Exploring IP Address Characteristics to Identify Router Deployments

Yue Xin

IP Address Alias Resolution

Why?

- Information is transmitted at the IP level, while AS/ ISP plan on Router level.

Problems Szenario:

- Routing politics Differences
- Unstable probing Samples
- IPv4 and IPv6 barrier

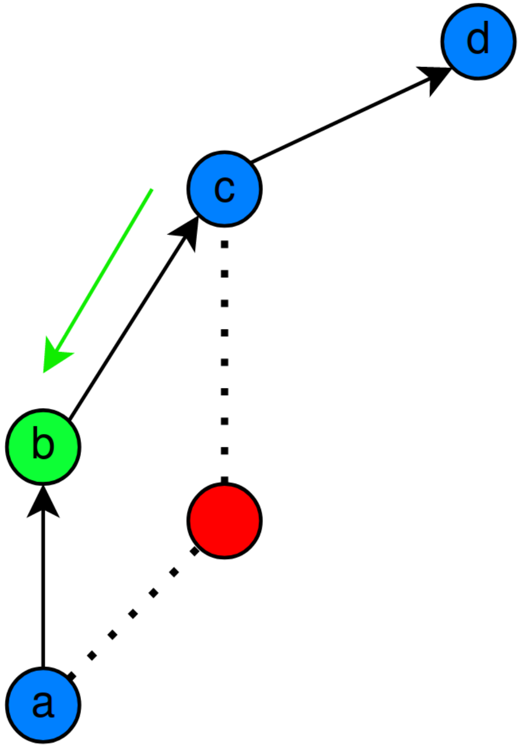
Problems Clustering:

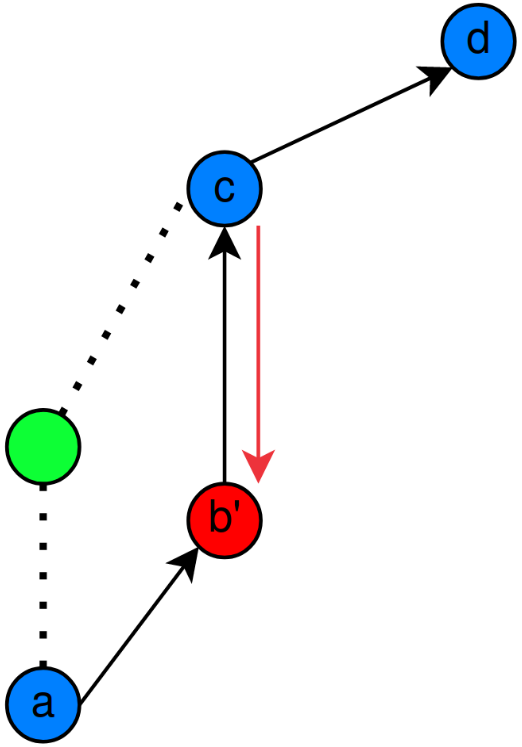
- Feature vector definition
- Distance definition
- Complexity of the algorithm

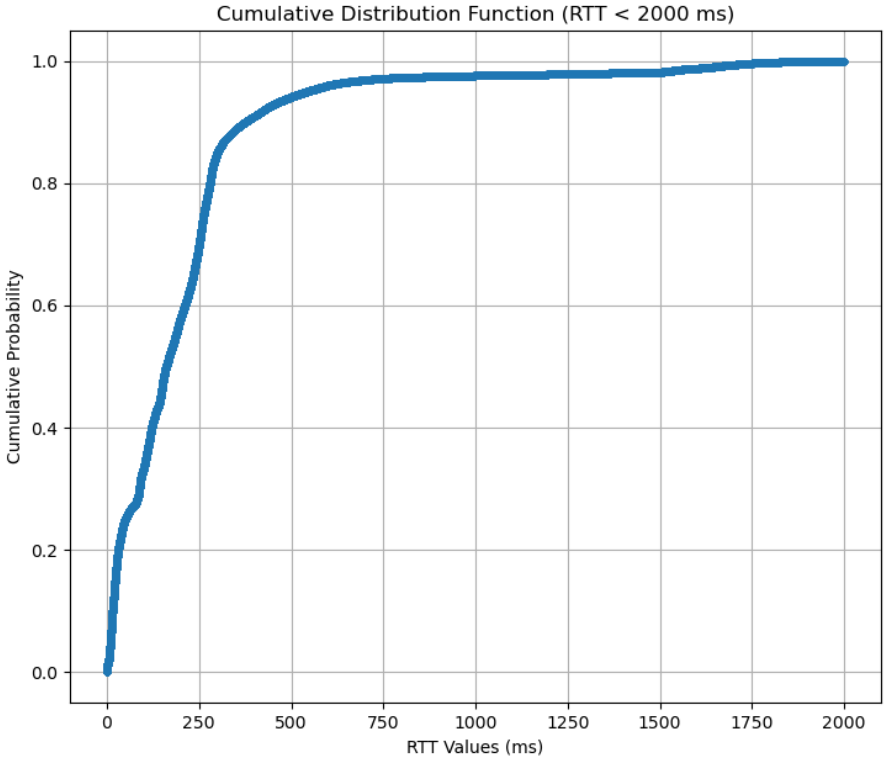
How was it before?

- Inference-based （prefix）
- Measurement based （APAR， MIDAR） (RTT, Loss Rate)
- Trafic-based

