Long Project 2024

Data Analytics and Collaborative Learning for Urban Mobility and Geospatial Characterization

03/14/2024

ENSEEIHT

THE N7 TEAM

HPC BigData
Brenda Tonleu

HPC BigData
Briag Rehel

Maxence Flaba
HPC BigData

Ying Liu
Multimedia

THE TOPIC

Data Analytics and Collaborative Learning for Urban Mobility and Geospatial Characterization

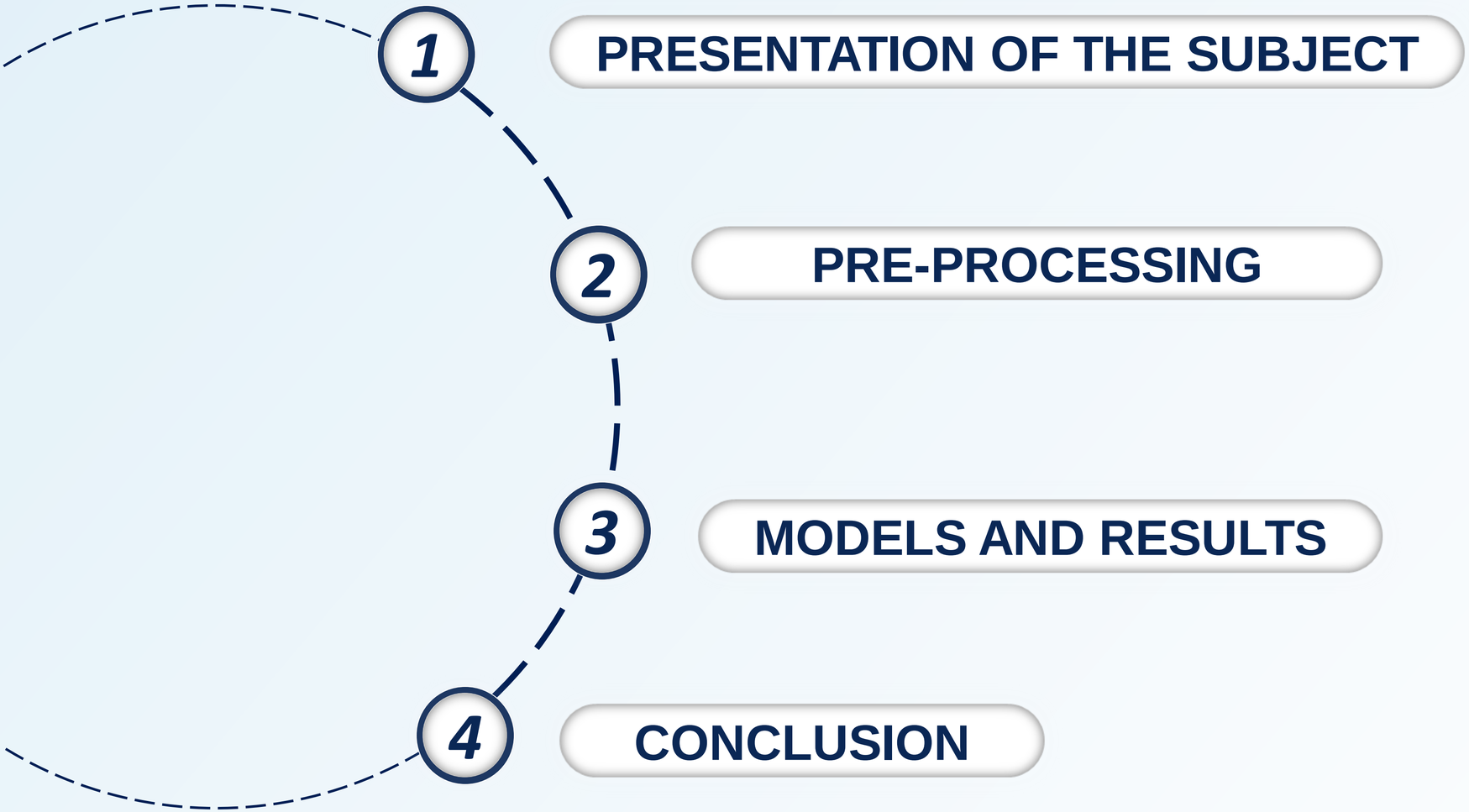
Instructor
Guilherme Iecker Ricardo

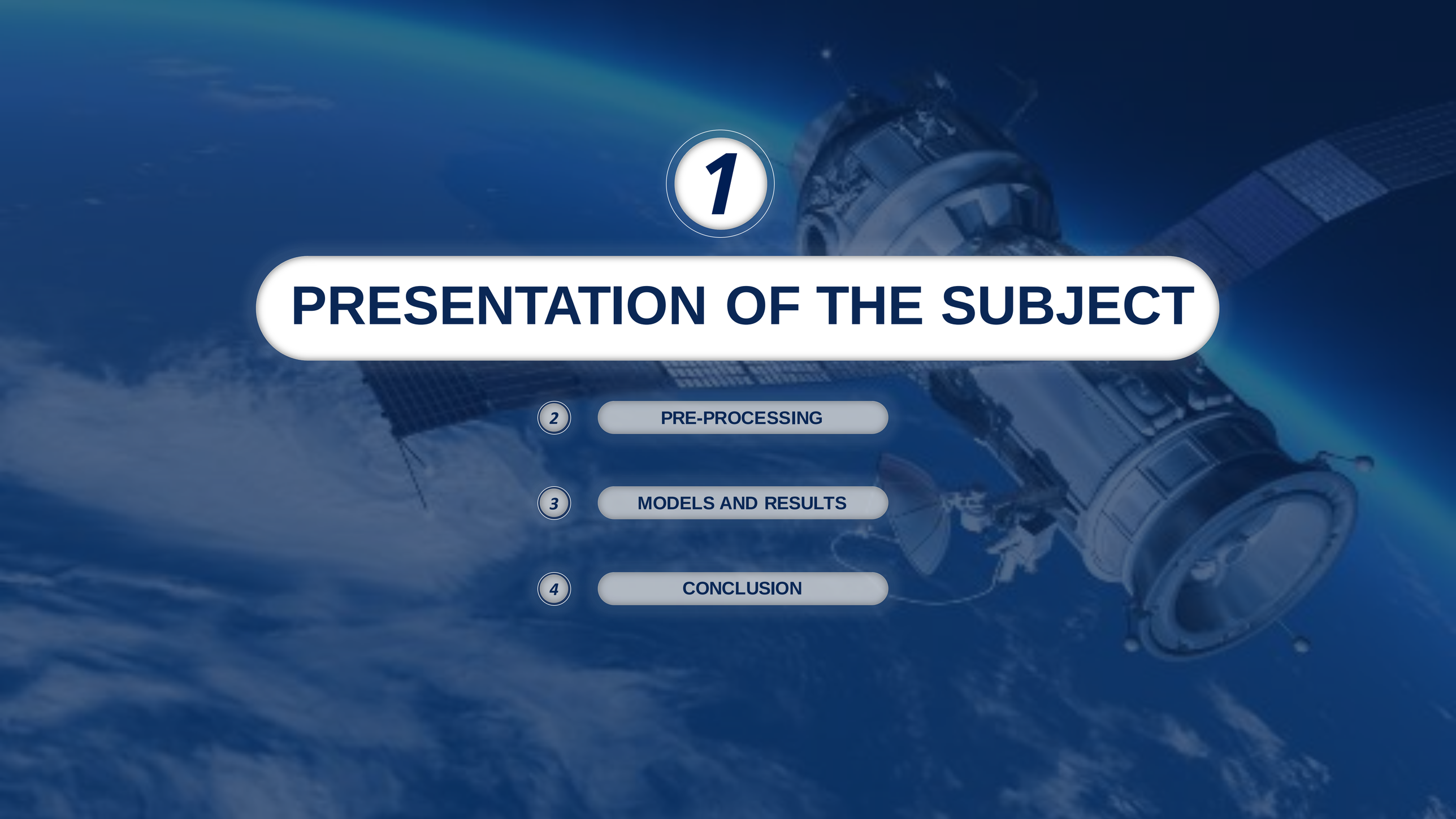
GAIN INSIGHTS INTO URBAN MOBILITY PATTERNS

