GPS Trajectory Data Clustering and Analysis

Report submitted to the SASTRA Deemed to be University

as the requirement for the course

CSE300: MINI PROJECT

Submitted by

Hema Kishore K

Reg. No.: 224003037, B.Tech CSE

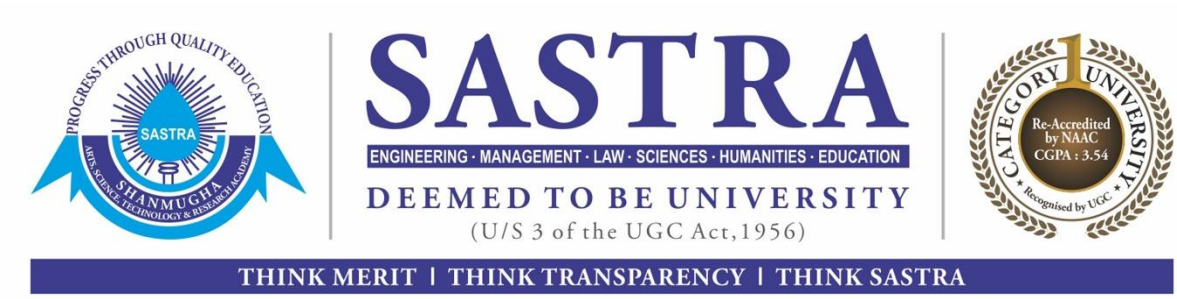
Sai Charan Velpuru

Reg. No.: 224003098, B.Tech CSE

Varshith Sai Naragam

Reg. No.: 224003157, B.Tech CSE

May 2023



Department of Computer Science and Engineering

Srinivasa Ramanujan Centre

Kumbakonam - 612001

Department of Computer Science and Engineering

Srinivasa Ramanujan Centre

Kumbakonam - 612001

Bonafide Certificate

This is to certify that the report titled "**GPS Trajectory Data Clustering and Analysis** " submitted as a requirement for the course**, CSE300 : MINI PROJECT** for B.Tech. is a bonafide record of the work done by Shri/Mr. **Hema Kishore K (Reg. No. 224003037, B.Tech CSE**),**Sai Charan Velpuru (Reg. No.: 224003098, B.Tech CSE), Varshith Sai Naragam (Reg. No.: 224003157, B.Tech CSE)**) during the academic year 2022-23, in the Srinivasa Ramanujan Centre, under my supervision.

Signature of Project Supervisor :

Name with Affiliation : Smt P.Venkateswari/AP-II/CSE/SRC/SASTRA

Date :

Project Based Work Viva voce held on _____

Examiner 1 **Examiner 2**

ACKNOWLEDGEMENTS

We would like to sincerely thank our Chancellor, **Prof. R. Sethuraman**, Vice Chancellor, **Dr. S. Vaidhyasubramaniam** and Registrar, **Dr. R. Chandramouli**, for allowing us to be a student of this esteemed institution.

We express our deepest thanks to **Dr. V. Ramaswamy**, Dean and **Dr. A. Alli Rani**, Associate Dean, Srinivasa Ramanujan Centre, for their constant support and invaluable suggestions whenever required by us without any reservations.

We are pleased to express our gratitude to our Guide **Smt.P.Venkateswari**, Assistant Professor, Department of CSE and one of the Project Coordinators, for her ever-encouraging spirit and meticulous guidance in completing this project.

We would also like to thank our panel members, **Smt.M.Martinaa** and **Smt.Glory Thephoral**, Assistant Professors of CSE department, for their valuable time and great support extended to us that helped us rectify our mistakes and complete this project.

We would like to record the humane approach and painstaking efforts of guidance of our Project Coordinators, **Dr. Durga Karthik, Dr.R.Bhavani** and all other departmental staff to whom we owe our hearty thanks forever.

Last but not least, without the support of our parents and friends, this project would never have become a reality. We dedicate this work to them with love and affection.

TABLE OF CONTENTS

List of Figures

ABBREVATIONS

GPS - Global Positioning System

MOR - Multi Output Regressor

RFR - Random Forest Regressor

OHE -One Hot Encoder

NumPy – Numerical Python

Abstract

The widespread use of smartphones has speed up the research on mobility by significantly expanding the types and amount of data that are available. GPS technology is one such source of mobility information, which is becoming more widely used and common in understanding of human mobility patterns by the research community. However, there is no standardised framework for using the machine learning methods to study the various mobility patterns created by non-Work, non-Home locations of Working and Nonworking users on Workdays and Offdays. The unsupervised machine learning method called k-means clustering, is used to obtain three user clusters for each type of day. The User Commonality and Average Frequency metrics are the two new metrics we suggest for analysing the clustering results. Using the proposed metrics, we can identify interesting user behaviours and gain a better understanding of the users' mobility patterns.

Keywords: Clustering, GPS Technology, Daily Characteristic Distance, Machine Learning, Mobility Patterns.

Specific Contribution :

Sai Charan V - Implemented Random Forest Algorithm and graphical analysis.

Varshith Sai N - Implemented K-Means Clustering Algorithm using elbow method.

Hema Kishore K – Data and algorithm architecture, Obtaining Mobility points on map.

Specific Learning :

Sai Charan Velpuru - Regression model of random forest.

Varshith Sai Naragam – Clustering and methodology.

Hema Kishore K – Concepts of data preprocessing and mapping.

Technical Limitations & Ethical Challenges faced:

Unable to use DB scan algorithm as it is unable to vary densities clusters with available data sets, so we switched to Random Forest algorithm with supporting paper which has same idea. Data set is not mentioned in base paper , they have used private data set which we felt very to find which has the ability to produce min points and epsilon values.

CHAPTER - 1

SUMMARY OF THE BASE PAPER

Title: Clustering and Analysis of GPS Trajectory Data Using Distance-Based Features

Journal Name: IEEE

Publisher: Zann Koh ,Yuren Zhou, Billy Pik Lik Lau, Ran Liu, Keng Hua Chong And Chau Yuen

Year: 2022

The title of the base paper is "**Clustering and Analysis of GPS Trajectory Data Using Distance-Based Features**". This paper was published in the year 2022.

1.1 SUMMARY:

- This paper gives a detailed information about the methodologies for identifying mobility patterns.
- It also has trajectory analysis using clustering algorithm and has different analysis methods of trajectory.
- Weekdays and Off days data and their trajectory is predicted in this study with different clusters and provided the usage of algorithm.
- This is used in traffic analysis by predicting the crowded areas and predicting traffic in advance using weekdays and off days analysis.
- Analysis of different vehicles is used here to predict the actual traffic using machine learning techniques and clustering approach.
- With the supporting paper which has the idea of Road traffic analysis by predicting trajectories using supervised and ensemble Random Forest regression algorithm.
- It contains both K-Means and Random Forest algorithm which is used to classify prediction and in regression.

- By taking this paper's algorithm we are going to support our base paper's idea and find the appropriate mobility patterns which is used in traffic analysis.

1.2 INTRODUCTION:

- GPS, often known as the Global Positioning System, has been available for a long and is increasingly being used in mobility studies. It has been discovered to be highly useful for obtaining spatiotemporal data of various sizes.

- We are seeking for some simple, simply understood qualities in our data that will allow us to interpret the clustering findings in a meaningful way. There are three types of mobility metrics: network-based, link-based, and movement-based metrics.

- Because our focus is on how users move around the system, we emphasized movement-based and link-based data such as visit frequency and mean squared distance rather than network-based metrics such as transmission count and energy usage.

- The problem of clustering and analyzing GPS trajectory data entails grouping comparable trajectories into clusters and deriving significant information from the resulting clusters.

- The GPS trajectory data is made up of a sequence of latitude and longitude coordinates that indicate the user's location at different periods in time.

- Each user may have many trajectories indicating different journeys or activities. Other information such as the timestamp, speed, and direction of movement may be included in the data.

- The purpose is to organise the trajectories into groups based on their spatial and temporal properties, and then analyse these groups to find meaningful patterns and insights.

- To do this, we must employ a combination of clustering techniques such as k-means, the elbow approach, and others.

- We can also use ensemble machine learning algorithm called Random Forest to identify mobility patterns by using regression which is accurate.

1.3 PROBLEM STATEMENT:

- In order to cluster and analyse GPS trajectory data, it is necessary to combine comparable trajectories into clusters and then draw conclusions from the resulting clusters.
- A set of latitude and longitude coordinates that show the user's location at various times make up the GPS trajectory data.

- Multiple trajectories for each user are possible; these represent various journeys or activities. The data might additionally include further details like the timestamp, movement speed, and direction. Other data contain train and test data sets and also taxi stand id as the taxi data set is more stable and maintains a correct record compared to previous one as it is survey based.

- It is intended to categorise the trajectories according to their spatial and temporal properties, then analyse these groups to find significant patterns and insights.

- We need to combine clustering methods like k-means, the elbow method, Random Forest, Multi output regressor to do this.

1.4 ARCHITECTURE DIAGRAM:

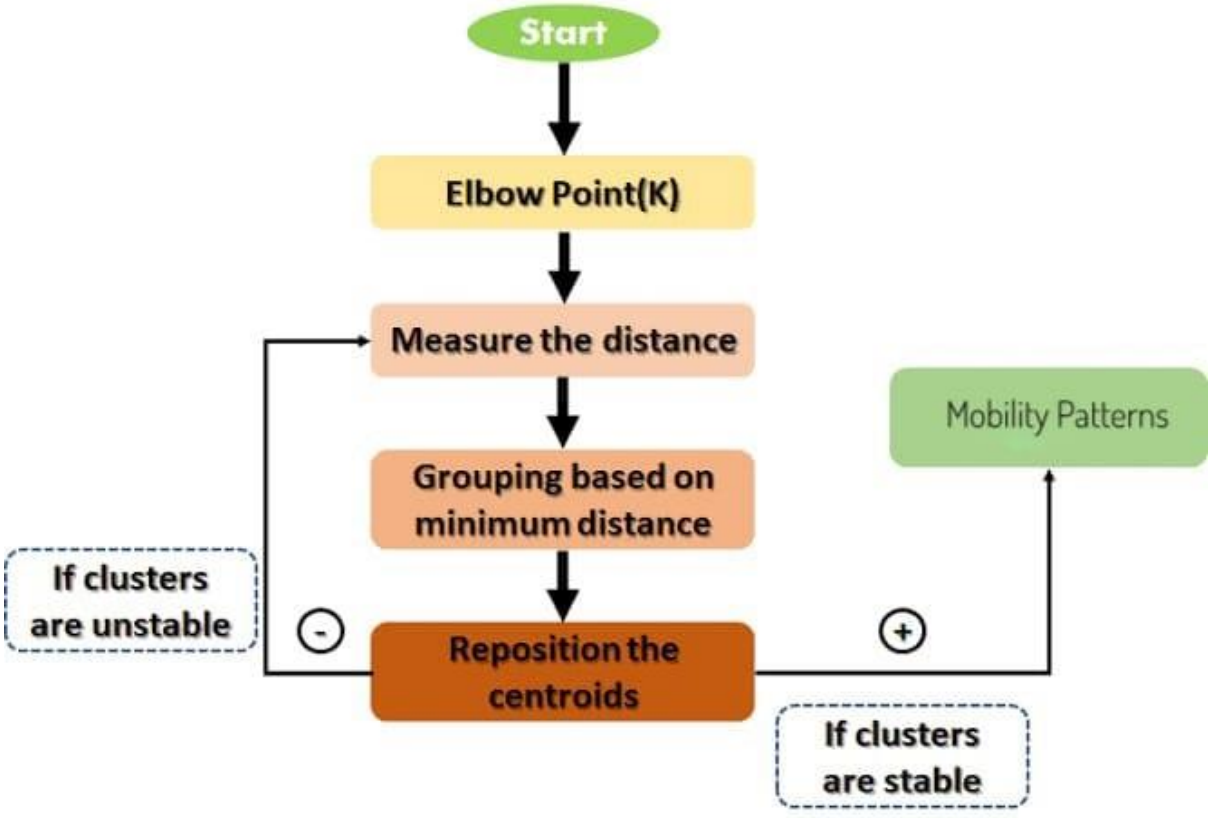


Fig. 1.4.1 K-Means Clustering Algorithm Architecture Diagram

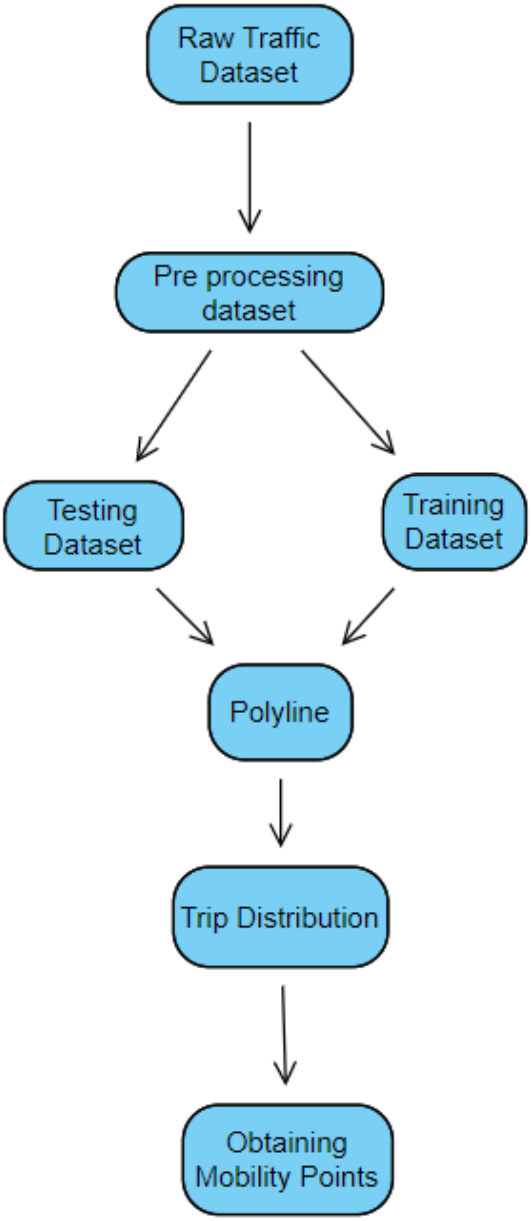


Fig 1.4.2 Random Forest Algorithm Architecture Diagram

1.5 DATA SET:

- First dataset was developed by Microsoft Research Asia by collecting the data of the GPS trajectories (or paths) represented by 24,876,978 points recorded by 182 users in the period from April 2007 to August 2012.

- Second dataset describes a complete year (from 01/07/2013 to 30/06/2014) of the trajectories for all the 442 taxis running in the city of Porto, in Portugal.
- It has 1704769 Rows x 22 columns.

1.6 Functions and techniques used in the source code:

1.OneHotEncoder(handle_unknown='ignore')
A pre-processing technique called the OneHotEncoder(OHE) is used to convert categorical data to numerical data that may be used in machine learning models. If there are unseen categories in the test data (categories that weren't included in the training data), the encoder will ignore them rather than generating an error because the handle_unknown parameter is set to 'ignore'.

2.scaler.fit_transform(X)
The method call scaler.fit_transform(X) scales the input data X using the StandardScaler instance scaler and returns the scaled data.

3. kmeans.fit(X)
It is a method call that fits the KMeans clustering algorithm to the input data X.

4. ax.scatter()
It is a method of an Axes object, which is a container that holds the plot elements in Matplotlib. It is used to create a scatter plot of data points on the Axes.

5. fig.add_subplot()
It a function in Matplotlib, a plotting library for Python. It is used to add a new subplot to a Matplotlib figure.

6.plt.scatter()
It a function in Matplotlib, a plotting library for Python. It is used to create a scatter plot of x-y pairs.

7. np.unique()
It a function in the NumPy library for Python. It is used to find the unique elements in an array.

8.**final_train.query()**
final_train is assumed to be a Pandas DataFrame object. Thequery() method of a Pandas DataFrame is used to filter the rows of the DataFrame based on a boolean expression.

9. sns.distplot()

It is a function in the Seaborn library, which is a popular data visualization library for Python. It is used to create a histogram with a density curve overlaid on top.

10.axs.flatten()

It is a method to flatten a NumPy array of subplots into a one-dimensional array.

11. axs[i].set_title()

It is a method that sets the title of a subplot in a Matplotlib figure.

