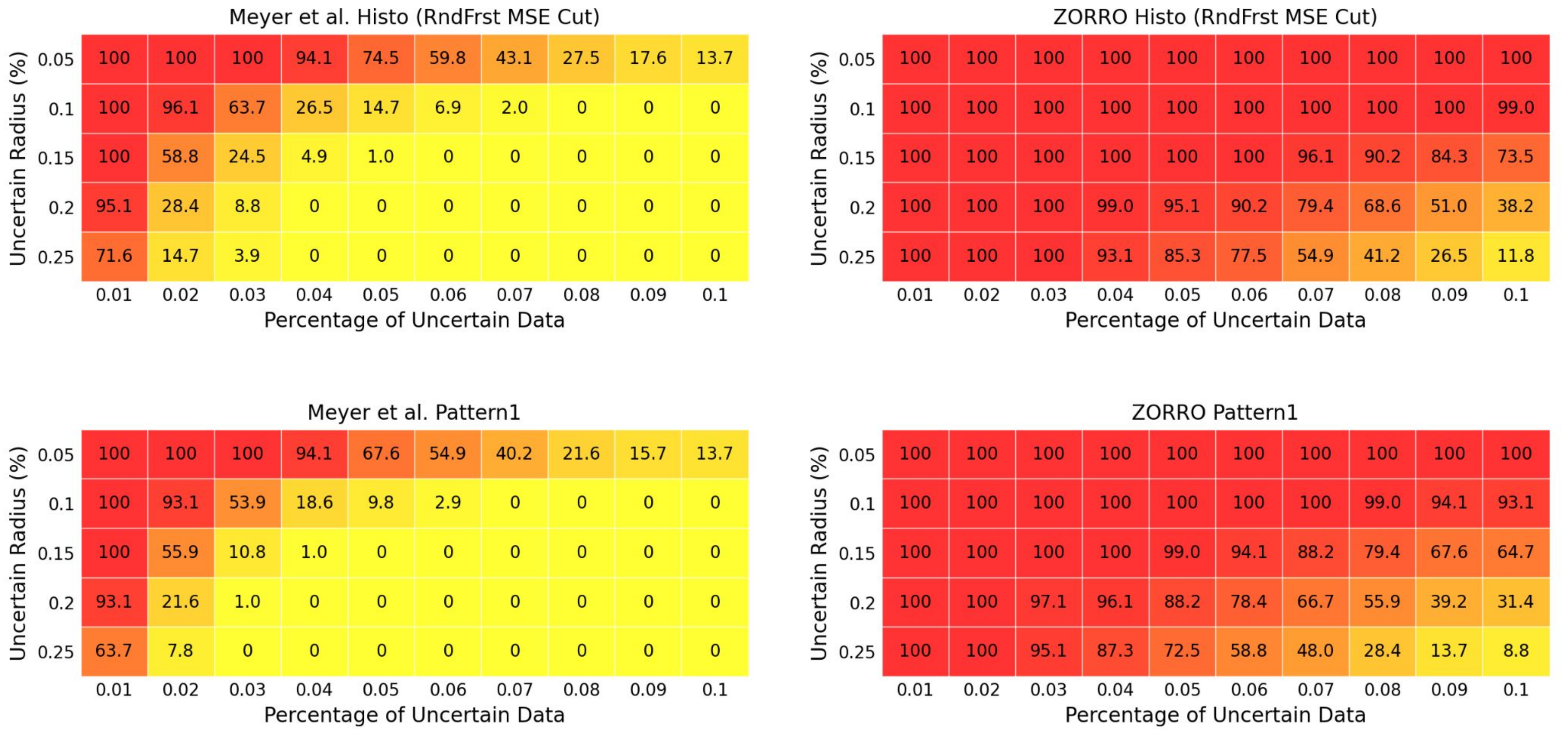
## 4. Pattern Mining

- Finally we utilized a pattern mining approach to discover all possible patterns for a given dataset before filtering out for the best patterns for error injection.
- We compare our results against the histogram method below:

