



fit@hcmus

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG - HCM
KHOA CÔNG NGHỆ THÔNG TIN

[MTH084] - Học máy với dữ liệu đồ thị

Đề tài 01: Tối ưu hoá đường đi cho xe điện

GVHD

GS.TS Lê Hoài Bắc

TS. Lê Ngọc Thành

ThS. Nguyễn Ngọc Đức

Học viên:

Vũ Công Minh - 23C11007

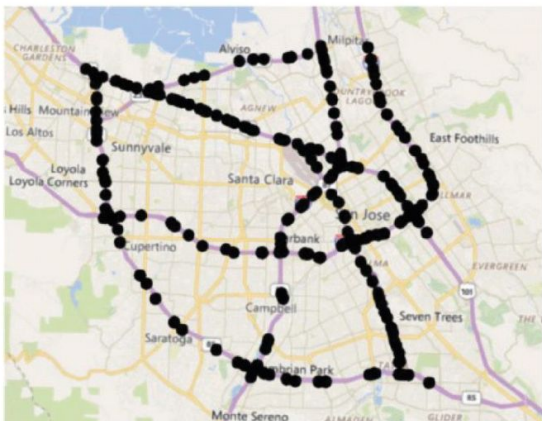
Mục lục

1. Giới thiệu bài toán
2. Cơ sở lý thuyết
3. Tập dữ liệu
4. Phương pháp thực hiện
5. Kết quả
6. Tổng kết

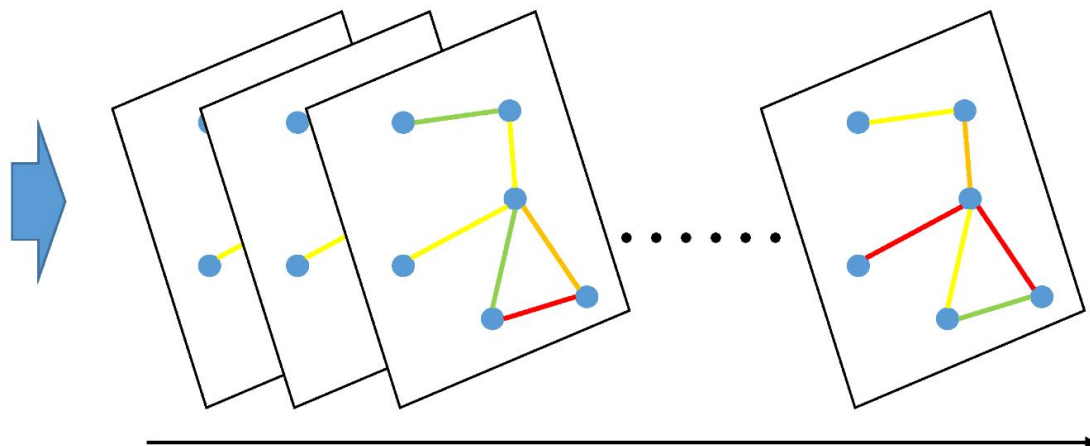
1. Giới thiệu

Sử dụng **GNN** nhằm **Tối ưu hoá** đường đi cho xe điện:

- + **Quãng đường** đi ngắn nhất
- + **Thời gian** di chuyển ngắn nhất
- + **Điện năng** tiêu hao ít nhất
- + (mở rộng: theo lưu lượng giao thông thời gian thực, khả năng tiếp nạp xăng trên đường đi,...)



Road Network with Traffic Sensors



Time

2. Cơ sở lý thuyết

DuETA (2022) Traffic Congestion Propagation Pattern Modeling via Efficient Graph Learning for ETA Prediction at Baidu Maps

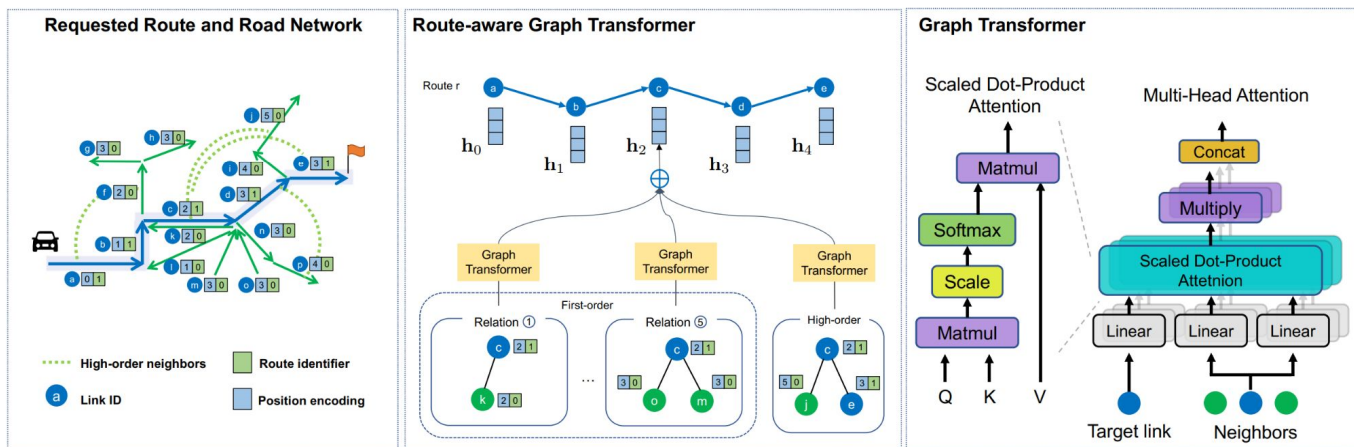


Figure 4: Architecture of the route-aware graph transformer.

