ODToolkit: A Toolkit for Building Occupancy Detection

Tianyu Zhang, Abdullah Al Zishan, and Omid Ardakanian

University of Alberta

ACM e-Energy

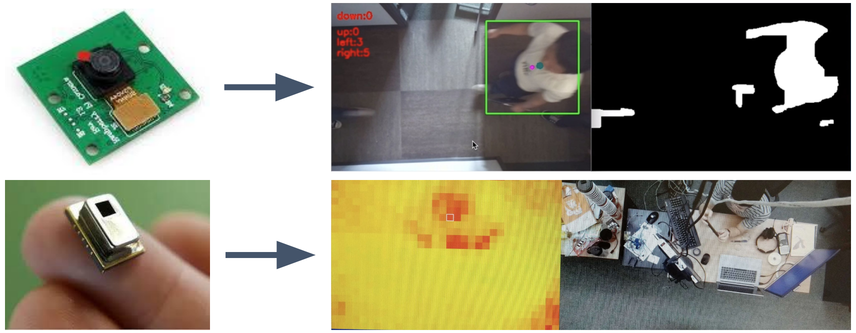
June 2019

Phoenix, United States

UNIVERSITY OF ALBERTA

Data-driven occupancy modelling

- Building occupancy detection is a well studied topic
  - using different sensing modalities
  - in office buildings, homes, schools, etc.
  - with various objectives and evaluation criteria



- Fine-grained occupancy information is essential for
  - **Energy-efficient** control of HVAC and lighting systems (e.g., demand-driven air circulation)
  - Safety and security
  - Space utilization
  - Automatic fault detection

Factors hindering development of new algorithms

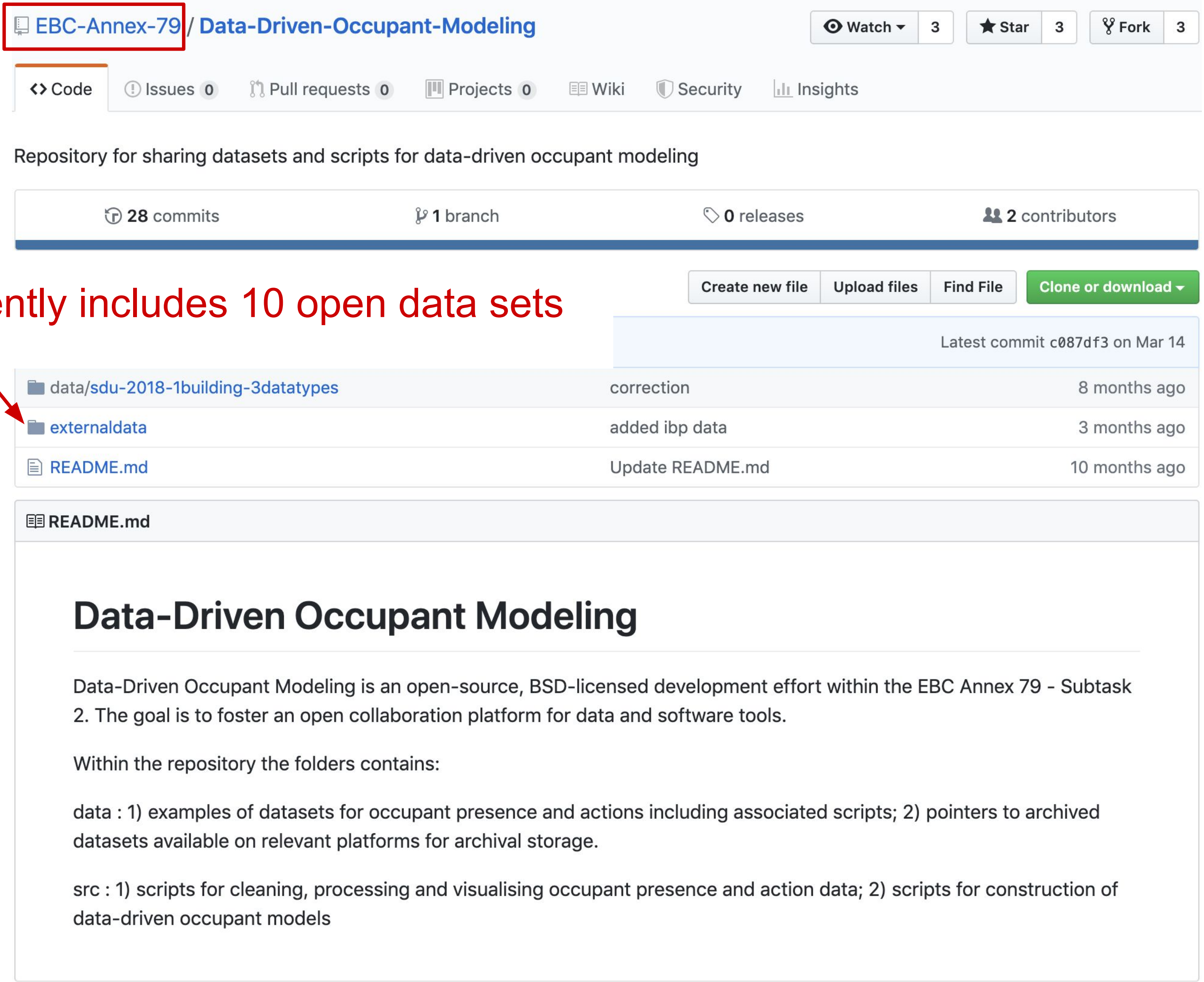
- lack of open source implementation of existing algorithms
- no standard data format
  - different modalities, time/space granularities, naming conventions, units, etc.
- lack of consensus on evaluation metrics
  - especially when it comes to comparing occupancy counting algorithms (±k people)

