SYNTHAPSE

24.01-26.01.2025

AI-Driven Tools for Optimizing Decentralized Network Performance

Organised by:

lab lab ai

lab lab NEXT

Technology Provider:

AI/ML API

Malik Zubair

Hamna Dawood

Piotr Żak

Analysis

```
# Connect provinces (example connections, modify as needed for real representation)
province_connections = [
    ("Arun", "Janakpur"),
    ("Janakpur", "Kathmandu"),
    ("Kathmandu", "Gandaki"),
    ("Gandaki", "Kapilavastu"),
    ("Kapilavastu", "Karnali"),
    ("Karnali", "Mahakali"),
    ("Mahakali", "Arun")
]
```

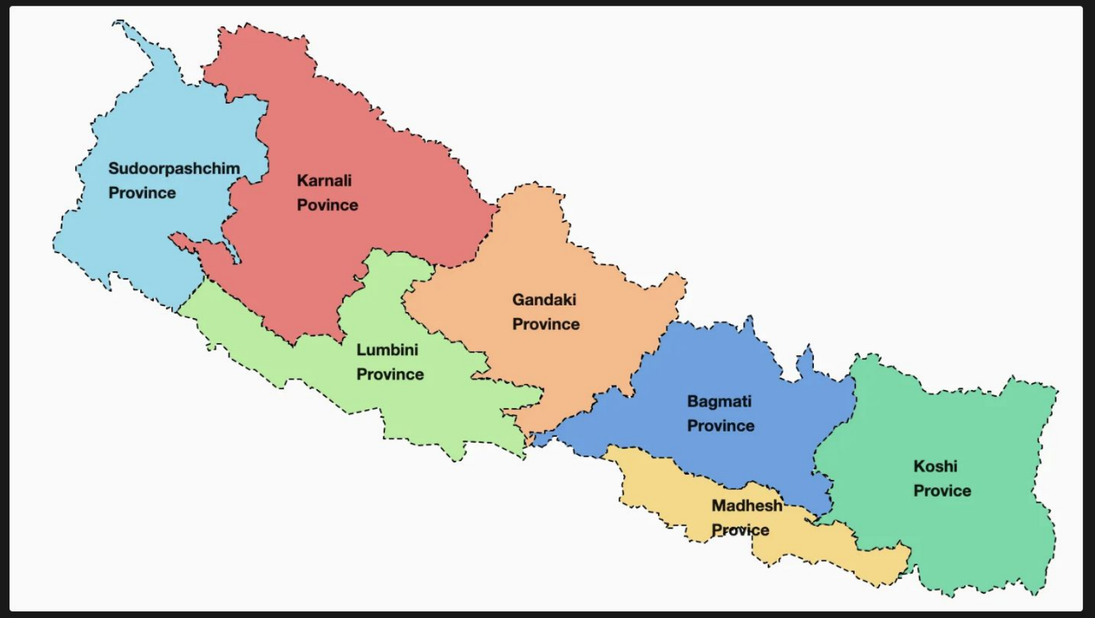
```
# Intermediate Level: Add cities/societal places for each province
cities = {
    "Arun": ["A1", "A2", "A3", "A4", "A5"],
    "Janakpur": ["B1", "B2", "B3", "B4", "B5"],
    "Kathmandu": ["C1", "C2", "C3", "C4", "C5"],
    "Gandaki": ["D1", "D2", "D3", "D4", "D5"],
    "Kapilavastu": ["E1", "E2", "E3", "E4", "E5"],
    "Karnali": ["F1", "F2", "F3", "F4", "F5"],
    "Mahakali": ["G1", "G2", "G3", "G4", "G5"]
}
```

7 Provinces Simulation

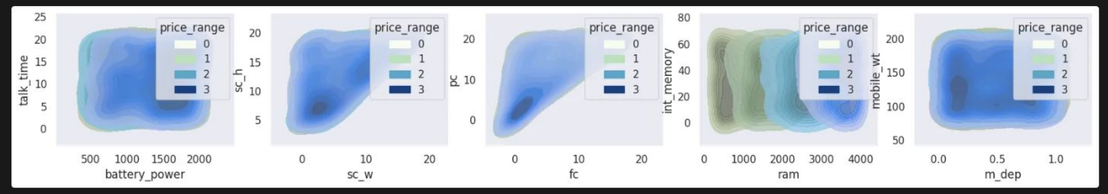
Data Distribution Agent.