12. MultiOutputRegressor(RandomForestRegressor(n_estimators=100, random_state=1))

It uses random forest regression as the primary estimator to build a multi-output regression model. For reproducibility, the RandomForestRegressor(RFR) function call specifies the random seed (random_state) and the number of decision trees in the forest (n_estimators). The MultiOutputRegressor(MOR) function uses this as a parameter and builds a wrapper for the random forest regressor to support multi-output regression.

13. forest.predict()

It is a method in scikit-learn used to predict the output of a machine learning model, specifically a random forest regressor.

CHAPTER - 2

MERITS AND DEMERITS OF BASE PAPER

2.1 MERITS:

1. Offers a way for analysing and grouping massive datasets of GPS trajectory data, which can be helpful in several industries including urban planning and transportation planning.

2. Has the potential to increase the precision and effectiveness of GPS tracking systems, which are becoming more crucial in numerous fields and applications.

3.In supporting paper, the authors have also provided a method for interpreting the Random Forest model, which allows for better understanding of the factors of using this algorithm.

4.The approach used in the paper is novel and could lead to further research in this area.

2.2 DEMERITS:

1.May be subject to privacy concerns, particularly if the data being analysed includes personally identifiable information or sensitive locations.

2.It could be difficult for academics or practitioners to implement if they lack the necessary computational power or specialised software.

3.May raise privacy issues, especially if the data being analysed contains sensitive or personally identifiable information.

4.In supporting paper, they have not provided a detailed discussion on the implementation of their model in real-world scenarios, which may limit its practical application.

CHAPTER - 3

SOURCE CODE

1)K-Means

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans, AgglomerativeClustering, SpectralClustering,
MeanShift


from sklearn.preprocessing import StandardScaler


import torch

import folium


# Load the GPS dataset

data = pd.read_csv('dataset_raw_full_Copy.csv', header=None)

data.columns = ['latitude', 'longitude', 'altitude', 'Date_Time']


# Split the Date_Time column into date and time columns

data[['date', 'time']] = data['Date_Time'].str.split(expand=True)

# Remove the original Date_Time column

data.drop('Date_Time', axis=1, inplace=True)

X = data[['latitude', 'longitude', 'altitude', 'date', 'time']].values


X


X = data[['latitude', 'longitude', 'altitude']].values


X


# Scale the data
```

```
scaler = StandardScaler()
X = scaler.fit_transform(X)

X

# Convert data to PyTorch tensor
X = torch.tensor(X, dtype=torch.float)

X

# Elbow method to find optimal number of clusters for K-means
wcss = [] #within cluster sum of square
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
plt.plot(range(1, 11), wcss)
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()

# K-means clustering
kmeans = KMeans(n_clusters=4)
kmeans.fit(X)
labels_km = kmeans.labels_

# Plot the clusters
plt.scatter(X[:, 0], X[:, 1], c=labels_km)
plt.title('K-means Clustering')
```

```
plt.xlabel('Latitude')
plt.ylabel('Longitude')
plt.show()

# Plot K-means clustering results with identified mobility patterns
fig = plt.figure(figsize=(8, 8))
ax = fig.add_subplot(111, projection='3d')
colors = ['red', 'blue', 'green', 'orange']
for i in range(kmeans.n_clusters):
    ax.scatter(X[labels_km == i, 0], X[labels_km == i, 1], X[labels_km == i, 2], s=20, c=colors[i], label=f'Cluster {i}')
    center = kmeans.cluster_centers_[i]
    ax.scatter(center[0], center[1], center[2], s=200, c=colors[i], marker='*', label=f'Cluster {i} centroid')
ax.legend()
ax.set_xlabel('Latitude')
ax.set_ylabel('Longitude')
ax.set_zlabel('Altitude')
ax.set_title('K-means clustering with identified mobility patterns')
plt.show()

# Analyze mobility patterns for each cluster
for i in range(kmeans.n_clusters):
    cluster_indices = np.where(labels_km == i)[0]
    cluster_data = data.iloc[cluster_indices]
    print(f'K-means Cluster {i+1}:')
    print(f'Average altitude: {np.mean(cluster_data["altitude"]):.2f} meters')

    distances = np.sqrt(
        np.sum(np.diff(cluster_data[['latitude', 'longitude']])**2, axis=1))
    print(f'Distance traveled (mean): {np.mean(distances):.2f} meters')
```

```
    print(f'Distance traveled (max): {np.max(distances):.2f} meters')
    print()
```

2)Random Forest

```
import zipfile
import pandas as pd
import datetime
import numpy as np
import re
import folium
import seaborn as sns
from matplotlib import pyplot as plt
from plotly.subplots import make_subplots
import warnings
warnings.filterwarnings('ignore')

from sklearn import preprocessing
from sklearn.preprocessing import OneHotEncoder
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.multioutput import MultiOutputRegressor
from sklearn.metrics import mean_squared_error,r2_score

from google.colab import drive
drive.mount('/content/drive')

zip_file_sample = zipfile.ZipFile(
    "/content/drive/MyDrive/sampleSubmission.csv.zip")
zip_file_train = zipfile.ZipFile(
    "/content/drive/MyDrive/train.csv.zip")
```

```
zip_file_test = zipfile.ZipFile(
    "/content/drive/MyDrive/test.csv.zip")
zip_file_GPSlocation = zipfile.ZipFile(
    "/content/drive/MyDrive/metaData_taxistandsID_name_GPSlocation.csv.zip")

sample = pd.read_csv(zip_file_sample.open('sampleSubmission.csv'))
train = pd.read_csv(zip_file_train.open("train.csv"))
test = pd.read_csv(zip_file_test.open("test.csv"))
location = pd.read_csv(zip_file_GPSlocation.open(
    "metaData_taxistandsID_name_GPSlocation.csv"))

sample

train

test

location

train.info()

test.info()

Percent_missing_train = train.isnull().sum() * 100 / len(train)
Percent_missing_train

Percent_missing_test = test.isnull().sum() * 100 / len(train)
Percent_missing_test

train["DAY_TYPE"].unique()
```

```
test["DAY_TYPE"].unique()

train = train.drop("DAY_TYPE", axis=1)

test = test.drop("DAY_TYPE", axis=1)

train

test

train.describe()

test.describe()

train.dtypes

test.dtypes

for colum in train:
    unq_vals = np.unique(train[colum])
    nr_vals = len(unq_vals)
    if nr_vals < 10:
        print("The number of unique values for features {} : {} --- {}".format(colum, nr_vals,
unq_vals))
    else:
        print("The number of unique values for features {} : {}".format(colum, nr_vals))

for colum in test:
    unq_vals = np.unique(test[colum])
    nr_vals = len(unq_vals)
```

```
    if nr_vals < 10:
        print("The number of unique values for features {} : {} --- {}".format(colum, nr_vals, unq_vals))
    else:
        print("The number of unique values for features {} : {}".format(colum, nr_vals))


train["TIMESTAMP"] = [float(time) for time in train["TIMESTAMP"]]
train["data_time"] = [datetime.datetime.fromtimestamp(time, datetime.timezone.utc) for time in train["TIMESTAMP"]]


test["TIMESTAMP"] = [float(time) for time in test["TIMESTAMP"]]
test["data_time"] = [datetime.datetime.fromtimestamp(time, datetime.timezone.utc) for time in test["TIMESTAMP"]]


train["data_time"].value_counts()


test["data_time"].value_counts()


train["year"] = train["data_time"].dt.year
train["month"] = train["data_time"].dt.month
train["week"] = train["data_time"].dt.week
train["day"] = train["data_time"].dt.day
train["hour"] = train["data_time"].dt.hour
train["min"] = train["data_time"].dt.minute
train["weekday"] = train["data_time"].dt.weekday


test["year"] = test["data_time"].dt.year
test["month"] = test["data_time"].dt.month
test["week"] = test["data_time"].dt.week
test["day"] = test["data_time"].dt.day
test["hour"] = test["data_time"].dt.hour
```

```
test["min"] = test["data_time"].dt.minute
test["weekday"] = test["data_time"].dt.weekday

encoder = OneHotEncoder(handle_unknown='ignore')

encoder_df = pd.DataFrame(encoder.fit_transform(train[['CALL_TYPE']]).toarray())

final_train = train.join(encoder_df)

final_train.rename(columns={0:'call_type_a', 1:'call_type_b',2:'call_type_c'}, inplace=True)

final_train

encoder = OneHotEncoder(handle_unknown='ignore')

encoder_df = pd.DataFrame(encoder.fit_transform(test[['CALL_TYPE']]).toarray())

final_test = test.join(encoder_df)

final_test.rename(columns={0:'call_type_a', 1:'call_type_b',2:'call_type_c'}, inplace=True)

final_test

lists_1st_lon = []
for i in range(0,len(final_train["POLYLINE"])):
    if final_train["POLYLINE"][i] == '[]':
        k=0
        lists_1st_lon.append(k)
    else:
        k = re.sub(r"[[|[|]|]|]]", "", final_train["POLYLINE"][i]).split(",")[0]
```

```
        lists_1st_lon.append(k)

final_train["lon_1st"] = lists_1st_lon

lists_1st_lat = []
for i in range(0,len(final_train["POLYLINE"])):
    if final_train["POLYLINE"][i] == '[]':
        k=0
        lists_1st_lat.append(k)
    else:
        k = re.sub(r"[[|[|]|]|]]", "", final_train["POLYLINE"][i]).split(",")[1]
        lists_1st_lat.append(k)

final_train["lat_1st"] = lists_1st_lat

lists_last_lon = []
for i in range(0,len(final_train["POLYLINE"])):
        if final_train["POLYLINE"][i] == '[]':
            k=0
            lists_last_lon.append(k)
        else:
            k = re.sub(r"[[|[|]|]|]]", "", final_train["POLYLINE"][i]).split(",")[-2]
            lists_last_lon.append(k)

final_train["lon_last"] = lists_last_lon

lists_last_lat = []
for i in range(0,len(final_train["POLYLINE"])):
    if final_train["POLYLINE"][i] == '[]':
        k=0
```

```
        lists_last_lat.append(k)
    else:
        k = re.sub(r"[[|[|]|]|]]", "", final_train["POLYLINE"][i]).split(",")[-1]
        lists_last_lat.append(k)

final_train["lat_last"] = lists_last_lat

final_train

train = final_train.query("lon_last != 0")

train["lon_1st"] = [float(k) for k in train["lon_1st"]]
train["lat_1st"] = [float(k) for k in train["lat_1st"]]
train["lon_last"] = [float(k) for k in train["lon_last"]]
train["lat_last"] = [float(k) for k in train["lat_last"]]
train['call_type_a']= [int(k) for k in train["call_type_a"]]
train['call_type_b'] =[int(k) for k in train["call_type_b"]]
train['call_type_c']= [int(k) for k in train["call_type_c"]]

train

lists_1st_lon = []
for i in range(0,len(final_test["POLYLINE"])):
    if final_test["POLYLINE"][i] == '[]':
        k=0
        lists_1st_lon.append(k)
    else:
        k = re.sub(r"[[|[|]|]|]]", "", final_test["POLYLINE"][i]).split(",")[0]
        lists_1st_lon.append(k)
```

```
final_test["lon_1st"] = lists_1st_lon

lists_1st_lat = []
for i in range(0,len(final_test["POLYLINE"])):
    if final_test["POLYLINE"][i] == '[]':
        k=0
        lists_1st_lat.append(k)
    else:
        k = re.sub(r"[[|[|]|]|]]", "", final_test["POLYLINE"][i]).split(",")[1]
        lists_1st_lat.append(k)

final_test["lat_1st"] = lists_1st_lat

lists_last_lon = []
for i in range(0,len(final_test["POLYLINE"])):
        if final_test["POLYLINE"][i] == '[]':
            k=0
            lists_last_lon.append(k)
        else:
            k = re.sub(r"[[|[|]|]|]]", "", final_test["POLYLINE"][i]).split(",")[-2]
            lists_last_lon.append(k)

final_test["lon_last"] = lists_last_lon

lists_last_lat = []
for i in range(0,len(final_test["POLYLINE"])):
    if final_test["POLYLINE"][i] == '[]':
        k=0
        lists_last_lat.append(k)
    else:
```

```
        k = re.sub(r"[[|[|]|]|]]", "", final_test["POLYLINE"][i]).split(",")[-1]
        lists_last_lat.append(k)

final_test["lat_last"] = lists_last_lat

final_test

test = final_test.query("lon_last != 0")

test["lon_1st"] = [float(k) for k in test["lon_1st"]]
test["lat_1st"] = [float(k) for k in test["lat_1st"]]
test["lon_last"] = [float(k) for k in test["lon_last"]]
test["lat_last"] = [float(k) for k in test["lat_last"]]
test['call_type_a']= [int(k) for k in test["call_type_a"]]
test['call_type_b'] =[int(k) for k in test["call_type_b"]]
test['call_type_c']= [int(k) for k in test["call_type_c"]]

test

year = [2013,2014]

fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(10,10))
axs = axs.flatten()

colors = ['blue', 'cyan']
for i, year in enumerate(year):
    data2 = train.loc[train.year == year,:]
    sns.distplot(data2.hour, ax=axs[i], color=colors[i%2])
    axs[i].set_title(f'Trip time distribution For Year {year} (Hour in the Day)')
```

```
plt.tight_layout()
plt.show()

months = [1,2,3,4,5,6,7,8,9,10,11,12]

fig, axs = plt.subplots(nrows=4, ncols=3, figsize=(20,10))
axs = axs.flatten()

for i, month in enumerate(months):
    data2 = train.loc[train.month == month,:].reset_index(drop = True)
    sns.distplot(data2.hour, ax=axs[i])
    axs[i].set_title(f'Trip time distribution For Month {month} (Hour in the Day)')

plt.tight_layout()
plt.show()

weekday = [0, 1, 2, 3, 4, 5, 6]

fig, axs = plt.subplots(nrows=3, ncols=3, figsize=(20,15))
axs = axs.flatten()

for i, day in enumerate(weekday):
    data2 = train.loc[train.weekday == day,:]
    sns.distplot(data2.hour, ax=axs[i])
    axs[i].set_title(f'Trip time distribution For weekday {day} (Hour in the Day)')

plt.tight_layout()
plt.show()

weekday = pd.DataFrame(data=train.groupby("weekday").TRIP_ID.count()).reset_index()
```

```
with plt.style.context("fivethirtyeight"):
    plt.figure(figsize=(10,6))
    plt.plot(weekday["weekday"], weekday["TRIP_ID"])
    plt.xlabel("weekday\n (0:Monday ~ 6:Sunday)")
    plt.ylabel("count")
    plt.ylim([200000, 300000])


countplt, ax = plt.subplots(figsize = (10,7))
ax =sns.countplot(x = train['CALL_TYPE'],data=train,palette='pastel' )
for rect in ax.patches:
    ax.text (rect.get_x() + rect.get_width()  / 2,rect.get_height()+
0.75,rect.get_height(),horizontalalignment='center', fontsize = 11)


del train['CALL_TYPE']


train['ORIGIN_CALL'] = train[['ORIGIN_CALL']].fillna('')
train['ORIGIN_STAND'] = train[['ORIGIN_STAND']].fillna('')


new_train = train.copy()
new_train


zfig, ax = plt.subplots(figsize=(15,15))
sns.heatmap(new_train.corr(), annot = True, vmin=-1, vmax=1, center= 0, cmap=
'coolwarm',square=True)


mapping_1st = pd.DataFrame({
    "date":train.head(10000)["data_time"].values,
    "lat":train.head(10000)["lat_1st"].values,
    "lon":train.head(10000)["lon_1st"].values
})
```

