Where Will Dockless Shared Bikes be Stacked? — Parking Hotspots Detection in a New City

Zhaoyang Liu
Shanghai Jiao Tong University
Shanghai, China
towardsun@sjtu.edu.cn

Yanyan Shen*
Shanghai Jiao Tong University
Shanghai, China
shenyy@sjtu.edu.cn

Yanmin Zhu*
Shanghai Jiao Tong University
Shanghai, China
yzhu@sjtu.edu.cn

ABSTRACT

Dockless shared bikes, which aim at providing a more flexible and convenient solution to the first-and-last mile connection, come into China and expand to other countries at a very impressing speed. The expansion of shared bike business in new cities brings many challenges among which, the most critical one is the parking chaos caused by too many bikes yet insufficient demands. To allow possible actions to be taken in advance, this paper studies the problem of detecting parking hotspots in a new city where no dockless shared bike has been deployed. We propose to measure road hotness by bike density with the help of the Kernal Density Estimation. We extract useful features from multi-source urban data and introduce a novel domain adaption network for transferring hotspots knowledge learned from one city with shared bikes to a new city. The extensive experimental results demonstrate the effectiveness of our proposed approach compared with various baselines.

CCS CONCEPTS

• **Applied computing** → **Transportation**; Forecasting; • **Information systems** → *Data mining*;

KEYWORDS

Dockless shared bikes; hotspots detection; urban computing; transfer learning

ACM Reference Format:
Zhaoyang Liu, Yanyan Shen[1], and Yanmin Zhu[1]. 2018. Where Will Dockless Shared Bikes be Stacked? — Parking Hotspots Detection in a New City. In *KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3219819.3219920

1 INTRODUCTION

The unprecedented booming of dockless shared bikes has reinvented bike riding business in China. For instance, Shanghai, the largest metropolis in China, currently has over 1.5 million dockless shared bikes on the streets. The rapid emergence of dockless shared bike services from companies such as Mobike[1] and ofo[2] allow users to find, ride and park bikes "anywhere" via GPS-based smartphone apps. As a flexible and convenient solution to the first-and-last mile connection, dockless shared bike business is spreading to new cities in China as well as expanding aggressively abroad.

During the expansion process of dockless shared bike programs, various challenging issues such as urban planning, transportation management and bike traffic turmoil, arise as side effects. Indiscriminate parking is undoubtedly one of the most emergent issues to be resolved. It has been reported that mountains of shared bikes have been strewn surrounding private buildings and blocked pedestrians in central areas of Shanghai [35]. Such parking chaos caused by too-many-bikes-yet-insufficient-demands raise great safety concerns with local residents. To address the issue, both the government and bike sharing operators have taken actions such as forbidding further bike deployment or forcing bikes to be returned to a designated parking location. Instead of bringing force timely amendments for the parking issue, a more constructive way is to prevent the issue from happening in the very beginning. More specifically, before delivering shared bikes into a new city, can we predict *the roads in the city that will be easily stacked with parking bikes so that necessary actions could be taken in advance?*

In this paper, we refer to the roads that are most likely to be stacked with shared bikes as *parking hotspots* (*hotspots* for short). Identifying hotspots in new cities before the actual bike deployment offers good benefits as follows. From sharing bike operators' perspective, this helps improve bike delivery strategy and avoid waste of overabundant bike deployment; for local governors, foreknowing potential bike congestions on the roads is of great value to renew bike regulations and establish electronic parking slots beforehand in order to ensure city tidiness and the safety of pedestrians.

Existing works on bike sharing systems mainly studied the problems of station demand prediction [11, 38], rebalance scheduling [22] and station site optimization [21, 26] where the systems are with docking stations. In [1], Bao et al. leveraged dockless shared bike trajectories for bike lane planning. However, none of these works addresses the bike parking issue in new cities during business expansion. To the best of our knowledge, this work is the first to detect hotspots with dockless shared bikes in new cities so as to prevent parking chaos before bike deployment.

Detecting hotspots with dockless shared bikes in a new city is challenging due to the following three factors. First, it is a non-trivial task to define the hotness of a road. Due to the arbitrary

*Corresponding author.