Start by 1 000 GB per province (then it be separable via 5 points inner province)

- AI Agent on global level (to manage the macro scale (province)
- AI Agent on province level (to manage the micro scale (within 5 province point)



1. **Gather the data**
2. **Analyse the data.**
   - k shivkumarganesh **Visualization of Starlink-Satellites-Launched Data**
   - k melissamonfared **Mobile Price Prediction▲EDA & Classification**



Internet Price:

- k carriech **World Internet Price-Data Visualization📊**

Geospatial data:

- k tumpanjawat **🌐Internet Use : Geo🗺️| Cluster🎛️|Time Series🪨**

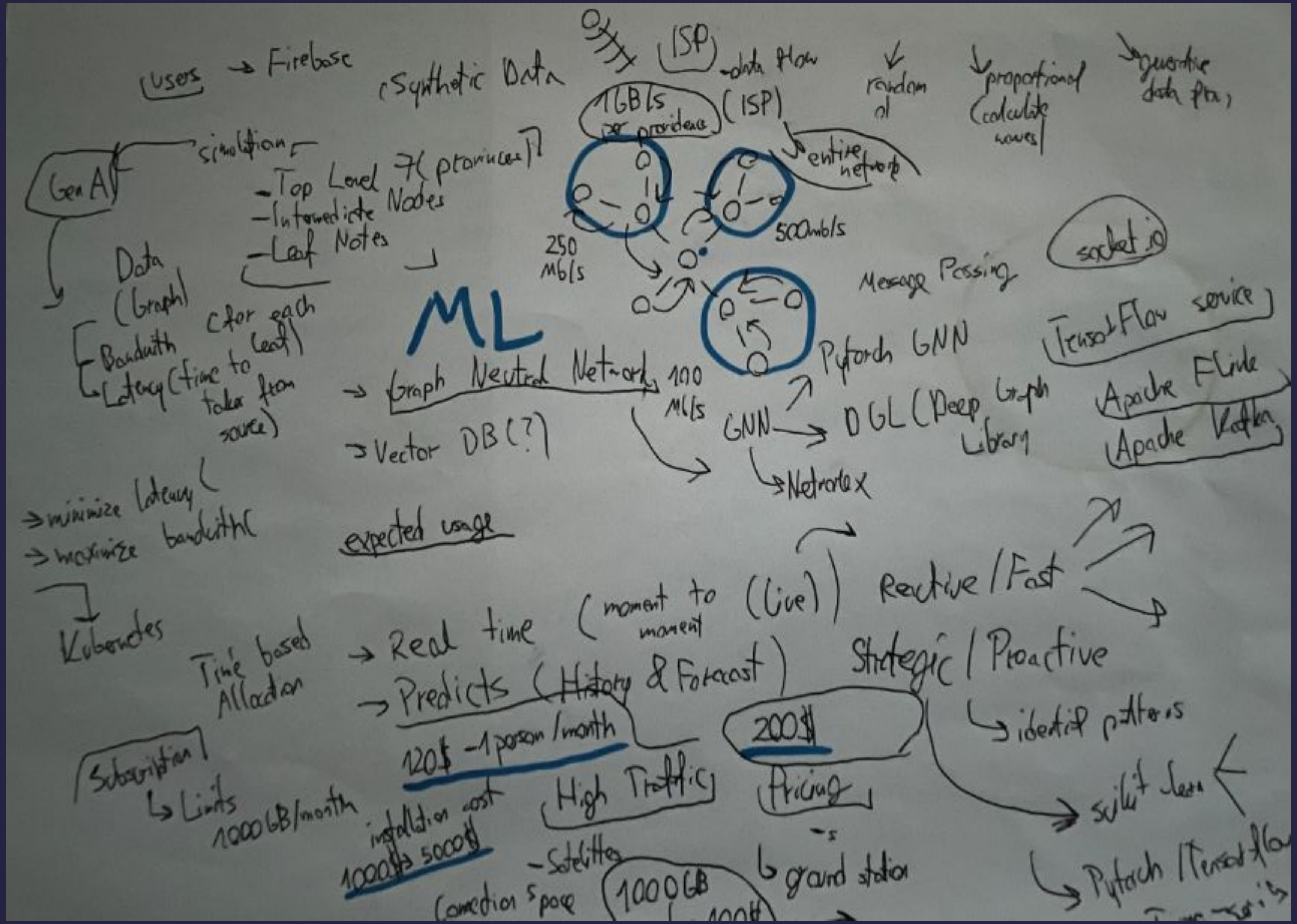
Internet Speed:

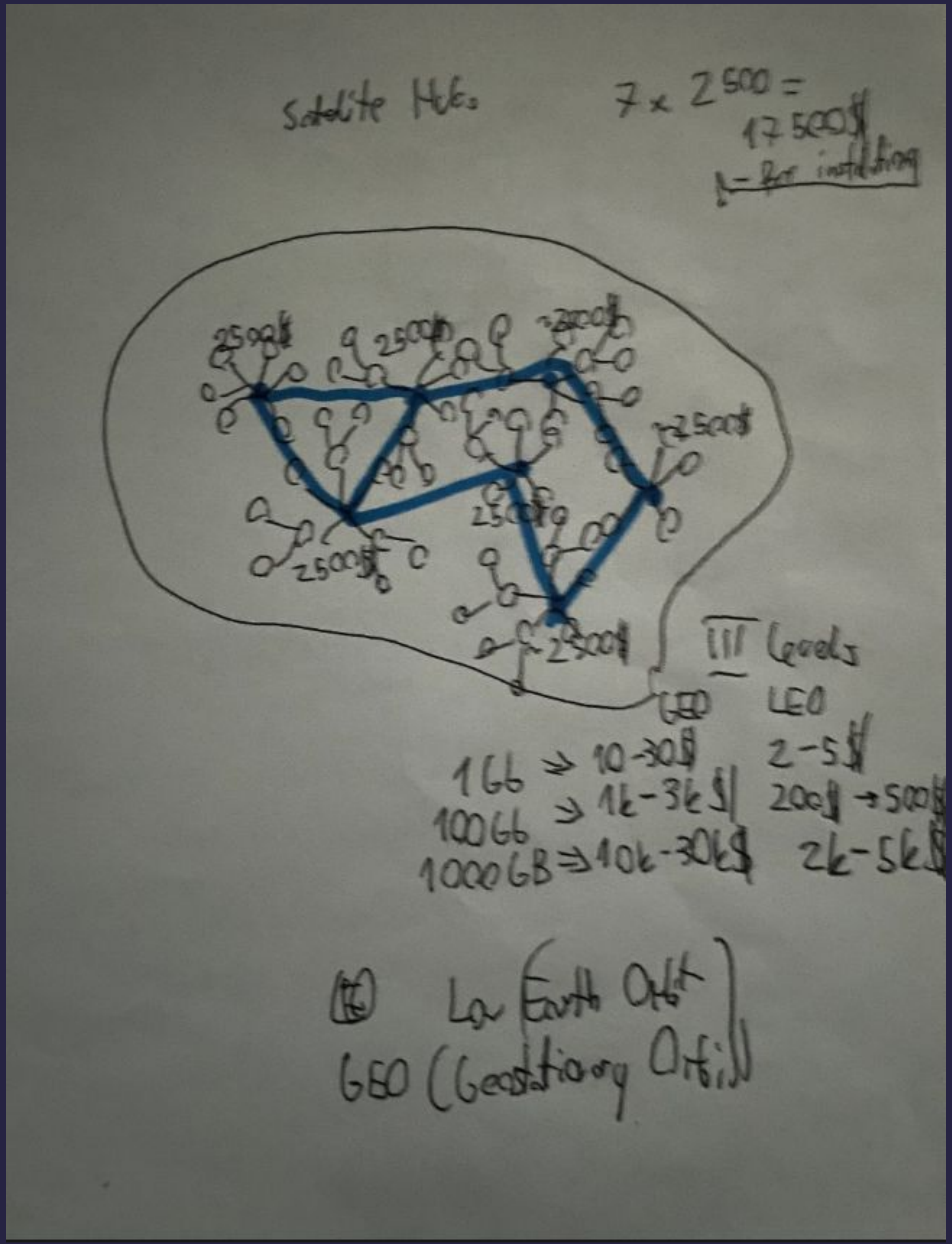
- https://www.kaggle.com/discussions/general/219630

Geo Internet Use:

- https://www.kaggle.com/code/tumpanjawat/internet-use-geo-cluster-time-series/notebook
- https://www.kaggle.com/code/aminawasiq/introduction-to-geospatial-data