Framework dự đoán thời gian di chuyển kết hợp thông tin tắc nghẽn với tương quan dài hạn giữa các đoạn đường (thông tin Thời gian + Không gian => Spatial-temporal GNN)

2. Cơ sở lý thuyết

Transformer-based Graph Neural Networks for Battery Range Prediction in AIoT Battery-Swap Services (2024) (GNN)

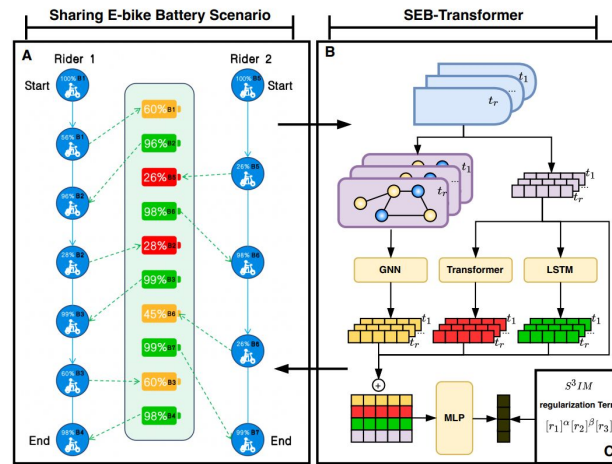
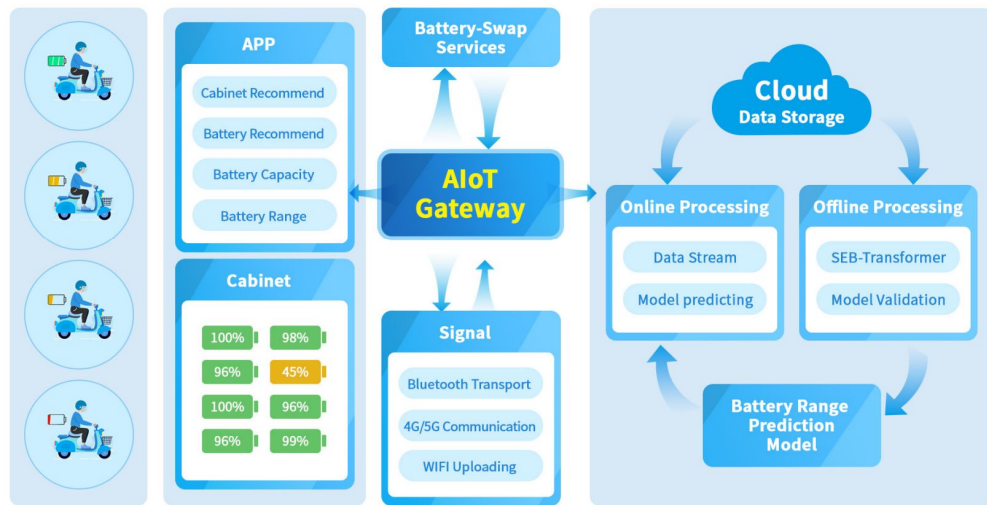
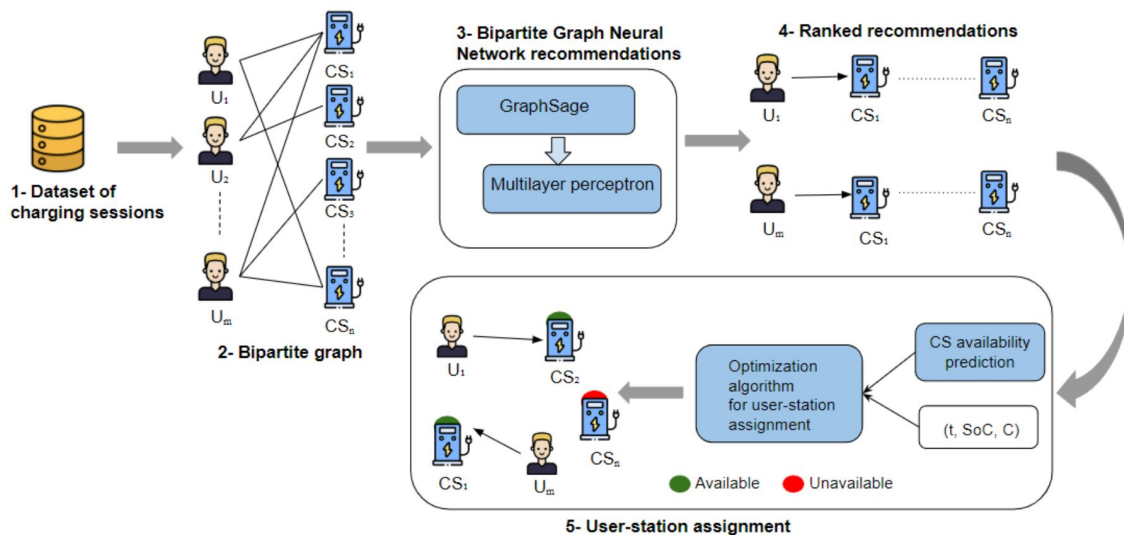


Fig. 2. Illustration of SEB scenario and SEB-Transformer.