this increases the effort to prepare data, evaluate models, and make sense of the performance results

Recent efforts to build an open collaboration platform

this is led by Mikkel Baun Kjærgaard, Bing Dong, and Salvatore Carlucci

currently includes 10 open data sets


EBC-Annex-79 / Data-Driven-Occupant-Modeling

Watch 3 | Star 3 | Fork 3

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Security | Insights

Repository for sharing datasets and scripts for data-driven occupant modeling

28 commits | 1 branch | 0 releases | 2 contributors

Create new file | Upload files | Find File | Clone or download

Latest commit c087df3 on Mar 14

data/sdu-2018-1building-3datatypes	correction	8 months ago
externaldata	added ibp data	3 months ago
README.md	Update README.md	10 months ago

README.md

Data-Driven Occupant Modeling

Data-Driven Occupant Modeling is an open-source, BSD-licensed development effort within the EBC Annex 79 - Subtask 2. The goal is to foster an open collaboration platform for data and software tools.

Within the repository the folders contains:

data : 1) examples of datasets for occupant presence and actions including associated scripts; 2) pointers to archived datasets available on relevant platforms for archival storage.

src : 1) scripts for cleaning, processing and visualising occupant presence and action data; 2) scripts for construction of data-driven occupant models

The need for a toolkit

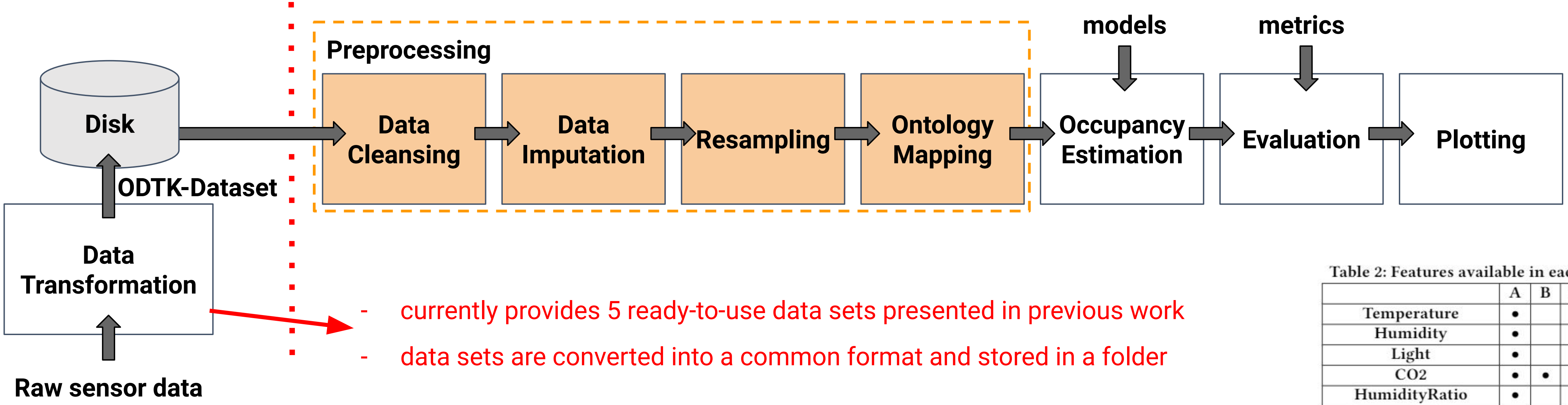
- Developed an open source toolkit for occupancy detection
- Similar to NILMTK for non-intrusive load monitoring [Batra'14]
- ODToolkit enables the comparison of data-driven occupancy detection algorithms in a reproducible manner across multiple buildings (possibly equipped with different sensors)

GitHub Documentation available on: https://odtoolkit.github.io/
Code available on: https://github.com/sustainable-computing/ODToolkit