KDD '18, August 19–23, 2018, London, United Kingdom

ACM ISBN 978-1-4503-5552-0/18/08…$15.00
https://doi.org/10.1145/3219819.3219920

[1]https://mobike.com
[2]https://www.ofo.com

road length, the total number of bikes parking along the road may not be an accurate indicator for road hotness. For instance, a long road typically involves more parking bikes, but it may not be hot or overloaded if the bikes are distributed over the entire pathway. Second, hotspots in a city are often changing over time and affected by user riding behaviors. Intuitively, most shared bikes are gathered along the roads near subway stations during morning rush hours but accumulated in those next to residential areas during evening rush hours. Finally, the locations of hotspots are closely related to city characteristics. Due to the dissimilar nature of different cities, the hotspots knowledge learned from bike distribution in one city may not be well adapted to another city, which is also known as the domain shift problem [29].

To address the challenges, in this paper, we propose to quantify road hotness based on a citywide bike probability density function. Regarding a city deployed with dockless shared bikes, we apply the Kernel Density Estimation to estimate the probability that a bike will be parked in any location of the city. The k roads with the highest bike density values in their centroid points are reported as *hotspots*, which alleviates the effects from varying road lengths. We conduct an empirical analysis over real-life dockless shared bike records from Mobike and observe that the variation in hotspots over time is relatively small thanks to the "short-riding-distance" nature of bike routes. Moreover, the proliferation of multi-source urban data such as POIs, road network, and satellite light disclose useful information on city characteristics that can be exploited to learn discriminative features for accurate hotspots detection.

As for the domain shift challenge, we introduce a novel Convolutional City Domain Adaptation Network named ConvCDAN for transferring hotspots knowledge learned from the source city to the target city without dockless shared bikes. ConvCDAN consists of three components: (1) a FeatureNet that learns discriminative features from the raw multi-source data, (2) a DensityNet that predicts bike density of a road such that the overall hotness ranking can be reserved, and (3) a DomainNet that promotes the features from FeatureNet to deeper representations that are domain-invariant. The collaboration of the three components effectively drives the model to obtain discriminative features for detecting hotspots in the target city accurately, when only bike records from the source city are supplied. The experimental results on real data demonstrate the superior performance of ConvCDAN on hotspots detection in a new city, compared with various baseline methods.

The main contributions of this paper are summarized as follows.

- To the best of our knowledge, we are the first to formulate the practical problem of parking hotspots detection for dockless shared bikes in new cities. We propose an end-to-end domain adaptation approach to this problem (Section 2).
- We apply the Kernel Density Estimation to measure road hotness based on the bike density. We also conduct an empirical analysis to summarize important temporal and spatial characteristics of hotspots (Section 3).
- We extract useful features from multi-source data, provide detailed feature analysis results and point out the domain shift challenge. We introduce a novel Convolutional City Domain Adaptation Network named ConvCDAN towards accurate hotspots detection in the target city (Section 4).
- We evaluate the performance of our proposed solution on real Mobike data collected from three cities in China, from 01/07/2017 to 31/07/2017 The results show that on average, ConvCDAN achieves 34% and 16.8% higher MAP@10 and NDCG@10 (against truth hotspots) compared with the competitive baselines (Section 5).

2 PRELIMINARIES

Let R_s and R_t denote the sets of roads in the source and target cities, respectively. Each road $r \in R_s \cup R_t$ is represented by a triplet of $(\phi, len, type)$, where ϕ contains a sequence of spatial points in the road, len is the road length and $type$ is the road type, e.g., trunk, pedestrian, motorway. We associate each road r with a bounding rectangle $B(r)$ that covers all the points ϕ in the road and a centroid point $C(r)$, which are defined as follows.

Definition 2.1 (Road Bounding Rectangle and Centroid). Given a road r, we define its *bounding rectangle* as $B(r) = (p_{r1}, p_{r2})$, where $p_{r1}.lon = \min\{p.lon \mid p \in \phi\}$, $p_{r1}.lat = \min\{p.lat \mid p \in \phi\}$, $p_{r2}.lon = \max\{p.lon \mid p \in \phi\}$, $p_{r2}.lat = \max\{p.lat \mid p \in \phi\}$. The *centroid point* $C(r)$ of road r satisfies: $C(r).lon = (p_{r1}.lon + p_{r2}.lon)/2$ and $C(r).lat = (p_{r1}.lat + p_{r2}.lat)/2$.