```
mapping_last = pd.DataFrame({
    "date":train.head(10000)["data_time"].values,
    "lat":train.head(10000)["lat_last"].values,
    "lon":train.head(10000)["lon_last"].values
})

por_map = folium.Map(location=[41.141412,-8.590324], tiles='Stamen Terrain',
zoom_start=11)

for i, r in mapping_1st.iterrows():
    folium.CircleMarker(location=[r["lat"],r["lon"]], radius=0.5,
color="red").add_to(por_map)

for i, r in mapping_last.iterrows():
    folium.CircleMarker(location=[r["lat"],r["lon"]], radius=0.5,
color="blue").add_to(por_map)

por_map

train["delta_lon"] = train["lon_last"] - train["lon_1st"]
train["delta_lat"] = train["lat_last"] - train["lat_1st"]

test["delta_lon"] = test["lon_last"] - test["lon_1st"]
test["delta_lat"] = test["lat_last"] - test["lat_1st"]

train

test

ml_train = train.copy()
```

```
def origin_call_flg(x):
    if x["ORIGIN_CALL"] == None:
        res = 0
    else:
        res = 1
    return res
ml_train["ORIGIN_CALL"] = ml_train.apply(origin_call_flg, axis=1)

def origin_stand_flg(x):
    if x["ORIGIN_STAND"] == None:
        res = 0
    else:
        res=1
    return res
ml_train["ORIGIN_STAND"] = ml_train.apply(origin_stand_flg, axis=1)

def miss_flg(x):
    if x["MISSING_DATA"] == "False":
        res = 0
    else:
        res = 1
    return res
ml_train["MISSING_DATA"] = ml_train.apply(miss_flg, axis=1)

ml_test = test.copy()
def origin_call_flg(x):
    if x["ORIGIN_CALL"] == None:
        res = 0
    else:
```

```
        res = 1
    return res
ml_test["ORIGIN_CALL"] = ml_test.apply(origin_call_flg, axis=1)

def origin_stand_flg(x):
    if x["ORIGIN_STAND"] == None:
        res = 0
    else:
        res=1
    return res
ml_test["ORIGIN_STAND"] = ml_test.apply(origin_stand_flg, axis=1)

def miss_flg(x):
    if x["MISSING_DATA"] == "False":
        res = 0
    else:
        res = 1
    return res
ml_test["MISSING_DATA"] = ml_test.apply(miss_flg, axis=1)

ml_train = ml_train.sample(136000)

X =
ml_train[["call_type_a","call_type_b","call_type_c",'ORIGIN_CALL','ORIGIN_STAND',
'MISSING_DATA', 'lon_1st', 'lat_1st', 'delta_lon', 'delta_lat']]

y = ml_train[["lon_last","lat_last"]]

X_Test =
ml_test[["call_type_a","call_type_b","call_type_c",'ORIGIN_CALL','ORIGIN_STAND',
'MISSING_DATA', 'lon_1st', 'lat_1st', 'delta_lon', 'delta_lat']]
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)

forest = MultiOutputRegressor(RandomForestRegressor(n_estimators=100,
random_state=1))

forest = forest.fit(X_train, y_train)
y_train_pred = forest.predict(X_train)
y_test_pred = forest.predict(X_test)

print("MSE train:{}".format(mean_squared_error(y_train, y_train_pred)))
print("MSE test;{}".format(mean_squared_error(y_test, y_test_pred)))
print("R2 score train:{}".format(r2_score(y_train, y_train_pred)))
print("R2 score test:{}".format(r2_score(y_test, y_test_pred)))

y_Test_pred = forest.predict(X_Test)
y_Test_pred[2]

submit_lat = y_Test_pred.T[1]
submit_lon = y_Test_pred.T[0]

submit = pd.DataFrame({"TRIP_ID":test["TRIP_ID"], "LATITUDE":submit_lat,
"LONGITUDE":submit_lon})