Sharing E-Bike Battery (SEB)-Transformer cho bài toán dự đoán
lượng mức pin sử dụng

2. Cơ sở lý thuyết

Optimizing EV Charging Recommendations Using Graph Neural Networks (2024) (GraphSage)

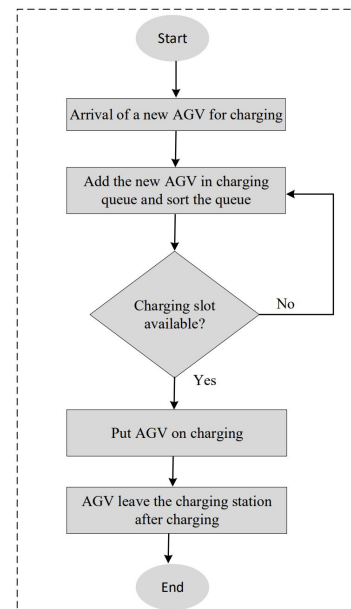
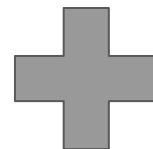
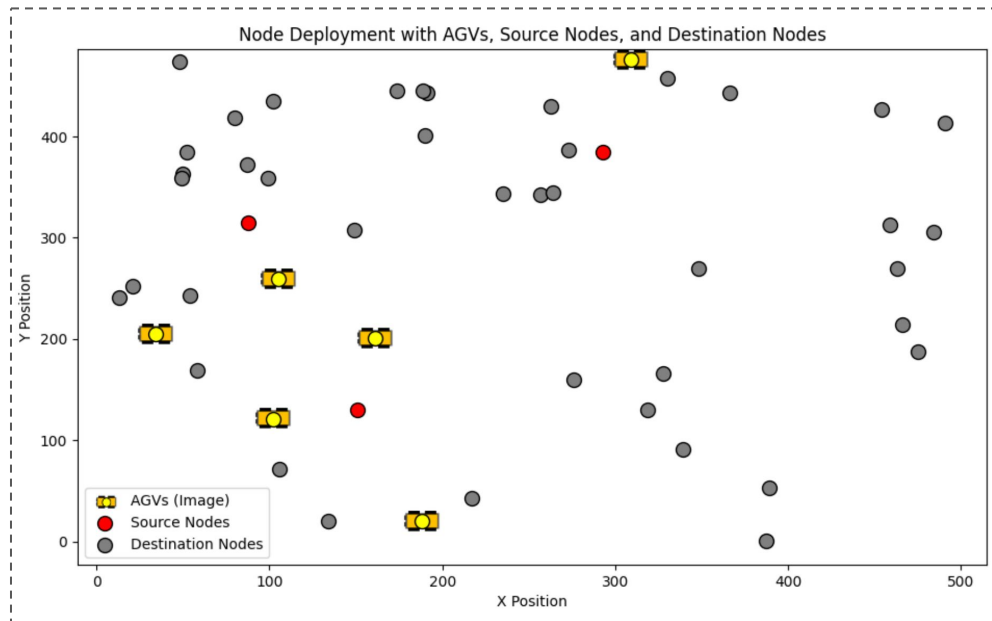


BipartiteSAGE: Huấn luyện với mô hình GraphSAGE (SAmple and aggreGatE) cho bài toán của đồ thị lưỡng phân (Bipartite Graphs) nhằm tối ưu trải nghiệm sạc nhiên liệu

2. Cơ sở lý thuyết

GRouteNet (2024)

A GNN-Based Model to Optimize Pathfinding and Smart Charging Management for Autonomous Guided Vehicles (GCN)



Cập nhật đồ thị theo thời gian thực

=> Thời gian xử lý lâu

=> Dữ liệu thực nghiệm nhỏ

=> Tối ưu vận hành logistics

<https://doi.org/10.3390/sym16121573>

Giảm thời gian “chờ đợi” một trạm điện/xăng

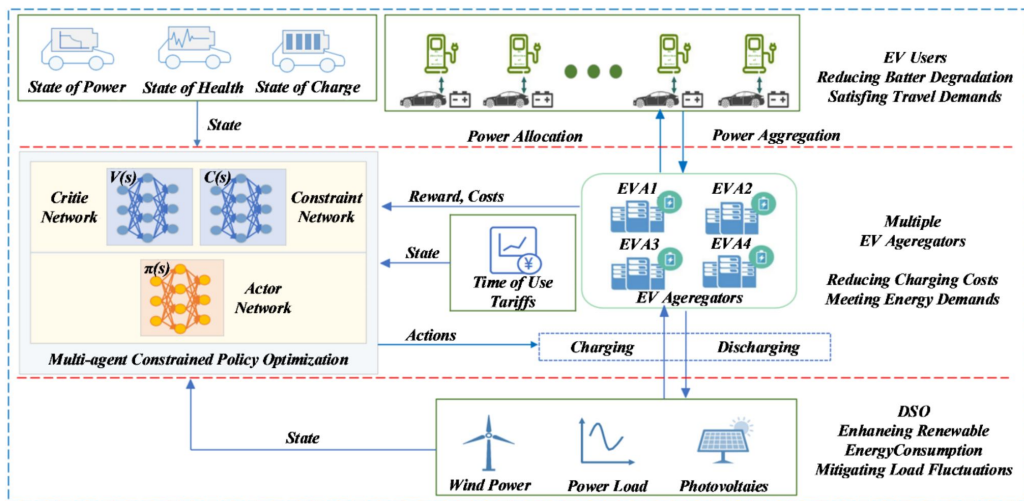
=> Không sạc đầy

=> Sạc khi còn nhiều nhiên liệu

2. Cơ sở lý thuyết

Impact and optimization of vehicle charging scheduling on regional clean energy power supply network management (2025)

(GNN)



=> Tập trung điều phối mạng lưới trạm sạc nhiên liệu nhằm tối.

2. Cơ sở lý thuyết

Áp dụng GNN cho việc tối ưu hóa nhiên liệu tiêu thụ, đường đi và thời gian di chuyển:

1. Tối ưu hóa tuyến đường:

- GNN có thể được sử dụng để phân tích mạng lưới giao thông, dự đoán tình trạng giao thông, và tìm ra tuyến đường tối ưu nhất cho xe điện. Điều này giúp giảm thiểu thời gian di chuyển và tránh ùn tắc giao thông.

2. Quản lý năng lượng dựa trên tuyến đường:

- GNN có thể giúp phân tích các yếu tố ảnh hưởng đến tiêu thụ năng lượng của xe điện trên từng tuyến đường, chẳng hạn như điều kiện mặt đường, độ dốc, và tốc độ trung bình.

