Taxi Demand Prediction using an LSTM-based Deep Sequence Model and Points of Interest

Bahman Askari, Tai Le Quy, Eirini Ntoutsi
Leibniz University Hannover & L3S Research Center, LUH, Hannover

Hannover, 13.07.2020

Content

- Introduction
- Methodology
- Dataset
- Experiments
- Results
- Conclusion and outlook

Introduction

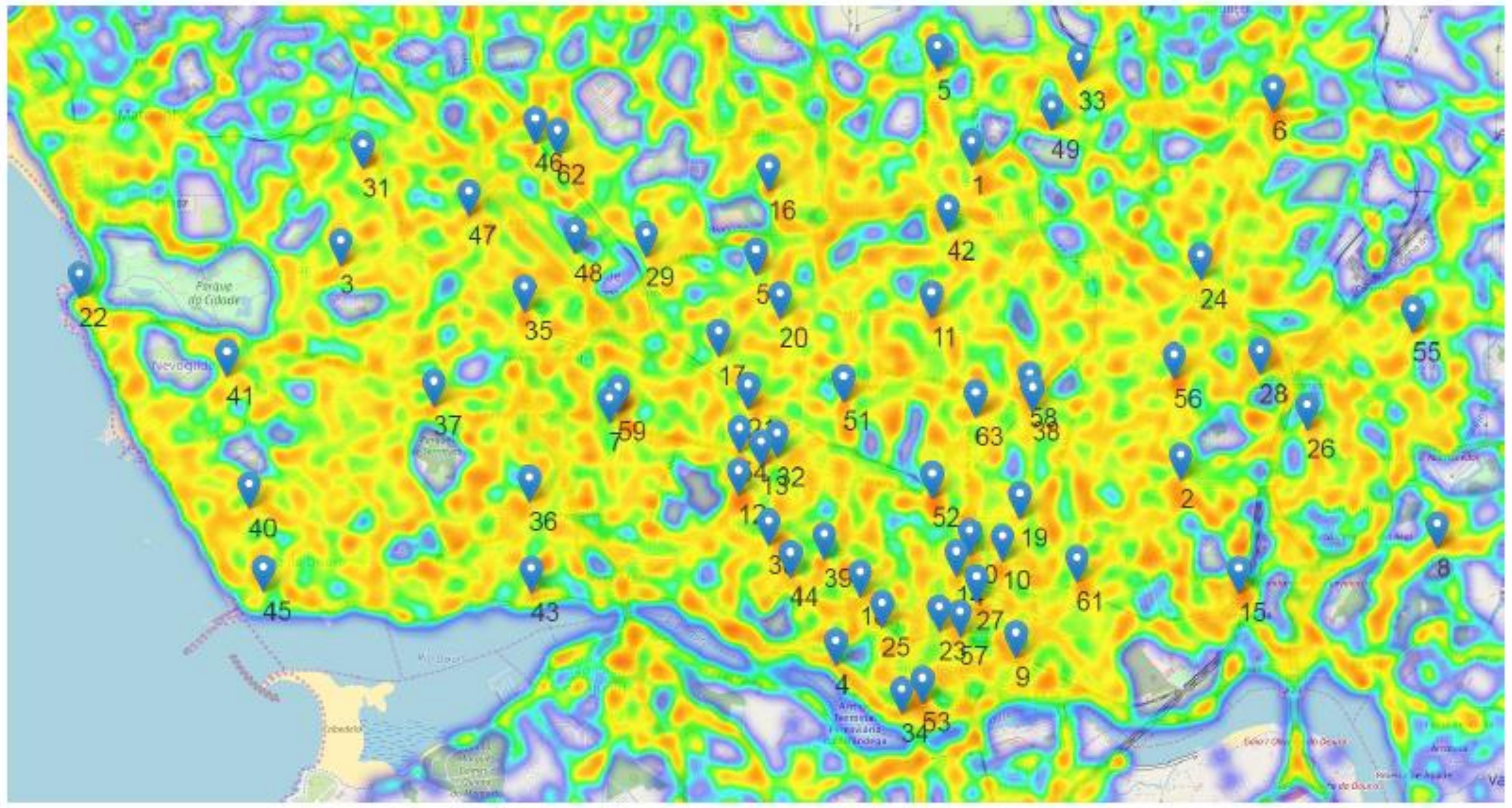
- Administration
  - Traffic control
  - Efficient transportation system
  - Avoid unnecessary energy consumption
- Taxi companies
  - Improve the levels of passenger satisfaction and maximal profit
  - Balance the relationship between the passenger demand and the number of running taxi vehicles
- Taxi-Passenger Demand Prediction
  - It’s useful for drivers in making decision moving to pick up passengers in a particular region in the city
  - Spatial information is useful for prediction task

Introduction

- Motivation
  - First Law of Geography:

    Everything is related to everything else, but near things are more related than distant things.
  - Point Of Interest (POI):

    Popular places such as bar, restaurant, hospital, etc.