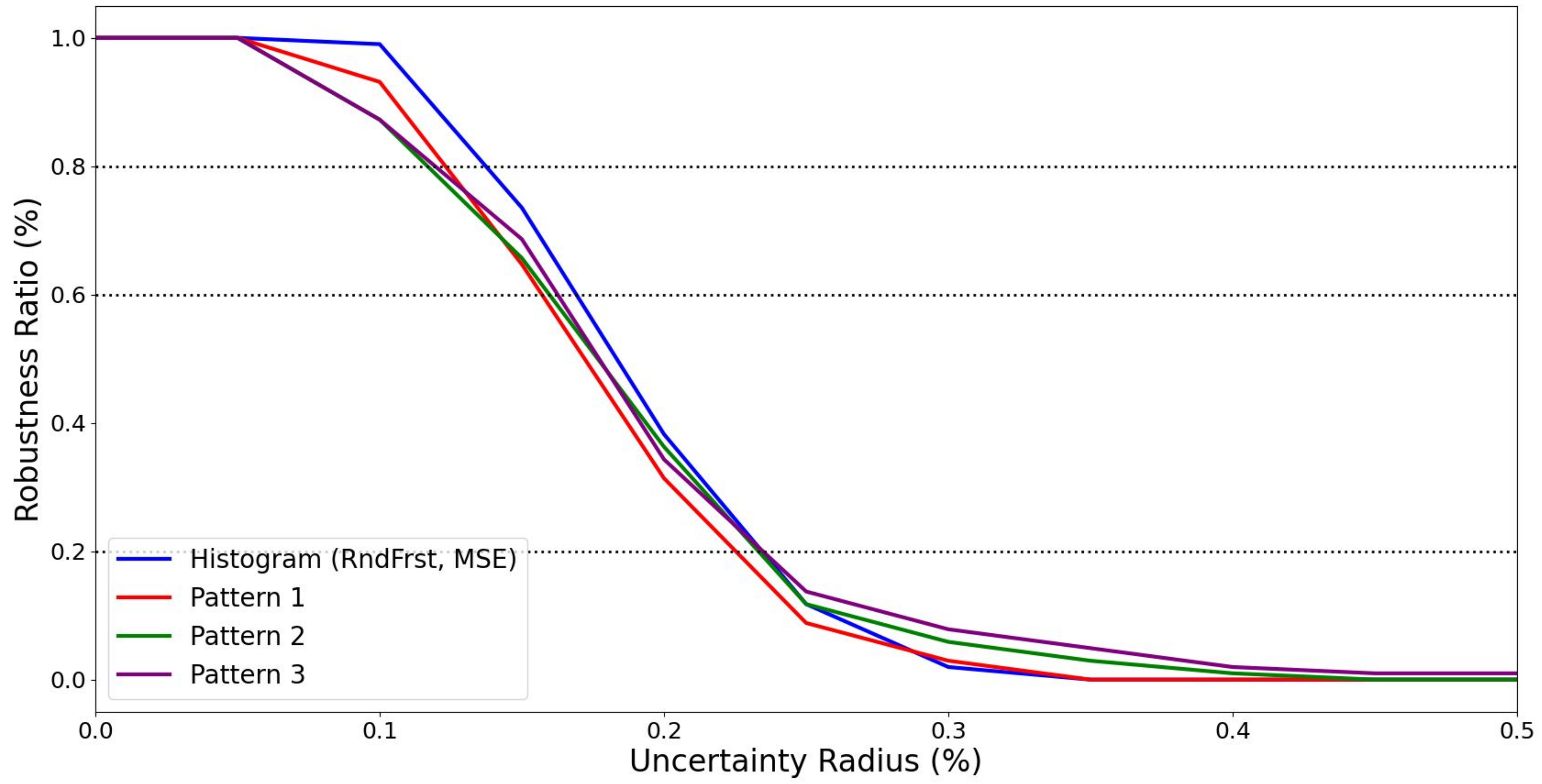
**Best Patterns Mined and Found:**

```
Pattern 1:
'Avg. # of Rooms per Dwelling' > 6.307842105263157
'Pupil-Teacher Ratio per Town' > 17.9

Pattern 2:
'Weighted Mean Distances to 5 Boston Employment Centres' < 2.3514777777777778
'Proportion of Local Population that is Lower Status' < 9.359473684210526

Pattern 3:
'Tax Rate per $10k' > 398.0
'Proportion of Local Population that is Lower Status' < 11.266842105263159
```