3. Tích hợp với hệ thống thông minh:

- GNN có thể tích hợp với các hệ thống thông minh trên xe điện để phân tích dữ liệu về tình trạng giao thông, điều kiện thời tiết, và các yếu tố khác ảnh hưởng đến thời gian di chuyển. Điều này giúp tối ưu hóa lộ trình và giảm thiểu thời gian chờ đợi.

Tuy nhiên, các nghiên cứu trên vẫn còn nhiều mặt hạn chế:

1. Hạn chế về dữ liệu và mô hình hóa

- Yêu cầu dữ liệu lớn và chất lượng cao và độ Phức tạp trong biểu diễn đồ thị

2. Vấn đề về khả năng mở rộng

- Khả năng mở rộng kém: Khi áp dụng GNN vào các hệ thống lớn như mạng lưới giao thông hoặc lưới điện, việc xử lý dữ liệu đồ thị với số lượng nút và cạnh lớn có thể dẫn đến chi phí tính toán cao và thời gian xử lý lâu. Điều này đặc biệt rõ ràng khi kích thước không gian trạng thái và hành động tăng thêm.
- Hạn chế trong tối ưu hóa toàn diện: Các phương pháp hiện tại vẫn chưa đạt được sự cân bằng hoàn hảo giữa giảm chi phí, đảm bảo sự hài lòng của người dùng, và duy trì ổn định lưới điện. Ví dụ, một số thuật toán tối ưu hóa chi phí nhưng lại gây ra vi phạm công suất lưới điện đáng kể hoặc giảm sự hài lòng của người dùng.

3. Tập dữ liệu

“Public”:

- + **Vehicle Energy Dataset (VED) & Extended VED (eVED)**
- + Electric Vehicle Trip Energy Consumption (Shanghai EV Dataset)
- + EV Energy Consumption and Speed Profiles (EVECS) – IEEE DataPort
- + ChargeCar Commute Database (Crowdsourced EV Trip Data)

Table 1: A list of related open-source vehicle trip datasets

Dataset name with citations of use in research community	Amount of records	Amount of features	Recorded from diverse data owners or not	Contains energy consumption related features or not
Battery and heating data in real driving cycles [2-4]	900,000+	48	No	Yes
Vehicle Energy Dataset [5-13]	20,000,000+	18	Yes	Yes
Panasonic 18650PF Li-ion Battery Data [14, 15]	1,000,000+	10	No	No
Battery cycle life data [16]	800,000+	6	No	No
EV battery level vs distance approximation	108	4	No	No
A New Dynamic Stochastic EV Model for Power System Planning Applications	364	2	Yes	No
Automotive Li-ion Cell Usage Data Set [17]	68,771	5	Yes	Yes
Electric vehicle speed, pedal, acceleration, voltage, and current data over four different roads [18]	163,984	6	No	Yes
Vehicle charging system testing for Chevrolet Volt 2013	29,315	4	No	Yes
Vehicle charging system testing for Ford C-Max Energi 2013	15,352	3	No	Yes
Vehicle charging system testing for Ford Focus Electric	3,601	4	No	Yes
Vehicle charging system testing for Ford Fusion Energi	1,663	4	No	Yes
Vehicle charging system testing for Nissan Leaf	3,343	6	No	Yes
Vehicle charging system testing for Toyota Prius Plug-in	1224	4	No	Yes

[\[2203.08630\] Extended vehicle energy dataset \(eVED\)](#)

Private:

3. Tập dữ liệu

Vehicle Energy Dataset (VED) - 2020

1. Dữ liệu tĩnh (Thông số kỹ thuật): Gồm 383 phương tiện

- **Xe xăng (264):** Các xe truyền thống chỉ sử dụng động cơ xăng.
- **Hybrid Electric Vehicle - HEVs (92):** Xe lai sử dụng cả động cơ xăng và động cơ điện, nhưng không cần sạc ngoài.
- **Plug-in Hybrid Electric Vehicles - PHEV / Fully Electric Vehicles - EVs (27):** Kết hợp giữa xe điện và sạc ngoài (PHEV) và xe sạc điện hoàn toàn (EV). Cụ thể, có 3 xe EV hoàn toàn trong bộ dữ liệu, tất cả đều là Nissan Leaf 2013 với dung lượng pin 24 kWh.

```
pd.read_excel ('VED-master\Data\VED_Static_Data_ICE&HEV.xlsx').head()
```

✓ 0.0s

VehId	Vehicle Type	Vehicle Class	Engine Configuration & Displacement	Transmission	Drive Wheels	Generalized_Weight
0	2	ICE	Car	4-FI 2.0L T/C	NO DATA	3500
1	5	HEV	Car	4-GAS/ELECTRIC 2.0L	NO DATA	3500
2	7	ICE	SUV	6-FI 3.6L	AUTOMATIC	4500
3	8	ICE	Car	4-FI 1.5L	5-SP MANUAL	2500
4	12	ICE	Car	4-FI 1.8L	NO DATA	2500

```
pd.read_excel('VED-master\Data\VED_Static_Data_PHEV&EV.xlsx').head()
```

✓ 0.0s

VehId	EngineType	Vehicle Class	Engine Configuration & Displacement	Transmission	Drive Wheels	Generalized_Weight
0	9	PHEV	Car	4-GAS/ELECTRIC 1.4L	NO DATA	4000
1	10	EV	Car	ELECTRIC	NO DATA	3500
2	11	PHEV	Car	4-GAS/ELECTRIC 2.0L	CVT	4000
3	371	PHEV	Car	4-GAS/ELECTRIC 2.0L	CVT	4000
4	379	PHEV	Car	4-GAS/ELECTRIC 1.4L	NO DATA	4000