Spatial distribution of the taxi-stands
Numbers 1- 63 indicate the IDs of the stands

- Goal
  - Develop LSTM model to predict the taxi-passenger demand using POI

Problem denition

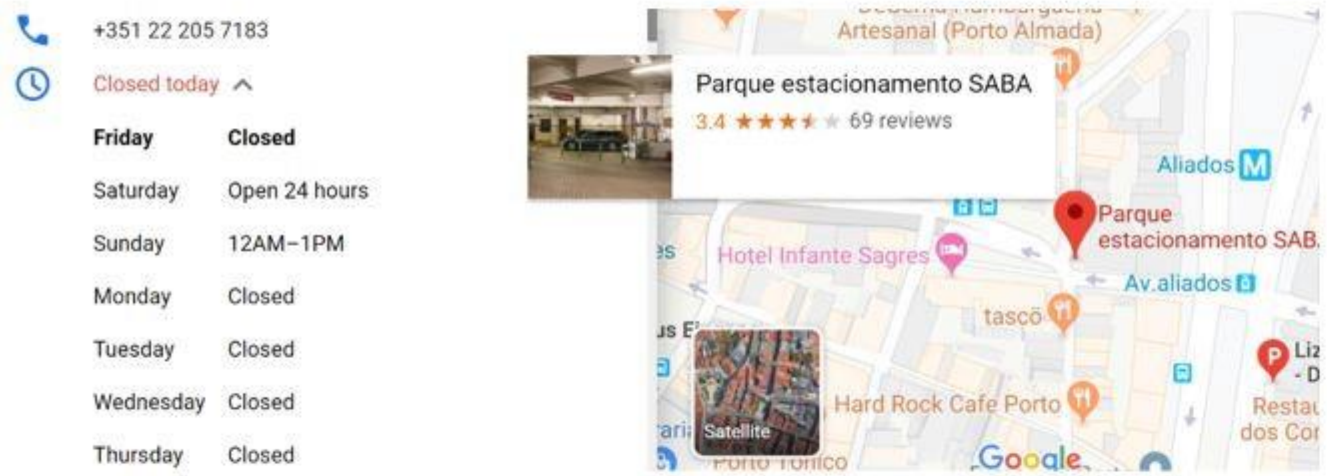
- Let $S = \{s_1; s_2; ..; s_N\}$ be the set of predened N taxi-stands in a city
- $X_s = \{X_{s;0}; X_{s;1}; ..; X_{s;t}\}$ to be a discrete time series modeling the taxi-demand for stand s
  - based on an aggregation period of P-minutes
- Our goal is to build a model which predicts the demand $X_{s;t+1}$ for the next time point $t + 1$ at taxi-stand s.

Methodology

❑ Point of Interest (POI)

Point of Interest around taxi-stands

(a) Popular time is available for this POI



(b) Popular times is not available for this POI

Popular times for POIs (Google Maps)

Methodology

- Finding the POIs

Algorithm 1: Finding POI in Porto

```
Input: Taxi-Stands Data TS , Business-Types BT
Output: POI-DATASET
for each stand i in TS do
    Radius[i] ← min(distance(i, j for each stand j in
    TS and i!=j))
    Coordinates[i] ← coordinate(stand i)
for each b in BT do
    for each t in TS do
        p ← search_places_by_coordinate
        (Coordinates[t], Radius[t], BT[b])
        POI ← populartimes.get_id(API-KEY, P)
    Add POI to POI-DATASET
Clean POI-DATASET //remove duplicate records
```

Methodology

- Convert POIs to time series
  - Let $POI_s = \{P_{s1}, P_{s2},\ldots, P_{sM}\}$ is the set of M point of interest of taxi-stand s, and $P_s = \{p_{s,0}, p_{s,1},\ldots, p_{s,t}\}$ is the time series for POIs of taxi-stand s.