Outline

- ODToolkit pipeline
    - Components of this toolkit
- Case studies
    - Does this toolkit facilitate the development of new algorithms for occupancy detection?
- Takeaways and future work

Pipeline



- currently provides 5 ready-to-use data sets presented in previous work
- data sets are converted into a common format and stored in a folder

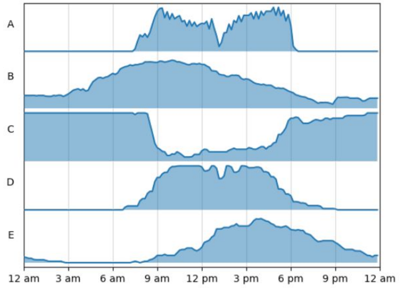


Table 1: Summary of 5 publicly available data sets imported and analyzed by ODToolkit

Date set	Granularity	Occupancy label	Collection method	Dropout rate	No. features	No. rooms	No. time slots	Pct. time occupied	Duration
A [18]	1 min	Binary	Camera	0%	6	3	20560	23.10%	~14.25 days
B [4]	1 min	Count	Camera	0%	3	4	97440	45.89%	~17 days
C [11]	1 min	Binary	App. (GPS)	0.14%	2	3	30240	72.28%	~7 days
D [35]	15 min	Binary	Manual	93.43%	10	24	35041	22.82%	~1 years
E [29]	10 sec	Count	Manual	0%	5	1	377549	23.99%	~43.66 days

Table 2: Features available in each data set

	A	B	C	D	E
Temperature	•			•	
Humidity	•			•	
Light	•			•	
CO2	•	•		•	
HumidityRatio	•				
DamperPosition		•			
LoadPower			•		
AirVelocity				•	
RadiantTemperature				•	
OutdoorTemperature				•	
OutdoorHumidity				•	
OutdoorAirVelocity				•	
VOC					•
Network					•
Bluetooth					•

Preprocessing

- Mark data points outside <u>**1.5 x IQR**</u> (interquartile range) as outliers
- Remove outliers
- Replace all NaN values (e.g., forward-filling algorithm, etc.)
- Change the sampling frequency (e.g., upsampling and downsampling)
- Convert the label of points to a standard name from the glossary (e.g., Temp. and RoomTemp will be replaced by IndoorTemperature)
    - Jaro distance is used to determine the similarity between names

Occupancy Estimation & Evaluation

- Implemented several baseline data-driven models
  - 7 supervised learning models are currently included
  - HMM, PF, SVM, RF, SNMF, ANN, LSTM

- Evaluation
  - 16 standard metrics available in the toolkit
  - 11 F-score metrics, RMSE, nRMSE, MAE, MAPE, MASE

- there is a built-in function to run all selected models on all selected data sets, evaluate them considering the selected metrics, plot the results

Case studies — extending the toolkit

- ODToolkit allows the user to add new **data sets**, and to extend the toolkit with new **models** and **evaluation metrics**

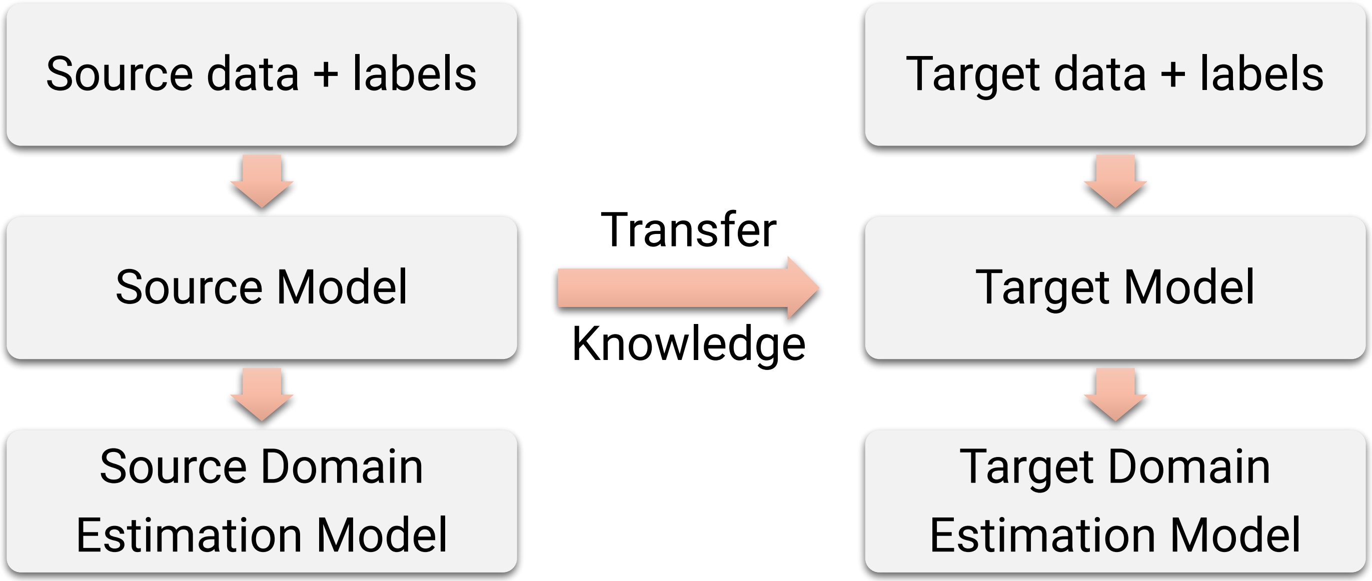
- To evaluate this toolkit, we implemented a new class of occupancy detection models (i.e., domain-adaptive models) and evaluated their performance with respect to the baseline models