3. Tập dữ liệu

Vehicle Energy Dataset (VED) - 2020

2. Dữ liệu động (Lịch sử di chuyển)

- Ghi lại quỹ đạo GPS của các phương tiện cùng với dữ liệu chuỗi thời gian về nhiên liệu, năng lượng, tốc độ và mức sử dụng năng lượng phụ trợ,...
- Được thu thập thông qua bộ ghi OBD-II trên xe từ **tháng 11/2017 đến tháng 11/2018** tại **Ann Arbor, Michigan, Mỹ**.
- Các tình huống lái xe trải dài từ đường cao tốc đến khu vực trung tâm thành phố đông đúc trong nhiều điều kiện lái xe và mùa khác nhau. Tổng cộng, VED tích lũy được khoảng 374.000 dặm.

```
pd.read_csv('VED-master\\Data\\VED_171129_week.csv').head(10)
```

	DayNum	VehId	Trip	Timestamp(ms)	Latitude[deg]	Longitude[deg]	Vehicle Speed[km/h]	MAF[g/sec]	Engine RPM[RPM]	Absolute Load[%]	OAT[DegC]	Fuel Rate[L/hr]	Air Conditioning Power[kW]	Air Conditioning Power[Watts]	Heater Power[Watts]	HV Battery Current[A]	HV Battery SOC[%]	HV Battery Voltage[V]	Short Term Fuel Trim Bank 1[%]	Short Term Fuel Trim Bank 2[%]	Long Term Fuel Trim Bank 1[%]	Long Term Fuel Trim Bank 2[%]
0	29.572386	8	805	0	42.266449	-83.706693	50.0	3.64	1610.0	16.862745	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	7.8125	NaN	7.8125	NaN
1	29.572386	8	805	1300	42.266449	-83.706693	46.0	3.21	1411.0	16.862745	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	7.8125	NaN	7.8125	NaN
2	29.572386	8	805	2400	42.266449	-83.706693	45.0	3.21	1411.0	16.862745	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	7.8125	NaN	7.8125	NaN
3	29.572386	8	805	3600	42.266449	-83.706693	44.0	2.90	1346.0	14.509804	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0000	NaN	7.8125	NaN
4	29.572386	8	805	5700	42.266449	-83.706693	44.0	2.93	1350.0	14.509804	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0000	NaN	7.8125	NaN
5	29.572386	8	805	7300	42.266449	-83.706693	45.0	2.93	1350.0	14.509804	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0000	NaN	7.8125	NaN
6	29.572386	8	805	7700	42.266449	-83.706693	45.0	3.02	1387.0	14.509804	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0000	NaN	7.8125	NaN
7	29.572386	8	805	8100	42.267909	-83.706651	45.0	3.02	1387.0	14.509804	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0000	NaN	7.8125	NaN
8	29.572386	8	805	9400	42.267909	-83.706651	46.0	3.02	1387.0	14.509804	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0000	NaN	7.8125	NaN
9	29.572386	8	805	9800	42.267909	-83.706651	46.0	3.08	1414.0	14.509804	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0000	NaN	7.8125	NaN

- **Đầu vào:** Đồ thị các điểm nút giao thông + Các thông tin của **một** chuyến đi (xe + điểm đi / đến)
- **Đầu ra:** Năng lượng tiêu hao của chuyến đi đó

3. Tập dữ liệu

Extended Vehicle Energy Dataset (eVED) - 2022

2. Tính toán ước lượng năng lượng tiêu hao

- Công thức gốc của VED:

$$\text{Energy consumption} = \frac{\text{DrivingLength}}{100} \cdot \text{MassAirFlow} \cdot \left(1 + \frac{\text{ShortTermFuelTrimBank1}}{100} + \frac{\text{LongTermFuelTrimBank1}}{100} \right) \cdot \frac{1}{\text{AirFuelRatio}}$$

- Công thức trong “Energy Consumption Prediction for Electric Vehicles Based on Real-World Data” (Energies 2015)

$$\begin{aligned} \Delta E &= \sum \Delta E_{\text{micro}} \\ &= \sum_{\text{segments}}^{\text{trip}} B_1 * \sum_i^{\text{segment}} \Delta s_i + B_2 * \sum_i^{\text{segment}} v_i^2 * \Delta s_i + B_3 |20 - T_{\text{segment}}| \\ &\quad * aux_{\text{segment}} * \Delta t_{\text{segment}} + B_4 * \left(\sum_i^{\text{segment}} CMF_i^+ \right) * \Delta s + B_5 \\ &\quad * \left(\sum_i^{\text{segment}} CMF_i^- \right) * \Delta s + B_6 * \sum_i^{\text{segment}} \Delta H_{\text{pos}_i} + B_7 * \sum_i^{\text{segment}} \Delta H_{\text{neg}_i} + \varepsilon \end{aligned} \quad (8)$$

3. Tập dữ liệu

Extended Vehicle Energy Dataset (eVED) - 2022

3. Tổng hợp thêm

- Thêm một số thuộc tính của điểm GPS đó như “Speed Limit”, “Focus Points”, “Intersection”, “Bus Stop”, ...

```
pd.read_csv('datarepo-eved-dataset-549403600cf5/data/eVED/eVED_171101_week.csv').head(10)
```

[41] ✓ 1.6s Python

C:\Users\MinhVu\AppData\Local\Temp\ipykernel_22844\3207840020.py:1: DtypeWarning: Columns (30) have mixed types. Specify dtype option on import or set low_memory=False.