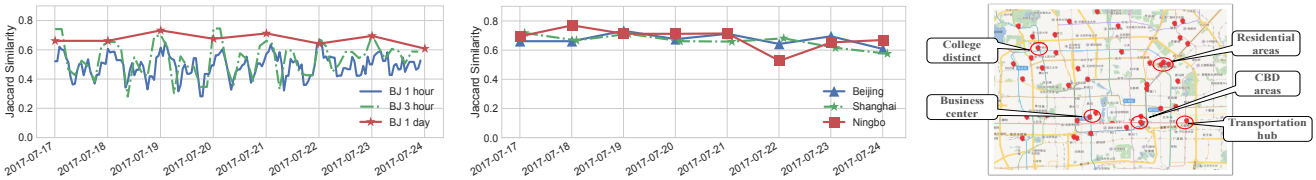
We denote by D the set of available dockless shared bike records in the source city, where each record in D represents a bike parking at a certain location during time t. To quantify the *hotness* of a road, a simple way is to measure the average number of bikes parking along the road. However, a longer road typically involves more bikes but may not be overloaded if the bikes are distributed over the entire pathway. Compared with the bike number, the bike density (i.e., the number of bikes per unit space) is a more precise signal that reflects the traffic condition of a road, without being affected by the road length. Hence, in this paper, we propose to compute bike density for each road as its hotness value.

Since dockless shared bikes could be parked in any places along the road and the number of accumulated bikes varies over time, we first extract raw bike records during time period t from D and map these records onto the city map accordingly. After that, we leverage the nonparametric model of Kernel Density Estimation (KDE) to compute a citywide bike density function f^t for time period t (see details in Section 3.1). To derive bike density of a road r, we focus on the average density value in road centroid over a sequence of time slots $\{1, \cdots, T\}$, as defined below.

Definition 2.2 (Average Bike Density $d_{avg}(r)$ of a Road). Given the estimated bike probability density function f^t during time period t, bike density $d^t(r)$ of road r at time t is computed by $f^t(C(r))$. The *average bike density* $d_{avg}(r)$ of road r satisfies:

$$d_{avg}(r) = \frac{1}{T}\sum_{t=1}^{T} f^t(C(r)) \tag{1}$$

It is important to note that (1) the variation in hotspots at different time is observed to be relatively small (see details in Section 3) and hence we average road hotness values over time, and (2) road hotness measured by Equation 1 only depends on road centroid and bike records, while other metrics such as mean density value over road points in ϕ could be affected by road point sampling methods.



(a) Jacccard similarity w.r.t. time window sizes (Beijing) (b) Jacccard similarity with 1-day time window (c) Top 30 hotspots (Beijing)

Figure 1: Temporal and spatial properties of hotspots

The goal of this paper is to identify spatial hot roads with parking bikes in the target city, which are also referred to as *hotspots*. As each city may have its own standard for hot roads, we focus on relative hotness ranking over all the roads in a city and try to report k roads with the highest average bike density values.

Definition 2.3 (Hotspots in a City). Let $d_{avg}(R)$ be the average bike density values for a set R of roads in a city. The hotspots $H_k(R)$ in the city is the k roads with the highest d_{avg} values, where k varies among cities and can be determined by the city scale.

Problem Statement. In this paper, we aim to transfer the knowledge of hotspots in a source city to a target city where no dockless shared bike has been deployed yet. Inspired by the high availability and abundant information encoded in multi-source urban data, we leverage them to facilitate hotspots detection in the target city. We now formally define the problem studied in this paper.

Definition 2.4 (Detecting Hotspots in the Target City). We consider two cities: source and target cities with road sets R_s and R_t, respectively. Given k, a set D of dockless shared bike records in the source city, multi-source data including POIs, satellite light, transportation, road networks collected in both cities, our problem is to predict bike density for each road in R_t and report k roads with the highest densities as hotspots $H_k(R_t)$ in the target city.

For ease of illustration, we denote by $\mathcal{S}$ and $\mathcal{T}$ the sets of feature instances extracted from multi-source data for $|R_s|$ and $|R_t|$ roads in the source and target cities, respectively. The instances in $\mathcal{S}$ are associated with ground-truth labels $\mathbf{Y} = \{y_i\}_{i=1}^{|R_s|}$ where each label is the corresponding average bike density value $d_{avg}(r_i)$.

3 HOTSPOTS DETECTION AND ANALYSIS

In this section, we present our hotspots detection algorithm and provide an empirical analysis of hotspots spatial-temporal properties using real dockless shared bike records from Mobike.