Domain adaptation - basic idea

transform a pre-trained model from a source domain to a related target domain after performing some modifications on it (adaptation process)

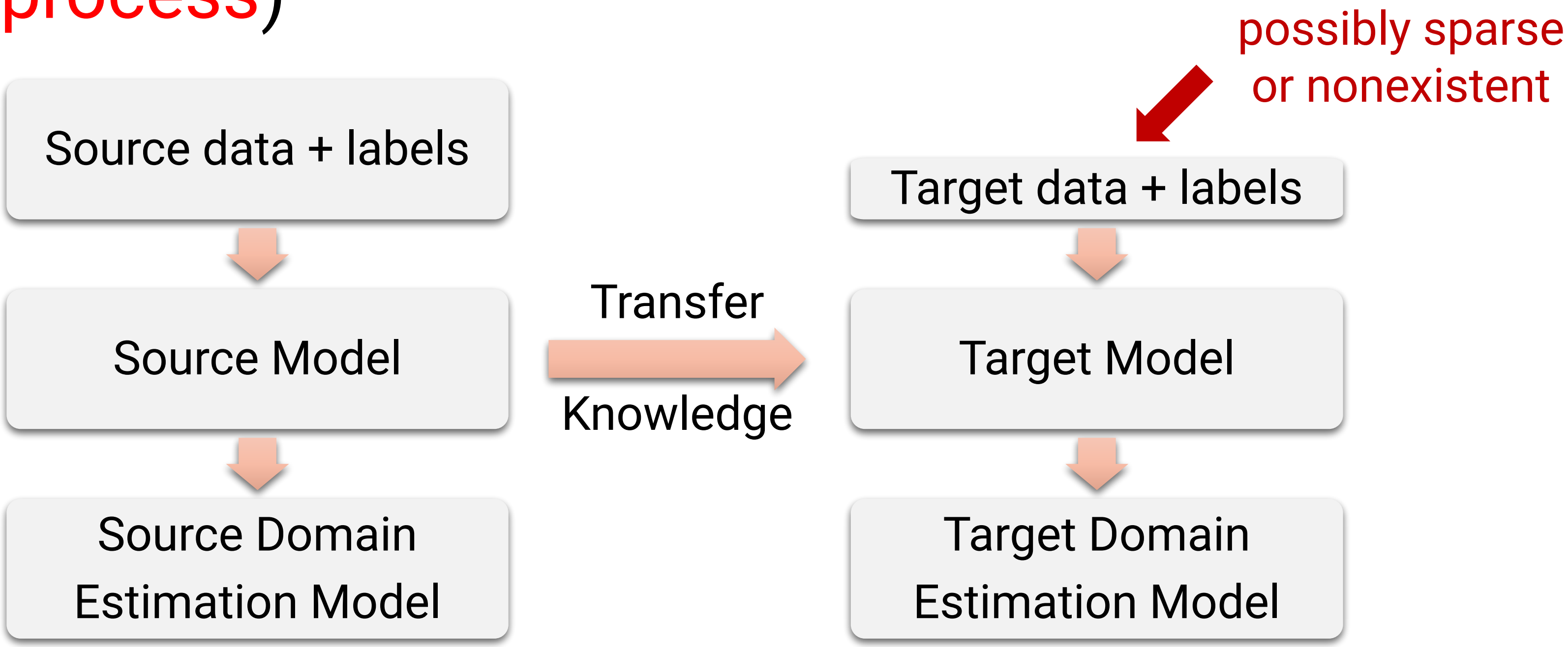
Domain adaptation - basic idea

transform a pre-trained model from a source domain to a related target domain after performing some modifications on it (adaptation process)