Algorithm 2: Converting POI-Dataset to time series

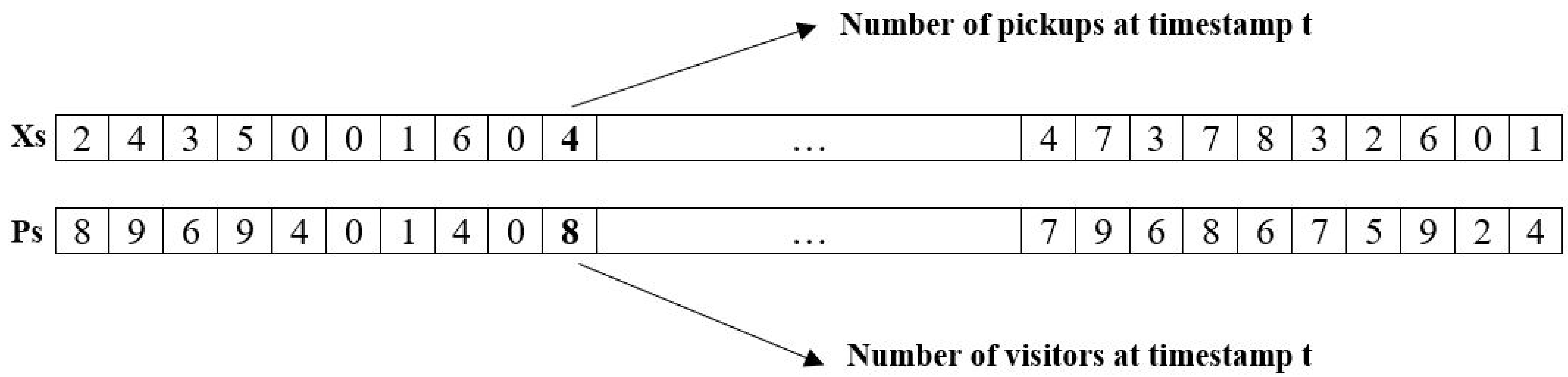
Input: POI-DATASET **PD**
Output: POI Time series **Ps(Len (TS), L)**
// **TS**: Taxi-Stands , **L**: Length=17520
for *each poi in PD* **do**
    visitlist ← convert POI to time series
    s ← index of closest taxi stand to the POI
    aggregate (Ps(s,:), visitlist)
return Ps

Methodology

❑ POI-LSTM Deep Sequence Model

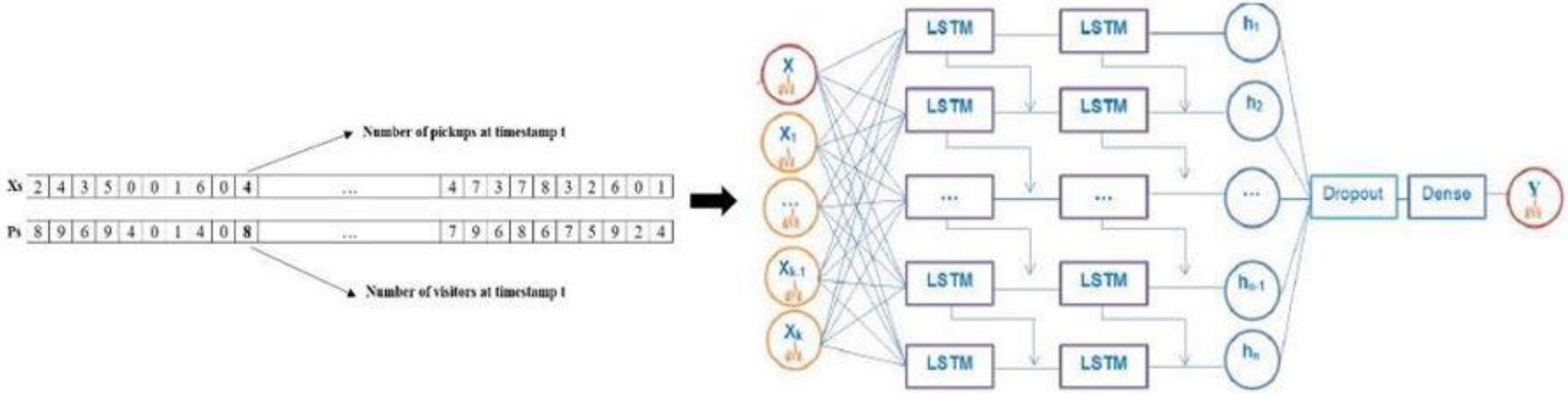
Algorithm 3: POI-LSTM Deep Sequence Model