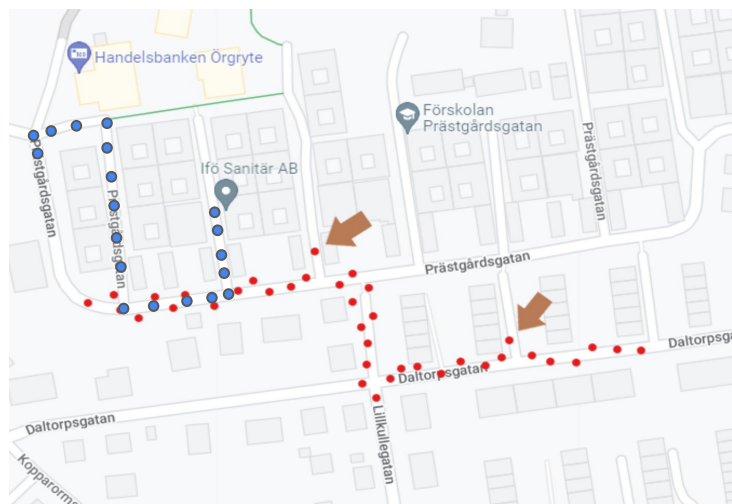
pd.read_csv('datarepo-eved-dataset-549403600cf5/data/eVED/eVED_171101_week.csv').head(10)

	Absolute Load[%]	OAT[DegC]	Fuel Rate[L/hr]	Air Conditioning Power[kW]	Air Conditioning Power[Watts]	Heater Power[Watts]	HV Battery Current[A]	HV Battery SOC[%]	HV Battery Voltage[V]	Short Term Fuel Trim Bank 1[%]	Short Term Fuel Trim Bank 2[%]	Long Term Fuel Trim Bank 1[%]	Long Term Fuel Trim Bank 2[%]	Elevation Raw[m]	Elevation Smoothed[m]	Gradient	Energy_Consumption	Matchtd Latitude[deg]	Matched Longitude[deg]	Match Type	Class of Speed Limit	Speed Limit[km/h]	Speed Limit with Direction[km/h]	Intersection	Bus Stops	Focus Points
1	49.019608	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	230.563034	230.563034	0.000000	NaN	42.277558	-83.698767	0	0.0	56	56.0	NaN	NaN	NaN
1	67.450981	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	230.563034	230.563034	0.000000	NaN	42.277558	-83.698767	1	0.0	56	56.0	NaN	NaN	NaN
1	67.450981	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	230.563034	230.563034	0.000000	NaN	42.277558	-83.698767	1	0.0	56	56.0	NaN	NaN	NaN
1	67.450981	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	230.563034	230.595203	0.000000	NaN	42.277558	-83.698767	1	0.0	56	56.0	NaN	NaN	NaN
1	67.450981	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	230.563034	230.627371	0.000415	NaN	42.277558	-83.698767	1	0.0	56	56.0	NaN	NaN	NaN
1	78.039215	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	230.723877	230.659540	0.000000	NaN	42.278255	-83.698812	0	0.0	64	64.0	NaN	NaN	NaN
1	78.039215	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	230.723877	230.691708	0.000000	NaN	42.278255	-83.698812	1	0.0	64	64.0	NaN	NaN	NaN
1	78.039215	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	230.723877	230.824719	0.000000	NaN	42.278255	-83.698812	1	0.0	64	64.0	NaN	NaN	NaN
1	78.039215	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	230.723877	230.925562	0.001193	NaN	42.278255	-83.698812	1	0.0	64	64.0	NaN	NaN	NaN
1	49.019608	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	231.228088	231.026404	0.000000	NaN	42.279011	-83.698912	0	0.0	64	64.0	NaN	NaN	NaN

4. Phương pháp thực hiện

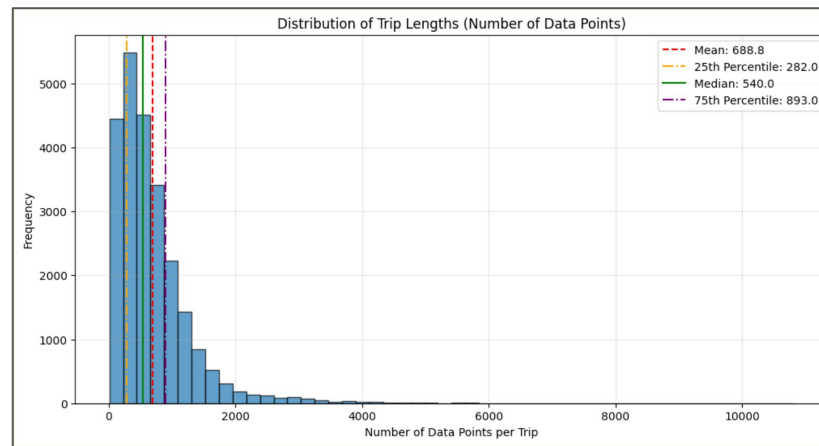
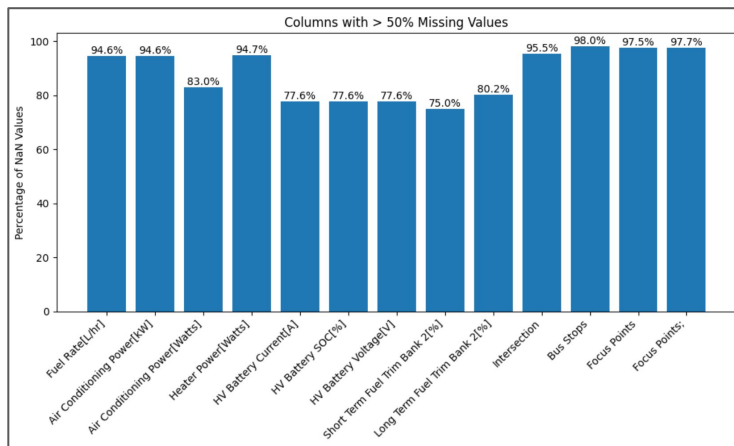
Phạm vi đồ án

- **Nút (lat, lon):** tọa độ của một điểm trên bản đồ (không phải nút giao).
- **Cạnh:** điểm nối giữa các tọa độ (đồ thị vô hướng).