Data Generation – ICMP - RTT curve balls

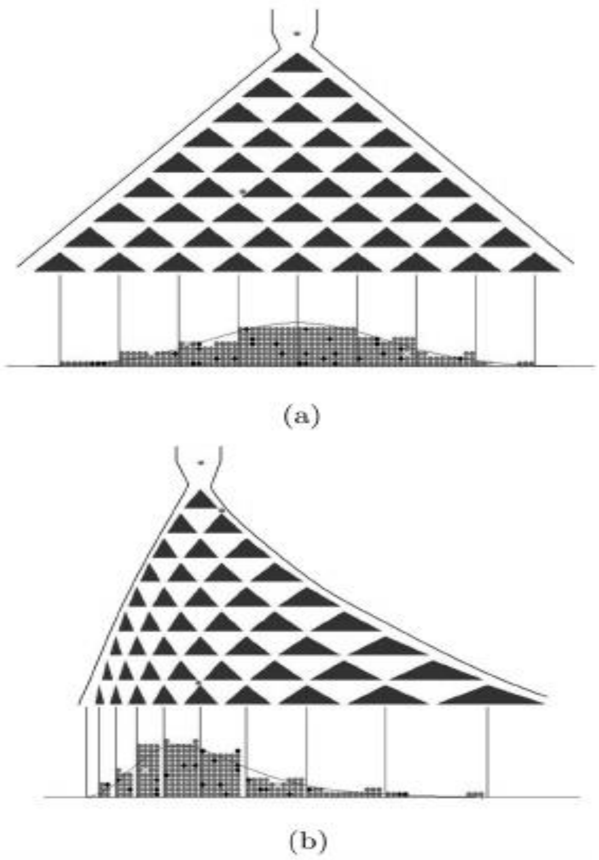


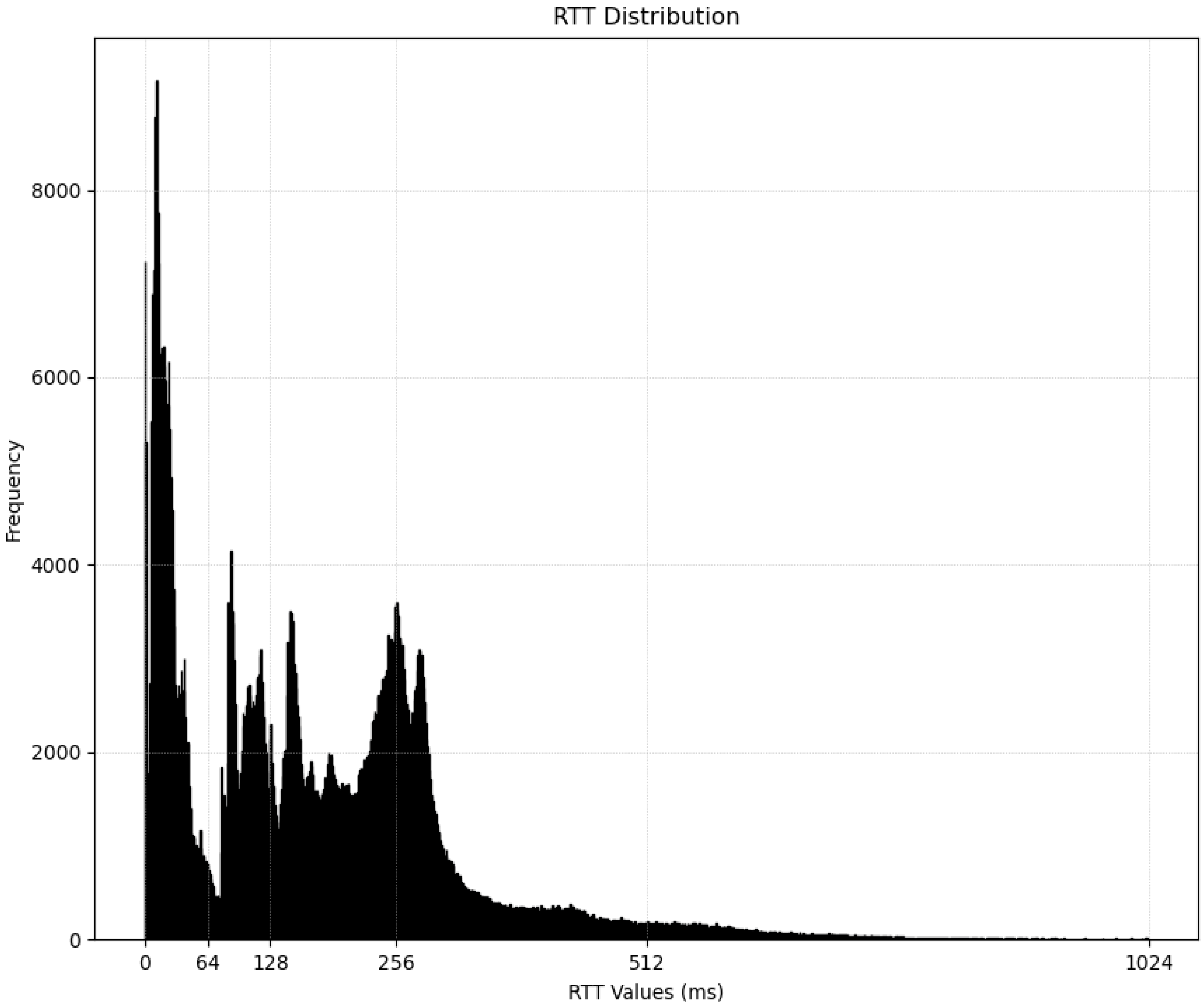




199,861,638 m/s×0.250s ≈ 50,000 km

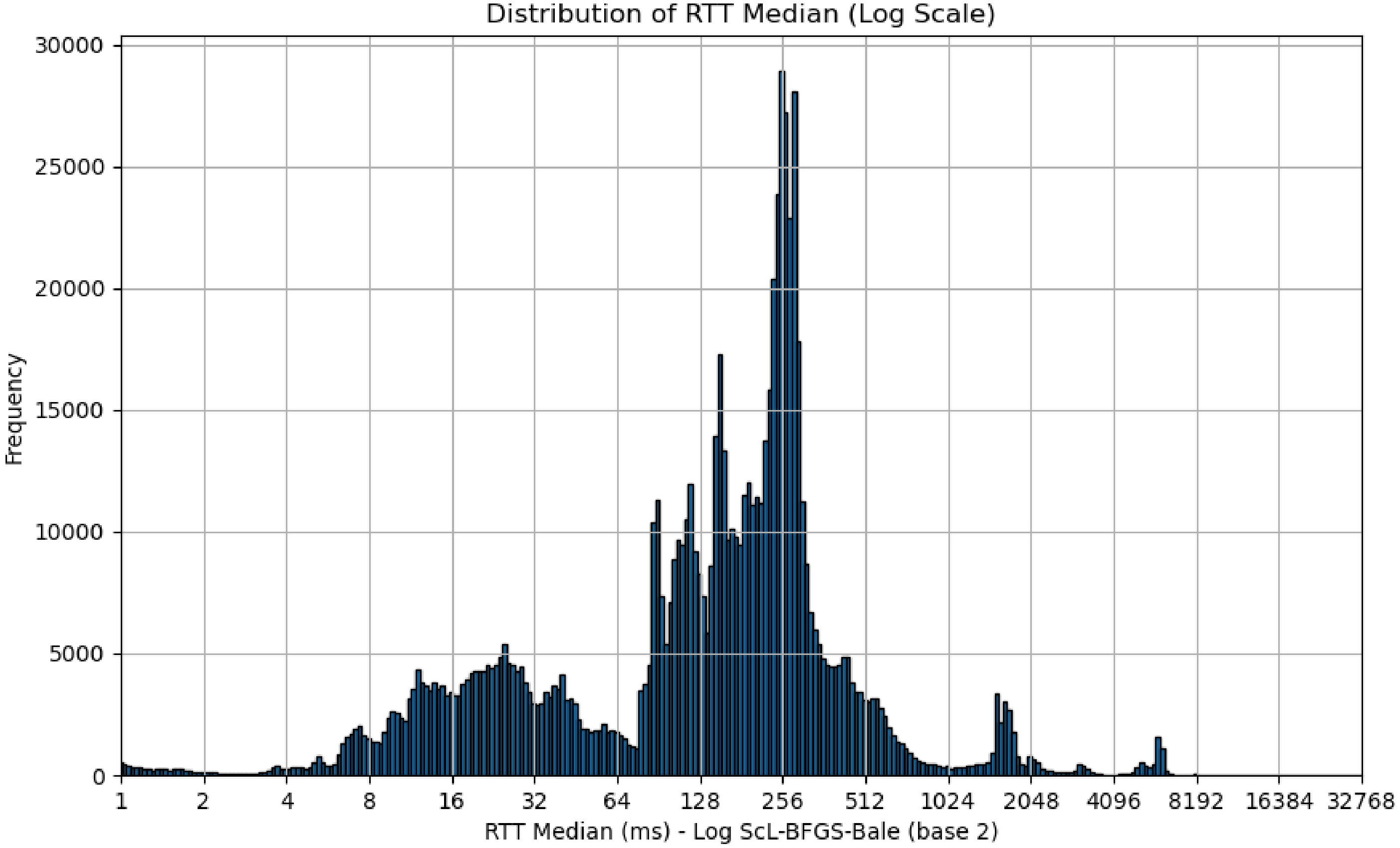
RTT

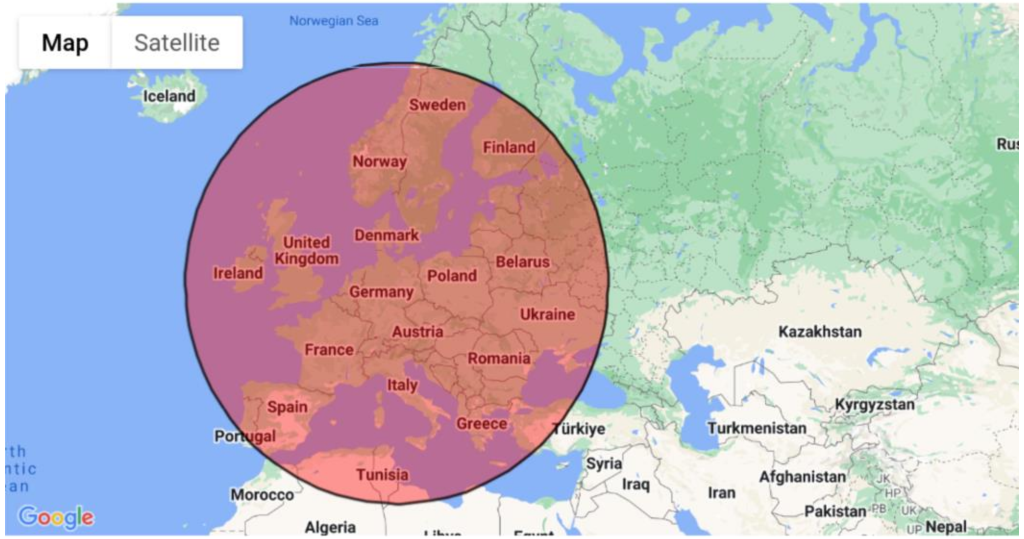




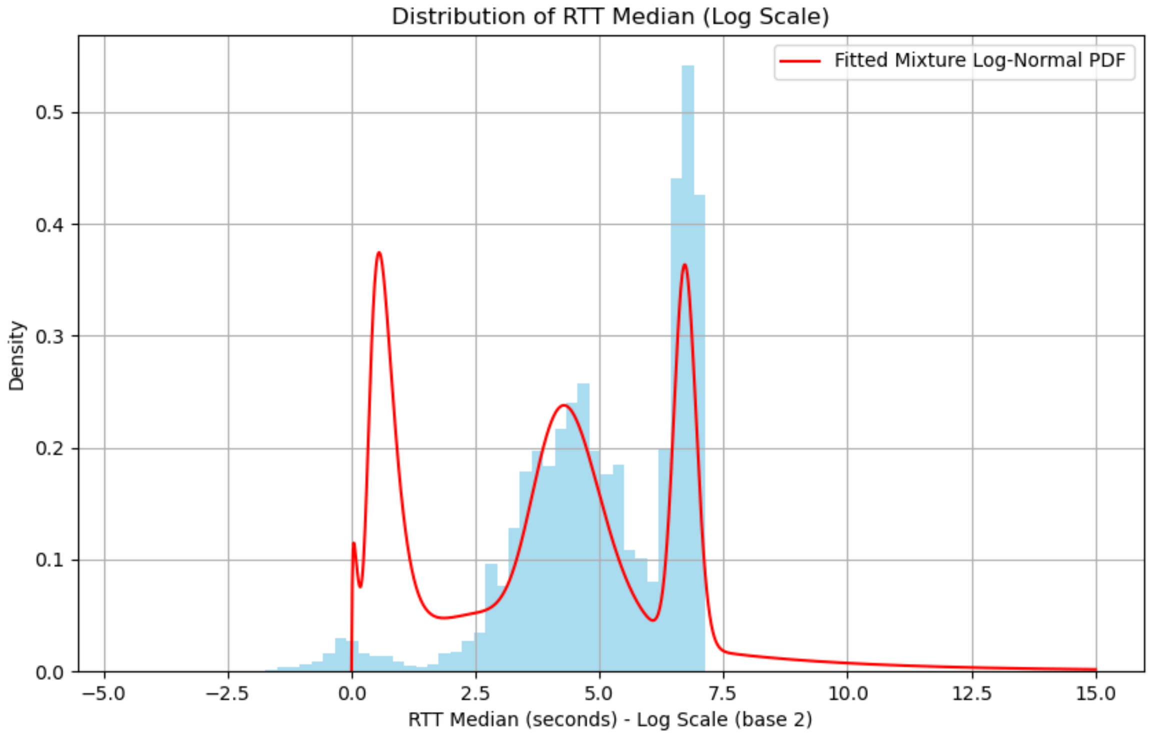
Population Distribution VS. RTT Distribution ?

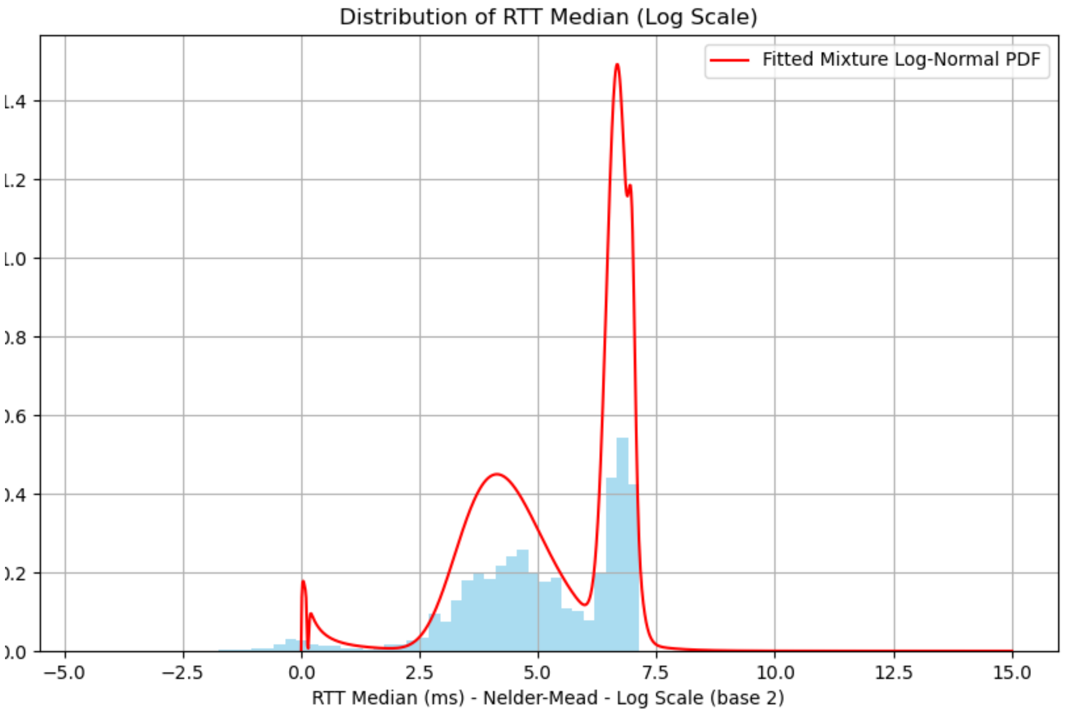
RTT - mixture of Log-Normal Distributions.