submit.to_csv('submission.csv', index=False)
sub = pd.read_csv("./submission.csv")
sub.head(50)
```

CHAPTER - 4

SNAPSHOTS

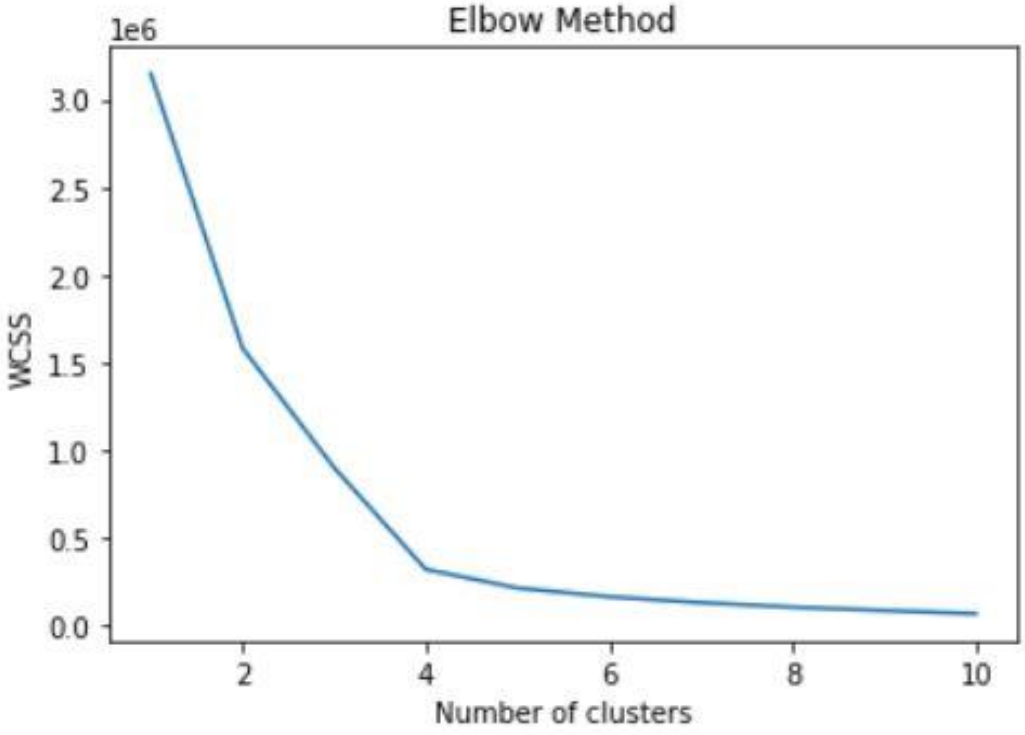


Fig 4.1 Clusters Obtained using Elbow Method

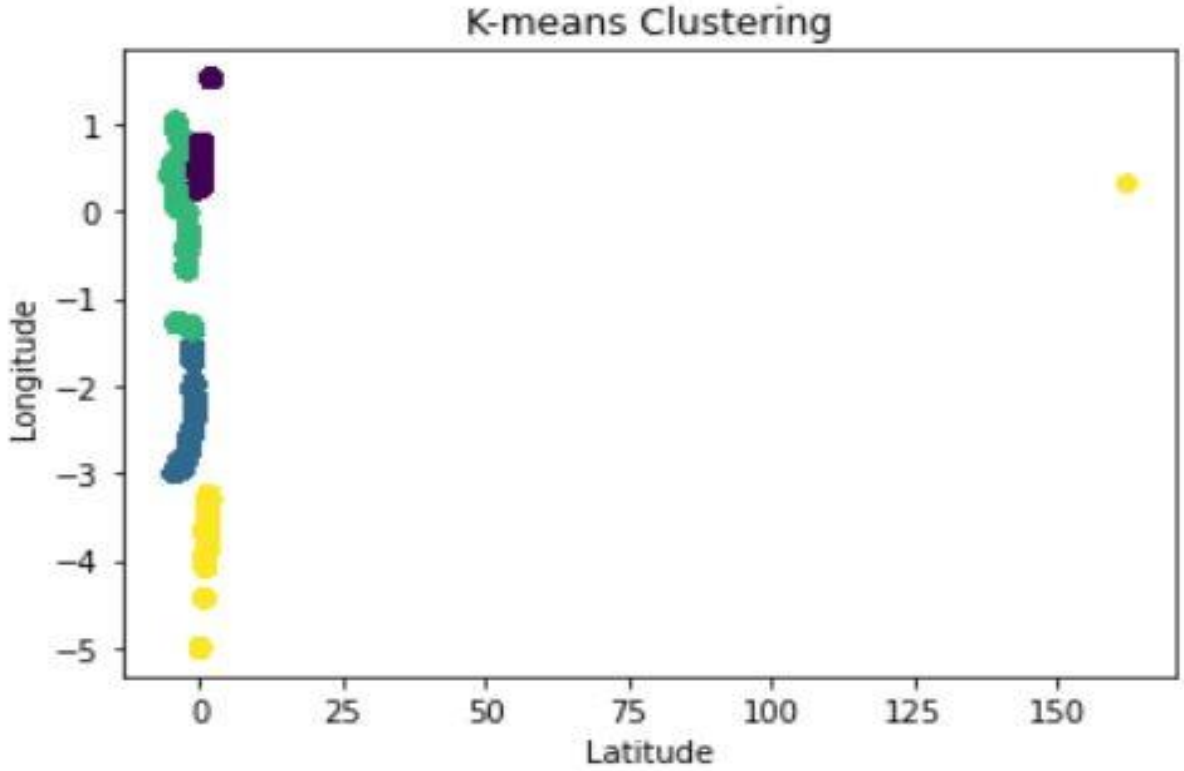


Fig 4.2 Latitudes and Longitudes

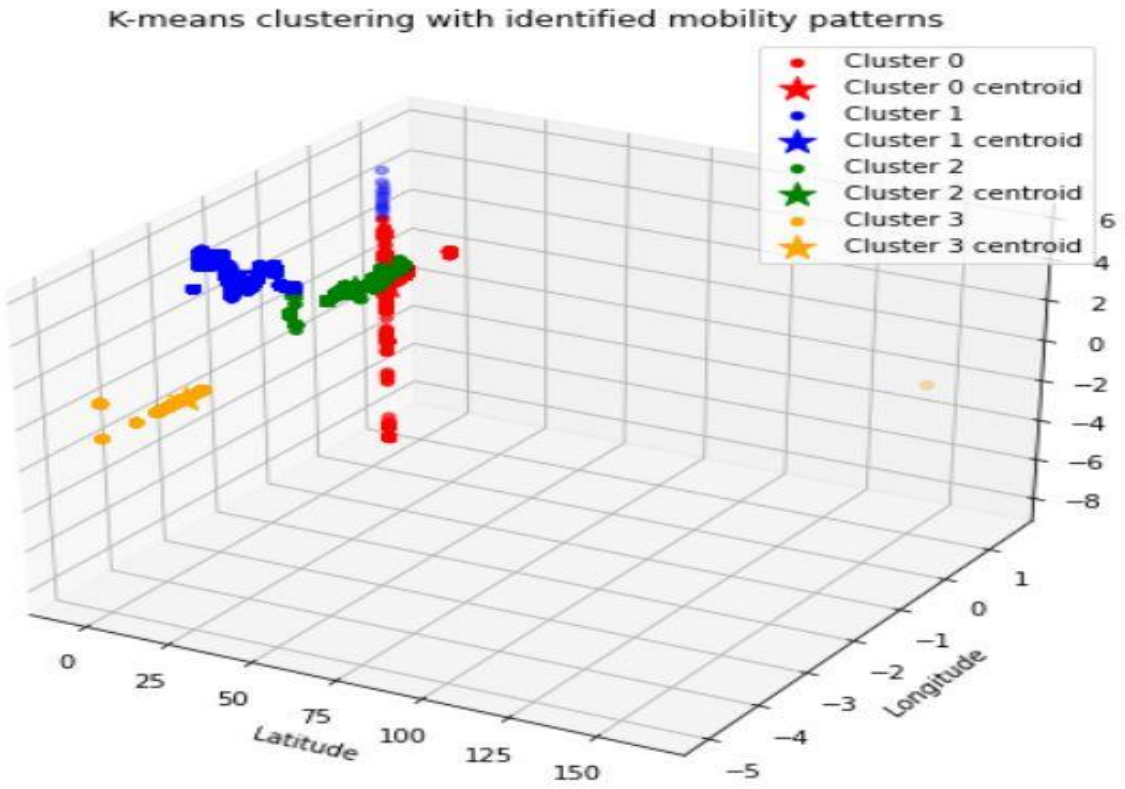


Fig 4.3 Mobility Patterns

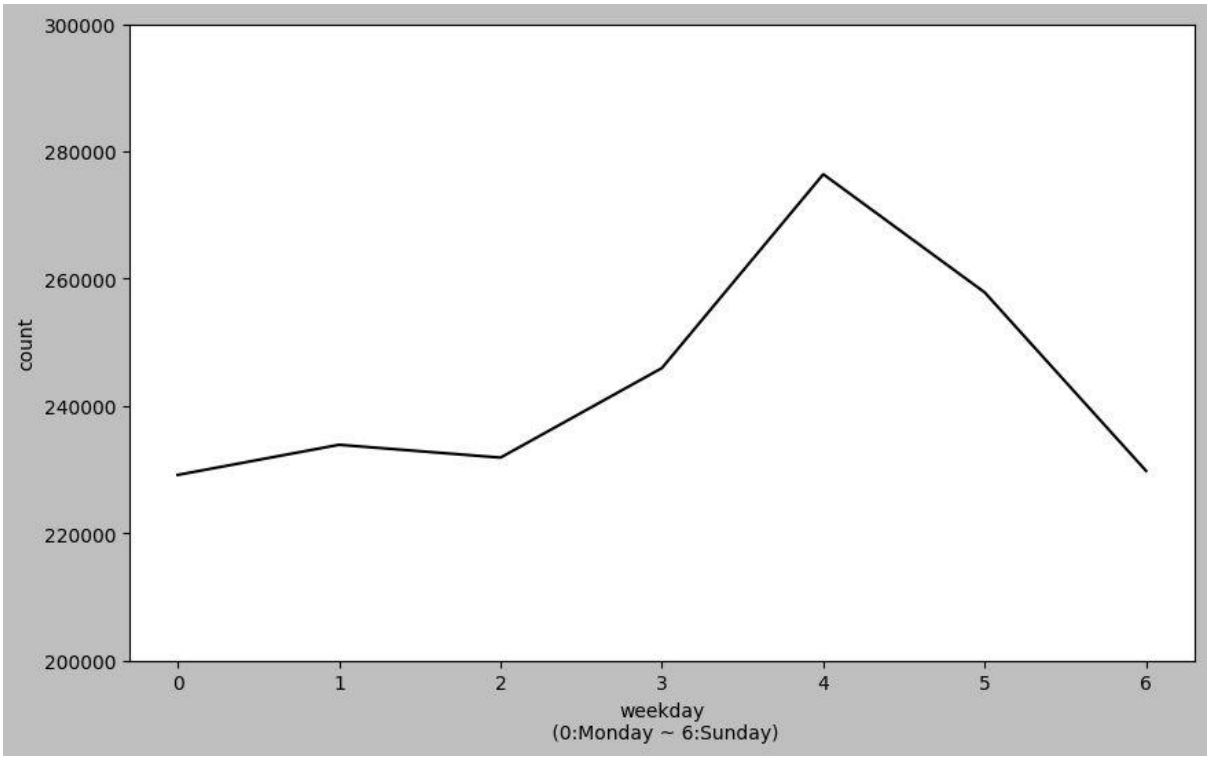


Fig 4.4 Weekday Count Graph

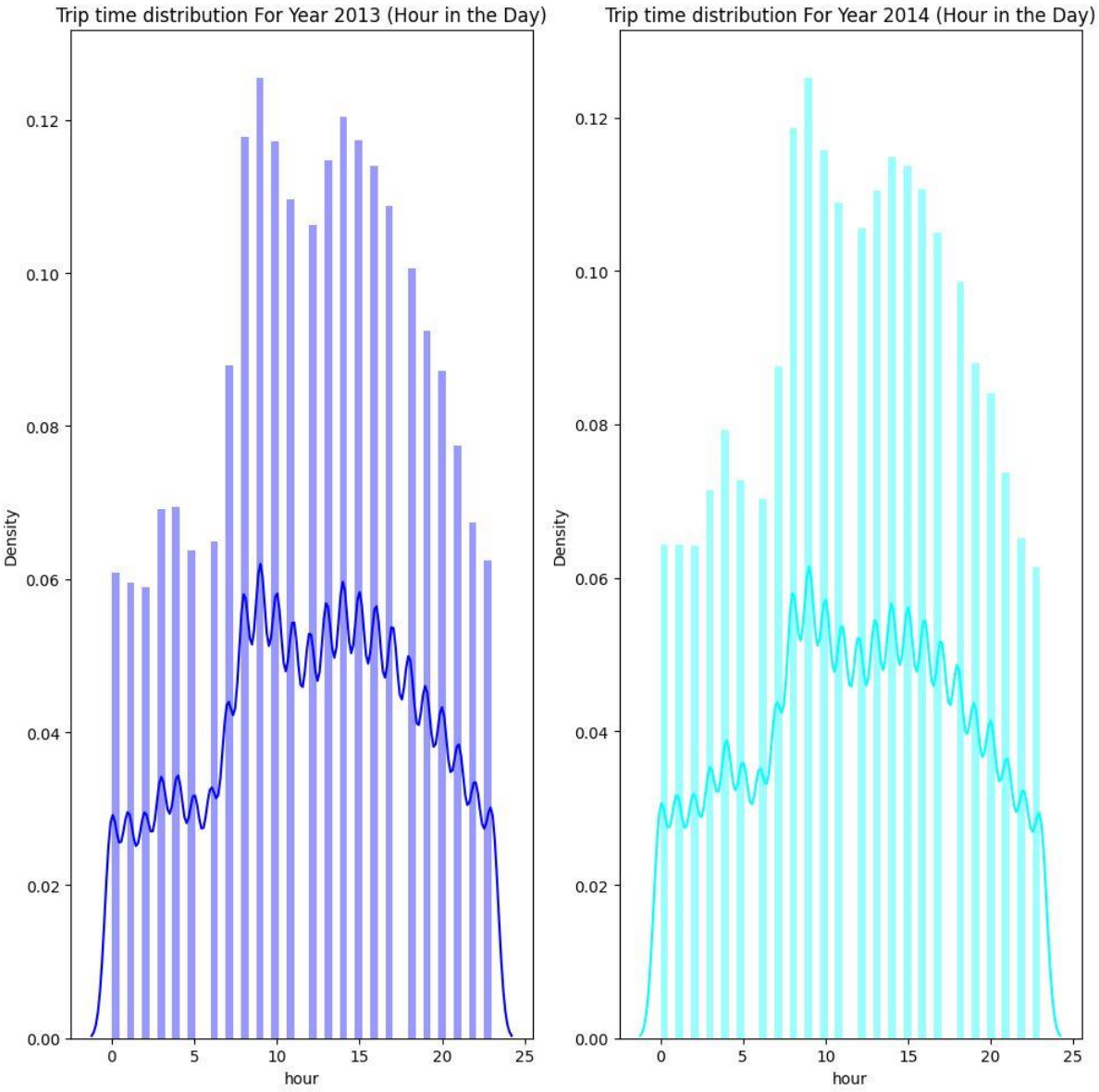


Fig 4.5 Trip Time Distribution Graph for Year

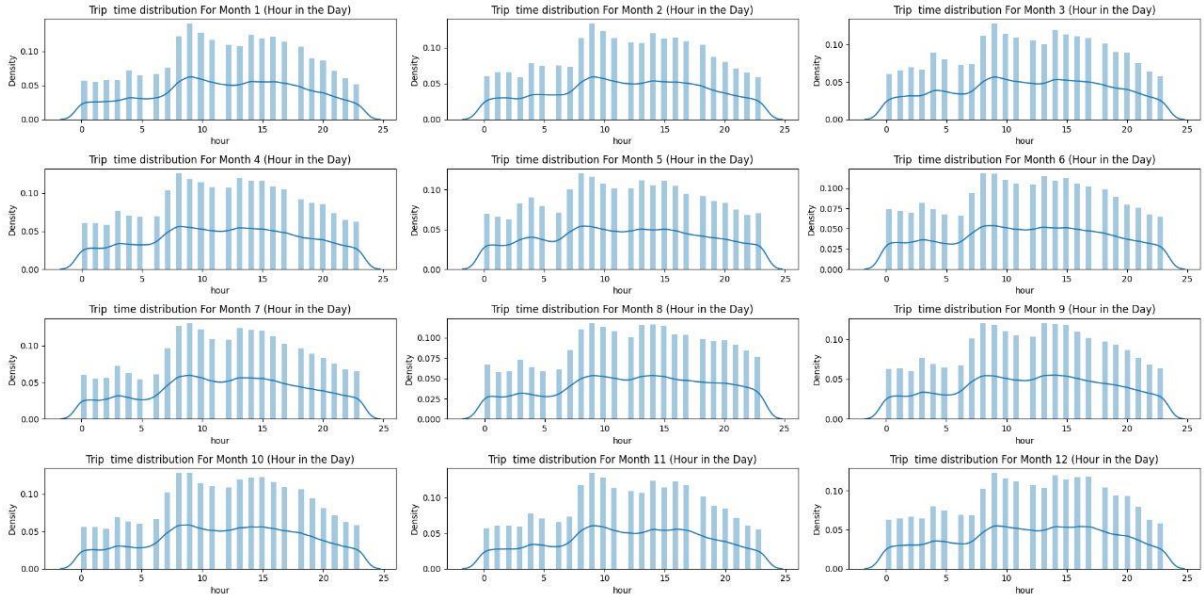


Fig 4.6 Trip Time Distribution Graph for Month

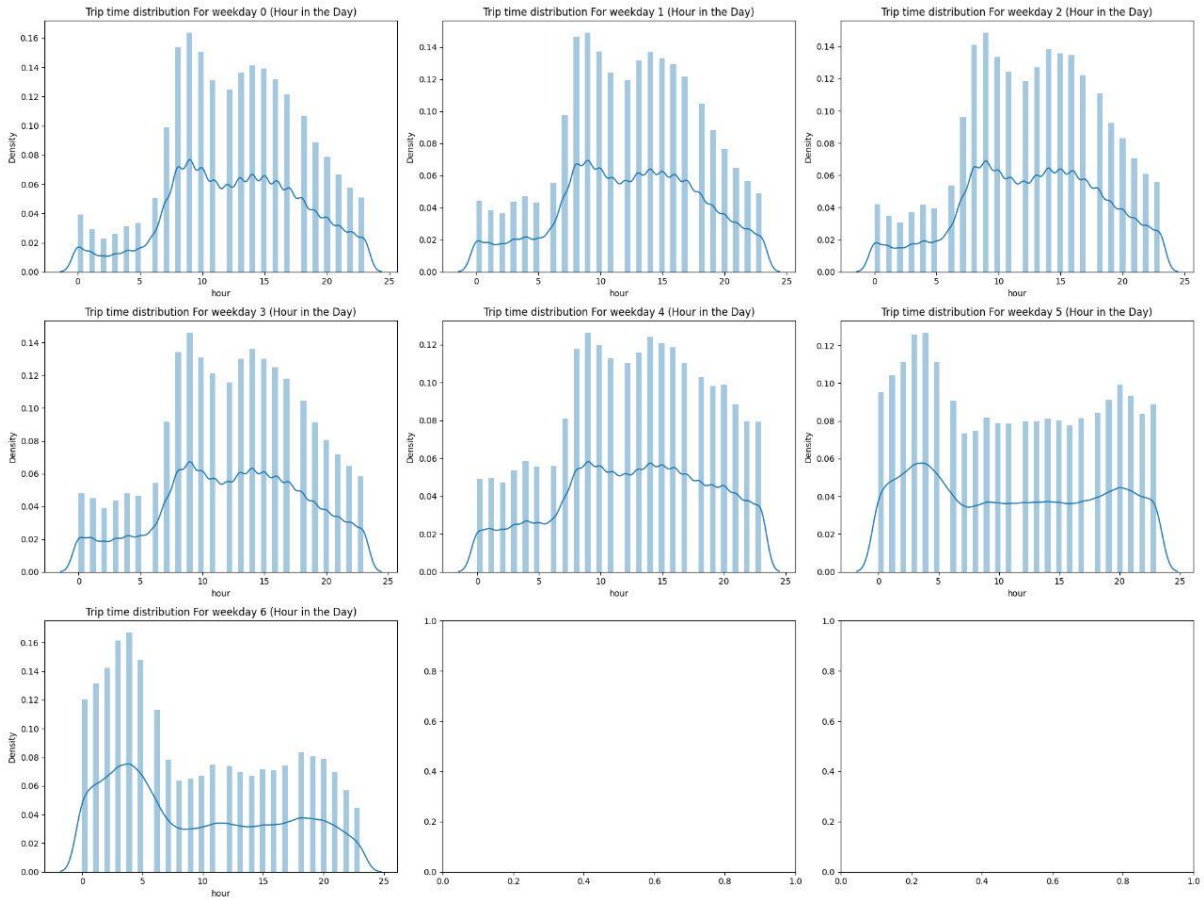


Fig 4.7 Trip Time Distribution Graph for Weekday

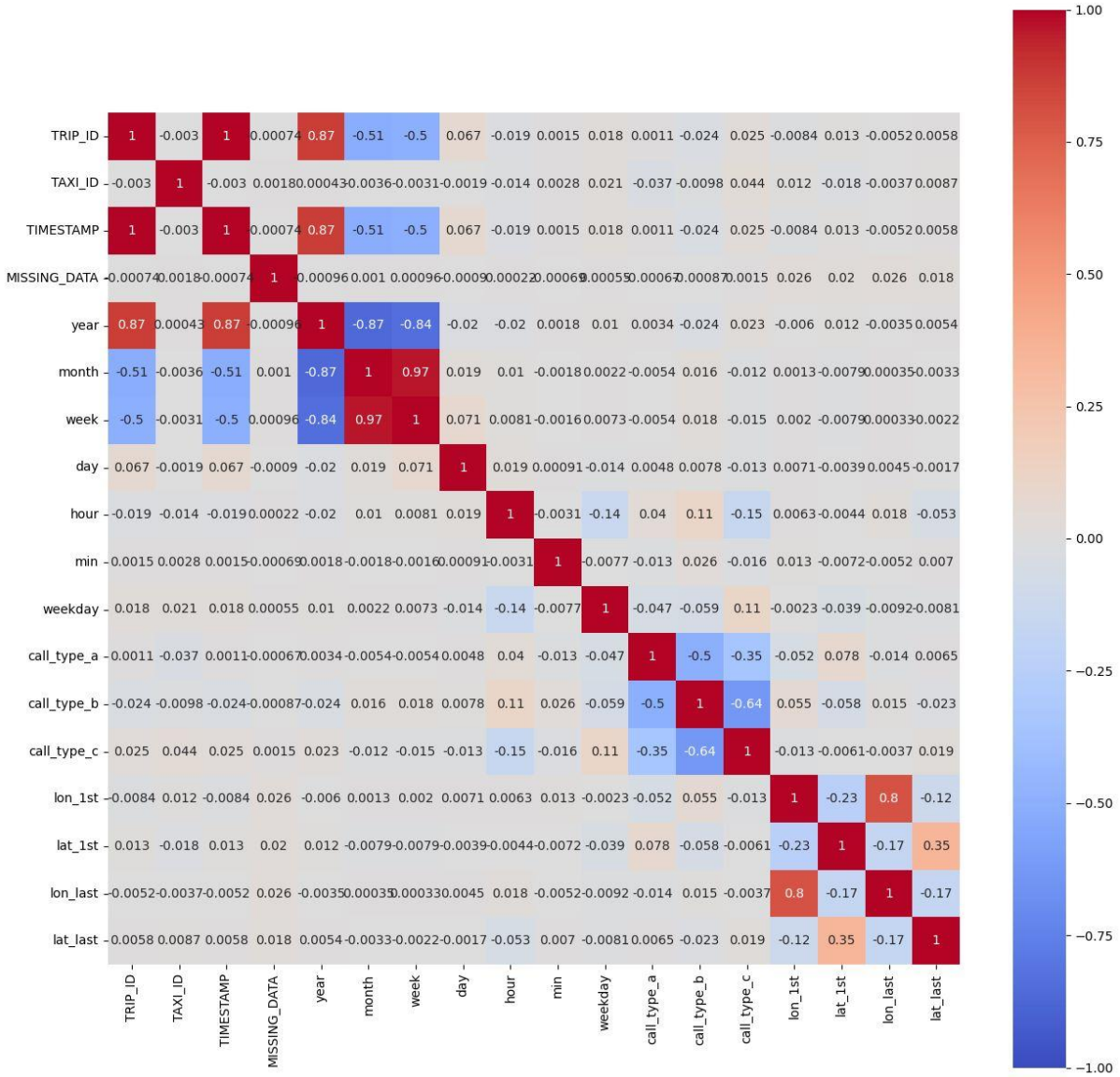


Fig 4.8 Heat Map

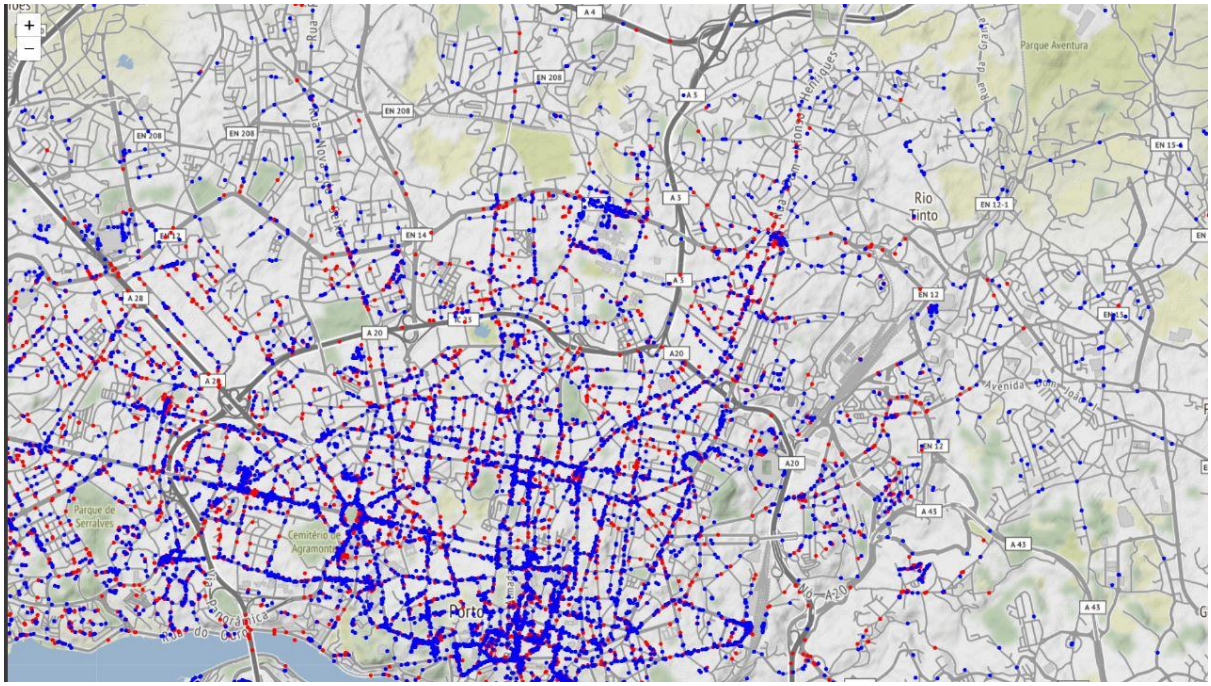


Fig 4.9 Mobility patterns on Porto city map.

CHAPTER - 5

RESULTS AND DISCUSSIONS

The GPS trajectory data clustering and analysis project produced helpful trajectories for analysing traffic. Using the K-Means Algorithm mobility patterns are identified and using random forest algorithm the trip time distribution for a day is obtained which is used to identify the busiest day of the week. The graph has the highest count of trips. Elbow point gives us the optimal number of clusters. Heat map gives us the corelation between features and gives best ones among all features. Latitudes and Longitudes are obtained using them mobility patterns are obtained. The graphs show trip distribution time comparing two years, months, and weekdays. The map has two coloured set of points. Red indicates pick up points and blue indicates drop points. The busiest day is found to be Friday.

CHAPTER – 6

CONCLUSION

In conclusion, K-Means clustering and Random Forest regression algorithms were used to analyse traffic at the beginning and end of trips, and the resulting data was shown on a map. Successful heat map feature analysis has also been accomplished, as well as monthly and weekly distribution time analysis. Using this mobility patterns and points that are obtained in the map we can do traffic analysis. This data helps in improving traffic management. Using the points obtained one can avoid busy routes and take less busy route to reach destination faster. The mobility patterns which are obtained help in picking up busy routes and areas. After finding out these areas we can take measures to avoid heavy traffic in such areas. Friday is the busiest day hence more measures should be taken on Friday to avoid traffic.

CHAPTER – 7

REFERENCES

- https://scikit-learn.org/stable/
- https://www.geeksforgeeks.org/machine-learning/
- https://www.w3schools.com/python/matplotlib_pyplot.asp
- https://openai.com/blog/chatgpt
- https://www.w3schools.com/python/pandas/pandas_intro.asp

CHAPTER 7

APPENDEX - A

BASE PAPER

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 7 November 2022, accepted 25 November 2022, date of publication 30 November 2022,
date of current version 5 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3225646

RESEARCH ARTICLE

Clustering and Analysis of GPS Trajectory Data Using Distance-Based Features

ZANN KOH[1], YUREN ZHOU[1], (Member, IEEE), BILLY PIK LIK LAU[1], (Member, IEEE), RAN LIU[1], (Senior Member, IEEE), KENG HUA CHONG[2], AND CHAU YUEN[1], (Fellow, IEEE)

[1]Engineering and Product Development Pillar, Singapore University of Technology and Design (SUTD), Singapore 487372
[2]Architecture and Sustainable Design Pillar, Singapore University of Technology and Design (SUTD), Singapore 487372

Corresponding author: Zann Koh (zann_koh@mymail.sutd.edu.sg)

This work was supported by the Singapore Ministry of National Development and the National Research Foundation, Prime Minister's Office under the Land and Liveability National Innovation Challenge (L2 NIC) Research Programme, under Award L2NICTDF1-2017-4.

ABSTRACT The proliferation of smartphones has accelerated mobility studies by largely increasing the type and volume of mobility data available. One such source of mobility data is from GPS technology, which is becoming increasingly common and helps the research community understand mobility patterns of people. However, there lacks a standardized framework for studying the different mobility patterns created by the non-Work, non-Home locations of Working and Nonworking users on Workdays and Offdays using machine learning methods. We propose a new mobility metric, Daily Characteristic Distance, and use it to generate features for each user together with Origin-Destination matrix features. We then use those features with an unsupervised machine learning method, *k*-means clustering, and obtain three clusters of users for each type of day (Workday and Offday). Finally, we propose two new metrics for the analysis of the clustering results, namely User Commonality and Average Frequency. By using the proposed metrics, interesting user behaviors can be discerned and it helps us to better understand the mobility patterns of the users.

INDEX TERMS Urban mobility, insight extraction, daily characteristic distance, GPS trajectories.

I. INTRODUCTION

Global Positioning System, or GPS for short, has been around for many years and is increasingly being used in the context of mobility studies. It has been found to be widely usable for collection of spatio-temporal data on different scales [1]. GPS mobility data has been used in many different fields and applications, such as finding efficient routes [2], understanding the progression of infectious diseases [3], and prediction or inferring of demographic information of users [4], [5].

Many studies analyze GPS data in conjunction with other data, such as demographic data [6], [7], supplementary survey data [1], Wi-Fi and geolocation data [8], or even sound and light data [9]. With increasing privacy concerns in recent years, it has become more difficult to obtain such data for large numbers of volunteers. Additionally, large volumes of human movement data are created without such supplementary data. To be eventually able to make use of this, we want to explore ways in which we can analyze GPS data without the need for additional supplementary data. Zhu et al. [10] found that the user's socio-demographic role can be predicted with high accuracy using long-term GPS data, which supports the idea that Working and Nonworking users may have different mobility patterns. In addition to this, Nahmias-Biran et al. [11] found several distinct clusters of activity-travel patterns in their GPS-enriched travel survey dataset, which included distinct temporal patterns of different Out-of-Work activities as well as different Leisure activities. This leads us to examine the mobility patterns of Workdays and Offdays separately.

Although there are many works that have used GPS data in many different applications, there lacks a study that compares the mobility patterns of Working and Nonworking users, with a focus on non-Home, non-Work locations, on Workdays as compared to Offdays. Some challenges faced in this research are ensuring a fair comparison between users who live and work at different locations, as well as a fair comparison between Workdays and Offdays. To this end, we propose

The associate editor coordinating the review of this manuscript and approving it for publication was Wentao Fan.

a new mobility metric that excludes the effects of Home and Work locations and uses the user's Home location as a reference point. We decided to use an unsupervised machine learning method - clustering, which finds groups in data without the need for labels or ground truth. We use the above-mentioned metric for each user in conjunction with other features as an input for the clustering algorithm.

Therefore, this paper has the following contributions:

- We propose a new mobility metric, Daily Characteristic Distance (DCD), as a fair basis to compare the mobility of Working and Nonworking users on Workdays and Offdays separately, even with differing distances between the Home and Work locations for different users.
- We use the DCD to extract features from users and use these features in conjunction with Origin-Destination (OD) matrix features to cluster users using k-means clustering on a real-world dataset collected in Singapore.
- Finally, we analyze the cluster results using two other analysis metrics that we have proposed - User Commonality and Average Frequency, which utilize information from within the clusters to gather higher-level insights.

The structure of the remaining sections will be as follows: Section II lists some related works in the field of GPS tracking and clustering. The dataset and preprocessing procedures are presented in Section III, while the proposed methodology is presented in Section IV. Section V shows the results and analysis of performing our proposed methodology on Workday data, while Section VI does the same for Offday data. Lastly, Section VII concludes the paper.

II. RELATED WORKS

This section will be split into three parts. The first part addresses past works on the analysis of human mobility via the usage of GPS data. The second part addresses the selection of clustering algorithms used for this paper. Lastly, the third part deals with mobility metrics.

A. WORKS USING GPS DATA

There have been several works focusing on the use of GPS data in mobility studies. Van der Spek et al. [1] used GPS to collect data in three European city centers, as well as track the activity data of 13 families in Almere (a city in The Netherlands) for one week and conclude that GPS offers wide usability in the collection of invaluable spatial-temporal data on different scales and in different settings, adding new layers of knowledge to urban studies.

Some studies make use of external data to supplement GPS data in order to gain additional insights. Sila-Nowicka et al. [6] performed an analysis of significant places identified from their GPS data in conjunction with additional social demographic data, while Long and Reuschke [7] even use detailed GPS data to analyze the effects of employment type (business owners or employees) on daily mobility. Marakkalage et al. [12] used a fusion of GPS data and Wi-Fi data to derive insights on neighbourhood activity and micro mobility.

GPS datasets may also provide an avenue for modelling human mobility if they are large enough. Alessandretti et al. [13] presented an extensive characterization of the statistical properties of GPS trajectories using a dataset collected from around 850 people lasting around 25 months, while Solmaz et al. [14] used GPS traces to model and simulate pedestrian mobility in disaster areas.

Other machine learning approaches in the analysis of GPS data include supervised learning, where Zheng et al. [15] proposed an approach based on supervised learning to infer people's motion modes from their GPS logs, as well as anomaly detection, where Wang et al. [16] proposed a hierarchical clustering method using an improved edit distance algorithm to detect anomalous taxi trajectories between selected pairs of origins and destinations.

From the above, we understand that GPS data can be a rich source of mobility information about users, even extending to other aspects of demographics. As our focus is more on unsupervised machine learning as compared to prediction, we turn our focus to the application of clustering methods in the analysis of GPS data.

B. CLUSTERING ALGORITHMS IN THE ANALYSIS OF GPS DATA

Some authors have performed clustering on trajectories to find common routes or popular locations. An example would be Cesario et al. [17], who proposed their own algorithm, Trajectory Pattern Mining (TPM), and used it to discover dense regions and popular sequential patterns within their dataset. Kumar et al. [18] also proposed their own novel algorithm, with the aim of discovering clusters of taxi routes. Dodge et al. [19] proposed their own framework to assess movement similarity using symbolic representation of movement parameters and used it to cluster trajectories of hurricanes and couriers according to their proposed similarity metrics. Tang et al. [20] used the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [21] algorithm to cluster locations of pick up and drop off points for their taxi GPS dataset, aiming to describe the taxi trips using statistical models. These above clustering methods are applied to the trajectories themselves and not to the users, which is less aligned with the aims of our study of the users' mobility patterns on Working and Nonworking days.