GEOSPATIAL CHARACTERIZATION

AID URBAN PLANNING AND OPTIMIZATION

EXECUTIVE SUMMARY





1 PRESENTATION OF THE SUBJECT

2 PRE-PROCESSING

3 MODELS AND RESULTS

4 CONCLUSION

WHAT IS AT STAKE ?

> *Those who control the **present**, control the **past** and those who control the **past** control the **future**.*
>
> - George Orwell, *1984*

Global Surveillance ***Private Life*** ***Free will***

Use for economic and political purposes to manipulate us

Use for humanitarian and epidemiological purposes to limit natural disasters or predict the evolution of a virus

A CONTROVERSIAL SUBJECT ?

Example

Using the trace function to contain the COVID 19 pandemic

Reduce consumption of energy in future cellular networks

Innovate to reduce the impact of mobility and transport by optimizing traffic

30%

The contribution of transport and mobility to global CO2 emissions (IPCC)

REFERENCE PAPERS

Understanding Mobile Traffic Patterns of Large Scale Cellular Towers in Urban Environment

5 Different geographical contexte of traffic patterns

- **Resident / Transport**
- **Office / Entertainment**
- **Comprehensive**

Idea: *Their model combines time, location and frequency information*

What Machine Learning Predictor Performs Best for Mobility Prediction in Cellular Networks?

4 Different predictors on the prediction of human mobilty patterns

- **XG Boost / SVM / DNN / Semi-Markov**

Best predictor : XG Boost **90.22%** accuracy and **20s** time complexity

Issue*: training/testing not done on real networks*

THE DATASET

3233 stations

9481 different users

6 Months of data from Shanghai

7.2 M Different connexions

2/3

Of users have less than 7 connectionx in 6 months

~50%

Of users only connected once in 6 months

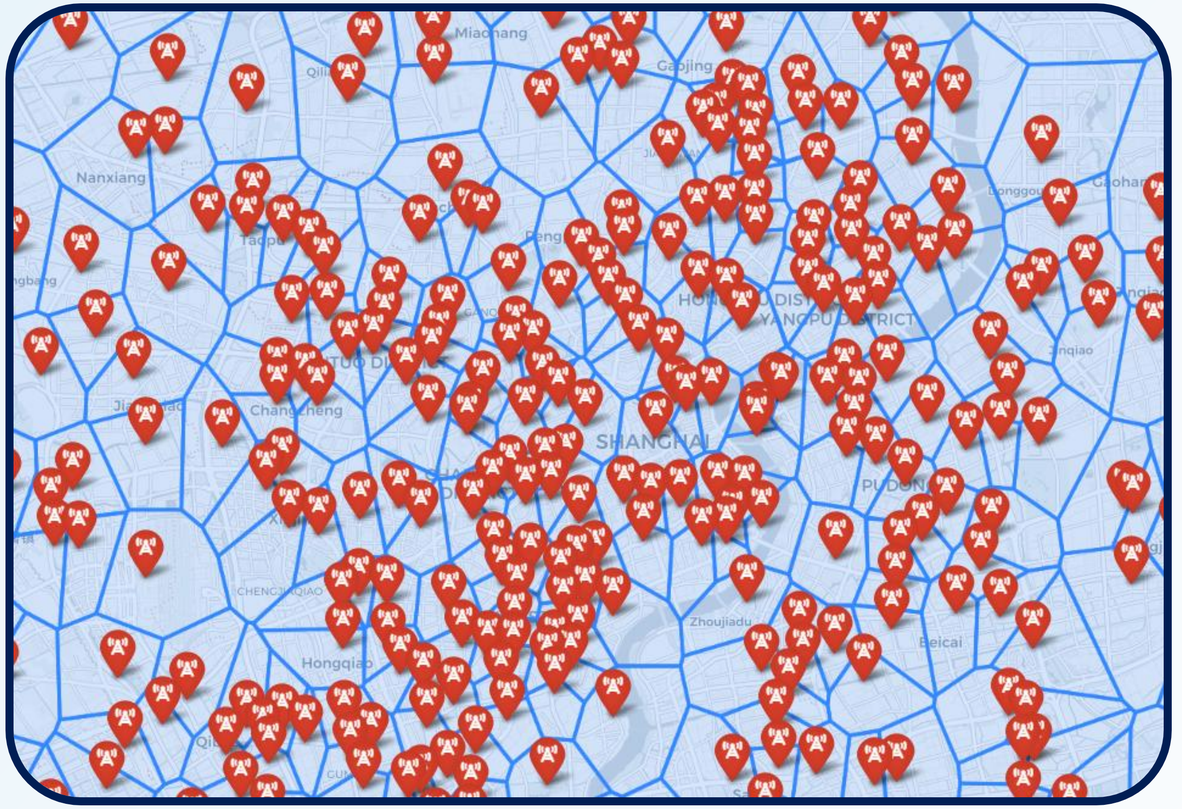
Privacy

The dataset is only meant for improvement of performances of telecom only

The dataset is anonymised using hash instead of user_id

It only contains necessary information

THE MAP



Map of the 500 most connected stations with its Voronoi diagram

WORK ORGANISATION

1 Pre-processing

2 Machine Learning Models

3 Conclusion

Tools used for project management

STUDY OF THE USER BEHAVIOR

Can we label the use of one station only using their connection pattern ?