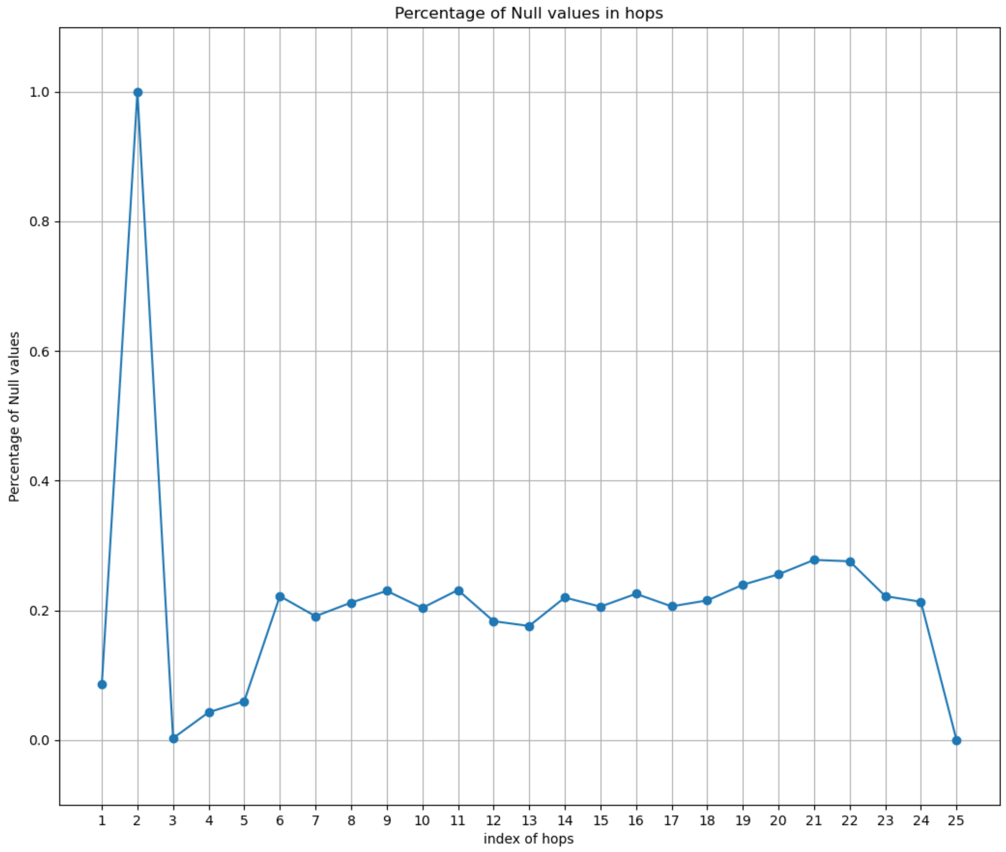


Fit this distribution

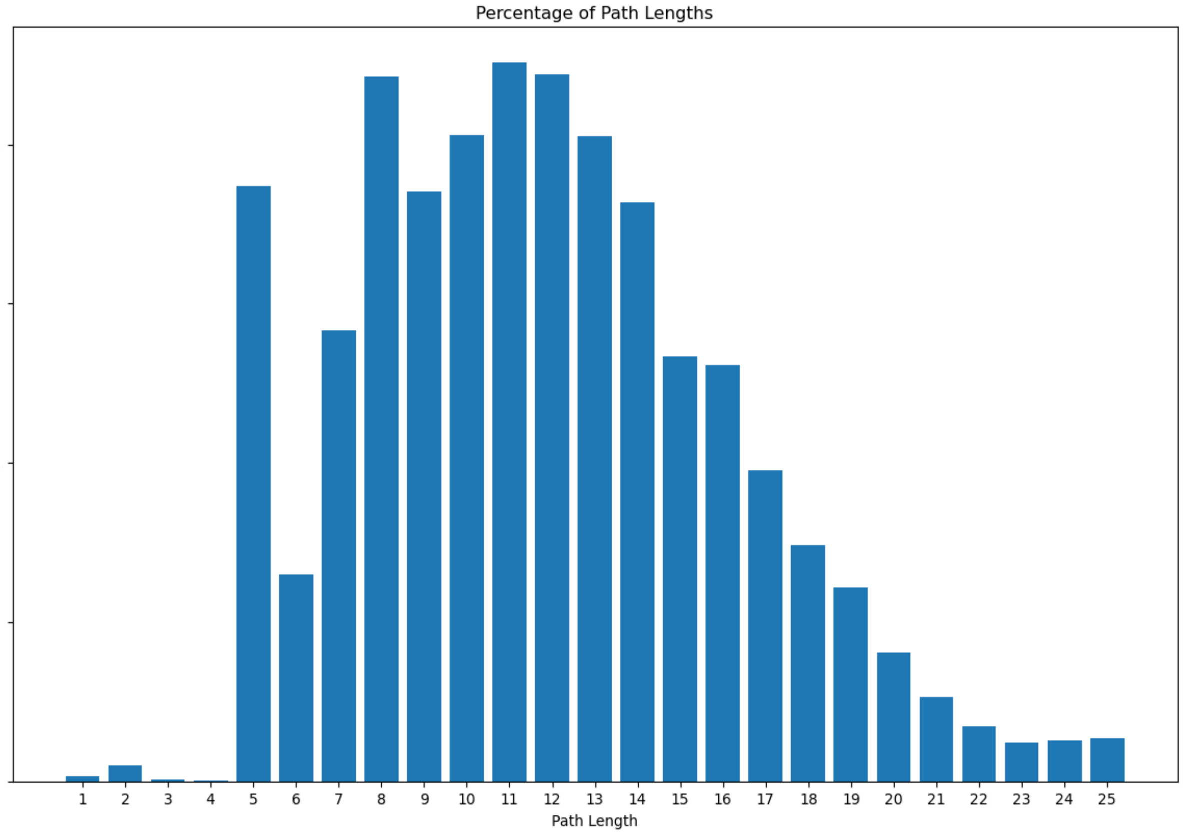




Null Values

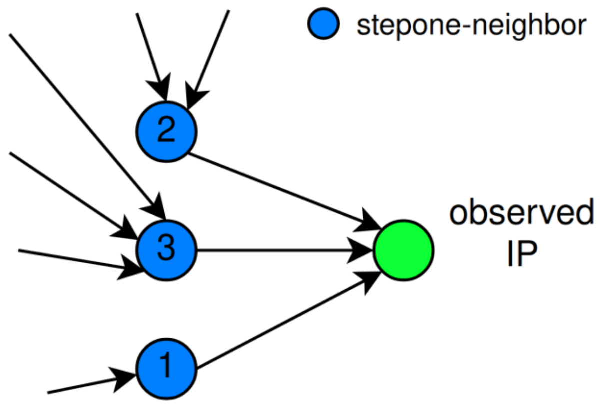


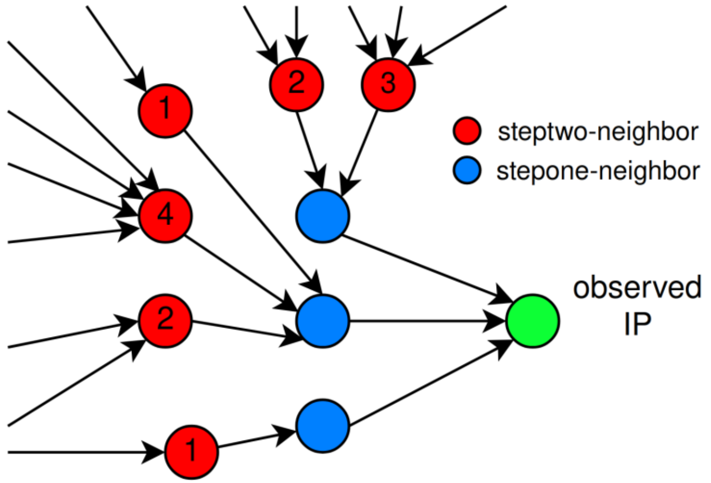
Percentage of Null values in hops
Percentage of Null values
1.0
0.8
0.6
0.4
0.2
0.0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
index of hops



Percentage of Path Lengths
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
Path Length

My Solution - Local Topology





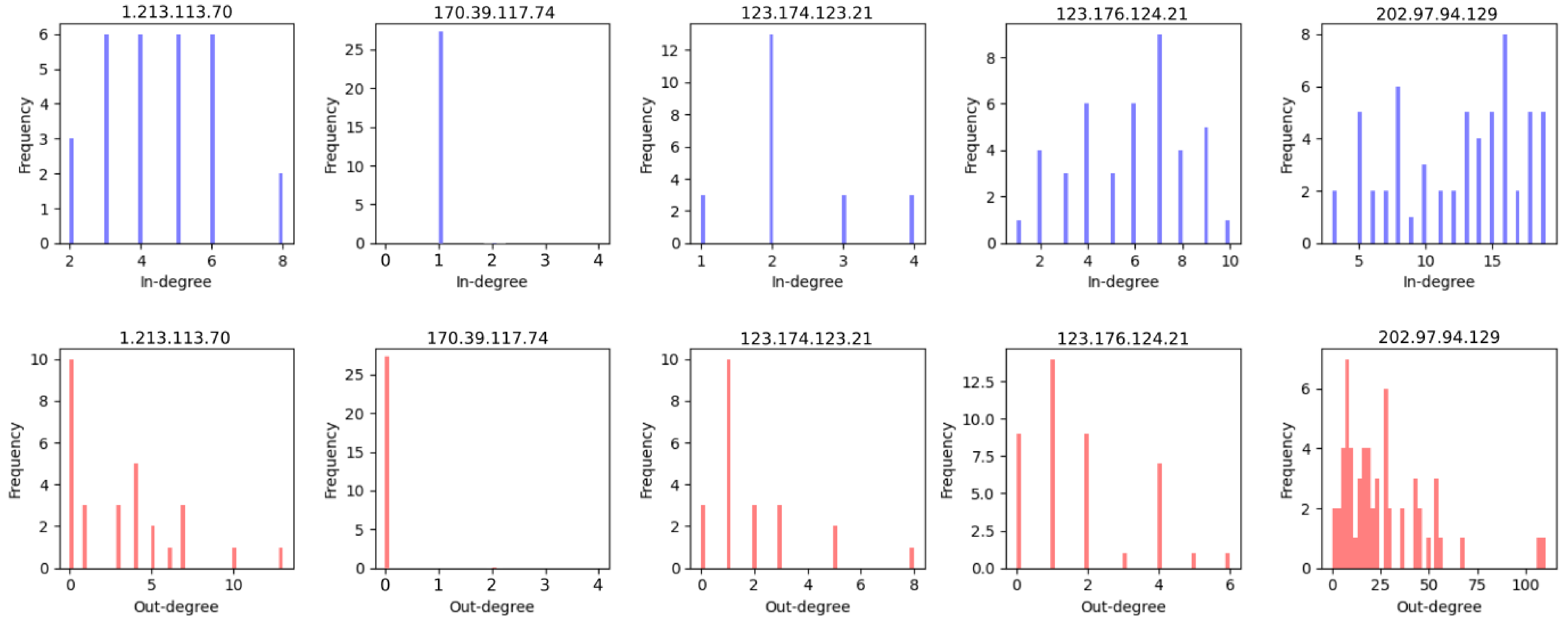
{ (2,1) , (3,1), (1,1) }

{ (3,1) , (2,1) , (1,1) , (4,1) , (2,1) , (1,1) }

Centroid: (2,1)