Another form of clustering in the analysis of GPS data is the grouping of users based on their travel patterns. Amichi et al. [22] used Gaussian mixture models [23] to identify three different groups of people based on their travel patterns - scouters, regulars, and routineers. However, this was mostly based on how often each user traveled to new locations as compared to returning to previously visited ones. In the long term, the number of recorded "visited" places will keep increasing, while the "new" locations will become few and far between, so this approach may not be applicable on a long term scale and is less suited for this study.

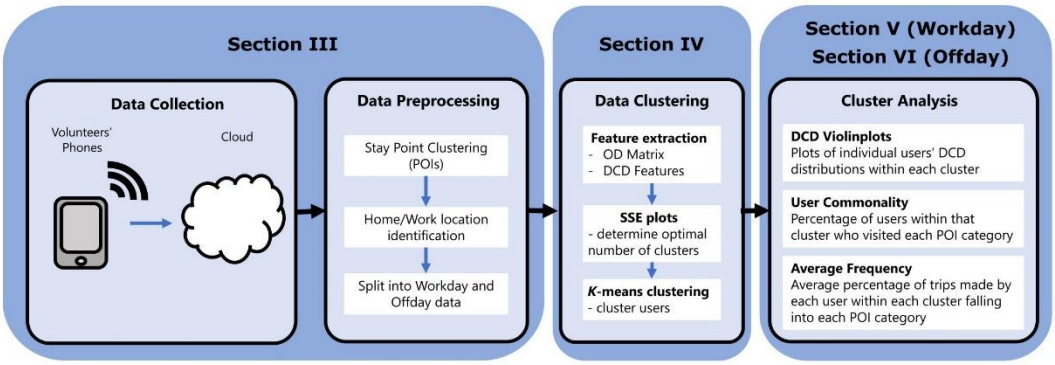


FIGURE 1. Flowchart depicting the data collection, processing, clustering, and analysis framework proposed by this paper.

Scherrer et al. [24] went through a rigorous selection process for parameters and algorithms before performing their clustering. Out of a total of four clustering algorithms, k-means clustering [25] was in at least the first two combinations in terms of their overall ranking, and they eventually used it to cluster users based on the large amount of data they gathered from a mobile application. They were able to draw conclusions from data that was gathered as a byproduct and hence without specific experimental aim or ground truth. As this is similar to our use case, we eventually decided to use k-means clustering.

C. EXISTING MOBILITY METRICS

We aim to find some simple, understandable features within our data that will result in meaningful interpretation of the clustering results. The review paper by Solmaz et al. [26] classifies mobility metrics into three different types - movement-based, link-based, and network-based metrics. As our focus is on how the users travel, we place an emphasis on movement-based and link-based metrics such as visit frequency and mean squared distance, rather than on the network-based metrics such as transmission count and energy consumption.

Movement-based metrics include visit frequency and mean squared distance. A commonly used metric that combines these is radius of gyration [27], which has been used in many papers [28], [29], [30]. It gives the characteristic distance traveled by a user within a specified time period and is calculated as the mean squared distance of the user's visited locations to the center of mass of those locations. We are interested in non-Home and non-Work locations, thus we adapted this formula based on what we have in our dataset to extract the relevant features for clustering.

For a link-based metric, those mentioned by Solmaz et al. [14] such as node density and intercontact time were difficult to apply in our dataset. We then considered Origin-Destination (OD) matrices, which have been commonly used in literature for analyzing flows between locations. For example, they have been used by Zhou et al. [31] and Koh et al. [32] to illustrate the probability of human flows between different fixed nodes. However, we believe that they can be used to describe an individual's probability of motion between different locations as well, much like the links of a Markov chain model, which has been shown to have relatively high prediction accuracy [33] for trajectories. Based on this, we propose a method of extracting OD matrix features in Section IV-A-2.

III. DATA COLLECTION AND PREPROCESSING

The overall data flow of this paper is summarized in Fig. 1. This section deals with the Data Collection and the Preprocessing parts.

Timestamped GPS data was collected through a user-installed smartphone application that runs in the background and collects GPS data adaptively. When moving, data is recorded about once every minute, whereas when the device is still or not moving, the data is recorded about once every five minutes. Data collection was carried out over a range of different periods of time for different users. Usable data was selected with the criteria of at least one month of valid data accounting for at least 50% of the recording duration for each user, resulting in the data from a total of 73 users being selected for use. Although the sample size is relatively small, it is due to strict criteria to ensure quality of the data used. Additionally, for this paper, we focus mainly on the framework, which can be extended to other datasets with larger sample size in the future.

Each detected point of the data consists of a latitude, longitude, start time, and end time. For each user, individual points at similar coordinates were clustered together using a validation based stay point detection algorithm [34] to identify points of interest (POIs). Home and Work locations for each user are then detected from this set of POIs using frequency and stay duration given the time of day, roughly based on a Monday to Friday workweek within standard office and retail hours. Taking into consideration that there

TABLE 1. Labels for the different POI types considered in the dataset.

Label	Description
Attraction	Places that tourists tend to visit
Healthcare	Hospitals, clinics etc.
Neighborhood Center	Community clubs, hawker centers, markets, etc.
Park	Public parks and gardens
Places of Worship	Temples, mosques, churches, etc.
Playground	Playgrounds
Recreational	Places that locals tend to visit for leisure
Shopping Mall	Shopping malls
Transportation	Train stations, bus interchanges, etc.
Residential	Condominiums, public housing estates, etc.

may be differences in mobility patterns on Workdays and off days for the Working population, the POI data was then separated into Workday data and Offday data. Workdays are taken to be days when the user was detected at their Work location. Some of the users who are not detected in a fixed 'Work' location during those standard hours are considered as Nonworking users and all of their data is considered to lie in the Offday category.

Next, each POI is manually labeled by its proximity to the nearest location with a specific type out of ten different categories. If it is more than 400m away from any location with known POI types, it is left unlabeled. The labels are as shown in Table 1.

Finally, to minimize the impact of the GPS inaccuracy, the data points were then assigned to specific areas called subzones, which are small sections of planning areas delineated by the Urban Redevelopment Authority of Singapore for statistical purposes [35]. These subzones were used in the extraction of clustering features, which can be found in Section IV-A-2.

IV. PROPOSED CLUSTERING METHODOLOGY

This section will go into details of the feature extraction and clustering processes. These processes differ slightly between the Workday and the Offday datasets. The aim of clustering these users is to find common types of users based on their mobility patterns, and thus possibly derive insights into common mobility patterns.

A. FEATURE EXTRACTION

For the purposes of clustering, we extract two main types of features from each user. One is derived from a proposed metric, Daily Characteristic Distance (DCD), while the other is derived from the Origin-Destination (OD) matrix of each user's individual trips.

1) DAILY CHARACTERISTIC DISTANCE

We are interested in mobility patterns for different users in the dataset. As different users have different Home and Work

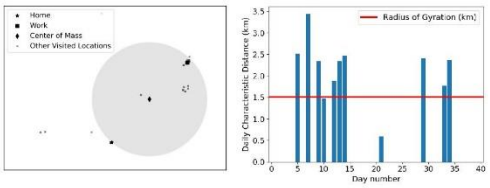


FIGURE 2. (a) Illustration of radius of gyration. The shaded gray circle is centered at the computed center of mass, with a radius equal to the computed radius of gyration. (b) Comparison between radius of gyration (single value, red line) and proposed Daily Characteristic Distance (set of values, bar plot) over the same time period.

coordinates, it is imperative to find a mobility metric that enables us to compare different users despite this spatial restriction. One such metric in the literature to quantify the mobility of individuals is the radius of gyration, which considers distances from the center of mass of a trajectory and is thus user-dependent. The radius of gyration r_g of a user a from the start of their dataset up to a certain time t was proposed by Gonzalez et al. [27] and expressed by Equation 1:

$$r_g^a(t) = \sqrt{\frac{1}{n_c^a(t)} \sum_{i=1}^{n_c^a} (\vec{r}_i^a - \vec{r}_{cm}^a)^2} \tag{1}$$

where $\vec{r}_i^a$ represents the $i = 1, \ldots, n_c^a(t)$ positions recorded for user a and $\vec{r}_{cm}^a = \frac{1}{n_c^a(t)} \sum_{i=1}^{n_c^a} \vec{r}_i^a$ is the center of mass of the trajectory.

For the purpose of comparison between different users in our dataset, the time t in the above equation is taken to be the entire duration of each user's dataset, as the duration varies between users. Thus, the value of $n_c^a(t)$, which originally refers to the number of recorded positions of user a up to time t, becomes the total number of locations N^a visited by user a and the time dependency is removed. The simplified equation is as shown in Equation 2.

$$r_g^a = \sqrt{\frac{1}{N^a} \sum_{i=1}^{N^a} (\vec{r}_i^a - \vec{r}_{cm}^a)^2} \tag{2}$$

An illustration for this metric is shown in Fig. 2(a) for a user with a 40-day dataset. As expected, the center of mass lies between the Home and Work location, as those locations are visited with a higher frequency than other locations.

However, there are some things that can be added to this metric, which it currently lacks. Firstly, as our dataset has labels for Home locations of each user, we can add more meaning to the metric by using the Home location of each user as the reference point for distance calculations instead of the computed center of mass of the user's visited locations. This will allow us to know the characteristic distance that the user travels from their Home, which may have more physical meaning than a computed center of mass of their trajectory. Secondly, the radius of gyration metric currently produces

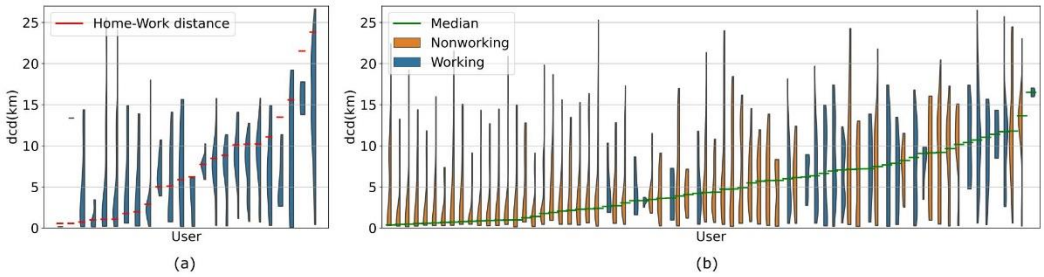


FIGURE 3. Violinplots illustrating the DCD distributions of users on (a) Workdays, consisting of only Working users, and (b) Offdays, consisting of both Working users on Offdays and Nonworking users on all days. (a) shows a moderately high correlation between DCD peaks and Home-Work distance of each user, while (b) shows a higher density of Working users with higher median DCD.

one value per user. We propose to break down the dataset into days and compute a value for each individual day, thus obtaining a distribution of the daily distances traveled by the user. Each day's characteristic distance may be affected by whether or not the user went to work on that day, which is part of what we want to investigate. Lastly, we want to investigate the locations that the users visit outside of Home and Work. Therefore, we manually negate the contribution of the Work and Home locations in the calculations by setting their distances to zero and removing the count of Home and Work visits from the total value of N^a. As our proposed new metric computes a characteristic distance for each day of the dataset, we call it Daily Characteristic Distance (DCD).

The DCD for day d of the dataset is given by Equation 3:

$$DCD_d = \sqrt{\frac{1}{n_d}\sum_{i=1}^{n_d} f_{id} \times (\vec{r}_{id} - \vec{r}_{home})^2} \tag{3}$$

where n_d is the number of unique POIs that the user traveled to on that day, f_{id} is the number of times the user traveled to location i where $i = 1, \ldots, n_d$ on that day, and $\vec{r}_{home}$ is given by the mean coordinates of the Home location of the user. Fig. 2(b) illustrates the difference between radius of gyration (one value per user) and DCD (a set of values per user). The days without bars have a value of zero, indicating that on those days the user traveled directly between Home and Work without visiting any other location.