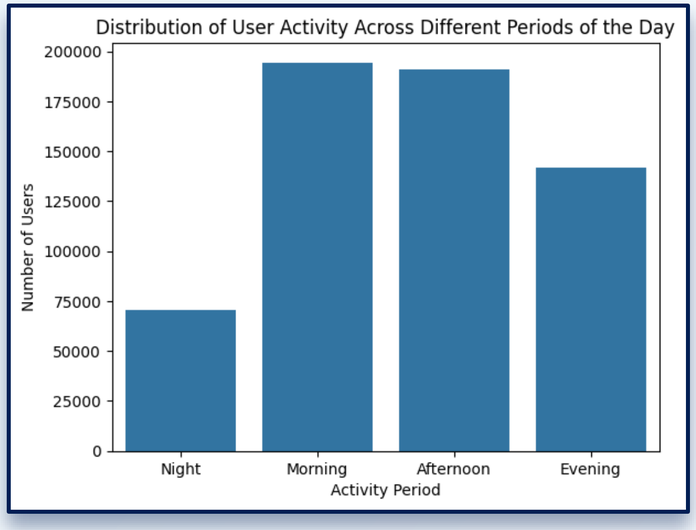
Number of User data : 5 695
Number of location : 2 771

```
All location encodings (vocab)
(30.935314, 121.733322) : 0
(31.144812, 121.119606) : 1
(31.381763, 121.301878) : 2
(31.056648, 121.404326) : 3
(31.100034, 121.663489) : 4
(31.040378, 121.255563) : 5
```

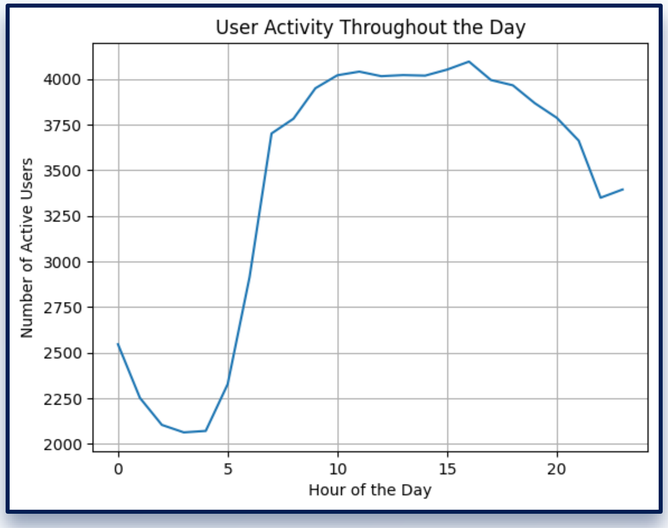
example of location encoding

```
User 774:
pos_count: [2, 2, 2, 2, 2]
hour: [14, 14, 14, 22, 22]
period: ['Afternoon', 'Afternoon', 'Afternoon', 'Evening', 'Evening']
pos_id: Shape = torch.Size([5])
pos_id_target: Shape = torch.Size([5])
input: Shape = torch.Size([5, 12])
time_target: Shape = torch.Size([5, 2])
```

example of user data



Night: 0am – 6am
Morning: 6am - 12pm
Afternoon: 12pm - 18pm
Evening: 18pm - 12am



The objective is to find relevant insights to add feature into our machine learning model : station_type

Users connection ratio for different location in different during the week

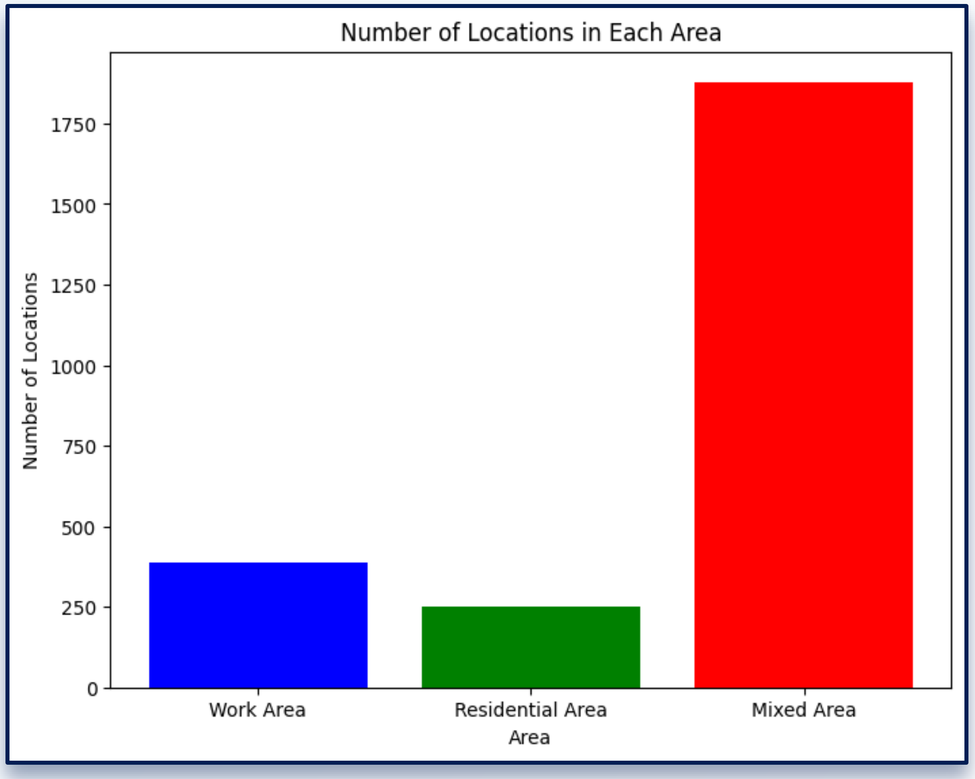
```
Frequency of activity for each location during different time periods (week):
period      Night    Morning  Afternoon     Evening
pos_id
0       11.445783  21.084337  43.373494   24.096386
1        6.521739  23.913043  45.652174   23.913043
2       11.594203  23.188406  28.985507   36.231884
3        5.319149  32.978723  40.425532   21.276596
4        8.108108  29.729730  35.135135   27.027027
```