**Heatmap Comparison between Histogram method and Best Pattern Found:**



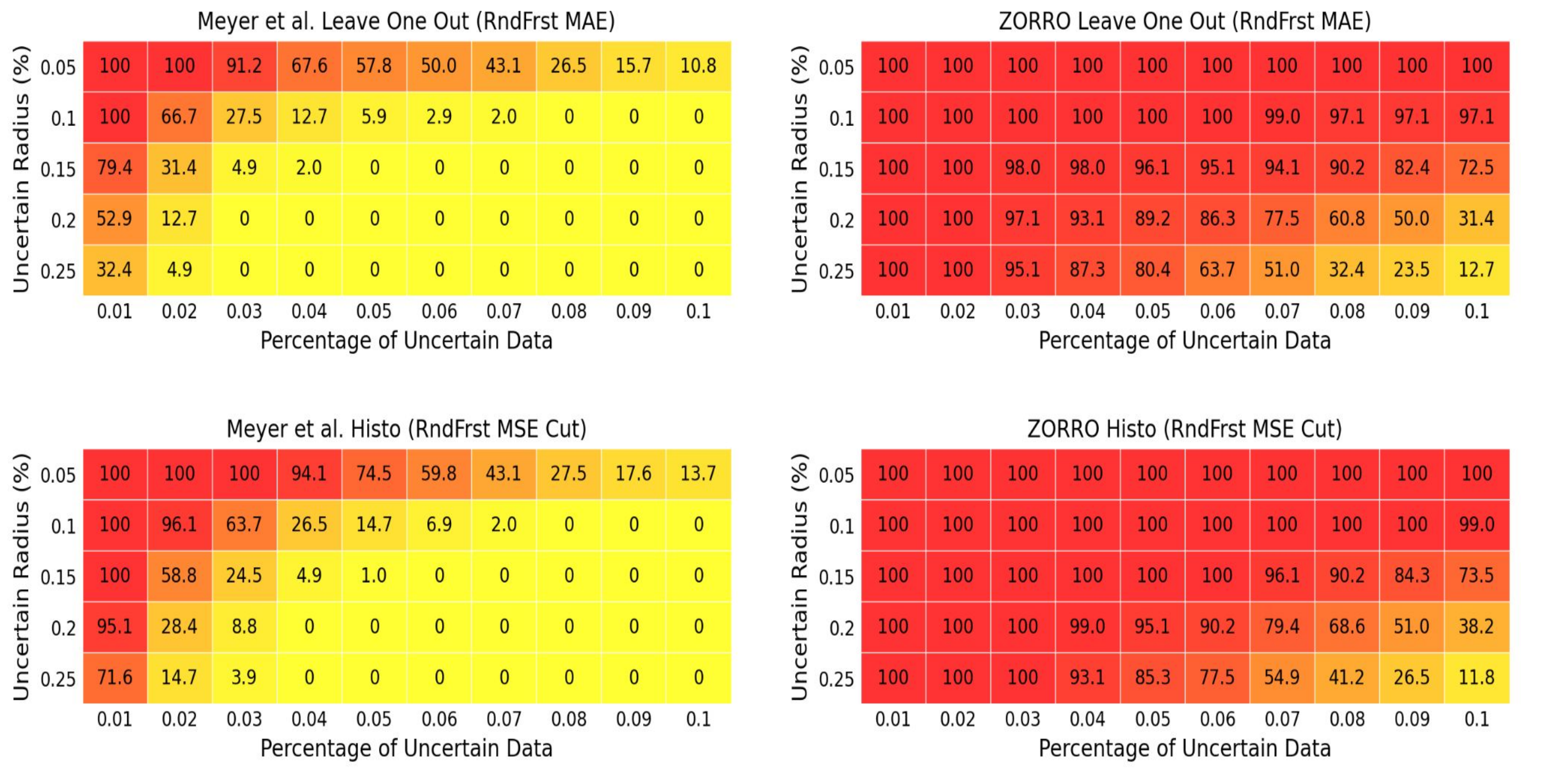
**Plotting Robustness Decreasing against patterns and Histo. method:**



## 3. Histogram

- The next method we attempted to capture the best representational indices was what we deemed the Histogram approach
- Below we compare our histogram approach, which utilizes a RndFrst model and MSE, to our previous LOO approach

**Heatmap Comparison between LOO method and Histogram method:**



## 5. Normalization

- Utilizing normalization across methods and datasets to determine the most effective error injection for each dataset along with the comparison of robustness across datasets

**Method Comparisons Across the Boston Dataset:**

```
Normalized robustness score for BOS_Pattern_Mining(Pattern 1) dataset is 0.0763
Normalized robustness score for BOS_Pattern_Mining(Pattern 3) dataset is 0.2783
Normalized robustness score for BOS_Pattern_Mining(Pattern 2) dataset is 0.7498
Normalized robustness score for BOS_histo(LinReg, MAE) dataset is 0.8172
Normalized robustness score for BOS_histo(Rndfrst, MSE) dataset is 0.8172
Normalized robustness score for BOS_histo(LinReg, MSE) dataset is 0.9182
Normalized robustness score for BOS_histo(Rndfrst, MAE) dataset is 3.3430
```

**Comparing the Robustness of "Boston" to other datasets:**

```
Normalized robustness score for Insurance dataset under worst case scenario is -0.2382
Normalized robustness score for FIRE dataset under worst case scenario is 0.2755
Normalized robustness score for BOS dataset under worst case scenario is 1.8923
Normalized robustness score for MPG dataset under worst case scenario is 2.0704
```

## 6. Conclusions

Possible next steps:

- better scaling of robustness across sets
- exploring the specific numerical differences in robustness between sets.