3.1 Detecting Hotspots with Kernel Density Estimation

According to Definition 2.2 and 2.3, an important part of our hotspots detection algorithm is a nonparametric model of Kernel Density Estimation (KDE) [31]. Given a set D of dockless shared bike records in a city with the road set R, we first estimate bike probability density function f^t within time window t by KDE and then evaluate average density values for the centroid points of roads in R.

We denote by $D^t = \{p^{(i)}\}_{i=1}^{|D^t|}$ the set of records of bike locations during time period t, which are considered as independent and identically distributed samples drawn from a distribution with an unknown density function f^t. We apply the standard KDE to estimate the shape of f^t, as expressed by:

$$f^t(p) \propto \sum_{i=1}^{|D^t|} K\left(\frac{d(p, p^{(i)})}{h}\right) \tag{2}$$

where $K : \mathbb{R} \to \mathbb{R}$ is a smooth kernel function, $h > 0$ is the smoothing bandwidth, and $d(\cdot, \cdot)$ is the Haversine distance [2] that measures the angular distance between points on a sphere. We choose to use the Gaussian kernel for K. The selection of bandwidth h has certain influence on the estimated density function, which is typically a trade-off between the bias and variance of the estimator. Instead of choosing the bandwidth parameter using the Silverman's rule [18], we employ a data-driven approach that computes the best bandwidth by maximizing likelihood via cross-validation.

Given $f^t(p)$ and a road $r \in R$, we can compute the density value in the centroid point of r during time t as: $d^t(r) = f^t(C(r))$, and derive road hotness $d_{avg}(r)$ according to Equation 1. The hotspots of the city $H_k(R)$ are k roads in R with the highest hotness values.

3.2 Analysing Temporal and Spatial Properties of Hotspots

We conduct an empirical analysis on a Mobike dataset with dockless shared bike records in three cities of China: Shanghai, Beijing, Ningbo, from 01/07/2017 to 31/07/2017. The details of the dataset will be presented in the experiments. We divide all the bike records into disjoint groups given a time window size which is set to 1 hour by default. We detect hotspots in each of the three cities according to the method mentioned in Section 3.1, and obtain the following observations on temporal and spatial properties of hotspots.

Temporal properties. We first examine any two sets of hotspots in consecutive time windows. Let $H_k^t(R)$ and $H_k^{t+1}(R)$ be the k hotspots in a city during time periods t and $t+1$, respectively. We compute Jaccard similarity between $H_k^t(R)$ and $H_k^{t+1}(R)$ as follows:

$$J(H_k^t(R), H_k^{t+1}(R)) = \frac{|H_k^t(R) \cap H_k^{t+1}(R)|}{|H_k^t(R) \cup H_k^{t+1}(R)|} \tag{3}$$

Intuitively, higher Jaccard similarity scores indicate a larger amount of overlap among hotspots detected in continuous time windows. For the illustration purpose, Figure 1(a) shows the Jaccard similarity scores for hotspots in Beijing from 17/07/2017 to 24/07/2017, where the time window size is set to 1-hour/3-hour/1-day and k is set to 100. We summarize the key observations: (1) consecutive sets of hotspots are similar with respect to different time window sizes, and the average Jaccard scores for 1-hour, 3-hour and 1-day

window sizes are 0.49, 0.54 and 0.67, respectively; (2) the trends of Jaccard scores for 1-hour/3-hour time window sizes exhibit periodicity, and the changes of hotspots during two peak hours are relatively larger.

We also plot Jaccard scores for hotspots detected in three cities with a window size of 1-day. From Figure 1(b), we can see that: (1) the hotspots in consecutive time windows are quite stable in different cities due to the large Jaccard scores; (2) on average, Shanghai has the smallest Jaccard similarity, followed by Beijing and finally Ningbo. This follows our intuition that the road network in Shanghai is more complex and may cause larger variation in hotspots.

Spatial properties. Figure 1(c) plots top 30 hotspots in Beijing with the highest average daily bike density values from 01/07/2017 to 31/07/2017. We observe several regularities: (1) many hotspots are spatially close to each other due to the connectivity of roads; (2) roads in business centers, transportation hubs and residential areas are more likely to be hotspots.

To sum up, the above observations indicate temporal stability of hotspots which could be explained by the fact of "short-riding-distance" of bike routes. Moreover, there exists a close relationship between hotspots and geographic information of the city, which inspires us to extract discriminative features from multiple urban data sources for hotspots detection.

4 DETECTING HOTSPOTS IN A NEW CITY

In this section, we first present the features considered in this work and comprehensive analysis results. We then describe the proposed neural approach to hotspots detection in the target city.