Users connection ratio for different location in different periods during the weekends

```
Frequency of activity for each location during different time periods (weekends):
period      Night    Morning  Afternoon     Evening
pos_id
0       37.837838  27.027027  16.216216   18.918919
1        1.562500  29.687500  26.562500   42.187500
2       21.052632  47.368421  26.315789    5.263158
3       18.000000  24.000000  32.000000   26.000000
4        6.521739  56.521739  26.086957   10.869565
```

Algorithm 1: Threshold

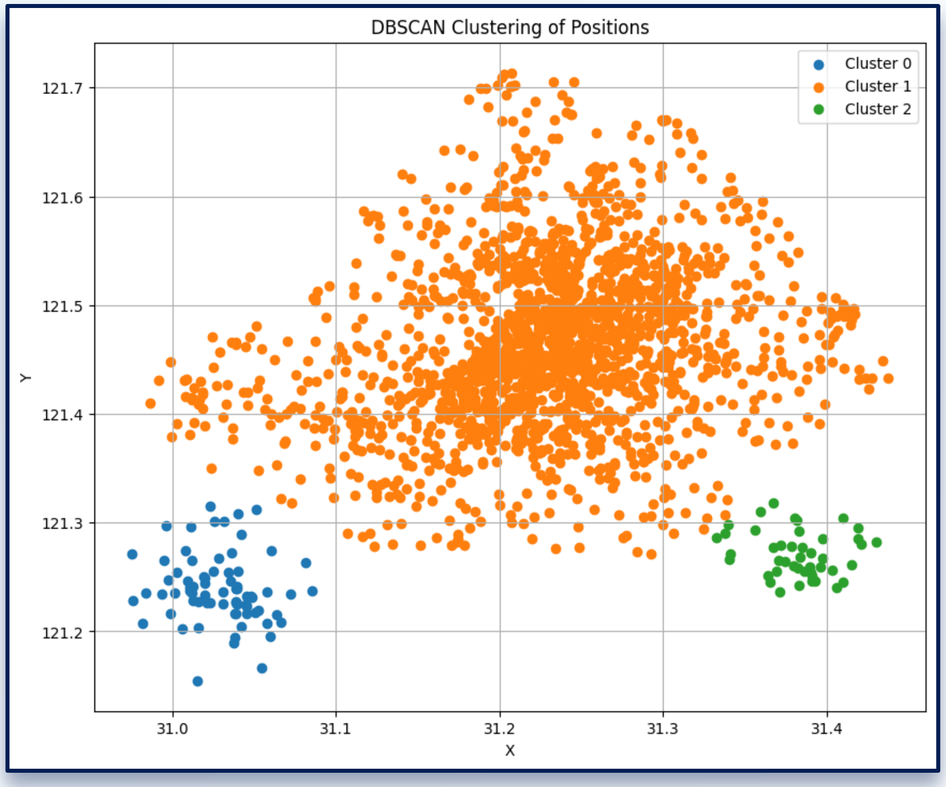
- 3 different locations : Work Area, Residential Area and Mixed Area
- 4 different time : Night, Morning, Afternoon, Evening
- **Work Area:**
  - > 40% Morning & Afternoon
  - < 30% Evening & Night
  - Maximum proportion during weekends < 30%
- **Residential Area:**
  - < 50% Morning & Afternoon
  - > 40% Evening & Night
  - Proportion of Morning and Afternoon during weekends is higher than during the week
- **Mixed Area:**
  - Otherwise

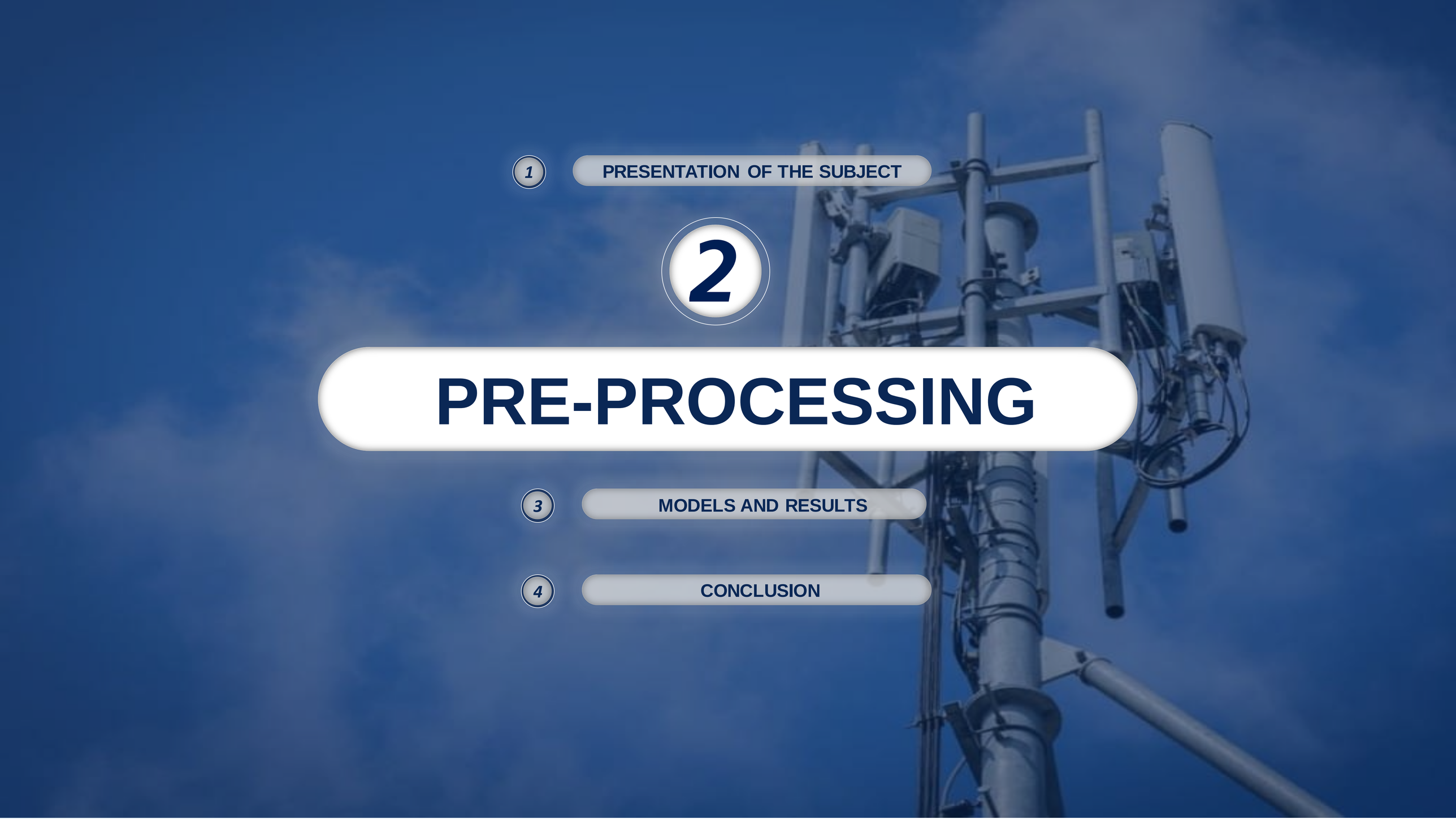



```
each area number:
Mixed Area: 1879
Residential Area: 252
Work Area: 387
```