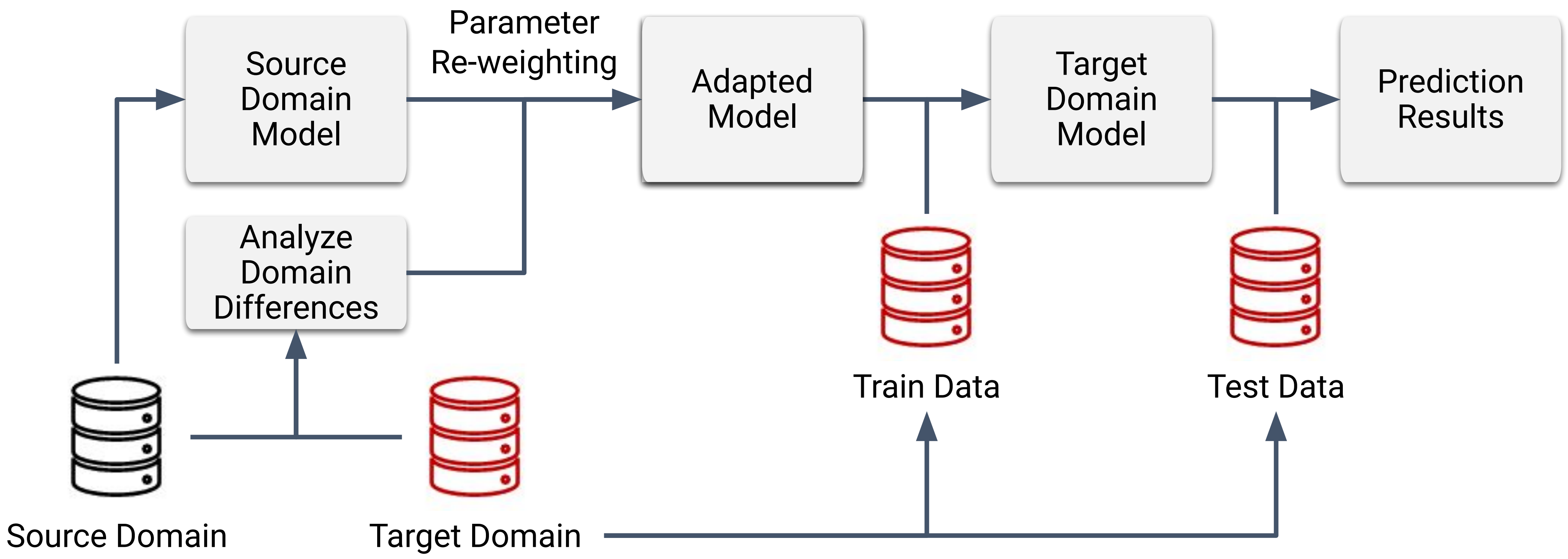
Domain adaptation - basic idea

transform a pre-trained model from a source domain to a related target domain after performing some modifications on it (adaptation process)