4.1 Feature Extraction

With the aim of predicting road hotness, we extract features for each road within its bounding rectangle. We mainly consider the following features extracted from multiple urban data including POIs, road network, satellite light and transportation.

POI Features $\mathbf{x_p}$. Each POI represents a city venue with its name, address, category and spatial coordinates. The number and diversity of POIs within the bounding rectangle of a road reflect its prosperity, and hence are related to bike density of the road. We thus extract POI features including: (1) *POI frequency in each category*, (2) *total number of POIs*, (3) *POI entropy*.

Road Network Features $\mathbf{x_r}$. Intuitively, the complexity of nearby road structure affects road hotness. For any road r, we propose to use two road network features: (1) *the type of road* r (in one-hot representation), (2) *the number of roads overlapped with the bounding rectangle* $B(r)$.

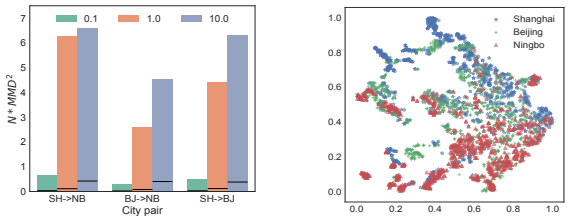
Satellite Light Features $\mathbf{x_l}$. Inspired by Jean's work [17] that leverages satellite images to predict area poverty, we identify two important features from the satellite data indicating prosperity degree around the road: (1) *average light intensity*, (2) *distance to the nearest light centre*, for which we sample top l% roads with the highest light intensity values, identify light centers by clustering selected roads using the DBSCAN [8] algorithm, and compute the distance from road centroid point to its nearest light centre.

Transportation Features $\mathbf{x_{tr}}$. Based on the observation that nearly 60% Mobikes in Shanghai/Beijing/Ningbo are parked less than $1km$ away from subway stations and most shared bike routes are less than $3km$, we extract two features: (1) *distance from road centroid to its nearest subway station*, (2) *the number of subway stations whose distances to the road centroid are less than* $3km$.

Business Centre Features $\mathbf{x_b}$. As discussed before, bike density can be affected by its surrounding business centres. We collect city business centres from AliTrip (https://www.alitrip.com), and divide them into four levels manually according to the location, people flow and per capita purchasing ability. We extract two business centre features for each road: (1) *distance to the nearest business centre*, (2) *the level of the nearest business centre*.

Table 1: Top 10 features with highest Pearson coefficients

Feature	Pearson	Feature	Pearson
# of POIs	0.62	POI shopping	0.46
dist to business centre	-0.61	POI life service	0.45
dist to subway station	-0.60	POI estate	0.44
POI entropy	0.60	POI food	0.43
POI company	0.54	light intensity	0.39



(a) MMD results (b) TSNE visualization

Figure 2: Domain analysis among cities

4.2 Feature Correlation and Domain Analysis

Feature Correlation Analysis. Table 1 lists 10 features that have the strongest correlations to road hotness (in Definition 2.2) measured by the Pearson correlation coefficient, where the coefficients are all above 0.39. This indicates the effectiveness of our extracted features from multi-source urban data for hotspots detection.

Domain Analysis between Cities. Intuitively, the feature domain of a city incorporates city-oriented characteristics. For example, the two tier-1 cities in China, Beijing and Shanghai, include more complex road networks and richer POI structures, compared with the tier-3 city Ningbo. The feature domain difference between cities poses a great challenge to transferring hotspots knowledge learned from one city to another.

To illustrate the *feature domain shift* between two cities, we apply maximum mean discrepancy (MMD) [28] to quantify the overall feature domain difference between the source city $\mathcal{S}$ and the target city $\mathcal{T}$. Specifically, the difference in two feature domains is computed by the squared distance between the embedding representations of two cities' instances in the RKHS space $\mathcal{H}$ [33]:

$$MMD(\mathcal{S}, \mathcal{T}) = \left\| \mu_s - \mu_t \right\|^2_{\mathcal{H}} \quad (4)$$

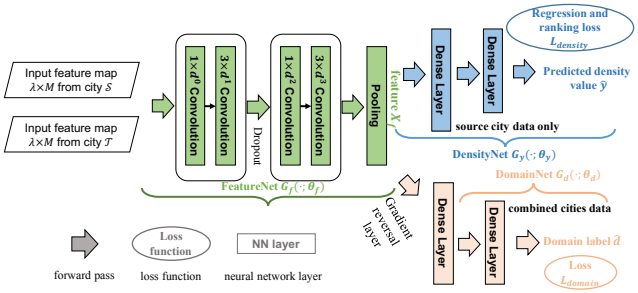


Figure 3: ConvCDAN architecture

where μ is the centroid vector by averaging feature representations of a city in the RKHS space.