Algorithm 2: DBSCAN

- **Parameters:**
  - Maximum distance of clustering = 0.05
  - Minimum number of samples = 35
- **fit_predict()**: cluster the standardized location data





1 PRESENTATION OF THE SUBJECT

2 PRE-PROCESSING

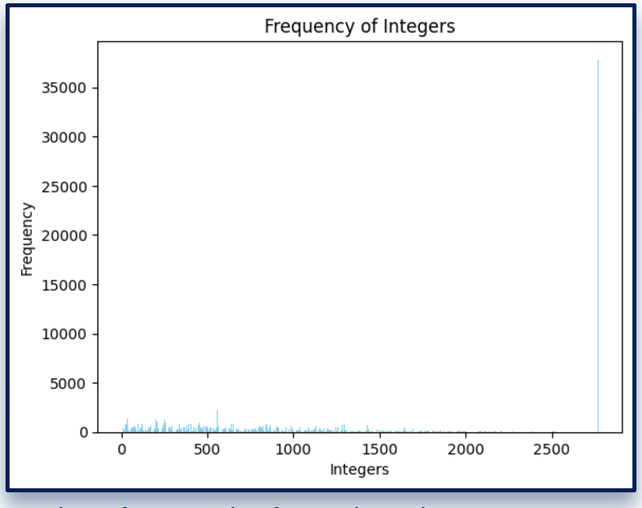
3 MODELS AND RESULTS

4 CONCLUSION

15 DAYS DATASET

The data pre-processing carried out on this dataset comprises 8 steps

- **Handling missing values** — Replacing missing positions with the previous one
- **Standardization of coordinates**
- **Creation of user sequences** — Drop first step until first connection; Drop sequence with at most one missing positions
- **Position encoding (pos_id)** — 20% threshold
- **Creation of additional insights**
- **Time encoding**
- **Target creation**



number of connection for each station

↗ ***Better results***

Loss of classic LSTM training from 18.84 to 7.76

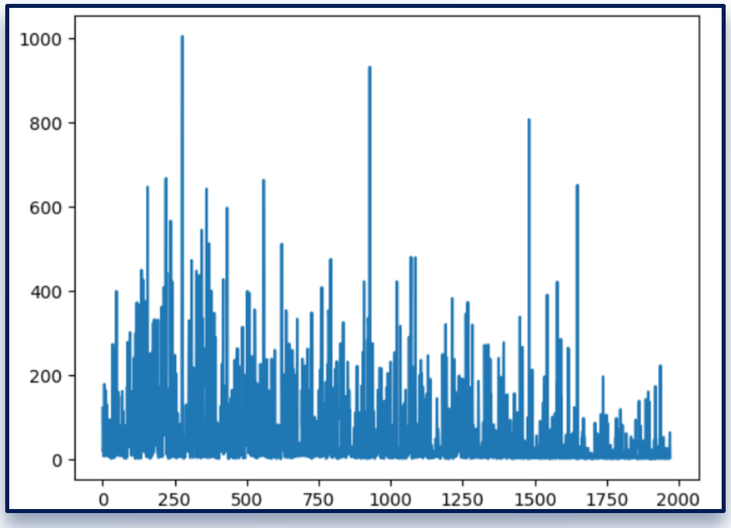
↗ ***Improvement***

Add learning rate scheduling to avoid overfitting

6 MONTHS DATASET

Second Pre-Processing

Too many missing values → **Removing missing values** → 72% naive accurancy



number of connection for each station

Final Pre-Processing

Too many repeated connections → Merge repeated connections → 0% naive accurancy

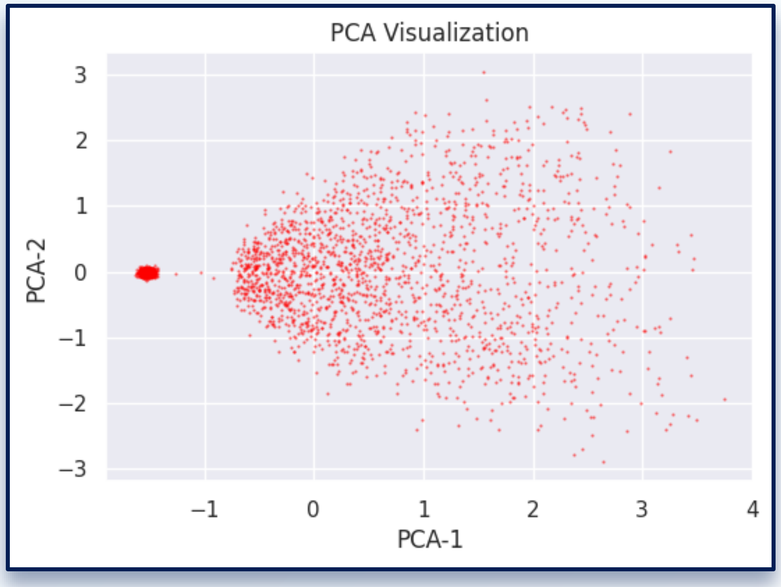
FEATURE ENGINEERING

OBJECTIVE: Enriching the dataset with geogtaphical structure and homophily

INPUT POSITION

- Directly with **coordinates** and station_id
- **Node embedding** using 2 different graphs :
  - Structured based on Voronoi : simple graph (no loops, no repeated edges)
    - 1 node = 1 station
    - 1 edge = 1 voronoi close to another
  - Using connections: directed edge-weighted looped graph
    - 1 node = 1 station
    - 1 edge from A to B = a user has been connected to B right after A