4. Phương pháp thực hiện

Tiền xử lý dữ liệu



DayNum	VehId	Trip	Timestamp(ms)	Vehicle Speed[km/h]	Engine RPM[RPM]	OAT[DegC]	Elevation Smoothed[m]	Gradient	Energy_Consumption	Matched Latitude[deg]	Matched Longitude[deg]	Speed Limit[km/h]	Vehicle_Type	Engine Configuration & Displacement	Generalized_Weight
1.058991	11	1485	0	40.439999	1336.0	6.25	258.927948	0.0	0.002591	42.300623	-83.73575	40	PHEV	4-GAS/ELECTRIC 2.0L	4000
1.058991	11	1485	200	40.439999	2214.0	6.25	258.927948	0.0	0.002591	42.300623	-83.73575	40	PHEV	4-GAS/ELECTRIC 2.0L	4000
1.058991	11	1485	300	40.439999	2214.0	6.25	258.927948	0.0	0.004696	42.300623	-83.73575	40	PHEV	4-GAS/ELECTRIC 2.0L	4000
1.058991	11	1485	600	42.599998	2214.0	6.25	258.927948	0.0	0.004696	42.300623	-83.73575	40	PHEV	4-GAS/ELECTRIC 2.0L	4000
1.058991	11	1485	1600	43.459999	2214.0	6.25	258.927948	0.0	0.004696	42.300623	-83.73575	40	PHEV	4-GAS/ELECTRIC 2.0L	4000

Dataframe shape: (1065501, 16)

Total unique vehicles: 32

Total unique trips: 1836

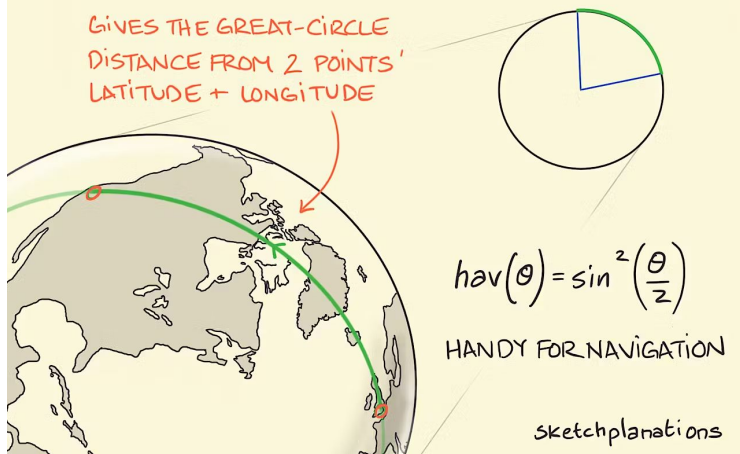
4. Phương pháp thực hiện

Tiền xử lý dữ liệu

THE HAVERSINE FORMULA

FINDING THE SHORTEST DISTANCE BETWEEN 2 POINTS ON A SPHERE

GIVES THE GREAT-CIRCLE DISTANCE FROM 2 POINTS' LATITUDE + LONGITUDE



$$\text{hav}(\theta) = \sin^2\left(\frac{\theta}{2}\right)$$

HANDY FOR NAVIGATION

sketchplanations

```
def haversine(lat1, lon1, lat2, lon2):  
    """  
    Calculate the great circle distance in kilometers between two points  
    on the earth specified in decimal degrees using the haversine formula.  
    """  
    # convert decimal degrees to radians  
    lat1, lon1, lat2, lon2 = map(radians, [lat1, lon1, lat2, lon2])  
    # differences  
    dlat = lat2 - lat1  
    dlon = lon2 - lon1  
    # haversine formula  
    a = sin(dlat/2)**2 + cos(lat1) * cos(lat2) * sin(dlon/2)**2  
    c = 2 * asin(sqrt(a))  
    r = 6371 # Radius of Earth in kilometers.  
    return c * r  
  
aggregations = {  
    'Energy_Consumption': 'sum', # Total energy consumed over the trip  
    'VehId': 'first', # Vehicle ID  
    'Vehicle Speed[km/h]': 'mean', # Average vehicle speed  
    'Engine RPM[RPM]': 'mean', # Average engine RPM  
    'OAT[DegC]': 'mean', # Average outside air temperature  
    'Elevation Smoothed[m]': 'mean', # Average elevation  
    'Timestamp(ms)': ['min', 'max'], # Start and end timestamps for time duration calculation  
    'Gradient': 'mean', # Average road gradient  
    'Matched Latitude[deg]': ['first', 'last'], # Starting and ending Latitude  
    'Matched Longitude[deg]': ['first', 'last'] # Starting and ending Longitude  
}  
  
# Group by TripID and apply the aggregation  
trip_df = full_df.groupby('TripID').agg(aggregations)
```