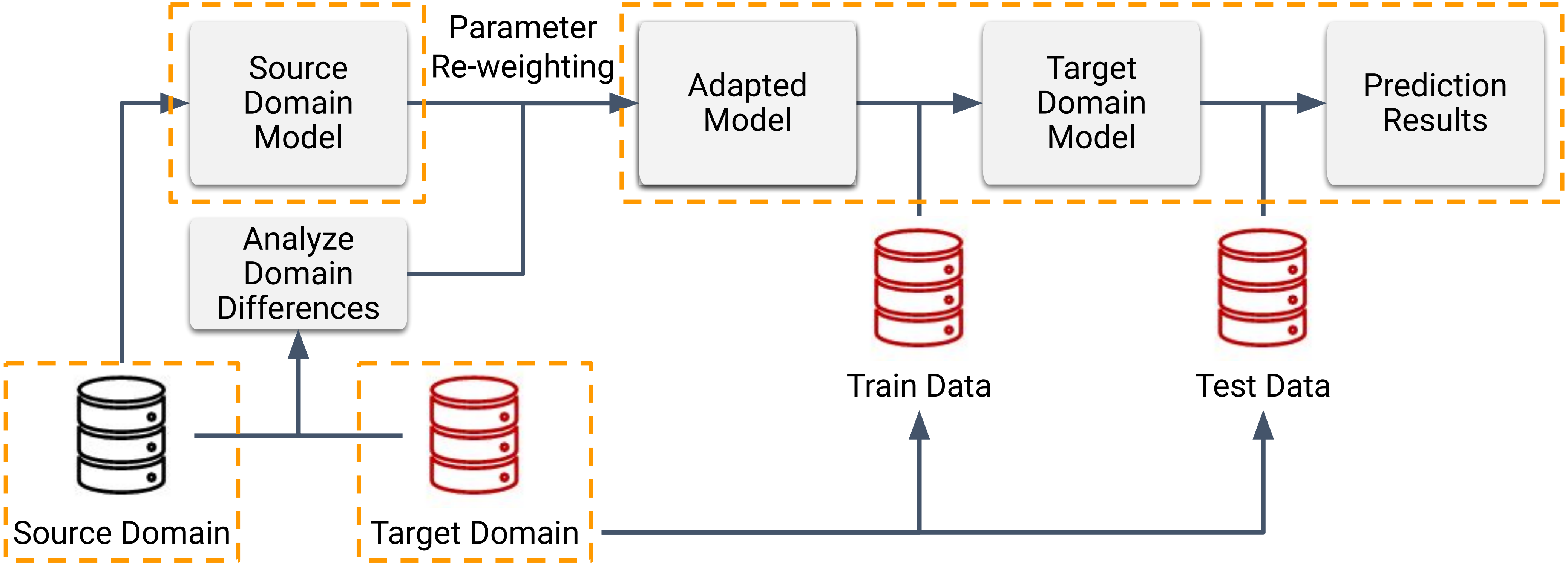
Case studies - extending the toolkit

- DA-LSTM and DA-PF are implemented by re-using the LSTM and PF models that were already included in the toolkit

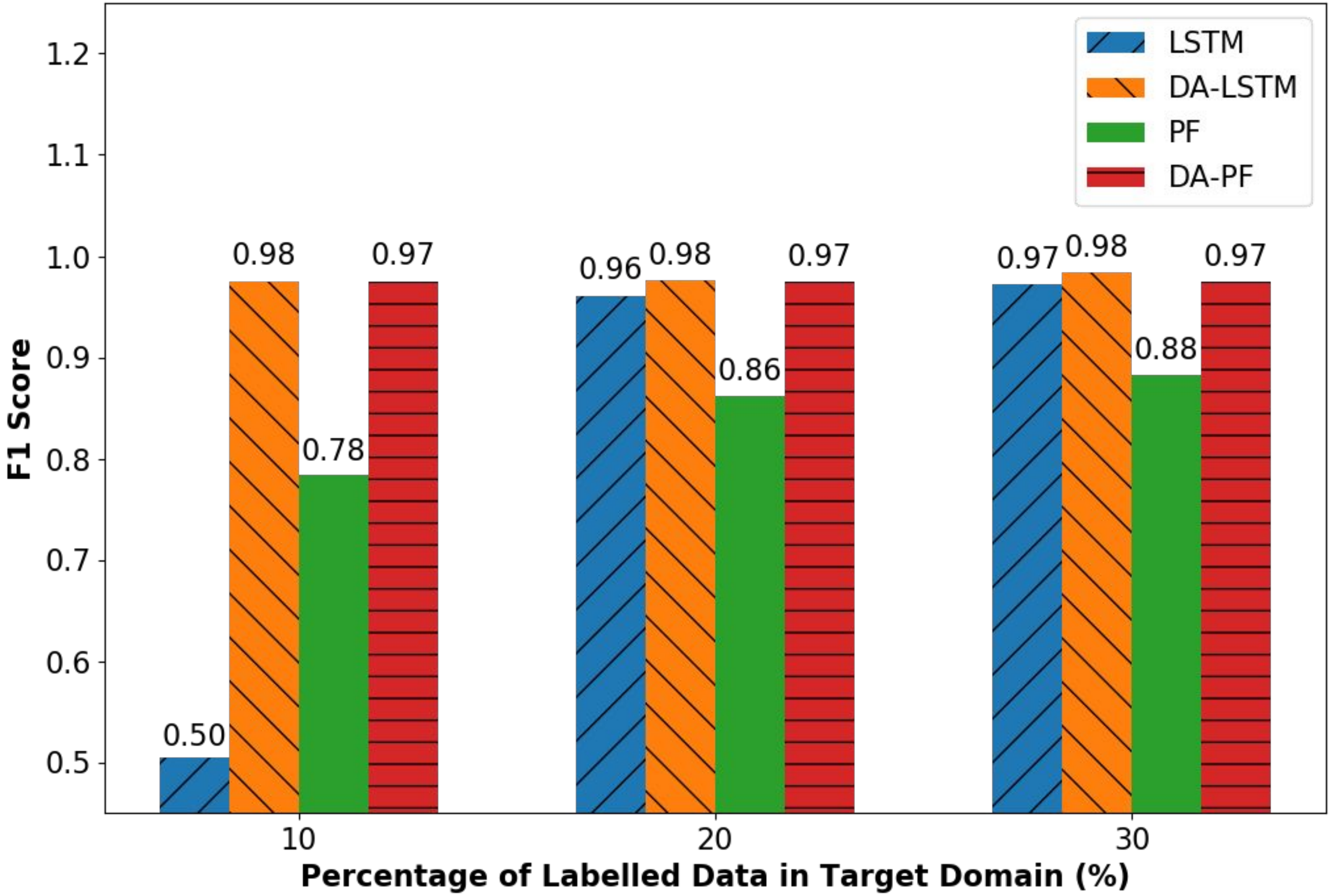


Case studies - extending the toolkit

- DA-LSTM and DA-PF are implemented by re-using the LSTM and PF models that were already included in the toolkit