The obtained DCD distributions of all users are plotted in Fig. 3. The Workday data of Working users is plotted in Fig. 3(a), while the Offday data (consisting of Working users during Offdays and Nonworking users on all days) is plotted in Fig. 3(b). In Fig. 3(a), we have sorted the distributions in ascending order of Home-Work distance of each user. From this, we can see that there generally seems to be a relationship between the Home-Work distance of the users and the location of the peaks of their DCD distributions. We calculated the Pearson's R-value between each user's Home-Work distance and the median of their DCD distribution and found that there was a moderately high R-value of 0.746, with a p-value of 2.90e−5.

For the Offday data, since there is no second location outside of Home that was fixed for every user, we could not apply this method. Instead, we have sorted the distributions in ascending order of their median point and colored the distributions according to whether the user is a Working or Nonworking user. From Fig. 3(b), we can see that there is a higher concentration of Working users (blue) at the side with higher median DCD. This may indicate that Working users tend to visit locations at further distances from their homes as compared to Nonworking users.

To use this new metric DCD as a clustering feature, we first break down all the data for all users into Workday and Offday data. We then separately compute the DCD values for each day and plot separate histograms of the DCD values over all relevant users on Workdays and Offdays, as shown in Fig. 4. From these histograms, we visually obtain the four thresholds of 0km (Home and/or Work only), 0 to 5km, 5 to 15km, and >15km by observing suitable valleys.

Each user's data is then broken down into Workday (if applicable) and Offday data. For each type of data, we calculate the percentage of DCD values that fall within each of the determined thresholds. This gives us four features for each type of data that add up to 1.0. An example of the features for one user, User 2, is shown in Table 2.

These four features will be used in conjunction with the 16 features derived from the Origin-Destination Matrix, explained below:

2) ORIGIN-DESTINATION MATRIX

From each user's trajectory, we can extract distances from each user's Home (and Work location if applicable) to the other POIs that the user visits. For Offdays, we can simply use the distance from Home to that POI as there is no other location that is common to all users. However, there is an additional important location for Workdays, which is the Work location of the user. Therefore, on Workdays, the distance value of each POI, referred to as minimum distance, is taken as the minimum of the distances between the POI and the user's Home location and between the POI and the

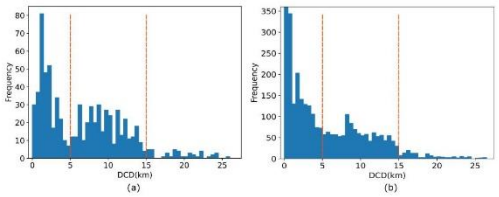


FIGURE 4. Histograms of the number of days within the whole dataset of (a) DCD value on Workdays, and (b) DCD value on Offdays. Note that (b) has been cropped vertically to show greater detail - the leftmost bar has an actual value of 3312, of which 3137 have a value of 0.

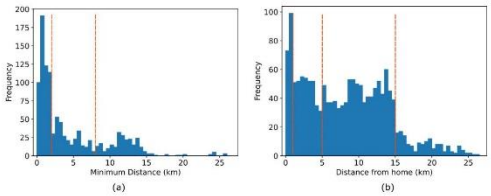


FIGURE 5. Histograms of the number of POIs visited over the whole dataset of (a) minimum distance between Home and Work to that location on Workdays, and (b) distance from home to that location on Offdays.

TABLE 2. Example of the four DCD features for Workday and Offday data.

Day Type	Home/Work	0-5km	5-15km	>15km
Workday	0.60	0.23	0.15	0.02
Offday	0.38	0.16	0.41	0.05

TABLE 3. Example of the 16 OD matrix features for Workday and Offday data.

(a) Workday features

Threshold	Home/Work	0-2km	2-8km	>8km
Home/Work	0.68	0.08	0.04	0.04
0-2km	0.08	0.00	0.00	0.00
2-8km	0.04	0.00	0.00	0.00
>8km	0.03	0.00	0.00	0.01

(b) Offday features

Threshold	0-1km	1-5km	5-15km	>15km
0-1km	0.00	0.08	0.23	0.03
1-5km	0.08	0.03	0.00	0.00
5-15km	0.19	0.04	0.29	0.00
>15km	0.02	0.00	0.01	0.00

user's Work location. This is to detect any specific locations that users may go to, that is not nearby to either their Home location or Work location and hence "out of the way" from the user's point of view. After extracting these distances separately from the Workday and Offday data, the corresponding histograms are plotted. These can be seen in Fig. 5. We obtain the thresholds visually by selecting suitable valleys in the histograms. The thresholds for Workdays are 0km (direct trips between Home and Work), 0-2km, 2-8km, and >8km, while the thresholds for Offdays are 0-1km, 1-5km, 5-15km, and >15km.

After getting these thresholds, the trips made by a user are now categorized based on these thresholds. We are interested in the combinations of movements that users make from threshold to threshold, and whether these will be a significant distinguishing factor between different users' mobility patterns. Taking an example of a user with Workday data, a trip consists of going from location A to location B, where threshold A is on the row of the matrix and threshold B is on the column of the matrix. If A is located within 0-2km and B is located within 2-8km, the number corresponding to the "0-2km" row and the "2-8km" column will be increased by 1. After the trips are all counted for a user, the matrix is normalized by the total number of trips counted for that user, such that all 16 elements of this matrix add up to 1.0. This is to make the data comparable between different users. An example of the resulting matrix using the Workday thresholds can be seen in Table 3. The Offday matrix and features are computed similarly.

We do not count trips occurring on different calendar dates (i.e. from the last POI on one day to the first POI the next morning), and we also do not count trips that occur within the same subzone (e.g. Home to Home).

These 16 features are then concatenated with the four features from the above DCD calculations to form the 20 features used in clustering, which will be discussed in detail in the next subsection.

B. PROPOSED CLUSTERING PROCESS

After extracting the features for each user as in the above subsection, we performed some initial test clustering, using Euclidean distance as a distance measure between users, to obtain the sum-of-squared errors (SSE) plot. This plot was used to decide on the number of clusters, k, to be used as the input parameter for k-means clustering [25].

The SSE plot measures the sum of all squared errors from the clustered points to their respective cluster centers after being grouped using each value of k. As the value of k increases, the SSE naturally decreases, but a good value for k would be one located at the 'elbow' of the plot, just before the decrease in SSE becomes less than proportionate to the increase in k. The SSE plots for our dataset can be seen in Fig. 6, where Fig. 6(a) shows the plot using the data from Workdays, while Fig. 6(b) shows the plot using the data from Offdays. From both SSE plots, the 'elbow' of the plot indicates that a good value of k to use would be $k = 3$. The detailed results are plotted in the following sections, with the Workday results presented first before Offday results.

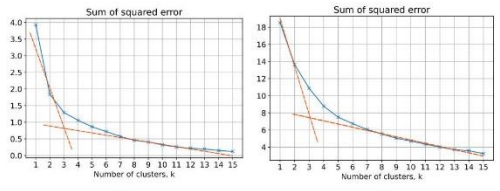


FIGURE 6. SSE plots used to derive (a) the optimal number of clusters for Workdays and (b) the optimal number of clusters for Offdays. Both plots indicate 3 as a suitable value for k, the number of clusters.

V. WORKDAY CLUSTERING AND ANALYSIS

This section focuses on the results obtained from clustering the Workday data of working users. The eventual aim of this clustering is to separate users into different clusters based on their workday data. Further analysis is then performed on the identified clusters, which consists of the DCD violinplots for each user in each cluster, as well as user commonality and average frequency heatmaps, which are explained in detail later on. The same process will be repeated for the Offday data in the next section.

A. CLUSTERING RESULTS–CENTROID VALUES

The centroid values of each cluster are shown in Fig. 7. These values represent the average percentage of trips within each threshold (for the first 16 values in the O-D matrix) and the last 4 values represent the percentage of days for each user that have DCD values within each of the 4 distance thresholds, as described in Section IV-A. The clusters are named W1, W2, and W3 respectively (W stands for Workday). Visually, it seems that the clusters are separated mainly based on the percentage of Home/Work trips out of the total number of trips taken by the user, with cluster W1 having the highest average percentage of direct Home-Work trips at 77% followed by W2 with 42% and W3 with 21%. The DCD features below each OD matrix show that there are similar average percentages of Home/Work Only trips and Home/Work Only days.

Users from Cluster W1 have a large majority - on average 72% - of their Workdays where they do not visit any other locations. The average percentage of their days spent with DCD at each distance threshold decreases with increasing distance.

Looking at Cluster W2, the DCD features are roughly evenly spread across the first three distance thresholds. Since there is a higher value of DCD being within 5-15km as compared to 0-5km, while the percentage of trips from the OD matrix indicate a higher emphasis on minimum distance between 0-2km, it is likely that some of the locations, which are 5-15km from their Home, are actually within 0-2km of their workplace, leading to a lower value for minimum distance.

For Cluster W3, the DCD features have a surprisingly high average value of 55% in the 5-15 km threshold as compared to 12% and 31% the other two clusters. It also has a much lower average value of 8% in the 0-5km threshold, as compared to 15% and 27% in the other clusters. As the average percentage of Home/Work direct trips from the OD matrix is also quite low at 21%, it can be interpreted that the users in this cluster usually travel quite far from their Home and Work locations.

Overall, the clusters can be described as mainly Home/Work Only (W1), frequent short trips in terms of Minimum Distance (W2), and mostly longer trips (W3).

B. CLUSTER ANALYSIS–DCD VIOLINPLOTS

Fig. 8 shows the DCD violinplot for each individual user in each cluster, sorted in ascending order of their Home-Work distance. These violinplots do not show the percentage of days spent only between Work and Home, as we are interested in the POIs that are not Home and not Work. From this figure, we observe from the yellow highlighted portion that most of the users in cluster W1 have a low Home-Work distance, below 5km. This may be a factor in these users having the highest percentage of direct trips between Home and Work out of the three clusters on Workdays. The users in cluster W2 have Home-Work distances in the middle range, and usually the peaks of their DCD distributions are located close to the Home-Work distances. This is also reflected in their OD matrix, in which this cluster has the highest percentage of trips within 0-2km of either their Home or Work location out of the three clusters. For Cluster W3, two out of the four users have a large Home-Work distance of over 20km. Three out of the four users have DCD peaks near their Home-Work distance, but those are not reflected in the centroid OD matrix, perhaps because they travel to other places that are the same distance from their Home as well as their Work location. These DCD plots are in agreement with the DCD features for Cluster W3, as the bulk of their DCD distributions are located within the 5-15km range.

C. CLUSTER ANALYSIS–USER COMMONALITY AND AVERAGE FREQUENCY