Input: Taxi-Stands **TS, Xs, Ps**
Output: POI-LSTM Prediction Model
for *each stand s in TS* **do**
    XPs← $Concatenate(Xs, Ps)$
    Train_s, Test_s← $Split(XPs)$
    Build and train LSTM with Train_s
    Execute prediction model with Test_s
    Evaluate the POI-LSTM model for stand s
return average(the evaluation metrics of all stands)



Methodology

- POI-LSTM architecture



Layers	Neurons	Optimization Function	Activation	Look Back	Epoch	Batch	Overfitting
1	200	AdaMax	tanh	5	25	100	0.7

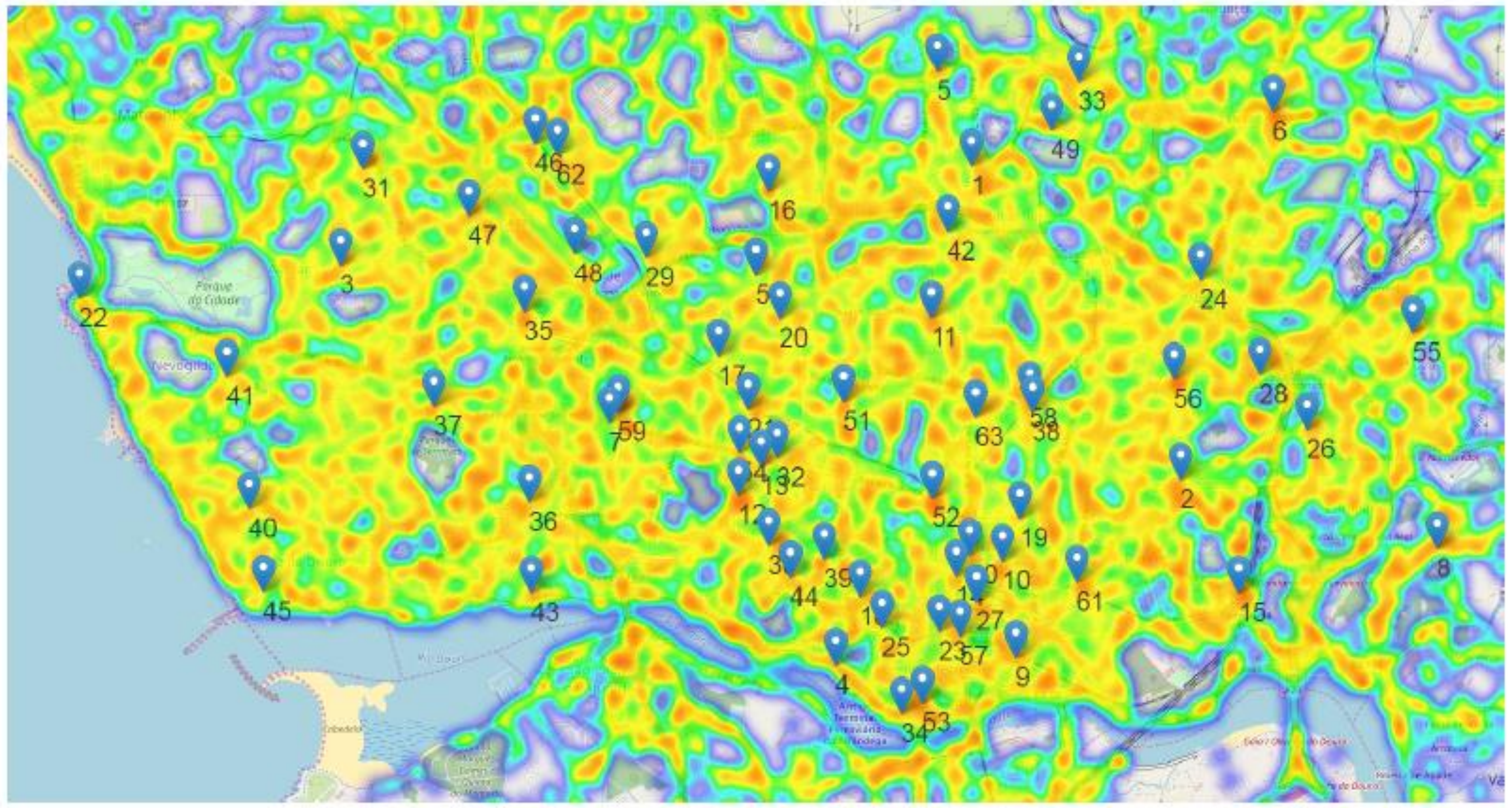
The architecture of the POI- LSTM.