Analysis

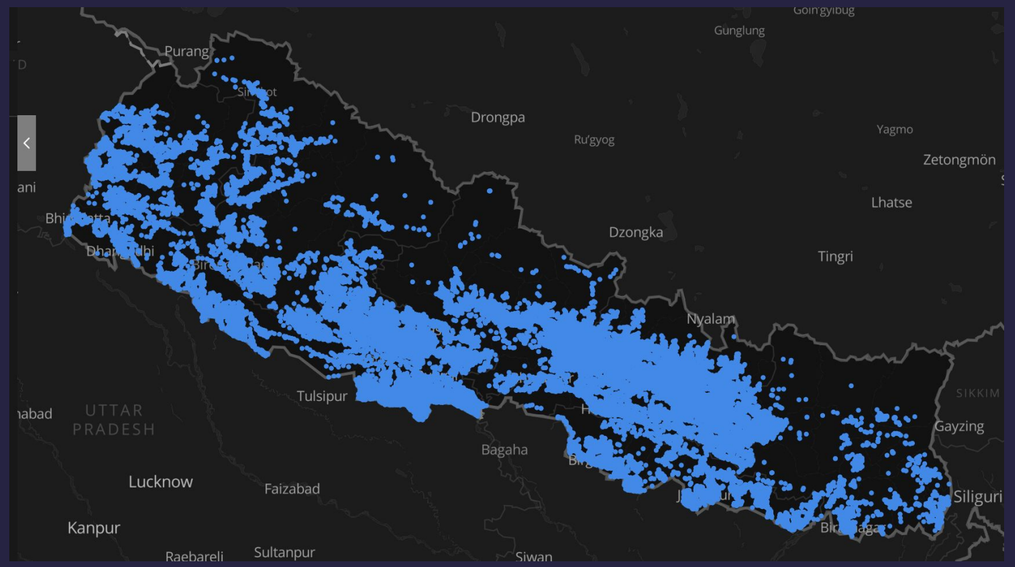

Users → Firebase
Synthetic Data
ISP
data flow
(ISP)
1GB/s per province
random
proportional (calculate waves)
entire network
Gen AI
simulation
Top Level 7 (provinces)
Intermediate Nodes
Leaf Notes
250 Mb/s
500mb/s
socket.io
Message Passing
ML
Data (Graph)
Bandwidth
Latency (time to take from source)
for each leaf
Graph Neural Network
100 Mb/s
Pytorch GNN
TensorFlow service
Vector DB (?)
GNN
DGL (Deep Graph Library)
Apache Flink
Apache Kafka
minimize latency
maximize bandwidth
expected usage
Kubernetes
Real time (moment to moment) (Live)
Reactive / Fast
Time based Allocation
Predicts (History & Forecast)
Strategic / Proactive
identify patterns
120$ - 1 person / month
200$
Subscription
Limits
1000GB/month
installation cost
High Traffic
Pricing
Satellite
ground station
1000GB
Pytorch / Tensorflow


Satellite Hubs
7 x 2500 =
17 500$
For installing
2500$
III Levels
GEO
LEO
1Gb → 10-30$
2-5$
100Gb → 1k-3k$
200$ → 500$
1000GB → 10k-30k$
2k-5k$
Low Earth Orbit
GEO (Geostationary Orbit)

t

Key numbers

- **217** nodes (in 3 level hierarchy) (in network)
- **1$** -> **137,93** NPR
- **250 Mbps** -> 1 Month **1,450 NPR**, 3 Month **3,600 NPR**, 12 Month **12,600 NPR**
- **300 Mbps** -> 1 Month **1,550 NPR**, 3 Month **3,600 NPR**, 12 Month **13,800 NPR**
- **10,000** locations throughout Nepal. Currently, there are over **14,000** of these Wi-Fi hotspots across Nepal.