No gain on accurancy



node embedding

TIME EMBEDDING

- Fixed and learned

No time embedding done with graphs

1. **Map** with the cluster centered on the shanghai old city
2. **Link** with the user behaviour of Ying
3. **Improvement** : add weight on nodes to remove loops

1 PRESENTATION OF THE SUBJECT

2 PRE-PROCESSING

3

MODELS AND RESULTS

4 CONCLUSION

UPGRADING THE MODEL

Adding embedding

- A vector linked to the position of the station
- The more they are closed, the more their vector are the same

Modified architecture with a linear regression layer

Sequencing classification on 10 epoch

- Then both classificaiton & regression on 10 epochs

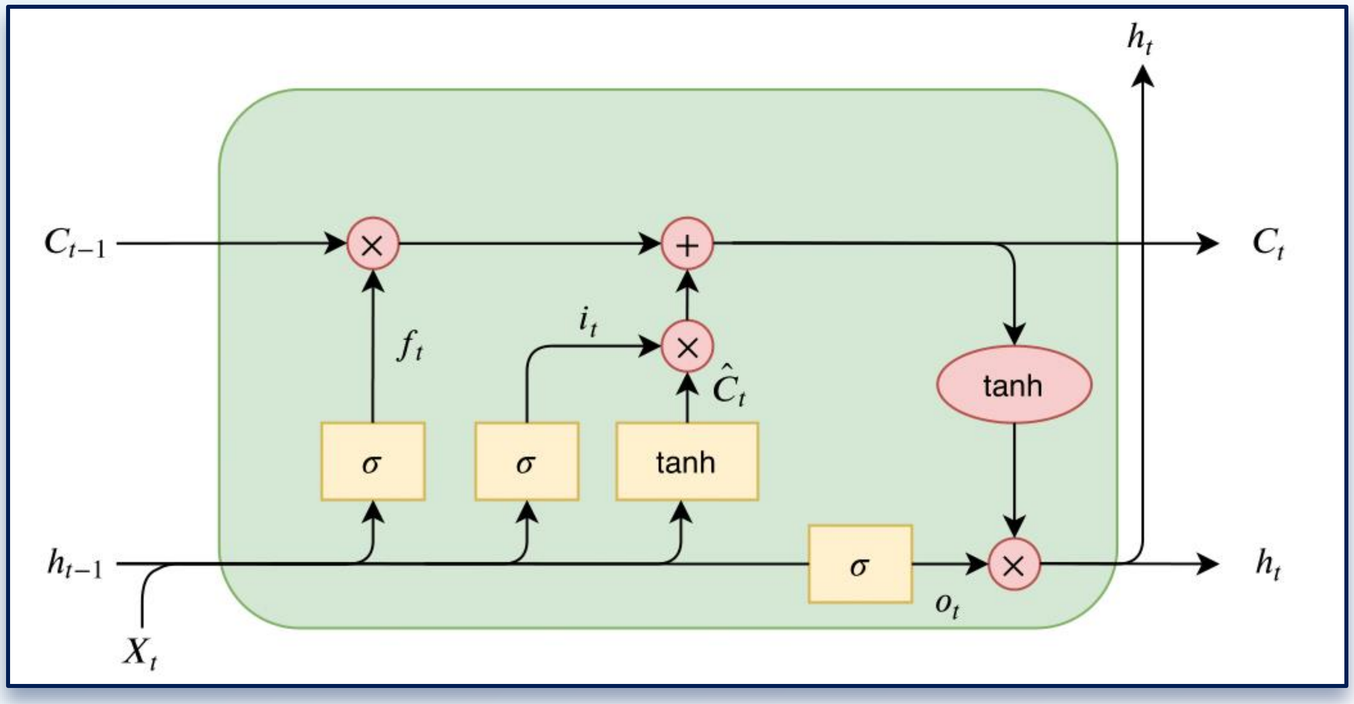
Training: maximal of batch
No impact of the learning rate scheduler on the accurancy

FIRST MODEL: LSTM

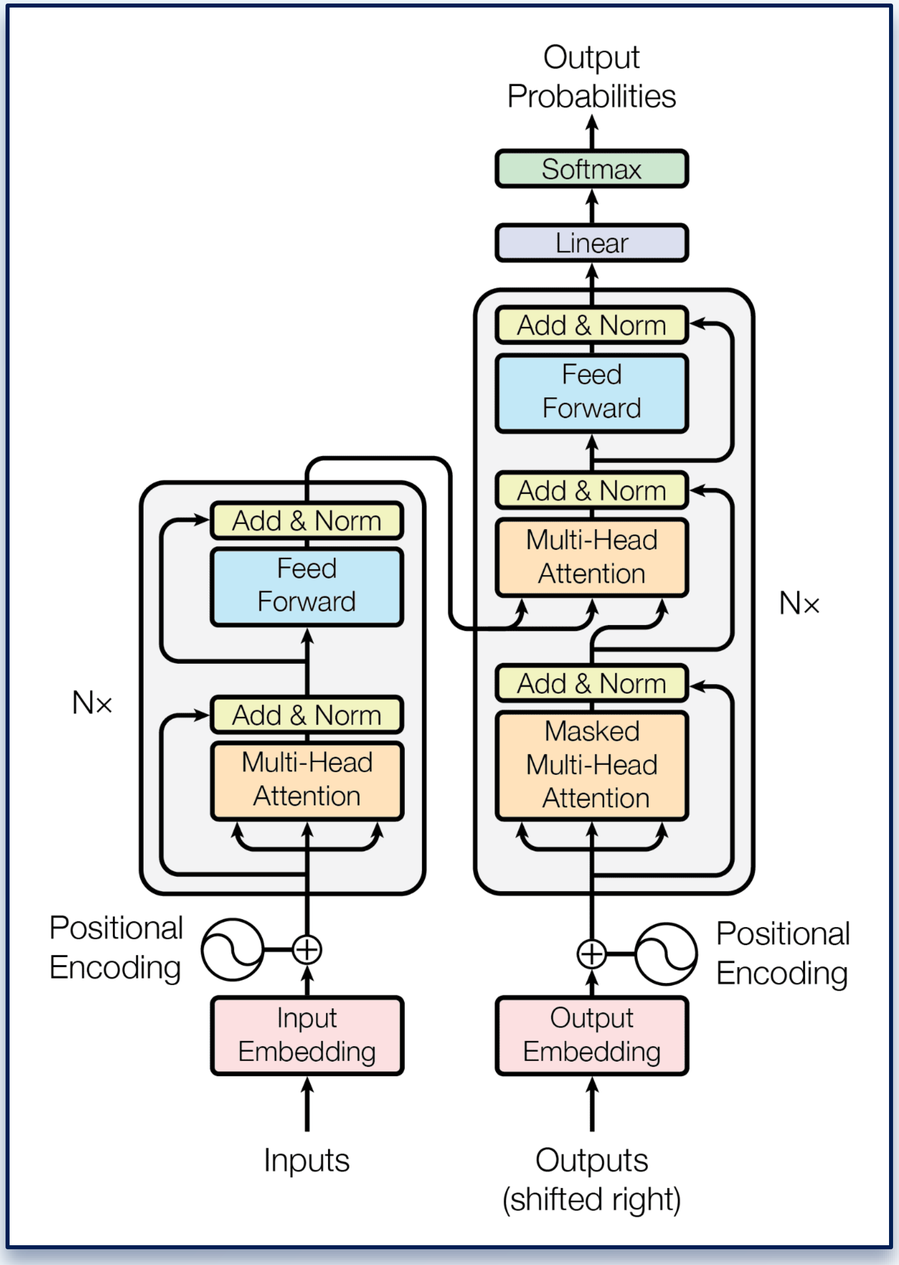
Classification & **Regression** at the same time