Given m_s and m_t training samples for source and target cities respectively, we can formulate a test statistic for the null hypothesis that both sets of samples stem from the same distribution, based on the empirical estimate of MMD as follows:

$$\widehat{MMD}(\mathcal{S}, \mathcal{T}) = \left\| \frac{1}{m_s} \sum_{i=1}^{m_s} \phi(s_i) - \frac{1}{m_t} \sum_{j=1}^{m_t} \phi(t_j) \right\|_{\mathcal{H}}^2 \tag{5}$$

where $\phi(\cdot)$ is the kernel function that maps the raw features into RKHS space.

To guarantee the comprehensiveness of the test statistic, we plot the MMD estimation results for three city pairs under the Gaussian kernel setting with bandwidths of 0.1, 1 and 10, respectively. As shown in Figure 2(a), the black solid lines are the thresholds for rejecting the hypothesis with a test power $\alpha = 0.05$. Our estimation results are much larger than the thresholds, which verify the existence of feature domain shift between cities.

Figure 2(b) shows the TSNE visualization [13] results over the feature domains for three cities. TSNE is an effective method to visualize the high-dimensional data distributions. When the number of feature dimension is reduced to 2, we can see the obvious feature distribution gaps and shifts among cities, especially between tier-1 cities (Shanghai, Beijing) and tier-three city (Ningbo). Both MMD and TSNE results inspire us to introduce a city domain adaptation method to address the shift challenge for hotspots detection in a new city.

4.3 City Domain Adaptation Network

We now present our Convolutional City Domain Adaptation Network named ConvCDAN for hotspots detection in a new city based on the features discussed in Section 4.1. Figure 3 shows the architecture of ConvCDAN that consists of three key components:

(1) *FeatureNet* learns latent features and their interactions based on the raw extracted features for a road and its successive ones;

(2) *DensityNet* predicts the average bike density of r (Definition 2.2) and tries to preserve the hotness ranking over all roads;

(3) *DomainNet* guides the network to learn domain-invariant feature representations for domain adaptation.

In what follows, we provide the details for the three components. Without otherwise specified, we use *Conv*, $Pool_{avg}$, *BN* and *Drop* to represent convolutional, average pooling, batch normalization [16] and dropout [34] operations, respectively.

4.3.1 FeatureNet G_f: learning deep feature representation. As most Mobike routes are shorter than $3km$, the bike density of a road can be easily influenced by its nearby roads. This is also verified in Figure 1(c) that hotspots tend to appear in groups. Hence, for each road r, we identify its *neighbor roads* according to the distance between road centroids, and denote by $L_\lambda(r)$ the set of $\lambda - 1$ nearest neighbor roads of r.

The input to FeatureNet G_f is a $\lambda \times M$ feature map $\mathbf{X}$, where the rows contain feature vectors for $\{r\} \cup L_\lambda(r)$ and M is the total feature dimension according to Section 4.1. The roads in the rows of $\mathbf{X}$ are ordered by their distances to the centroid point of r. Besides the input layer, FeatureNet contains two convolutional blocks and one pooling layer for deep feature representation learning.

Each convolutional block includes two convolutional layers: $1 \times d^{l-1}$ *Conv* to enhance non-linear expression discriminability of the network [20], followed by $3 \times d^l$ *Conv* to learn latent feature interactions. We denote by tensor $\mathbf{W}^l_{k,t,d}$ the k convolutional filters with a size of $t \times d$ in layer l. The definition of each convolutional block is as follows:

$$\mathbf{Z}^l = Conv(\mathbf{W}^l_{k^l,1,d^{l-1}}, \mathbf{X}^{l-1}) + \mathbf{b}^l \tag{6}$$

$$\mathbf{X}^l = \sigma(BN(\mathbf{Z}^l)) \tag{7}$$

$$\mathbf{Z}^{l+1} = Conv(\mathbf{W}^{l+1}_{k^{l+1},3,d^l}, \mathbf{X}^l) + \mathbf{b}^{l+1} \tag{8}$$