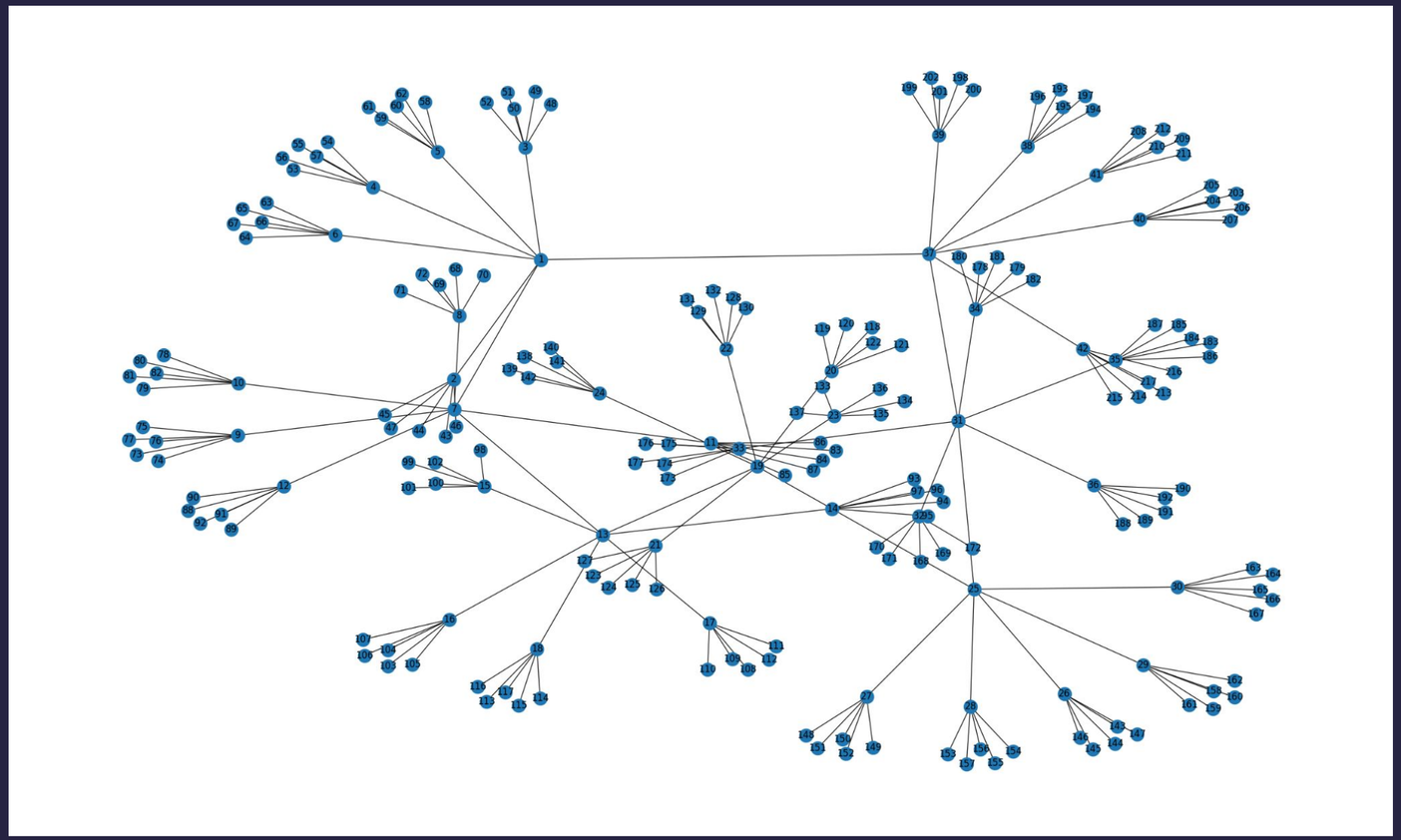
Technology

Key Features:

- Swarm Intelligence in ISP & SNP
- Automated traffic
- Predictive maintenance
- Real time monitoring

Tools:

- NetworkX
- GNN
- Kafka
- Prometheus + Grafana



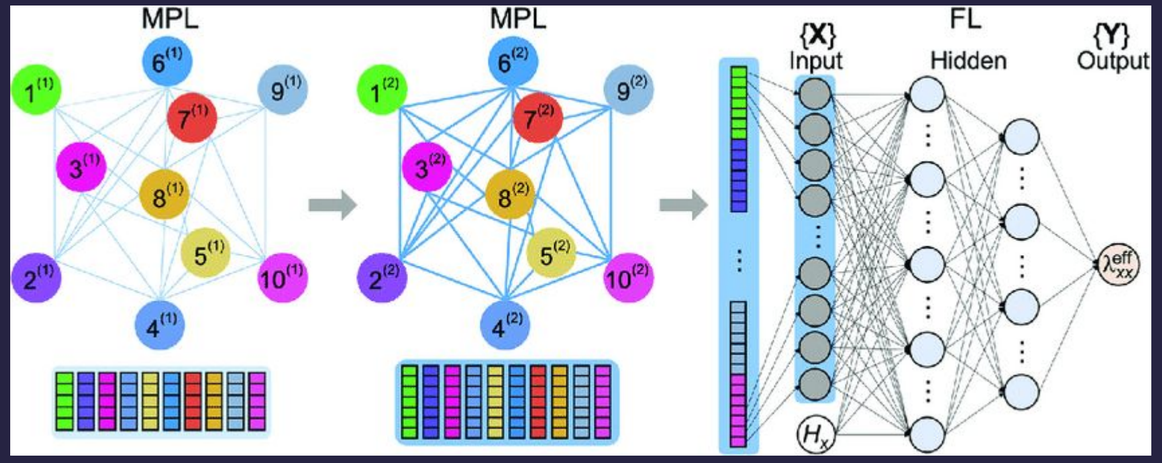
t

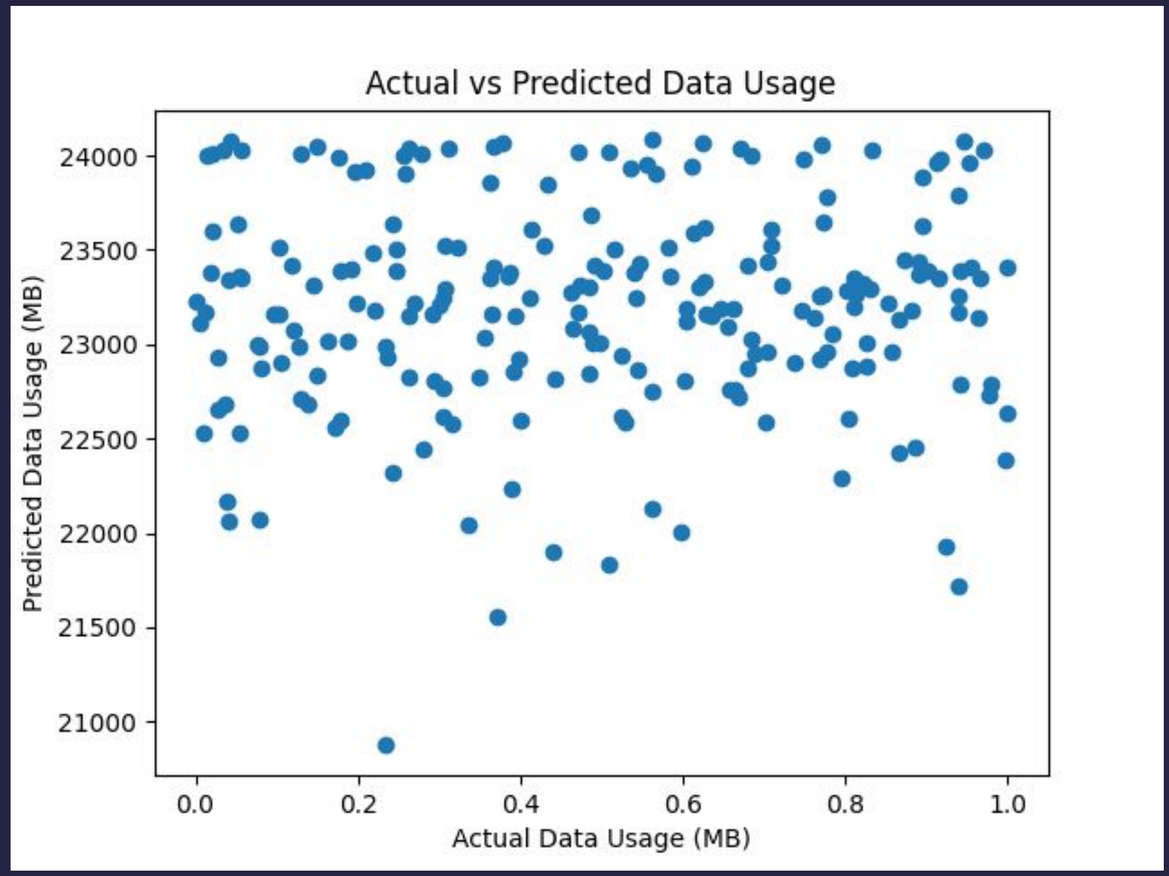
GNN (Generational Neural Network)

For GNN there is need collecting history data per node in the **features**:

- mb_data_usage_last_quarter (1-15min)
- mb_data_usage_last_two_quarter (15-30min)
- mb_datA_usage_last_three_quarter (30-45min)