Dataset

- Porto city in Portugal
  - Period: July 2013 to June 2014
  - Records: 1.710.670
  - 9 features
  - 63 taxi-stands
- Two versions of dataset for experiment
  - D1: all trips departing from taxi-stands (817.861 instances)
  - D2: all trips (1.706.572 instances).
    - Assign trips (do not start from a taxi-stand) to their closest taxi-stand based on distance.

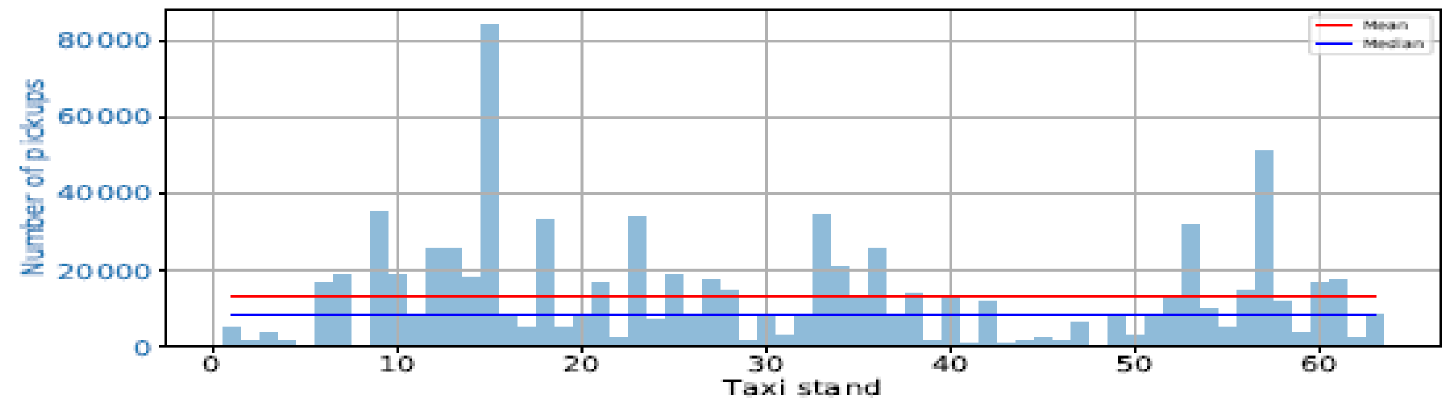
Dataset Characteristics

- Spatial distribution

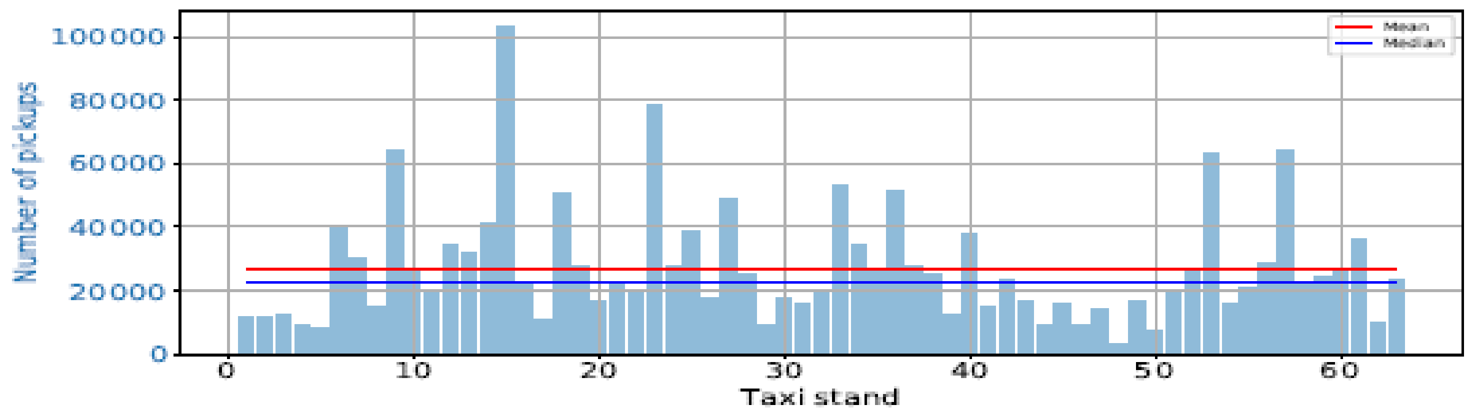


Spatial distribution of the taxi-stands
Numbers 1- 63 indicate the IDs of the stands