Case studies - performance evaluation



Case studies - comparing different supervised learning models

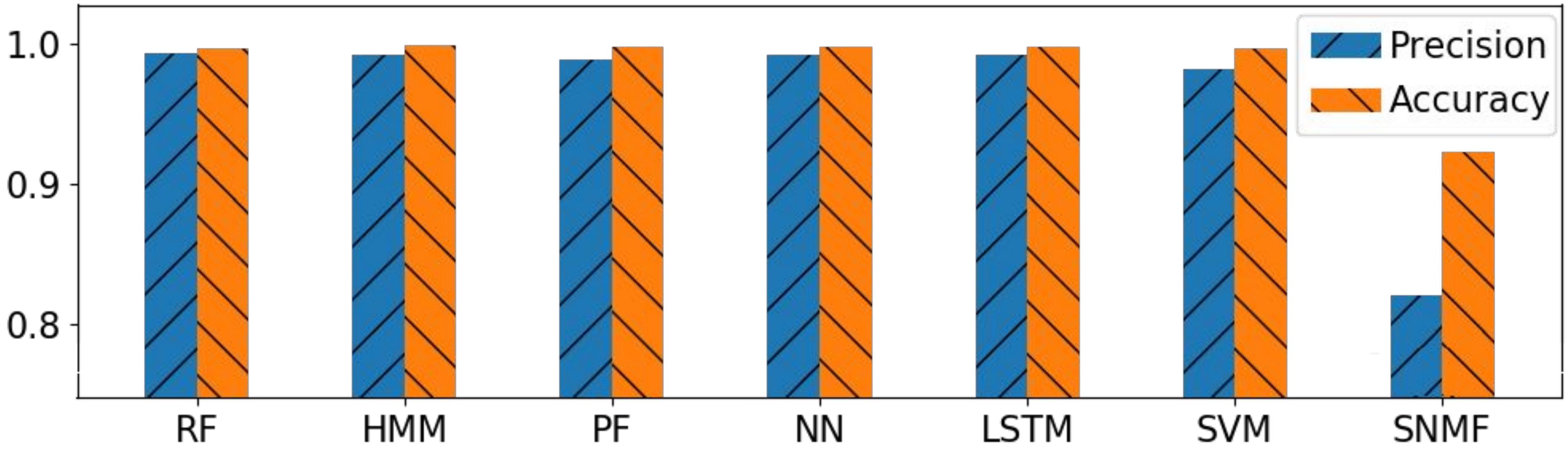
- **Task 1**: Find the best binary occupancy detection model on one data set and across all data sets
- **Task 2**: Compare the results of binary occupancy detection and occupancy count determination models
- **Task 3**: Evaluate the robustness of the model

Criteria for choosing the best binary occupancy detection model:

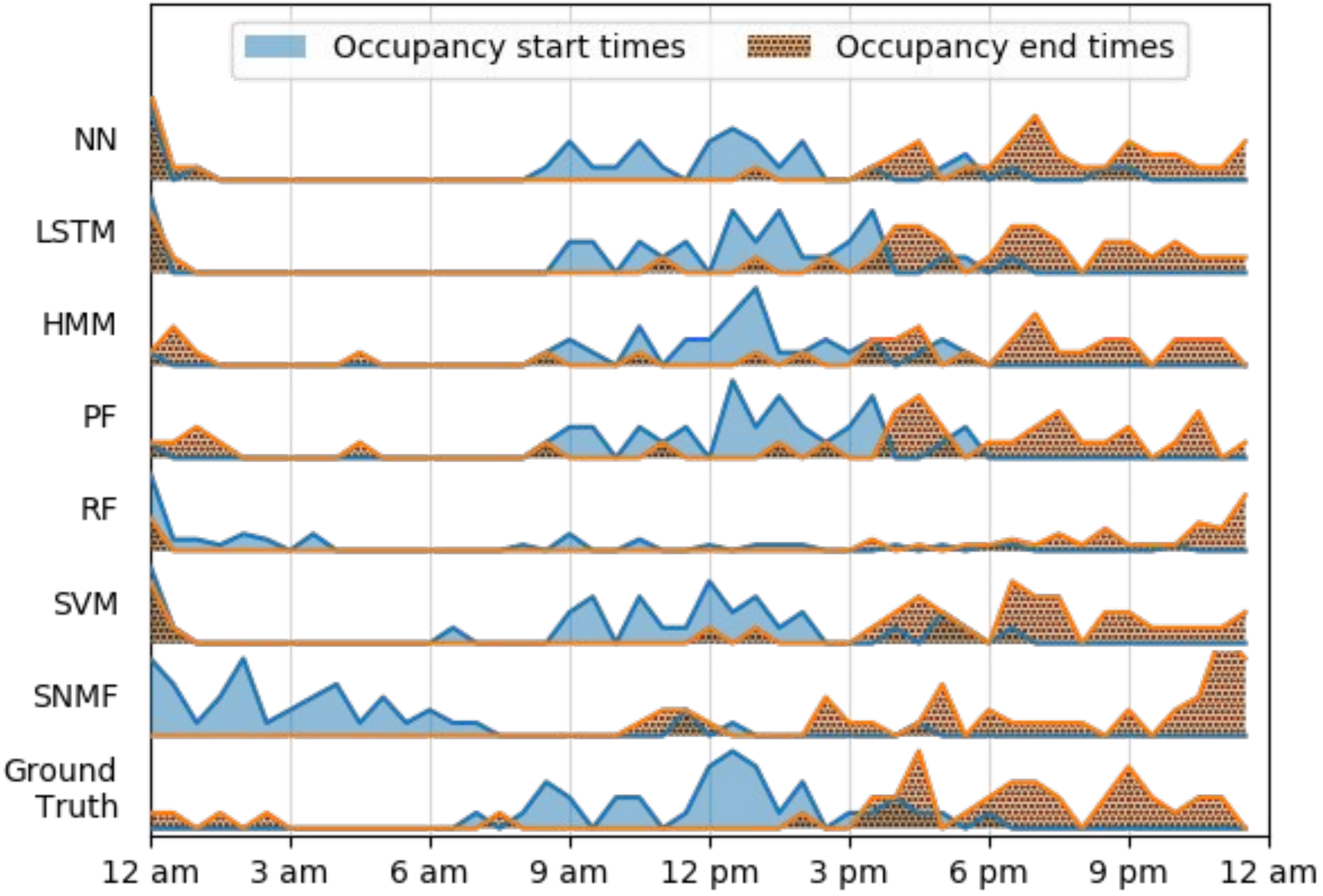
  - Highest overall score, better estimation of occupancy start/end times