Prediction label is: Predicted Data Usage for Next Quarter (15min) (MB)





t

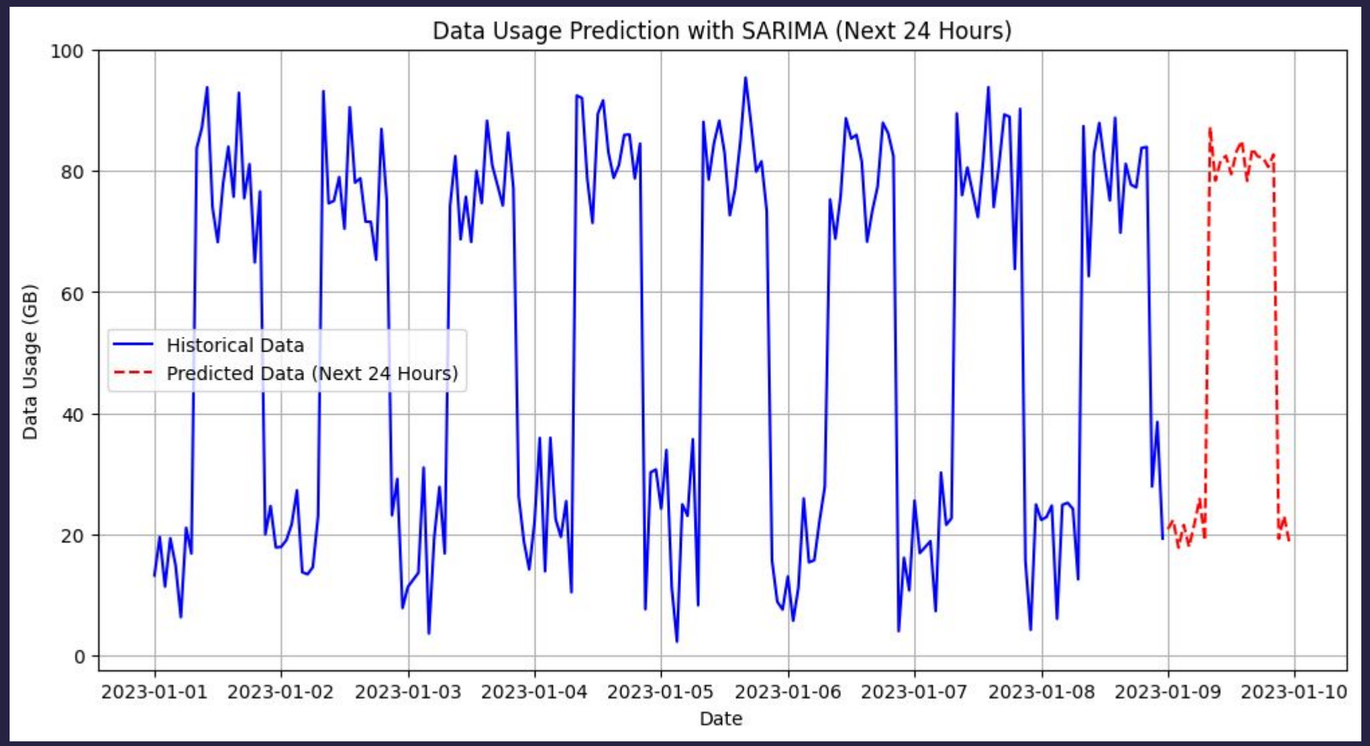
Time Series

The another approach was to predict next 24h of usage in Google Colab with Sarima.