Pickup distribution

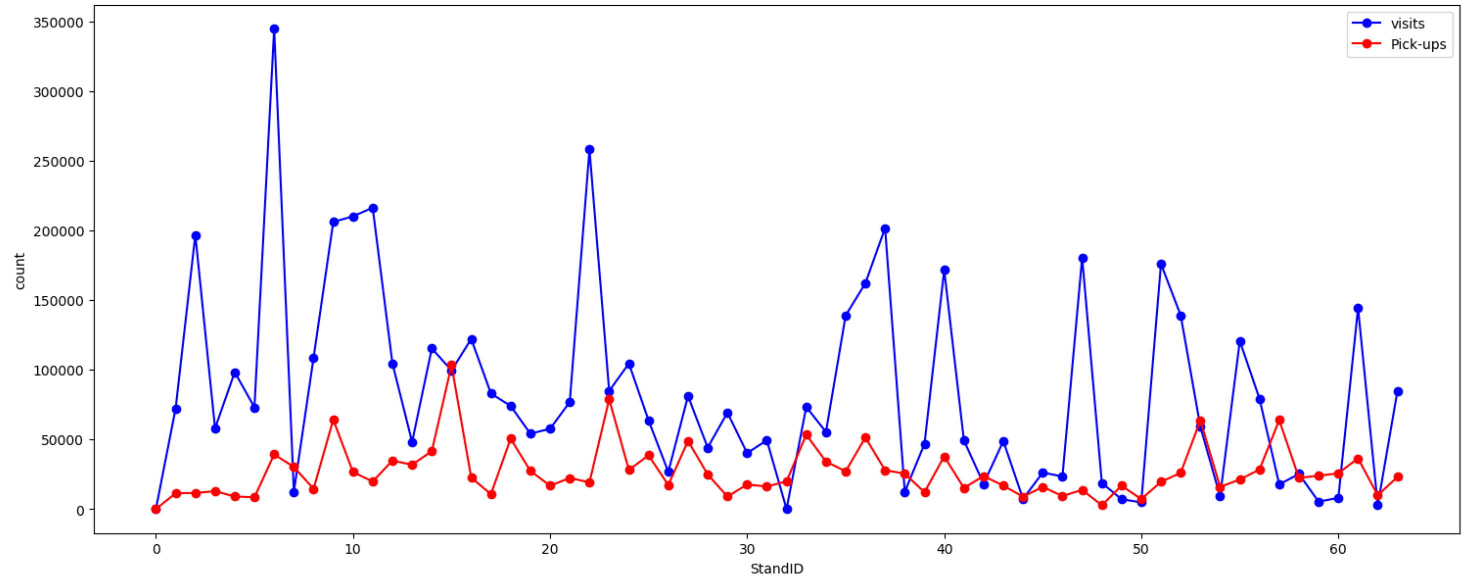


Pickup distribution per taxi-stand on D1

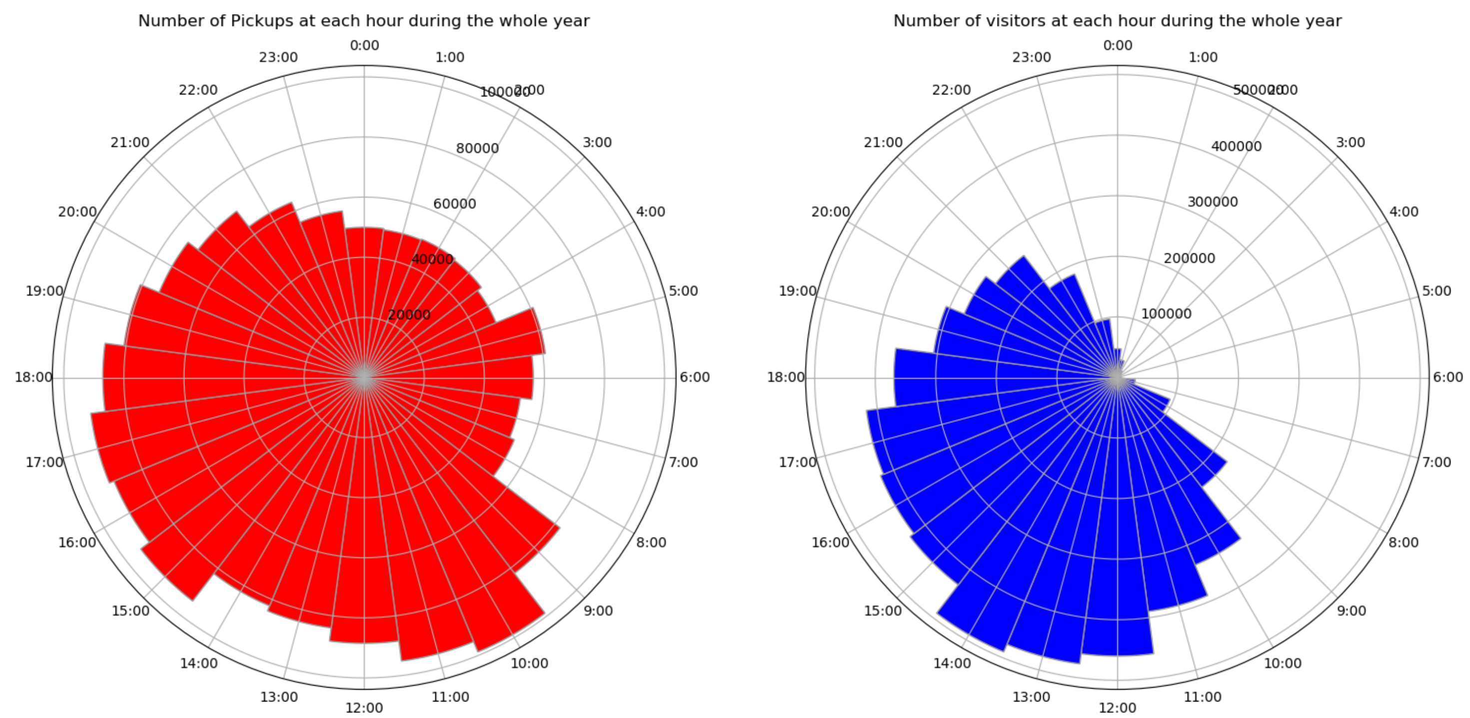


Pickup distribution per taxi-stand on D2

Dataset Characteristics



Number of visitors and pickups at each taxi-stand



Number of Pickups and POIs over 24 hours

Experiment

- Experimental setup
  - Set aggregation period at 30 minutes
  - 70% data for training, 30% data for testing (by the time series)
- Baselines
  - Linear Regression
  - Random Forest Regression
  - XGBoost Regression
  - LSTM
  - Neighborhood-LSTM
- Evaluation measure
  - symmetric Mean Absolute Percentage Error (sMAPE)

$$SMAPE_s = \frac{100\%}{T} \sum_{i=1}^{T} \frac{\left|Y_{s,i} - \hat{Y}_{s,i}\right|}{\left|Y_{s,i}\right| + \left|\hat{Y}_{s,i}\right| + c}$$

in which, Y_s and $\hat{Y}_s$ are the true and predicted demand, c=1

  - Mean Squared Error (MSE)

$$\mathrm{MSE}_s = \frac{1}{T} \sum_{i=1}^{T} \left(Y_{s,i} - \hat{Y}_{s,i}\right)^2$$

  - Overall

$$MSE = \frac{\sum_{i=1}^{N} MSE_i}{N}$$

$$SMAPE = \frac{\sum_{i=1}^{N} SMAPE_i}{N}$$

Results

- Taxi-demand prediction quality results

MODEL	MSE		SMAPE	
Timestamp: 30 minutes	Train	Test	Train	Test
Linear Regression	1.61	1.76	24.37	24.52
Random Forest Regression	*0.38*	1.66	*16.83*	24.25
XGBoost Regression	1.39	1.59	23.90	23.91
LSTM	1.66	1.84	18.37	18.54
Neighborhood-augmented LSTM	1.49	1.68	17.32	17.64
POI-LSTM	**1.26**	**1.41**	**17.12**	**17.25**