$$\mathbf{X}^{l+1} = \sigma(BN(\mathbf{Z}^{l+1})) \tag{9}$$

where $\mathbf{X}^{l+1}$ is the block output, the $\mathbf{b}$ terms are bias vectors, σ is the non-linear activation function for which we use *ReLU*. Note that $d^l = k^l$ holds, i.e., the output feature map dimension of layer-l is equal to the number of the filters applied by the convolutional operation in layer l. And d^0 is equal to the feature dimension M of the input feature map $\mathbf{X}$. To prevent overfitting, we also apply *Drop* to randomly drop neurons between two convolutional blocks.

In G_f, we perform the global average pooling operation $Pool_{avg}$ on the feature map produced by two convolutional blocks, and get the final feature representation $\mathbf{X_f}$ as the output of FeatureNet which will be fed to DensityNet and DomainNet simultaneously.

4.3.2 DensityNet G_y: predicting density and ranking. The aim of DensityNet is to predict average bike density for road r as its hotness value. As we only know densities for roads in the source city, during the training phase, we feed DensityNet with the transformed feature representations $\{\mathbf{X_f}\}$ from FeatureNet for roads in R_s.

DensityNet consists of two fully connected feed-forward layers and one output layer. Formally, given the feature representation $\mathbf{X_f}$ for road r, DensityNet predicts its average bike density value $\hat{y}$ based on the following operations:

$$\mathbf{Z}_{y1} = \sigma(\mathbf{W}_{y1}\mathbf{X_f} + \mathbf{b}_{y1}) \tag{10}$$

$$\mathbf{Z}_{y2} = \sigma(\mathbf{W}_{y2}\mathbf{Z}_{y1} + \mathbf{b}_{y2}) \tag{11}$$

$$\hat{y} = \mathbf{h}_y^{\mathrm{T}}\mathbf{Z}_{y2} \tag{12}$$

where the $\mathbf{W}$ and $\mathbf{b}$ terms are weight matrices and vectors respectively, σ is the *ReLU* activation function, and $\mathbf{h}_y$ denotes the neuron weights in the output layer.

A simple way to optimize DensityNet is to minimize the regression loss over labeled training instances from the source city. However, a tiny error in the predicted density values may result in a huge mistake in the final hotness ranking list [30]. In order to find hotspots in the target city accurately, we propose to combine regression loss L_{reg} with ranking loss L_{rank} to obtain the loss function $L_{density}$ for DensityNet as follows:

$$L_{density} = (1-\alpha)L_{reg} + \alpha L_{rank} \tag{13}$$

where $\alpha \in [0,1]$ is a hyperparameter to be tuned on validation set.

We next formally define L_{reg} and L_{rank} in Equation 13. Recall that $\mathcal{S}$ and $\mathcal{T}$ contain $\lambda \times M$ feature maps for roads in source and target cities, respectively. Each feature instance $\mathbf{X}$ in $\mathcal{S}$ is associated with a ground-truth density y. We have:

$$L_{reg} = \frac{1}{|\mathcal{S}|}\sum_{\mathbf{X}\in\mathcal{S}}(\hat{y}-y)^2 \tag{14}$$

where $\hat{y} = G_y(G_f(\mathbf{X},\theta_f),\theta_y)$ is the predicted density.

For L_{rank}, we define $o_{ij} = y_i - y_j$ and $\hat{o}_{ij} = \hat{y}_i - \hat{y}_j$ for any two training instances $\mathbf{X}_i, \mathbf{X}_j$ in $\mathcal{S}$ that satisfy $y_i > y_j$. Let $P_{ij} = \frac{e^{o_{ij}}}{1+e^{o_{ij}}}$ (or $\hat{P}_{ij} = \frac{e^{\hat{o}_{ij}}}{1+e^{\hat{o}_{ij}}}$) be the probability (or the predicted probability) that instance i is ranked higher than j, respectively. L_{rank} is then defined based on the cross entropy function as follows.

$$L_{rank} = \frac{\sum_{i,j\wedge i\neq j} -P_{ij}\log\hat{P}_{ij} - (1-P_{ij})\log(1-\hat{P}_{ij}))}{|\mathcal{S}|(|\mathcal{S}|-1)} \tag{15}$$

4.3.3 DomainNet G_d: learning transferable features for domain adaptation. Applying FeatureNet and DensityNet trained on instances from the source city may suffer from poor performance for hotspots detection in the target city due to the feature distribution shift between cities. To address the problem, one method is to match source and target feature distributions upon learning a new domain-invariant feature space. Many works [23, 27, 36] realize this new space by minimizing MMD, a metric for the dissimilarity between two feature distributions. However, this process is typically non-trivial because the source and target feature distributions are often high-dimensional and constantly changing during the training phase.