Using 2 layers **LSTM** (RNN)

PROBLEM: Doing both at the same lead to **results that can be improved**



TRANSFORMERS



RESULTS

Features

Second version of pre-processing : **before** merging connections and without missing values

Valid Accurancy

Naive
72.20%

LSTM
70.75%

Transformers
69.94%

90% wrong when the next station is not the previous one

HYPERPARAMETER TUNING

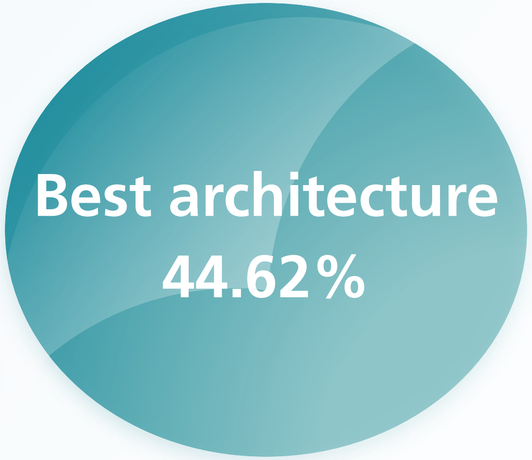
Problem:
We can’t compare each choice together

Solution:
Let the Hyperparameter search policy select the best feature
Hyperopt based on tree of parzen estimators

Features

Final version of pre-processing : **after** merging connections and without missing values