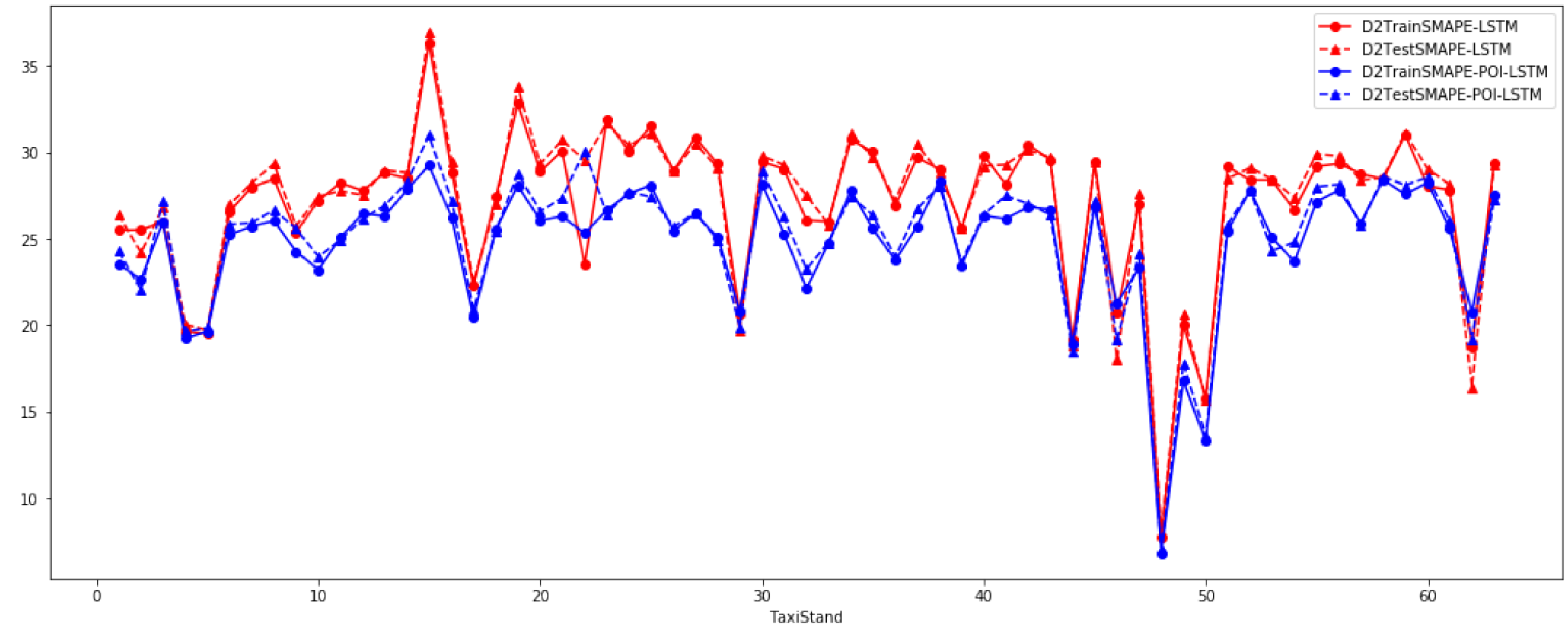
Prediction quality of the different models on D1

MODEL	MSE		SMAPE	
Timestamp: 30 minutes	Train	Test	Train	Test
Linear Regression	4.21	5.988	30.78	31.23
Random Forest Regression	*0.71*	5.50	*18.49*	31.03
XGBoost Regression	3.60	5.45	30.47	30.51
LSTM	4.16	6.66	27.03	27.22
Neighborhood-augmented LSTM	3.84	6.44	25.88	26.07
POI-LSTM	**2.57**	**5.27**	**24.73**	**25.08**

Prediction quality of the different models on D2

Results

- Performance of models on taxi-stand



sMAPE error rate over D2 dataset

Conclusion and outlook

- We propose a LSTM model
  - Consider Point of interest for prediction
  - The proposed model achieved better results comparing the traditional models
  - Deep sequence learning methods are suitable for this kind of estimation task
- Future work
  - Learn locally per stand and re-tune globally the predictions in the city
  - Adding other features such as events and weather condition can result to better predictions for both training and unseen data

Thank you for your attention!
Questions?

baskari@l3s.de, tai@l3s.de