The next two parts of cluster analysis, what we will call User Commonality and Average Frequency, are shown in Fig. 9 (a) and (b) respectively. Both of these types of analysis use the same distance thresholds for minimum distance that were used for the OD matrix features, broken down into each of the ten different POI categories that were labeled in the data. To represent User Commonality, each square in Fig. 9(a) shows the percentage of users within the cluster who fulfilled each minimum distance and POI label combination at least once in their trajectory. The aim of this is to see whether there is any minimum distance and POI label combination that is favored by the users in each cluster. The value of each heatmap square u_{jk}, in row j and column k, is given by Equation 4:

$$u_{jk} = \frac{n_{jk}}{n_c} \tag{4}$$

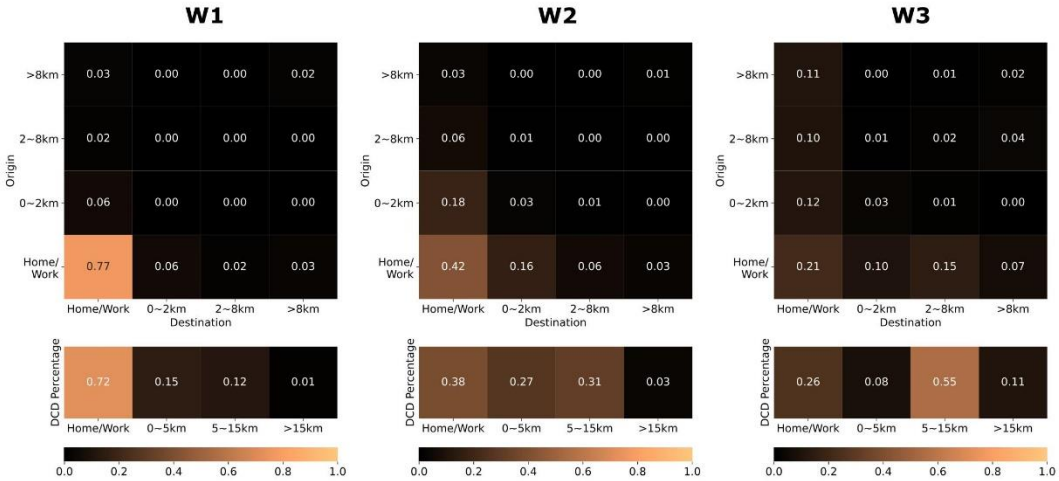


FIGURE 7. Centroid values of the three clusters obtained from clustering Workday data. Cluster W1 has the highest percentage of trips directly between Home and Work, as well as the highest percentage of days spent only at Home or Work. The other two clusters W2 and W3 are in descending order of percentage of trips directly between Home and Work.

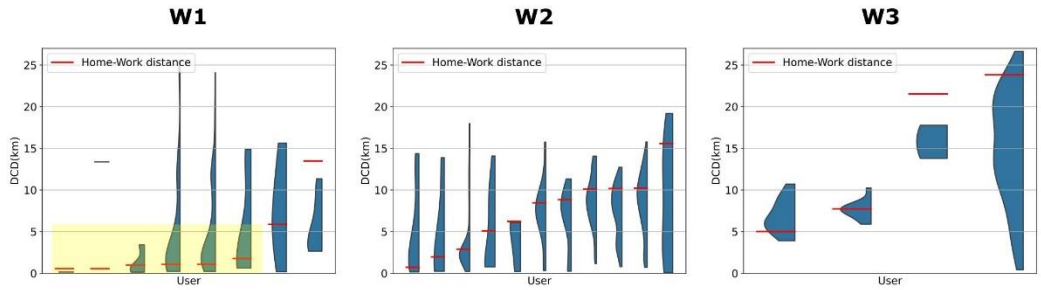


FIGURE 8. Violinplots illustrating each user's DCD distribution within each cluster on Workdays.

where n_{jk} is the number of users within the cluster who visited a POI at distance threshold j with label k, and n_c is the total number of users in that cluster.

Meanwhile, Fig. 9(b) shows the Average Frequency, taken as a percentage of the user's total trips and averaged over all users in the cluster, of each minimum distance and POI label combination. The value of each heatmap square f_{jk}, in row j and column k, is given by Equations 5 and 6:

$$P_{ijk} = \frac{p_{ijk}}{p_i} \tag{5}$$

$$f_{jk} = \frac{\sum_{i=1}^{n_c} P_{ijk}}{n_c} \tag{6}$$

where P_{ijk} is the percentage frequency of user i visiting a POI at distance threshold j and with label k, p_{ijk} is the number of POIs that user i visited at distance threshold j with label k, p_i is the total number of labeled POIs visited by user i, and n_c is the total number of users within the cluster.

The advantage of using these two analysis metrics is that they both make use of the data available within the clusters, and thus do not require an external source of ground truth, to highlight meaningful differences between the clusters that may not be apparent at first glance.

From Fig. 9(a), it can be seen that there is no single distance threshold and POI label combination that is visited by 100% of the users in each cluster, except for Shopping Malls at a minimum distance of >8km for Cluster W3. However, quite a high percentage of users in the other two clusters visit this distance threshold/POI label combination as well. Other common combinations that appear in all three clusters are: Neighborhood Center, Shopping Mall, and Residential, all within the 0-2km threshold. The distinguishing features of the clusters can be summarized as follows: Cluster W1

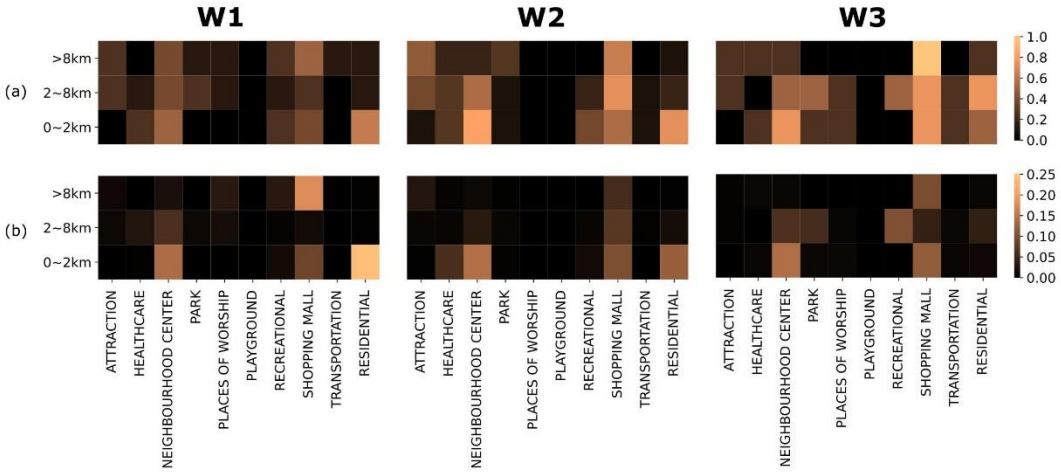


FIGURE 9. Heatmaps for each of the three Workday clusters showing (a) User Commonality and (b) Average Frequency. The colormap scales for (b) are narrowed to 0.25 to better show the contrast between the different squares.

has a visible percentage at Recreational at >8km minimum distance as compared to the others. More of the users in cluster W2 visit Attractions at a minimum distance of larger than 8km as compared to the other clusters. Users in Cluster W3 have a higher frequency of trips to various places over a variety of distance thresholds, such as Parks, Recreational, and Residential areas within the 2-8km range, as well as Healthcare, Neighbourhood Center, and Shopping Mall in the >8km range.

From Fig. 9(b), it can be seen that the POI label with common frequency among the three clusters is Neighborhood Center at 0-2km. Cluster W1 has highest Average Frequency at Residential POIs within 0-2km and Shopping Malls at >8km. In comparison to Cluster W2, which has the highest relative frequency of shopping mall trips at 0-2km, this indicates that the users in Cluster 1 may be more willing to go a further distance on their shopping trips. Cluster W2 has a visible frequency at Healthcare at 0-2km, something which is not seen in the other two clusters. The users in Cluster W2 may visit Healthcare locations more frequently, and it makes sense that they would primarily visit Healthcare locations that are nearer to either their Home or Work locations. Cluster W3 has a visible frequency at the Park and Recreational POIs within the 2-8km threshold, which is not observed in the Cluster W1 and Cluster W2. This may indicate that users in Cluster W3 make trips to areas related to leisure more frequently than the users in the other clusters.

Comparing the two parts of Fig. 9, we can see that although there are some distance and label combinations that have more users in each cluster that visit them, it does not necessarily mean that they visit them frequently. The label/distance combinations that are visited frequently are a subset of those that are visited commonly by users.

VI. OFFDAY CLUSTERING AND ANALYSIS

This section describes the results obtained from clustering the Offday data of all users. The process is the same as the one used for the Workday data in Section V. The three clusters here are labeled O1, O2, and O3, with 'O' standing for 'Offday'.

A. CLUSTERING RESULTS—CENTROID VALUES

The values of each cluster's centroids are plotted in Fig. 10. We can observe that these clusters show similar trends to the Workday clusters in that there are those that stay mostly at Home Only (O1), those that make mostly short trips (O2), and those that make mostly longer trips (O3). The users in Cluster O1 spent 71% of their Offdays only at their Home location. Cluster O2 users spent on average 21% of their days at their Home location, and 58% of their days have a DCD value of 0-5km. The average percentages for Cluster O3 are more evenly split between the Home Only and the first two distance categories, with the highest being 39% of days with DCD values of 5-15km.

When observed together with each cluster's corresponding OD matrix, we see that Cluster O2 actually has the highest average percentage of trips within 0-1km at 46% compared to 32% for Cluster O1. Additionally, we see that the percentages of trips going between the 0-1km threshold and further thresholds is actually higher in Cluster O1 than Cluster O2. A possible reason for this could be that although the users in O1 stay at home only for more days than those in O2, they tend to travel further when they do go out, whereas those in O2 could go out on more days but stay within 0-1km for most of their trips. The users in cluster O3 seem to have more of a balance between staying at home and going out to near or further places.

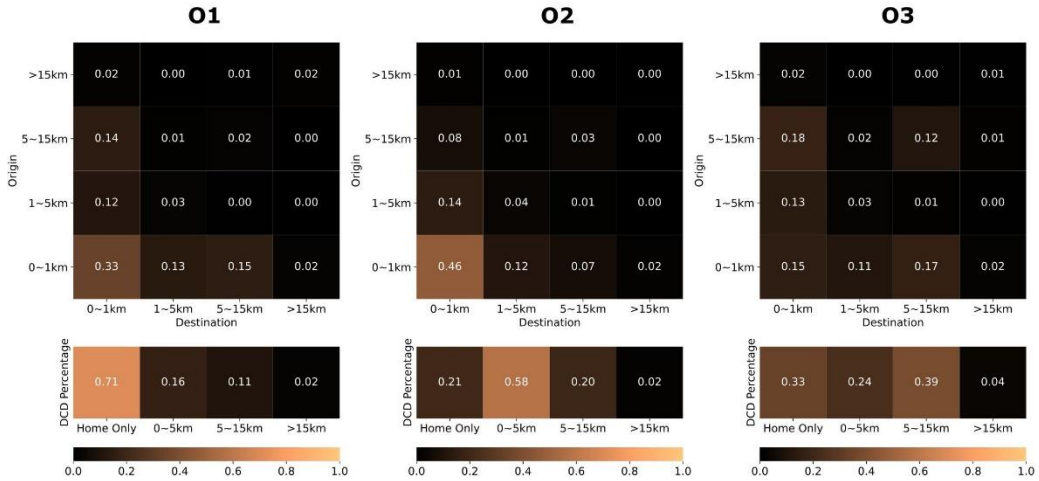


FIGURE 10. Centroid values of the three clusters obtained from clustering Offday data. Cluster (a) has the highest percentage of days spent at Home only, while cluster (b) has the highest percentage of days with DCD between 0 to 5 km, meaning they went to at least one other non-Home location. Cluster (c) has the highest percentage of days with DCD in the 5 to 15km range, indicating that they generally travel the furthest on Offdays.

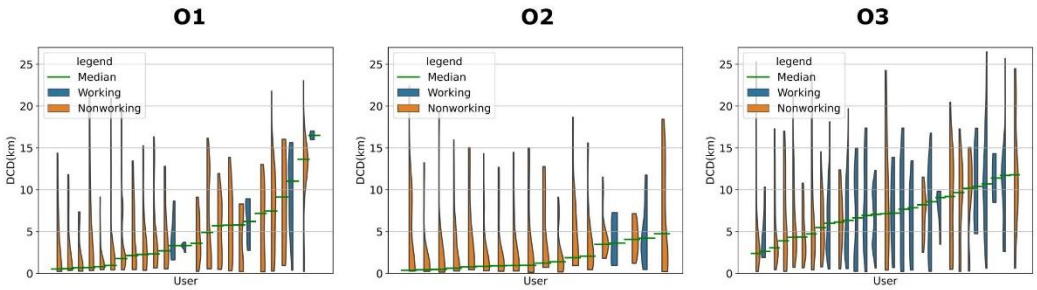


FIGURE 11. Violinplots illustrating each user's DCD distribution within each cluster on Offdays.

B. CLUSTER ANALYSIS—DCD VIOLINPLOTS

The violinplots representing the DCD distribution of each user within each cluster have been plotted in Fig. 11. Similarly to before, the Home Only days are not reflected on this plot as we are more interested on days in which the users do go out.

Cluster O1 and Cluster O2 both contain dominantly Nonworking users, while the bulk of the Working users are in Cluster O3. Qualitatively speaking, Cluster O1 seems to lie in the middle of Clusters O2 and O3. The median DCDs of the users in Cluster O2 are limited to the 0-5km range, which agrees with the DCD features observed in Fig. 10 and further emphasizes that this group of users makes mostly short trips. Although the median values of Cluster O3 are not always higher than those in O2, the bulk of the DCD distributions for Cluster O3 lies above 5km, which is the distance threshold for longer trips in this case.

C. CLUSTER ANALYSIS—USER COMMONALITY AND AVERAGE FREQUENCY

User Commonality and Average Frequency of each cluster is obtained as described earlier in Section V-C, and plotted in Fig. 12. From Fig 12(a), we observe that there are the same three main POI labels that are commonly visited by users, namely Neighborhood Center, Shopping Mall, and Residential. There is a higher percentage of users in Cluster O3 who visit Parks and Recreational areas between 1-15km, as well as visiting Attractions that are in the 5-15km range from their homes. This may imply that users who have a higher median DCD tend to visit a variety of locations on Offdays, which are also further from their Home locations.

Looking at Fig. 12(b), we see that for all three clusters, the three highest frequency labels are the same as the highest commonality labels. However, for Shopping Malls, the highest frequency distance threshold differs for each cluster.

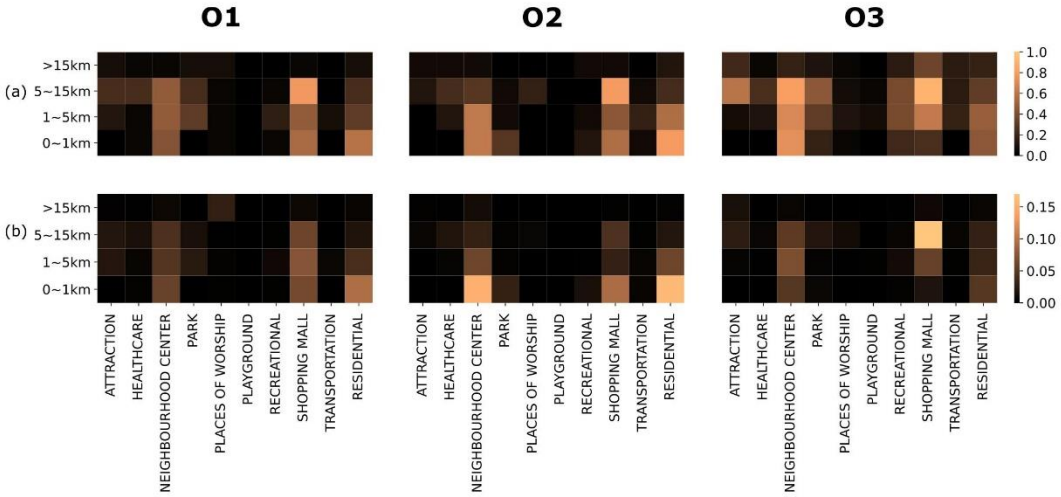


FIGURE 12. Heatmaps for each of the three Offday clusters showing (a) User Commonality and (b) Average Frequency. The colormap scales for (b) are narrowed to 0.17 to better show the contrast between the different squares.

For Cluster O1, the frequency is higher for Shopping Malls in the 1-5km range. For Cluster O2, the frequency is concentrated at the 0-1km range for Shopping Malls and similarly for Neighborhood Center and Residential areas. We can infer that the frequent short trips for this cluster are mainly for the purpose of visiting locations with those three labels. For Cluster O3, the frequency is prominently concentrated at Shopping Malls in the 5-15km range, and the frequency for Residential areas is much lower than for the other two clusters. This implies that shopping malls are a common destination further away from their home and work for these users in Cluster O3.

VII. CONCLUSION

In this paper, we investigated the differences between the GPS trajectory patterns of Workday and Offday data, as well as Working and Nonworking users. To do so, we proposed a new mobility metric based on radius of gyration, named Daily Characteristic Distance (DCD), to zoom in on the locations outside of Home and Work if applicable that the users visited. We discover that Working users' median DCD on Workdays is highly correlated to the distance between their Home and Work locations, and that Working users generally have a higher median DCD on Offdays as compared to Nonworking users.

We then used features derived from DCD in conjunction with those derived from the users' Origin-Destination matrices to cluster the users in our dataset. We find that we can group users' mobility into three types for both Workdays and Offdays. The three types are mainly those that mainly stick to Home (and Work if applicable), those that make frequent short trips, and those that make longer trips. We also propose two new types of metric for cluster analysis, namely User Commonality and Average Frequency, to better assess the labels and distances of different locations that are favored by the users in different clusters. We discover that three main POI labels are favored regardless of cluster - Neighborhood Centers, Shopping Malls, and Residential areas, but the main differences between clusters are the distance thresholds of these POI labels, as well as the presence or absence of some other labels such as Attraction, Parks, and Recreational areas. Urban planners could use this framework on their own target datasets as a case study to discover the types of places that would be beneficial to locate nearby their intended residential environment. It is important to note that, while our proposed framework is general, the results that we have obtained are dependent on our data that we have gathered in Singapore, and thus results may differ widely if our framework is used on data from other countries.