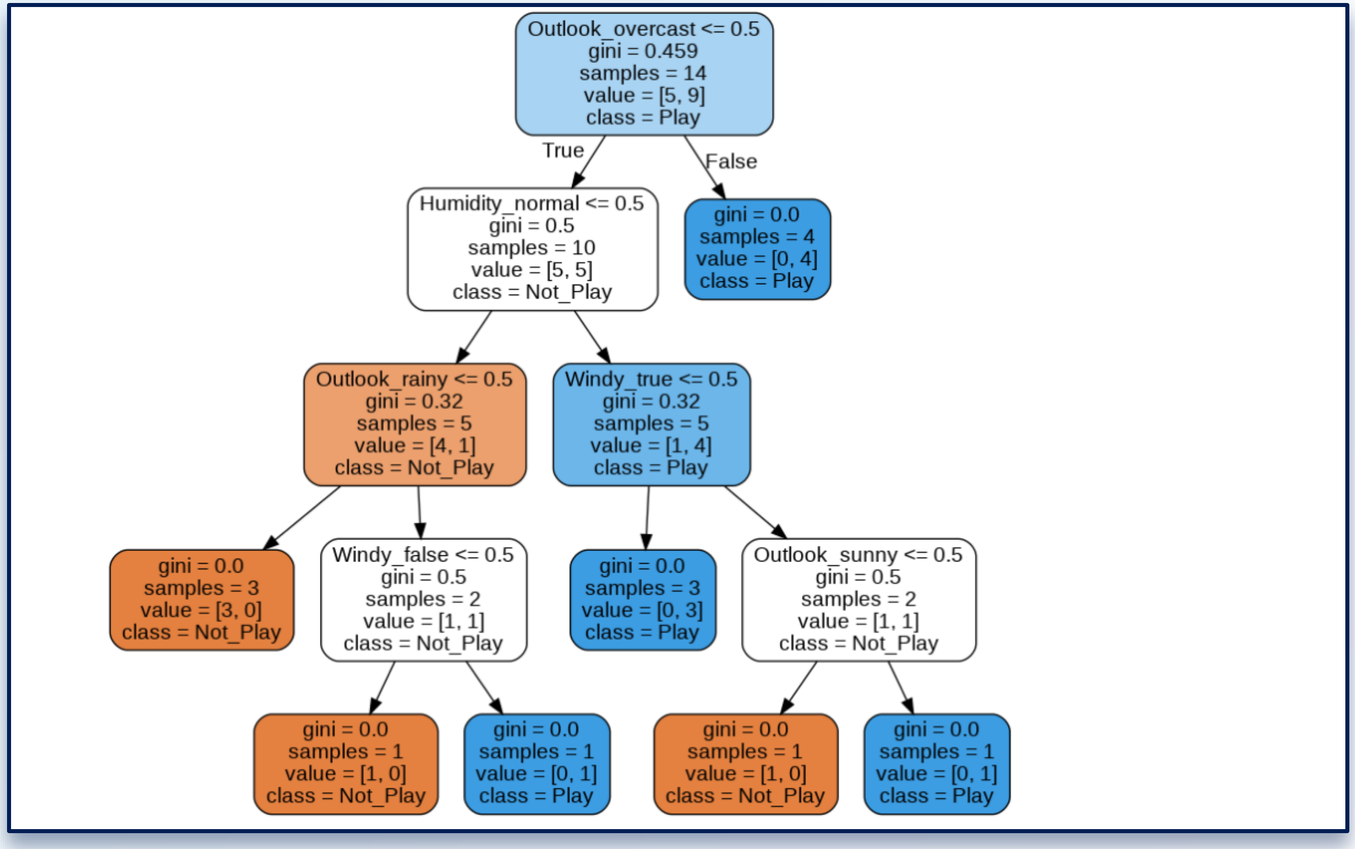
Valid Accurancy out 200 trials



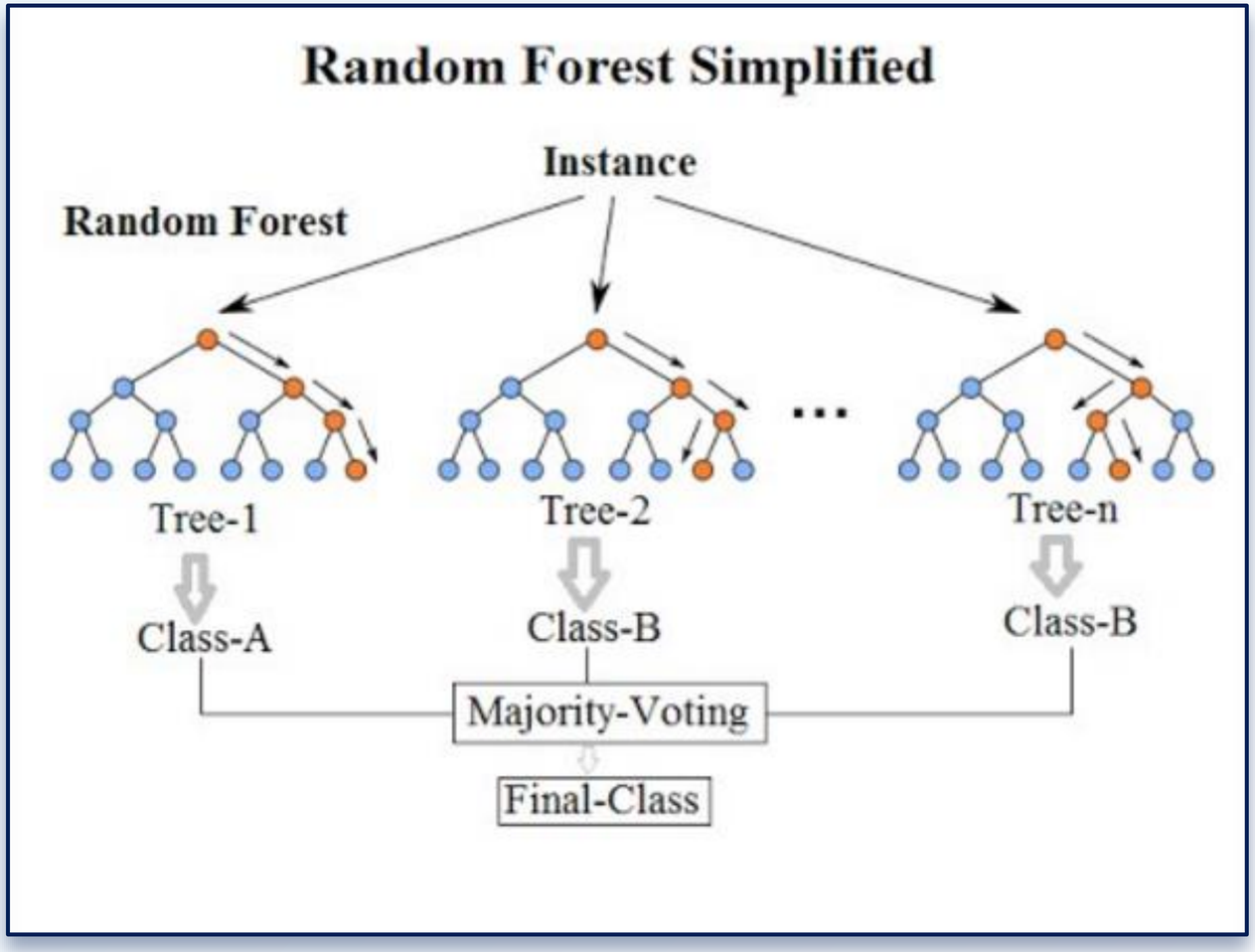
Tree-based models: Decision Tree, Random Forest

1. **Supervised learning** used for classification task
2. Built by **recursively** selecting the best feature to split the data based on impurity (gini or entropy)

Decision Tree classifier



Tree-based models: Decision Tree, Random Forest



Tree-based models: Decision Tree, Random Forest

Find the best parameters with **Grid Search CV**

Decision Tree

Criterion :
gini

Max depth :
25

Min samples split :
15

Splitter :
random

Random Forest

Number estimators :
50

Min samples leaf :
5

Max Features
0.7

Tree-based models: Decision Tree, Random Forest

Features

Second version of pre-processing : **before** merging connections and without missing values

Valid Accurancy

Decision Tree
69.43%

Random Forest
70.81%

93% wrong when the next station is not the previous one

Tree-based models: Decision Tree, Random Forest

Features

Final version of pre-processing : **after** merging connections and without missing values

Valid Accurancy

Decision Tree
37.29%

Random Forest
40.47%

OTHER TRIES AND IDEAS

- Using a pre-trained GPT-2 : 35% valid accuracy
- Curriculum learning

1 PRESENTATION OF THE SUBJECT

2 PRE-PROCESSING

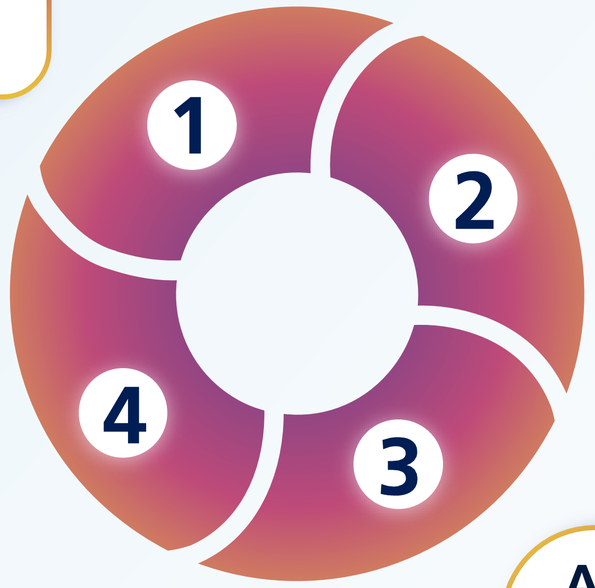
3 MODELS AND RESULTS

4

CONCLUSION

CONCLUSION

MIX OF LSTM AND TRANSFORMERS



44.62% ACCURACY

NOT EASY TO PREDICT HUMAN MOVEMENT

A LOT OF VARIATIONS BETWEEN SEQUENCES

QUESTIONS