(13/6 , 1)

In-degree , Out-degree



Distance

Distance:

- Euclidean Distance of Geometric Centroids
- „Distance“ of Distribution

$$a = [1, 1, 1, 2, 2]$$

$$b = [1, 1, 1, 2, 2]$$

$$c = [1, 1, 1, 2, 2, 99]$$

$$d = [1, 1, 1, 2, 2, 3, 5, 8]$$

$$e = [1, 1, 1, 2, 2, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9]$$

$$f = [8, 9, 9, 100]$$

Wasserstein Distance

$$W(P, Q) = \sum_{i=1}^{n} |F_P(x_i) - F_Q(x_i)|$$

F = Cumulative Distribution Functions (CDFs)

Table 1: Wasserstein Distance

	a	b	c	d	e
b	0.0000				
c	16.2667	16.2667			
d	1.4750	1.4750	15.7917		
e	6.4824	6.4824	20.2157	5.0074	
f	30.1000	30.1000	13.8333	28.6250	23.8235

Jensen-Shannon Divergence

$$JSD(P\|Q) = \frac{1}{2}(D_{KL}(P\|M) + D_{KL}(Q\|M))$$

$$D_{KL}(P\|Q) = \sum_i P(i) \log\left(\frac{P(i)}{Q(i)}\right)$$

Table 2: Jensen-Shannon Divergence

	a	b	c	d	e
b	0.0000				
c	34.3108	34.3108			
d	5.5452	5.5452	33.0887		
e	90.4557	90.4557	109.2773	82.1875	
f	36.3518	36.3518	70.6625	41.8969	126.8075

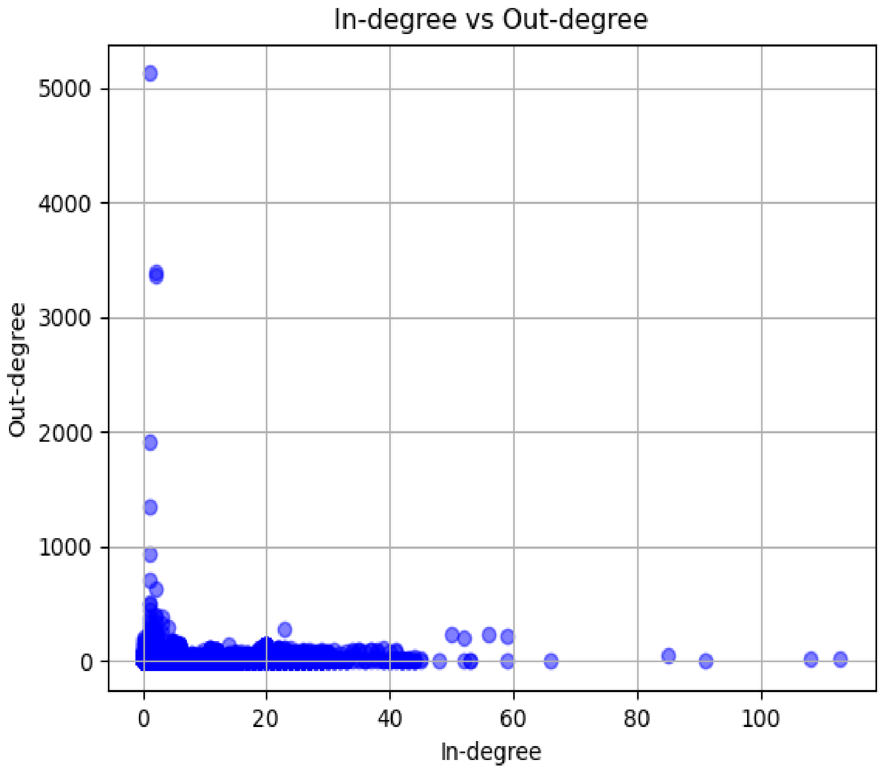
Jaccard Distance

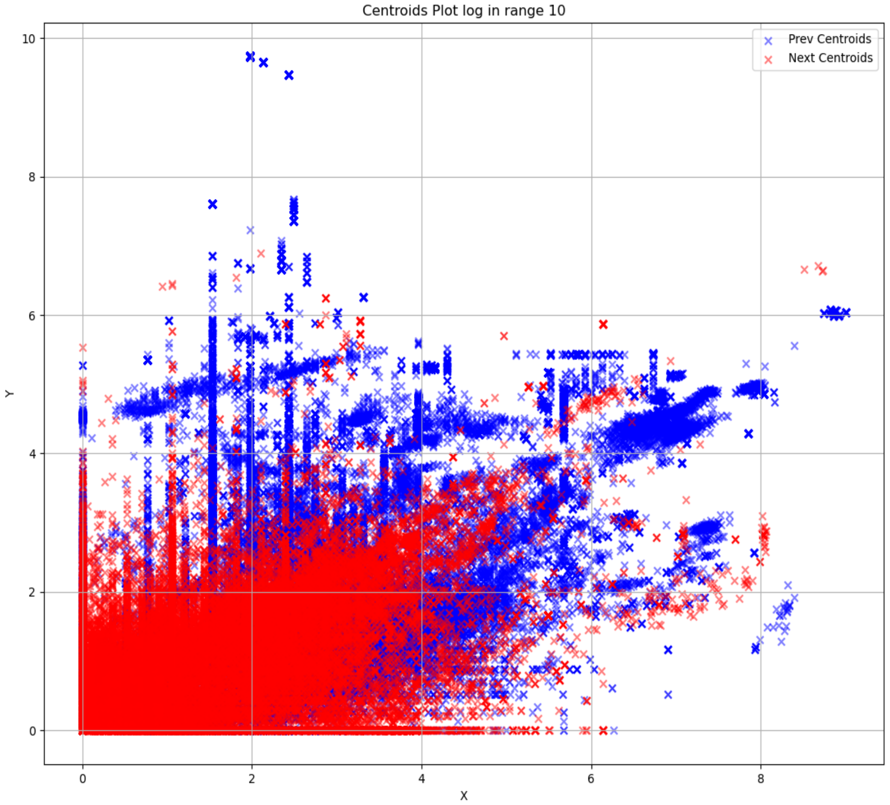
$$D_J(A,B) = 1 - \frac{|A \cap B|}{|A \cup B|}$$

Table 5: Customized Jaccard Distance

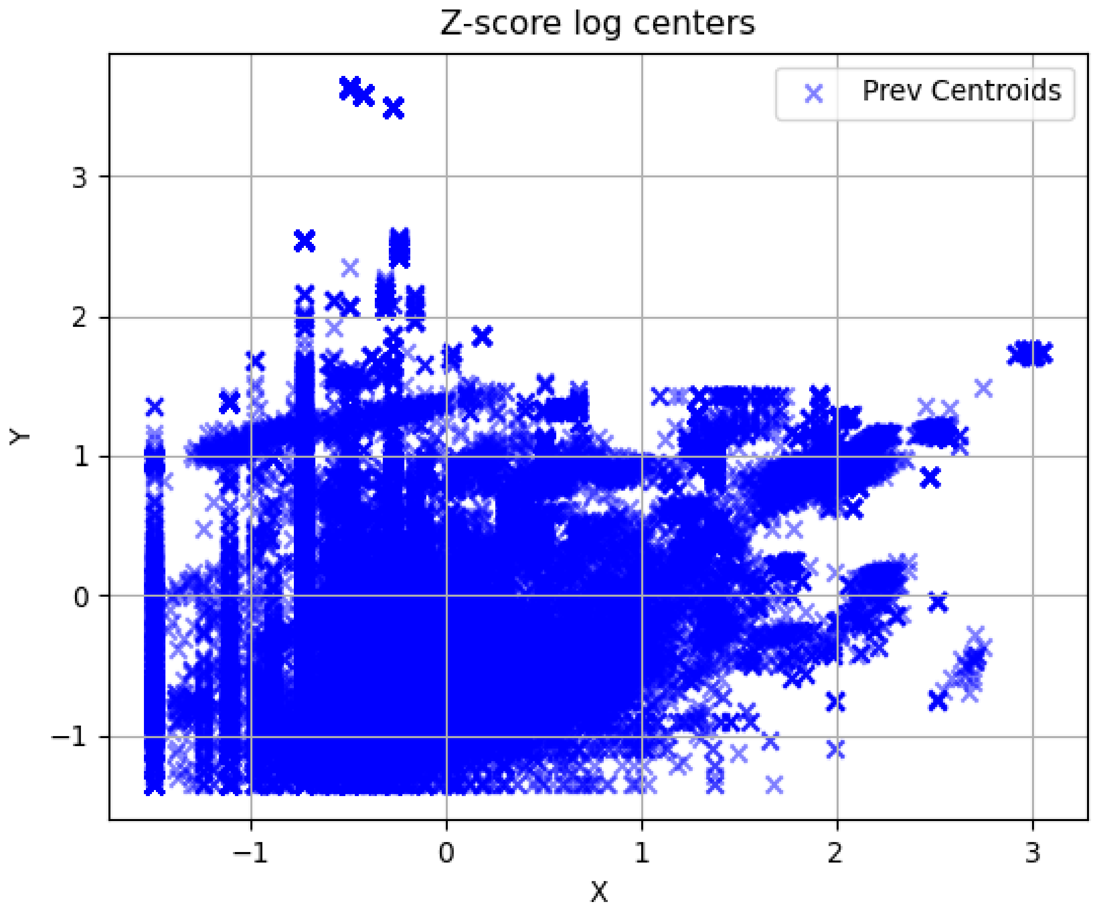
	a	b	c	d	e
b	0				
c	0.0909	0.0909			
d	0.2307	0.2307	0.2857		
e	0.7435	0.7435	0.75	0.7619	
f	1	1	1	0.8333	0.8947