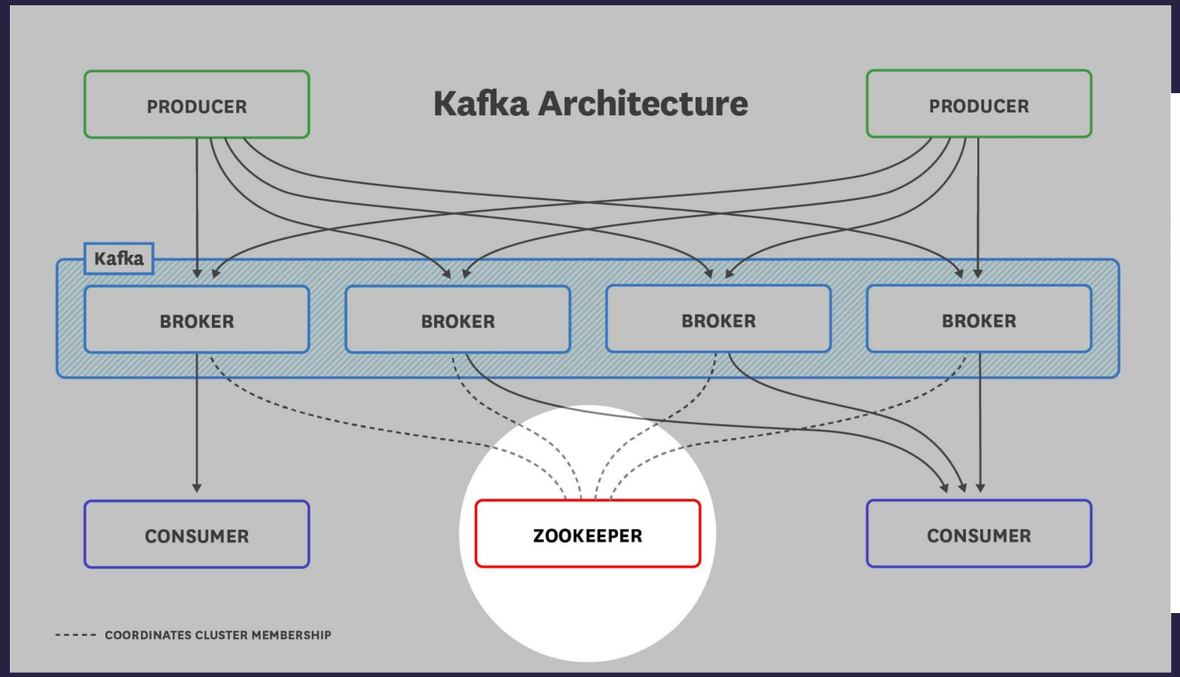
This data was concentrated in diagram within one province (Arun)

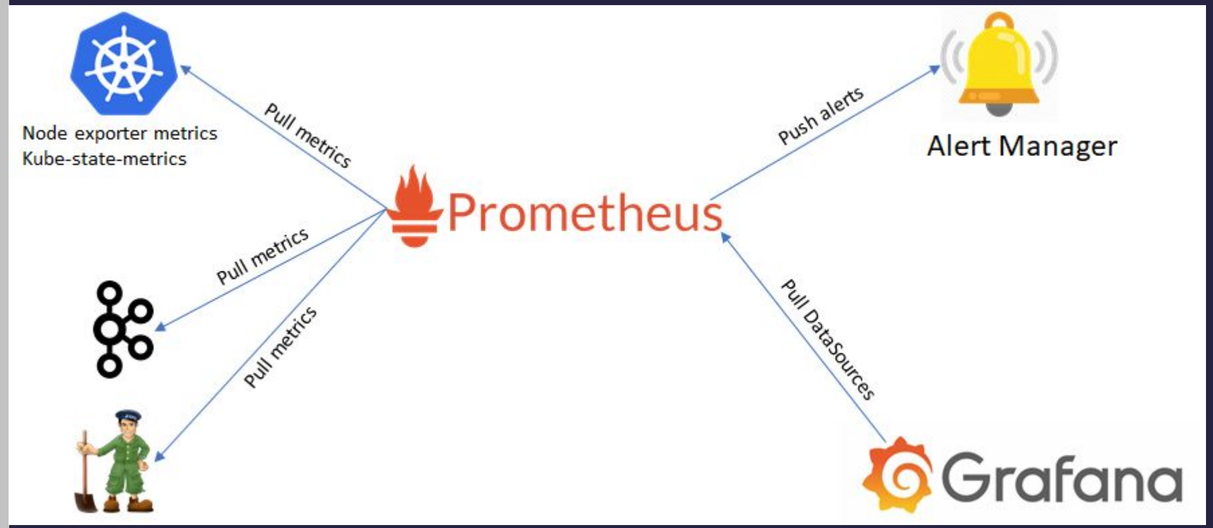
```
# node_name,
# data_usage (MB),
# bandwidth,
# latency,
# neighbors

nodes_data = [
    ("Arun", 120, 50, 10, 7),
    ("A1", 70, 30, 7, 6),
    ("A1_Point1", 20, 20, 5, 1),
    ("A1_Point2", 20, 20, 5, 1),
    ("A1_Point3", 20, 20, 5, 1),
    ("A1_Point4", 20, 20, 5, 1),
    ("A1_Point5", 20, 20, 5, 1)
]
```



Monitoring
(Kafka + Prometheus + Grafana)





Real time communication between Nodes