4. Phương pháp thực hiện

Tiền xử lý dữ liệu

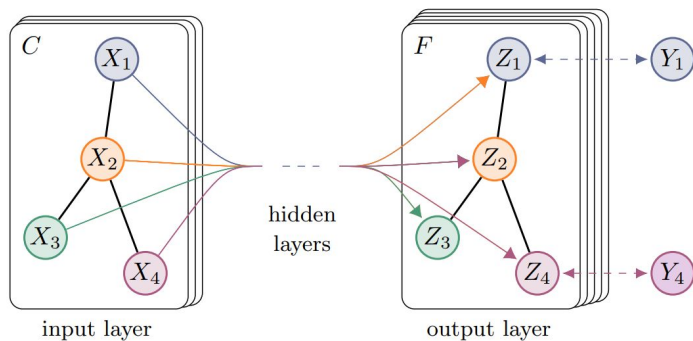
```
def get_hour_category(daynum):  
    """  
    Convert a DayNum float to a datetime object.  
    DayNum 1 corresponds to Nov 1st, 2017, 00:00:00.  
    """  
  
    # Reference date: DayNum 1 = Nov 1st, 2017, 00:00:00  
    reference_date = datetime(2017, 11, 1, 0, 0, 0)  
  
    # Calculate the timedelta from the reference date  
    delta = timedelta(days=daynum - 1)  
    # Add the timedelta to the reference date  
    result_datetime = reference_date + delta  
    # Extract the hour from the datetime  
    hour = result_datetime.hour  
  
    # Categorize based on hour ranges  
    if 7 <= hour < 17: # 7am-5pm  
        return 1 # peak hour  
    elif 17 <= hour < 21: # 5pm-9pm  
        return 2 # off-peak hour  
    else: # Other hours  
        return 3 # night hour
```

```
numeric_cols = ['Energy_Consumption',  
                'Vehicle Speed[km/h]',  
                'Engine RPM[RPM]',  
                'OAT[DegC]',  
                'Elevation Smoothed[m]',  
                'Gradient',  
                'Matched Latitude[deg]',  
                'Matched Longitude[deg]',  
                'Speed Limit[km/h]',  
                'Generalized_Weight']  
  
categorical_cols = ['Vehicle_Type', 'Engine Configuration & Displacement', 'time_category']  
  
# Make a copy of the DataFrame.  
full_df = full_trip_df.copy()  
  
# Scale numeric features.  
scaler = StandardScaler()  
full_df[numeric_cols] = scaler.fit_transform(full_df[numeric_cols])  
  
# One-hot encode categorical features.  
full_df = pd.get_dummies(full_df, columns=categorical_cols, dtype=float)  
  
# If necessary, reindex the DataFrame so that node indices go from 0 to N-1.  
full_df = full_df.reset_index(drop=True)
```

4. Phương pháp thực hiện

So sánh mô hình GNN

- **GCN:** base-lines => Tích hợp thông tin từ hàng xóm
- **GraphSage:** Inductive Representation Learning => tạo biểu diễn cho các nút chưa được nhìn thấy



(a) Graph Convolutional Network

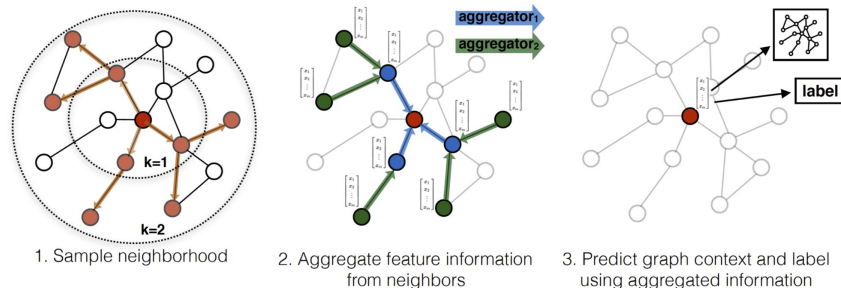


Figure 1: Visual illustration of the GraphSAGE sample and aggregate approach.

<https://arxiv.org/abs/1609.02907>

<https://arxiv.org/abs/1706.02216>

4. Phương pháp thực hiện

Cách tiếp cận

	StepByStep	Combined
Input	<ul style="list-style-type: none">- VehId- Start (lat, lon)- End (lat, lon)- Recent_Performance- Trip_Agg_Metrics	<ul style="list-style-type: none">- VehId- Start (lat, lon)- End (lat, lon)- Recent_Performance- Trip_snapshot_lists
Model	GCN & GraphSage (100 epochs)	GCN & GraphSage (10 epochs - 50% records)
Output	Est_EnergyConsumption	List of snapshots [(lat_01, lon_01, energy_01), ...]

4. Phương pháp thực hiện

Độ đo

$$\text{MSE} = \overset{\text{Mean}}{\frac{1}{n} \sum_{i=1}^n} \left(\overset{\text{Error}}{Y_i - \hat{Y}_i} \right) \overset{\text{Squared}}{^2}$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|$$

Where:

\hat{y}_i = Predicted value for the i^{th} data point

y_i = Actual value for the i^{th} data point

n = number of observations

$$\begin{aligned} r2_score &= 1 - \frac{\text{total_error_model}}{\text{total_error_baseline}} \\ &= 1 - \frac{\sum_{i=1}^N (\text{predicted}_i - \text{actual}_i)^2}{\sum_{i=1}^N (\text{average_value} - \text{actual}_i)^2} \end{aligned}$$

5. Kết quả

StepByStep:

Test Evaluation Metrics:

GCN Model:

MSE: 1.7962, MAE: 1.0326, R^2 : -0.7515

GraphSAGE Model:

MSE: 2.1194, MAE: 1.1286, R^2 : -1.0667

Combined:

Test Evaluation Metrics:

Multi-Task GCN Model:

Total Loss: 175.5310 | Route Loss: 174.4086 | Energy Loss: 1.1224 | Energy MSE: 1.1224

Multi-Task GraphSAGE Model:

Total Loss: 178.8403 | Route Loss: 177.5712 | Energy Loss: 1.2691 | Energy MSE: 1.2691

5. Kết luận

- Xử lý dữ liệu chưa phù hợp
- Cả 2 mô hình rất **khó** học điểm **dữ liệu mới** và **đồ thị dạng vòng**.
- Đồ án còn giới hạn về tài nguyên tính toán.
- Đối với cách tiếp cận Combined, cần nghiên cứu thêm các cách biểu diễn phù hợp ở Hidden-Space.
- Đồ án chưa hỗ trợ real-time, thay đổi theo điều kiện động.

Kết thúc