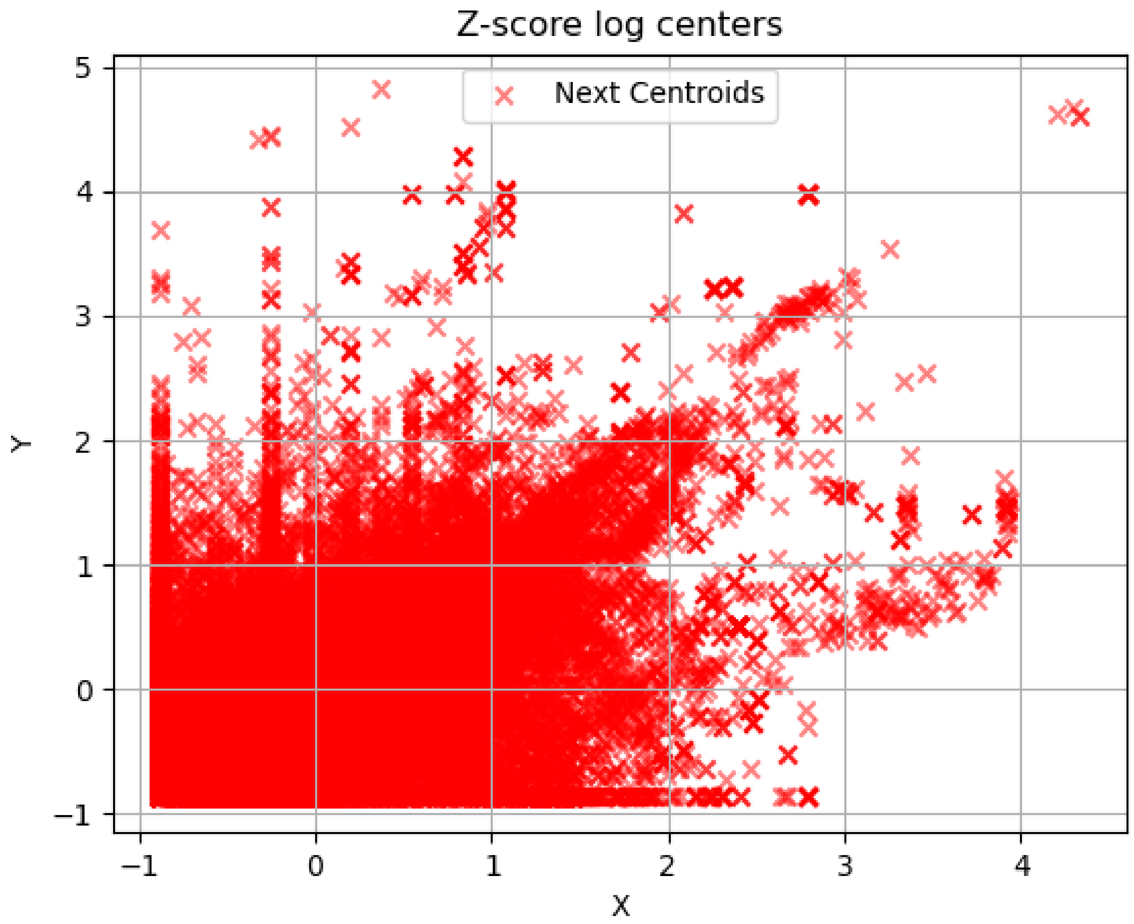
Z-score log scaled centroid



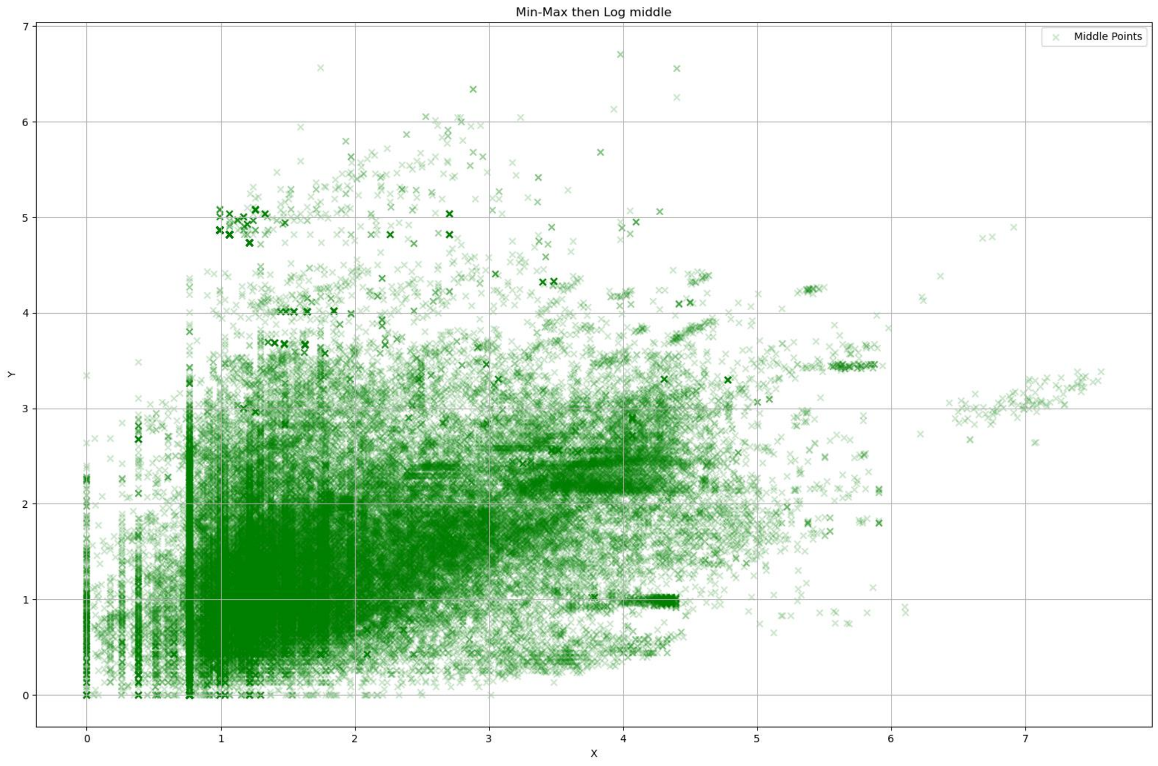


Z-score log scaled centroid

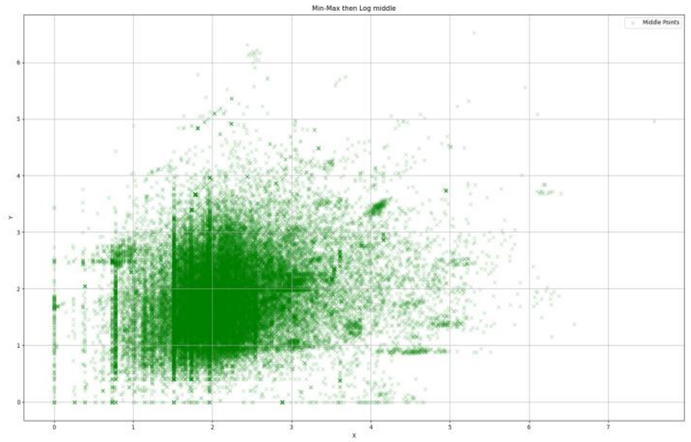




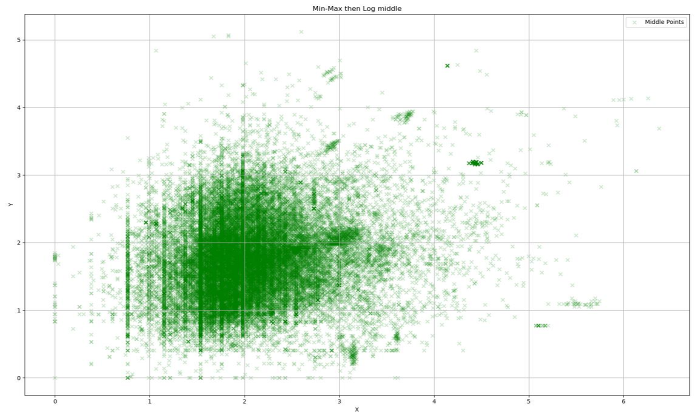
Neighbours, not observed IP



Step one neighbour centroids

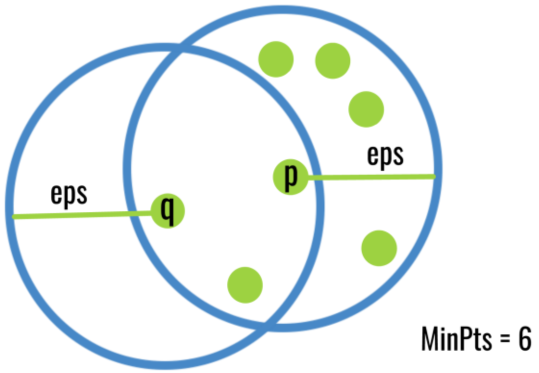


Step two

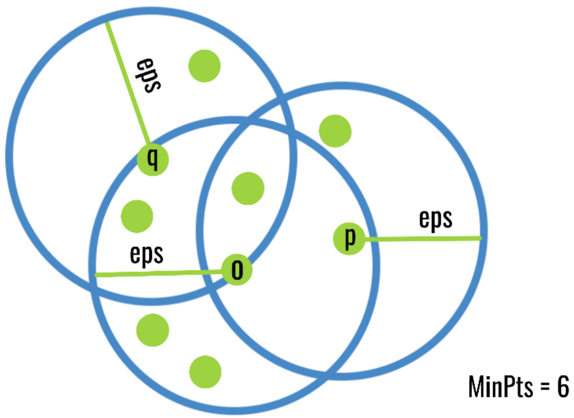


Step three