Case studies - comparing different supervised learning models

		Data set				
		A	**B**	**C**	**D**	**E**
RF	**Accuracy**	0.9959	0.6416	0.7624	0.8234	0.9088
	F1 Score	0.9854	0.6608	0.8364	0.6151	0.7399
NN	**Accuracy**	0.9981	0.6172	0.8884	0.8000	0.9662
	F1 Score	0.9932	0.7141	0.9279	0.1957	0.8945
LSTM	**Accuracy**	0.9981	0.6719	0.8472	0.8068	0.9412
	F1 Score	0.9932	0.7478	0.8816	0.1436	0.8947



Case studies - comparing different supervised learning models



Takeaways

- We present the design and implementation of ODToolkit, and discuss how it can be extended to incorporate new **data sets**, **algorithms**, and **metrics**
- We extend the toolkit with three new domain-adaptive occupancy detection algorithms and evaluate their performance
- We investigate how using the toolkit reduces the time and effort required to build new models

Directions for future work

- Collect more available data sets and models
- Separate huge data set into small chunks so that they could fit in the RAM

We encourage the community to use, improve and extend this toolkit by adding their occupancy estimation models and/or data sets

GitHub Documentation available on: https://odtoolkit.github.io/
Code available on: https://github.com/sustainable-computing/ODToolkit

Case studies — extending the toolkit

```python
import odtk

# Load two sample data sets from the package
dataset = odtk.data.load_sample(["umons-all", "sdu-all"])

# Use two models to perform occupancy estimation
# Use all binary evaluation metrics to evaluate the model
result = odtk.evaluation.Result()
result.set_result(odtk.easy_set_experiment(dataset,
                                           models=["RandomForest", "NN"],
                                           evaluation_metrics="all",
                                           thread_num=1)[0])


# Plot the scores in a bar chart
odtk.plot.plot_result(result,
                      metric="F1Score",
                      threshold="<= 1",
                      file_name="one_dataset_one_model_all_metrics")
```

Case studies — extending the toolkit

```
e.g.

self.learning_rate = 0.1
self.step_size = 0.9
```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
# If you want to put your model into the ODToolkit folder, use this
from .superclass import *
# If you want to put your model into your own folder, use this
from odtk.model.superclass import *

# Sample for supervised-learning model
class YourModelName(NormalModel):
    def __init__(self,
                 name_for_train_dataset,
                 name_for_test_dataset):
        # all changeable parameters
        self.name_for_train_dataset = name_for_train_dataset
        self.name_for_test_dataset = name_for_test_dataset

        # ... Any other parameters defines here

    # the model must have a method called run, and return the predicted result
    def run(self):
        # Your model goes here
        # Use odtk.data.dataset.Dataset() as data type

        # ...

        # Result must be a numpy.ndarray with shape of (num_of_rows, 1)
        return predict_occupancy
```

```
e.g.

result = \
    self.name_for_test_dataset.data.sum(axis=1)
predict_occupancy = result.reshape((-1, 1))
```