Currently, our work examines the users' data and clusters them separately for Workdays and Offdays. There could be more insights to be drawn from linking both the Workday and Offday mobility features of the individual Working users together and examining the resulting features for new correlations. This could be a part of future work.

ACKNOWLEDGMENT

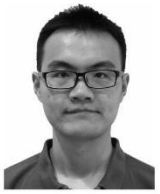
Any opinion, findings, and conclusion or recommendations expressed in this material are those of the author(s) and do not reflect the views of the Singapore Ministry of National Development and National Research Foundation, Prime Minister's Office, Singapore.

REFERENCES

[1] S. Van der Spek, J. Van Schaick, P. De Bois, and R. De Haan, "Sensing human activity: GPS tracking," *Sensors*, vol. 9, no. 4, pp. 3033–3055, Apr. 2009.

[2] N. Ta, Y. Zhao, and Y. Chai, "Built environment, peak hours and route choice efficiency: An investigation of commuting efficiency using GPS data," *J. Transp. Geogr.*, vol. 57, pp. 161–170, Dec. 2016.
[3] M. Hast, K. M. Searle, M. Chaponda, J. Lupiya, J. Lubinda, J. Sikalima, T. Kobayashi, T. Shields, M. Mulenga, J. Lessler, and W. J. Moss, "The use of GPS data loggers to describe the impact of spatio–temporal movement patterns on malaria control in a high-transmission area of Northern Zambia," *Int. J. Health Geograph.*, vol. 18, no. 1, pp. 1–18, Dec. 2019.
[4] A. Solomon, A. Bar, C. Yanai, B. Shapira, and L. Rokach, "Predict demographic information using Word2vec on spatial trajectories," in *Proc. 26th Conf. User Modeling, Adaptation Personalization*, Jul. 2018, pp. 331–339.
[5] L. Wu, L. Yang, Z. Huang, Y. Wang, Y. Chai, X. Peng, and Y. Liu, "Inferring demographics from human trajectories and geographical context," *Comput., Environ. Urban Syst.*, vol. 77, Sep. 2019, Art. no. 101368.
[6] K. Siła-Nowicka, J. Vandrol, T. Oshan, J. A. Long, U. Demšar, and A. S. Fotheringham, "Analysis of human mobility patterns from GPS trajectories and contextual information," *Int. J. Geograph. Inf. Sci.*, vol. 30, no. 5, pp. 881–906, May 2016.
[7] J. Long and D. Reuschke, "Daily mobility patterns of small business owners and homeworkers in post-industrial cities," *Comput., Environ. Urban Syst.*, vol. 85, Jan. 2021, Art. no. 101564.
[8] N. Brouwers and M. Woehrle, "Dwelling in the canyons: Dwelling detection in urban environments using GPS, Wi-Fi, and geolocation," *Pervas. Mobile Comput.*, vol. 9, no. 5, pp. 665–680, Oct. 2013.
[9] S. H. Marakkalage, S. Sarica, B. P. L. Lau, S. K. Viswanath, T. Balasubramaniam, C. Yuen, B. Yuen, J. Luo, and R. Nayak, "Understanding the lifestyle of older population: Mobile crowdsensing approach," *IEEE Trans. Computat. Social Syst.*, vol. 6, no. 1, pp. 82–95, Feb. 2019.
[10] L. Zhu, J. Gonder, and L. Lin, "Prediction of individual social-demographic role based on travel behavior variability using long-term GPS data," *J. Adv. Transp.*, vol. 2017, pp. 1–13, Jan. 2017.
[11] B.-H. Nahmias-Biran, Y. Han, S. Bekhor, F. Zhao, C. Zegras, and M. Ben-Akiva, "Enriching activity-based models using smartphone-based travel surveys," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2672, no. 42, pp. 280–291, Dec. 2018.
[12] S. H. Marakkalage, B. P. L. Lau, Y. Zhou, R. Liu, C. Yuen, W. Q. Yow, and K. H. Chong, "WiFi fingerprint clustering for urban mobility analysis," *IEEE Access*, vol. 9, pp. 69527–69538, 2021.
[13] L. Alessandretti, P. Sapiezynski, S. Lehmann, and A. Baronchelli, "Multi-scale spatio–temporal analysis of human mobility," *PLoS ONE*, vol. 12, no. 2, Feb. 2017, Art. no. e0171686.
[14] G. Solmaz and D. Turgut, "Modeling pedestrian mobility in disaster areas," *Pervas. Mobile Comput.*, vol. 40, pp. 104–122, Sep. 2017.
[15] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma, "Understanding mobility based on GPS data," in *Proc. 10th Int. Conf. Ubiquitous Comput.*, 2008, pp. 312–321.
[16] Y. Wang, K. Qin, Y. Chen, and P. Zhao, "Detecting anomalous trajectories and behavior patterns using hierarchical clustering from taxi GPS data," *ISPRS Int. J. Geo-Inf.*, vol. 7, no. 1, p. 25, 2018.
[17] E. Cesario, C. Comito, and D. Talia, "An approach for the discovery and validation of urban mobility patterns," *Pervas. Mobile Comput.*, vol. 42, pp. 77–92, Dec. 2017.
[18] D. Kumar, H. Wu, S. Rajasegarar, C. Leckie, S. Krishnaswamy, and M. Palaniswami, "Fast and scalable big data trajectory clustering for understanding urban mobility," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 11, pp. 3709–3722, Nov. 2018.
[19] S. Dodge, P. Laube, and R. Weibel, "Movement similarity assessment using symbolic representation of trajectories," *Int. J. Geograph. Inf. Sci.*, vol. 26, no. 9, pp. 1563–1588, 2012.
[20] J. Tang, F. Liu, Y. Wang, and H. Wang, "Uncovering urban human mobility from large scale taxi GPS data," *Phys. A, Stat. Mech. Appl.*, vol. 438, pp. 140–153, Nov. 2015.
[21] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, vol. 96, 1996, pp. 226–231.
[22] L. Amichi, A. C. Viana, M. Crovella, and A. F. Loureiro, "Mobility profiling: Identifying scouters in the crowd," in *Proc. 15th Int. Conf. Emerg. Netw. Exp. Technol.*, Dec. 2019, pp. 9–11.
[23] D. A. Reynolds, "Gaussian mixture models," *Encyclopedia Biometrics*, vol. 741, pp. 1–5, Jul. 2009.
[24] L. Scherrer, M. Tomko, P. Ranacher, and R. Weibel, "Travelers or locals? Identifying meaningful sub-populations from human movement data in the absence of ground truth," *EPJ Data Sci.*, vol. 7, no. 1, p. 19, Dec. 2018.
[25] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.
[26] G. Solmaz and D. Turgut, "A survey of human mobility models," *IEEE Access*, vol. 7, pp. 125711–125731, 2019.
[27] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, p. 779, 2008.
[28] L. Pappalardo, F. Simini, S. Rinzivillo, D. Pedreschi, F. Giannotti, and A.-L. Barabási, "Returners and explorers dichotomy in human mobility," *Nature Commun.*, vol. 6, no. 1, pp. 1–8, Nov. 2015.
[29] E. Pepe, P. Bajardi, L. Gauvin, F. Privitera, B. Lake, C. Cattuto, and M. Tizzoni, "COVID-19 outbreak response, a dataset to assess mobility changes in Italy following national lockdown," *Sci. Data*, vol. 7, no. 1, pp. 1–7, 2020.
[30] Y. Xu, A. Belyi, I. Bojic, and C. Ratti, "Human mobility and socioeconomic status: Analysis of Singapore and Boston," *Comput., Environ. Urban Syst.*, vol. 72, pp. 51–67, Nov. 2018.
[31] Y. Zhou, B. P. L. Lau, Z. Koh, C. Yuen, and B. K. K. Ng, "Understanding crowd behaviors in a social event by passive WiFi sensing and data mining," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4442–4454, May 2020.
[32] Z. Koh, Y. Zhou, B. P. L. Lau, C. Yuen, B. Tuncer, and K. H. Chong, "Multiple-perspective clustering of passive Wi-Fi sensing trajectory data," *IEEE Trans. Big Data*, vol. 8, no. 5, pp. 1312–1325, Oct. 2022.
[33] X. Lu, E. Wetter, N. Bharti, A. J. Tatem, and L. Bengtsson, "Approaching the limit of predictability in human mobility," *Sci. Rep.*, vol. 3, no. 1, pp. 1–9, Dec. 2013.
[34] B. P. L. Lau, M. S. Hasala, V. S. Kadaba, B. Thirunavukarasu, C. Yuen, B. Yuen, and R. Nayak, "Extracting point of interest and classifying environment for low sampling crowd sensing smartphone sensor data," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2017, pp. 201–206.
[35] Urban Redevelopment Authority. (2019). *Master Plan 2019 Subzone Boundary (No Sea)*. Accessed: Aug. 26, 2020. [Online]. Available: https://data.gov.sg/dataset/master-plan-2019-subzone-boundary-no-sea

ZANN KOH received the B.Eng. degree in engineering and product development from the Singapore University of Technology and Design, Singapore, in 2017, where she is currently pursuing the Ph.D. degree with the Singapore University of Technology and Design, under the supervision of Chau Yuen. Her current research interests include big data analysis, data discovery, urban human mobility, and unsupervised machine learning.



YUREN ZHOU (Member, IEEE) received the B.Eng. degree in electrical engineering from Harbin Institute of Technology, Harbin, China, in 2014, and the Ph.D. degree from the Singapore University of Technology and Design, Singapore, in 2019, with a focus on data mining and smart city applications. He is currently a Postdoctoral Research Fellow with the Singapore University of Technology and Design. His current research interests include big data analytics and its application in urban human mobility, building energy management, and the Internet of Things.

BILLY PIK LIK LAU (Member, IEEE) received the degree in computer science and the M.Phil. degree in computer science from Curtin University, Perth, WA, Australia, in 2010 and 2014, respectively. He is currently pursuing the Ph.D. degree with the Singapore University of Technology and Design, Singapore, with Dr. Chau Yuen. He studied the cooperation rate between agents in multiagents systems during master's studies. His current research interests include urban science, big data analysis, data knowledge discovery, the Internet of Things, and unsupervised machine learning.

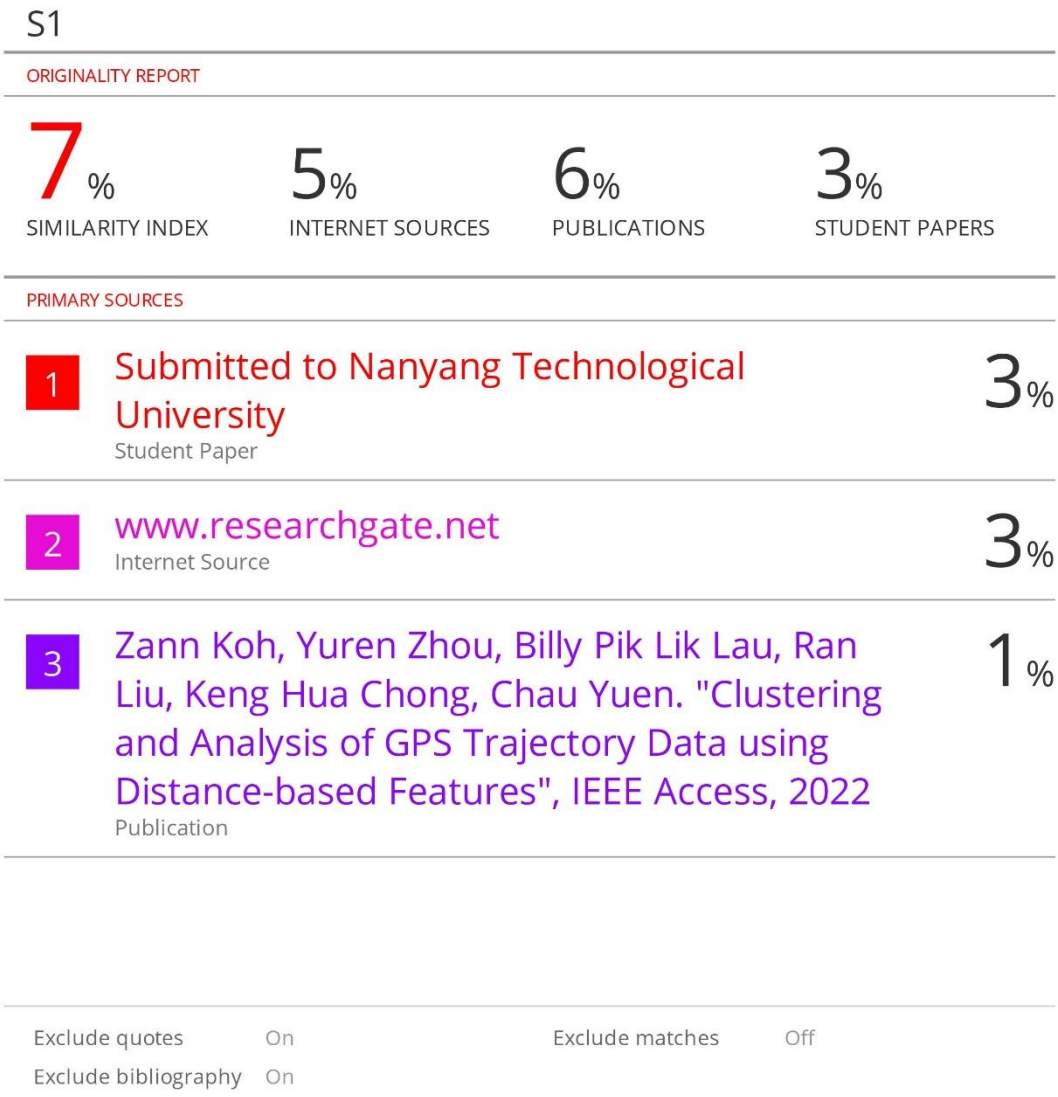
RAN LIU (Senior Member, IEEE) received the B.S. degree from the Southwest University of Science and Technology, China, in 2007, and the Ph.D. degree from the University of Tuebingen, Germany, in 2014, under the supervision of Prof. Andreas Zell. Since 2014, he has been a Research Fellow under the supervision of Prof. Chau Yuen with the MIT International Design Center, Singapore University of Technology and Design, Singapore. His research interests include robotics, indoor positioning, and SLAM.

KENG HUA CHONG is an Associate Professor of architecture and sustainable design with the Singapore University of Technology and Design (SUTD), where he directs the Social Urban Research Groupe (SURGe) and co-leads the Opportunity Laboratory (O-Lab). He has several key publications and projects, including *Creative Ageing Cities*, *Second Beginnings*, and the New Urban Kampung Research Programme. His research interests include social architecture, ageing population, liveable place, and data-driven collaborative design.

CHAU YUEN (Fellow, IEEE) received the B.Eng. and Ph.D. degrees from Nanyang Technological University, Singapore, in 2000 and 2004, respectively. He is currently an Associate Professor with the Singapore University of Technology and Design. He was a Postdoctoral Fellow at Lucent Technologies Bell Labs, Murray Hill, NJ, USA, in 2005. He was a Visiting Assistant Professor at The Hong Kong Polytechnic University, in 2008. From 2006 to 2010, he was a Senior Research Engineer at the Institute for Infocomm Research (I2R, Singapore), where he was involved in an industrial project on developing an 802.11n Wireless LAN systems, and participated actively in 3Gpp Long Term Evolution (LTE) and LTE-Advanced (LTE-A) Standardization. He has been with the Singapore University of Technology and Design, since 2010. He was a recipient of the Lee Kuan Yew Gold Medal, the Institution of Electrical Engineers Book Prize, the Institute of Engineering of Singapore Gold Medal, the Merck Sharp and Dohme Gold Medal, and twice the recipient of the Hewlett Packard Prize. He received the IEEE Asia-Pacific Outstanding Young Researcher Award, in 2012. He is an Editor of the IEEE Transaction on Communications and the IEEE Transactions on Vehicular Technology and was awarded the Top Associate Editor, from 2009 to 2015.

APPENDIX – B

PLAGARISM REPORT



S1

ORIGINALITY REPORT

7%	5%	6%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Nanyang Technological University Student Paper	3%
2	www.researchgate.net Internet Source	3%
3	Zann Koh, Yuren Zhou, Billy Pik Lik Lau, Ran Liu, Keng Hua Chong, Chau Yuen. "Clustering and Analysis of GPS Trajectory Data using Distance-based Features", IEEE Access, 2022 Publication	1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		