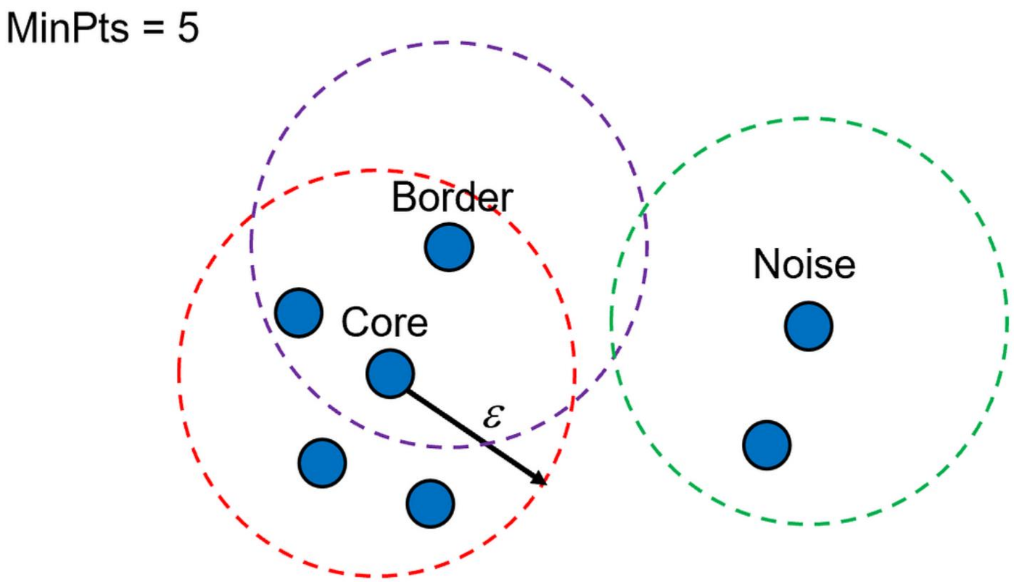
DBSCAN

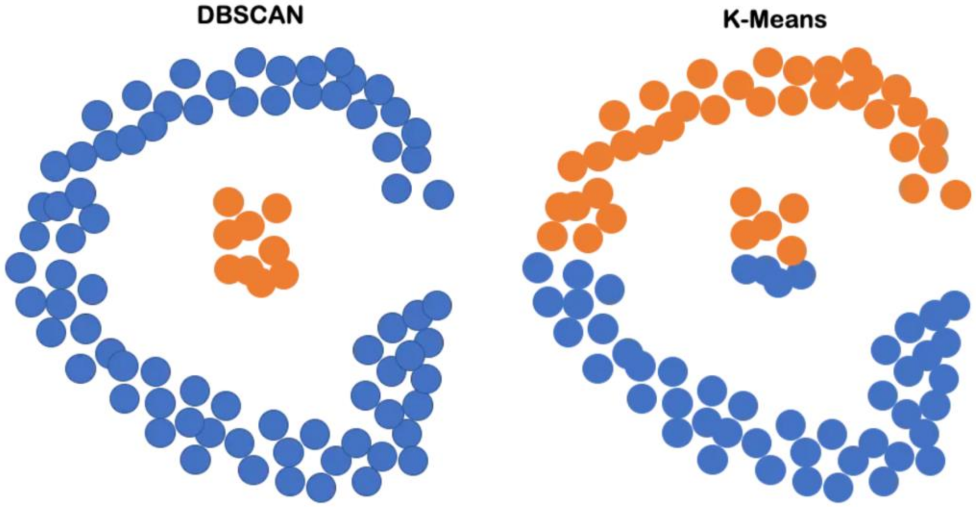


Directly density reachable



Density connectivity

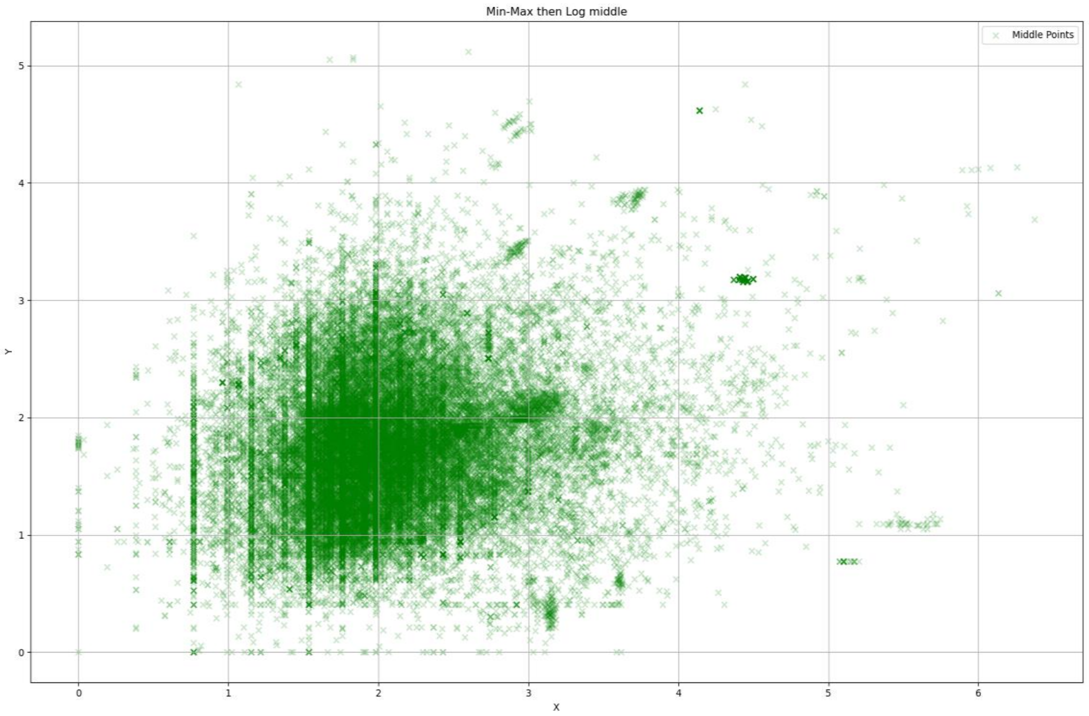


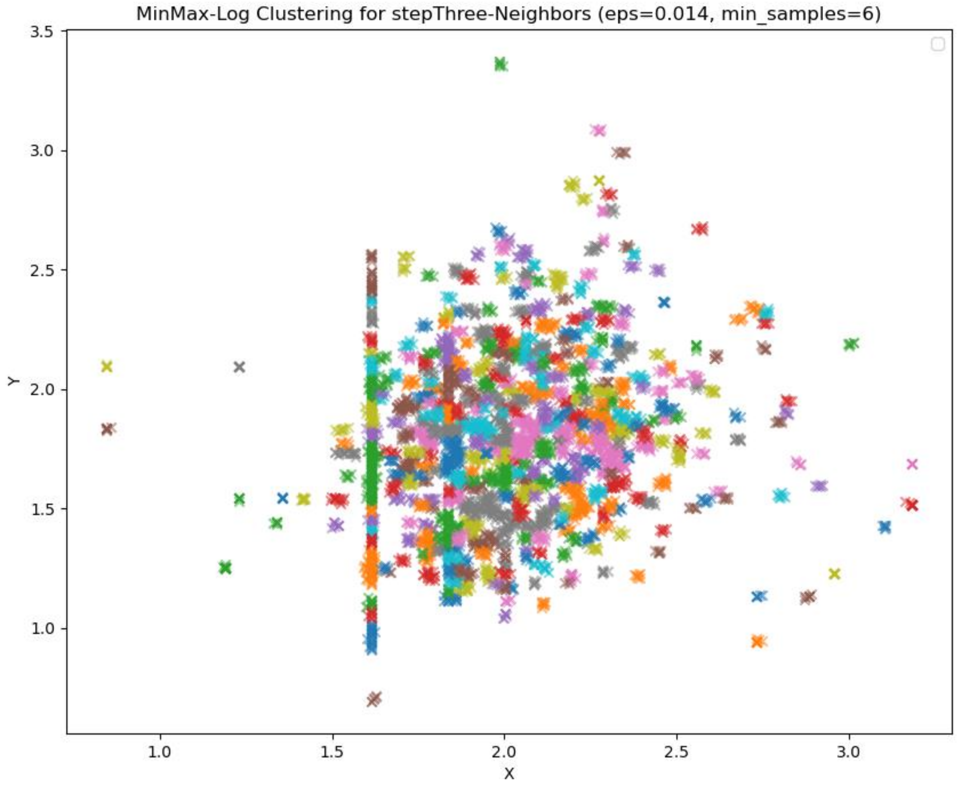


DBSCAN vs K-means. Dots are samples, the X-axis is feature 1 and the Y-axis is feature 2. (Image by author)

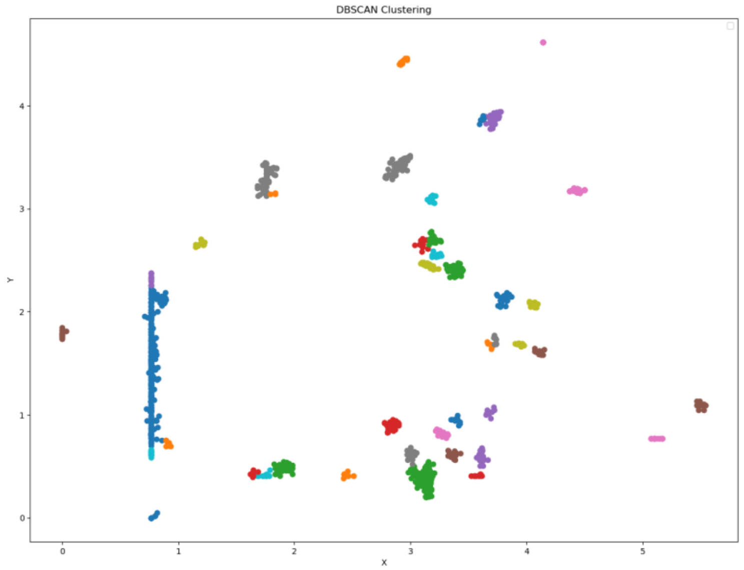
Picture by Dr. Yufeng published in Towards Data science

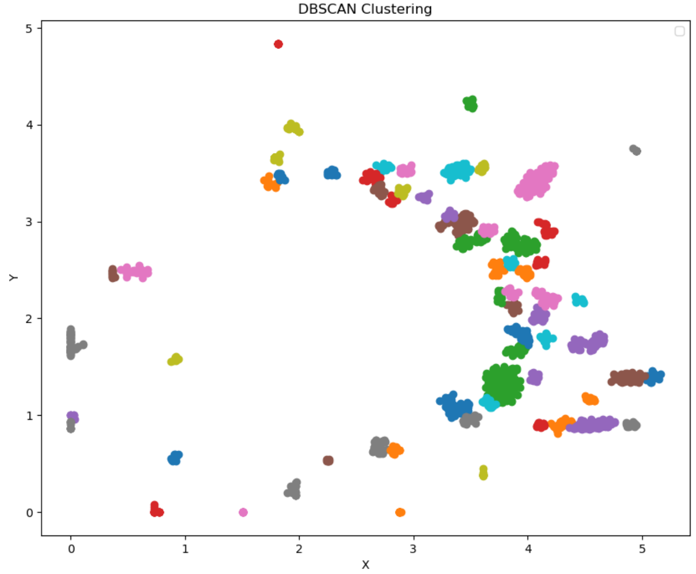
Apply DBSCAN

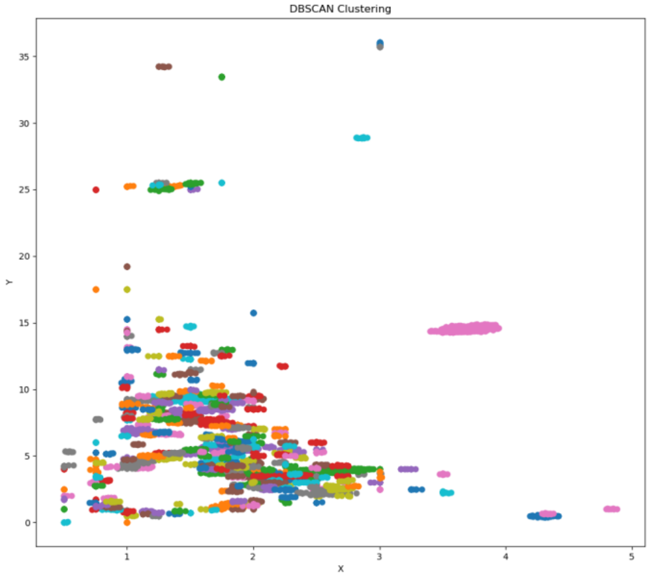




Apply DBSCAN with different radii





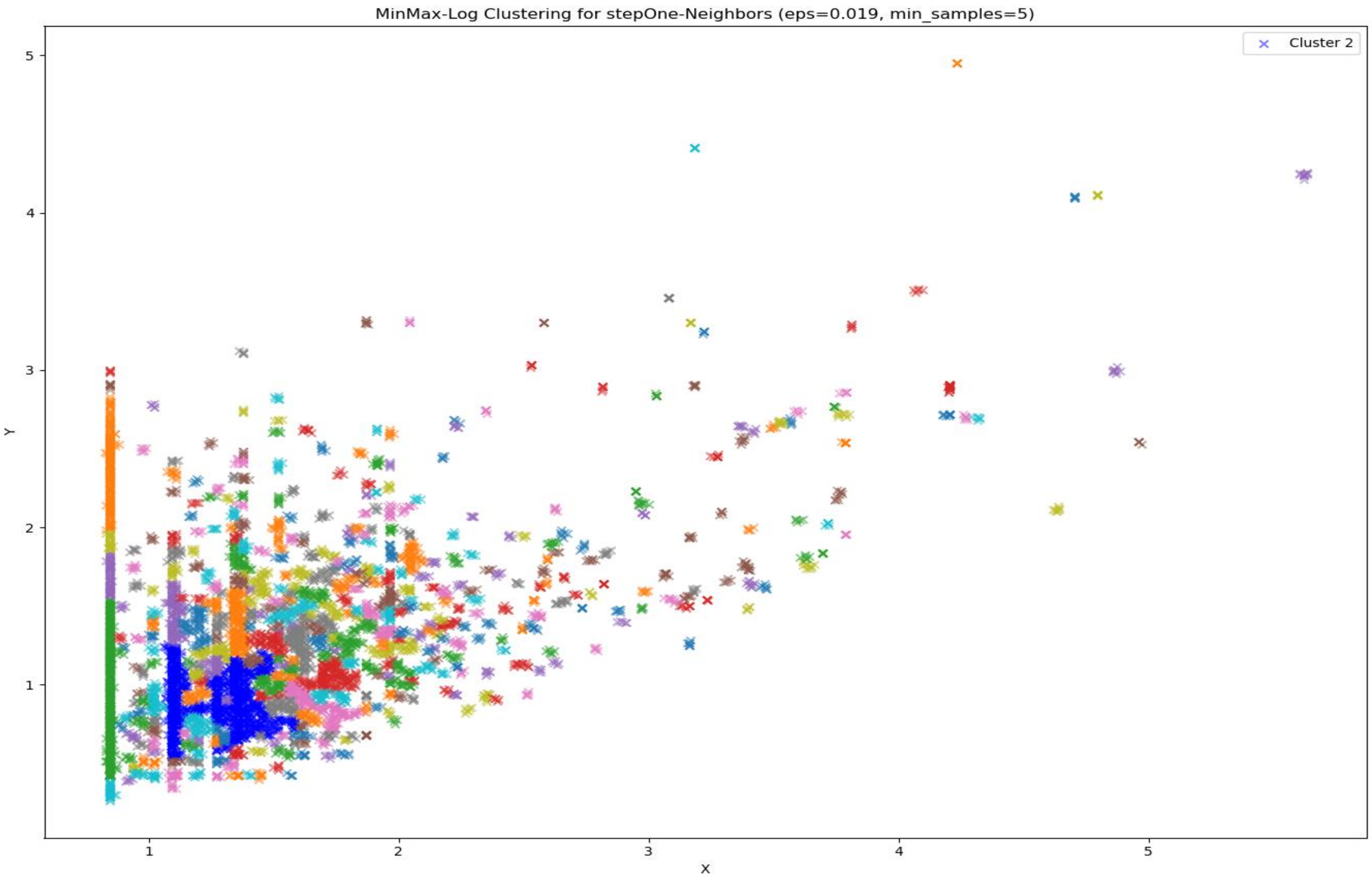


Apply DBSCAN iterativly and with RTT

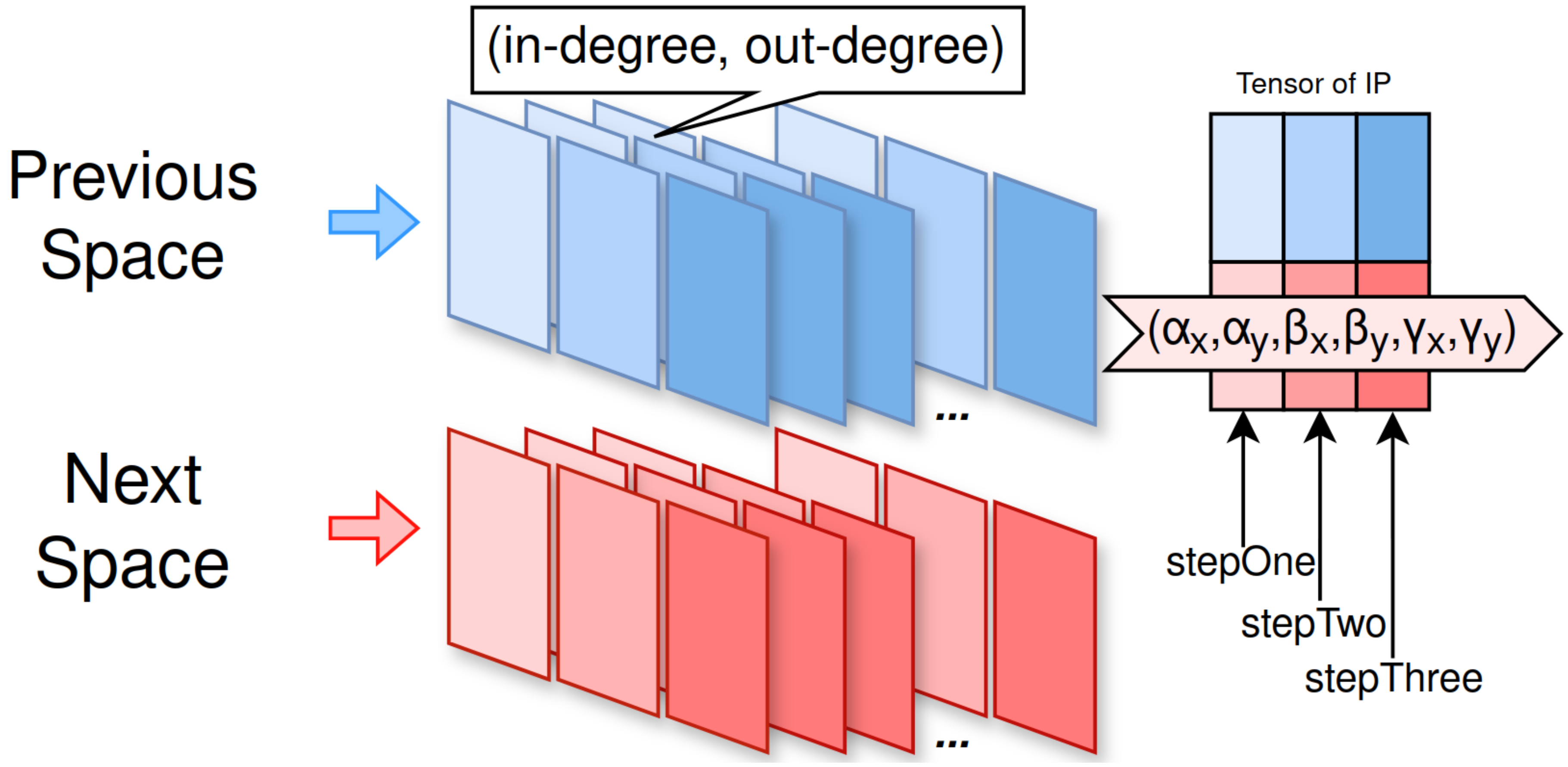
ϵ from large to small

Algorithm 3 Iterative RTT Enhanced DBSCAN

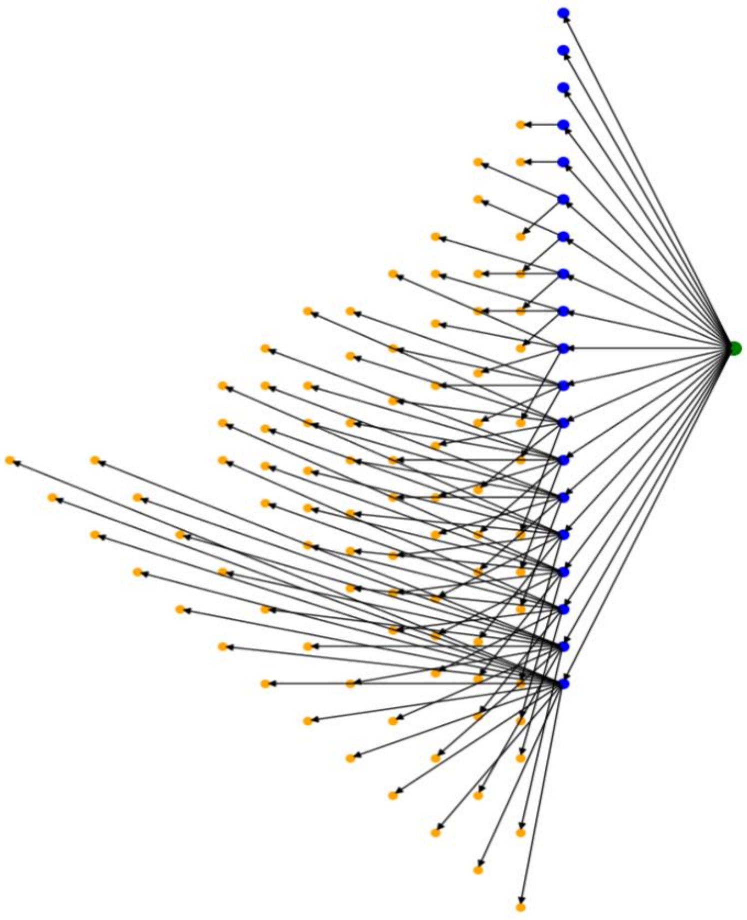
```
function IterativeRTTEnhancedDBSCAN(D, P, ϵ, ϵ_rtt)
    Initialize value of threshold N
    Initialize value of step size step
    Pool ← D
    Δ ← 1
    while Δ > 0 do
        PoolSizeBefore ← size of Pool
        Clusters, Noise ← RTT-DBSCAN(Pool, MinPts, ϵ, ϵ_rtt)
        for cluster in Clusters do
            if size of cluster ≤ N then
                add cluster to Result
            else
                add cluster to LargeClusters
            end if
        end for
        Pool ← Noise ∪ LargeClusters
        Δ = PoolSizeBefore − size of Pool
        ϵ = ϵ − step
    end while
    return Result
end function
```


MinMax-Log Clustering for stepOne-Neighbors (eps=0.019, min_samples=5)
Cluster 2
Y
X
1
2
3
4
5
1
2
3
4
5

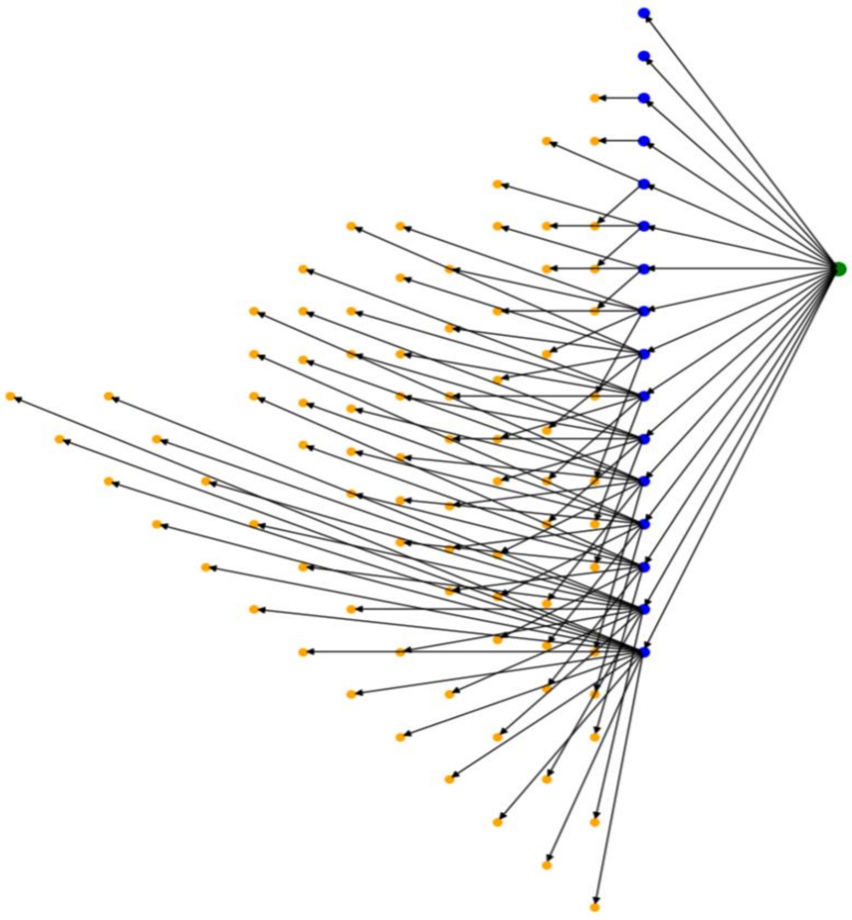
My Solution - Local Topology



Results



66.109.3.233



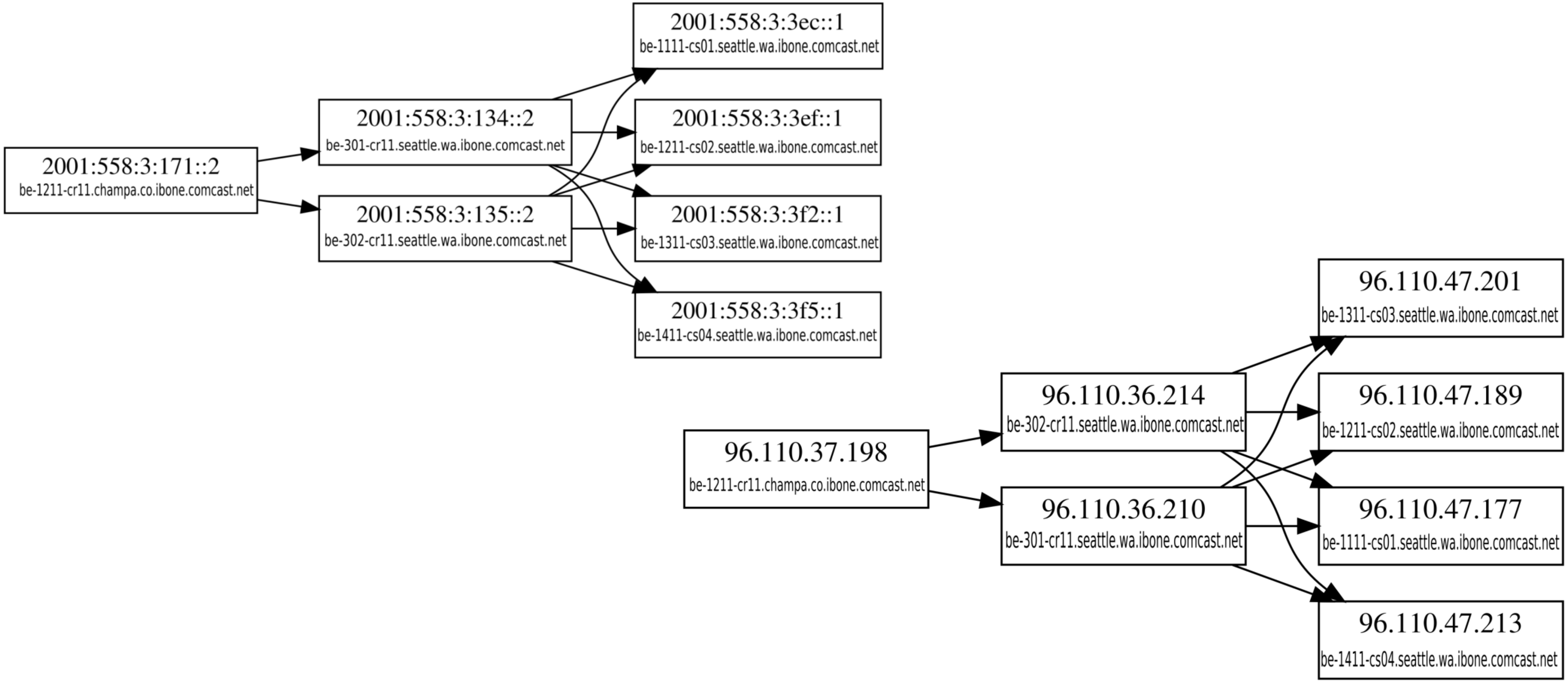
209.18.43.73

DNS

Table 8: Cluster "62" IP and DNS

	IP Address	DNS Name
1	2001:558:3:170::2	be-1111-cr11.champa.co.ibone.comcast.net
2	2001:558:3:172::2	be-1311-cr11.champa.co.ibone.comcast.net
3	96.110.37.202	be-1311-cr11.champa.co.ibone.comcast.net
4	96.110.37.206	be-1411-cr11.champa.co.ibone.comcast.net
5	58.138.106.18	osk004ipgw01.IIJ.Net
6	96.110.37.198	be-1211-cr11.champa.co.ibone.comcast.net
7	2001:558:3:173::2	be-1411-cr11.champa.co.ibone.comcast.net
8	96.110.37.194	be-1111-cr11.champa.co.ibone.comcast.net
9	2001:558:3:171::2	be-1211-cr11.champa.co.ibone.comcast.net

IPv4 , IPv6


2001:558:3:171::2
be-1211-cr11.champa.co.ibone.comcast.net
2001:558:3:134::2
be-301-cr11.seattle.wa.ibone.comcast.net
2001:558:3:135::2
be-302-cr11.seattle.wa.ibone.comcast.net
2001:558:3:3ec::1
be-1111-cs01.seattle.wa.ibone.comcast.net
2001:558:3:3ef::1
be-1211-cs02.seattle.wa.ibone.comcast.net
2001:558:3:3f2::1
be-1311-cs03.seattle.wa.ibone.comcast.net
2001:558:3:3f5::1
be-1411-cs04.seattle.wa.ibone.comcast.net
96.110.37.198
be-1211-cr11.champa.co.ibone.comcast.net
96.110.36.214
be-302-cr11.seattle.wa.ibone.comcast.net
96.110.36.210
be-301-cr11.seattle.wa.ibone.comcast.net
96.110.47.201
be-1311-cs03.seattle.wa.ibone.comcast.net
96.110.47.189
be-1211-cs02.seattle.wa.ibone.comcast.net
96.110.47.177
be-1111-cs01.seattle.wa.ibone.comcast.net
96.110.47.213
be-1411-cs04.seattle.wa.ibone.comcast.net

Future Works

1. Pattern explicit definition
   - „Bridge, dumbbell, star“ .
   - Router typ definition on Patterns.
   - Routing Rules Detection/ Prediction
   - Relationships between ASes.
2. Supervised Learning
3. Dynamic Sampling
   - Update, Delete…
4. Fault Detection
   - Pattern - „Should be a EBGP, act like a IGP, why?“
   - Dynamy – „It was a EGP yesterday, looks like a IGP today, why?“
5. #ALLHands