Following the Ganin's work [9], we introduce DomainNet G_d in ConvCDAN – a domain classifier network that supervises FeatureNet towards learning both discriminative and transferable features. Specifically, DomainNet absorbs the feature representations produced by FeatureNet for roads in two cities and promotes them to deeper representations that are invariant in two domains. To do this, we design G_d as a combination of two fully connected feedforward layers and an output layer, to classify which domain an input feature representation belongs to. Given an output $\mathbf{X_f} = G_f(\mathbf{X},\theta_f)$ from FeatureNet, DomainNet is formally defined as follows.

$$\mathbf{Z}_{d1} = \sigma(\mathbf{W}_{d1}\mathbf{X_f} + \mathbf{b}_{d1}) \tag{16}$$

$$\mathbf{Z}_{d2} = \sigma(\mathbf{W}_{d2}\mathbf{Z}_{d1} + \mathbf{b}_{d2}) \tag{17}$$

$$\hat{d} = sigmoid(\mathbf{h}_d^{\mathrm{T}}\mathbf{Z}_{d2}) \tag{18}$$

where the $\mathbf{W}$ and $\mathbf{b}$ terms are weight matrices and bias vectors respectively, $\mathbf{h}_d$ denotes the neuron weights of the output layer and $\hat{d}$ is the predicted domain label.

The parameters θ_d in DomainNet are optimized by maximizing the binary cross-entropy loss L_{domain} for domain classification:

$$L_{domain} = \frac{1}{|\mathcal{S}|+|\mathcal{T}|}\sum_{\mathbf{X}\in\mathcal{S}\cup\mathcal{T}} -d\log\hat{d} - (1-d)\log(1-\hat{d}) \tag{19}$$

where d is the ground-truth domain label 0/1 indicating the instance is from the target/source city.

4.3.4 ConvCDAN optimization. ConvCDAN involves three sets of parameters for the three subnets, namely θ_f, θ_y and θ_d. The parameters are optimized jointly by minimizing $L_{density}$ to learn discriminative features for density prediction and maximizing L_{domain} to learn domain-invariant features for domain adaptation. Hence, the overall loss function is defined as follows:

$$\begin{aligned} L(\theta_f,\theta_y,\theta_d) =& (1-\alpha)L_{reg}(\mathbf{X},y;\theta_f,\theta_y,\mathbf{X}\in\mathcal{S}) \\ &+ \alpha L_{rank}(\mathbf{X},y;\theta_f,\theta_y,\mathbf{X}\in\mathcal{S}) \\ &- \lambda L_{domain}(\mathbf{X},y;\theta_f,\theta_d,\mathbf{X}\in\mathcal{S}\cup\mathcal{T}) \end{aligned} \tag{20}$$

where the hyperparameter λ controls the trade-off between density prediction and domain classification.

It is worth noticing that the loss for DomainNet cannot be optimized by the stochastic gradient descent method directly. This is because the parameters θ_d in DomainNet needs to be updated to enhance the classification ability. When the gradients $\frac{\partial L_{domain}}{\partial\theta_f}$ are backpropagated from DomainNet to FeatureNet, they are reversed and the reversed gradients drive FeatureNet to learn new feature representations that are even harder to be distinguished by DomainNet, i.e., the parameters θ_f are updated to increase domain classification error. To solve the problem, we adopt the **gradient reversal layer** (GRL) proposed by [9] for ConvCDAN optimization. GRL has no parameters. During forward propagation, GRL acts as an identity mapping, while it multiplies each gradient from the subsequent level by $-\lambda$ and passes the result to the preceding layer during backward propagation. As shown in Figure 3, we insert GRL between FeatureNet and DomainNet, based on which, we are able to optimize ConvCDAN using the standard SGD process.

In our implementation, we apply the Adam optimizer [19] to update parameters $\theta_f, \theta_y, \theta_d$ iteratively. The initial learning rate is set to 0.0001, and the batch size is set to 2048 by default to better optimize ranking loss between road pairs.

5 EXPERIMENTS

5.1 Experimental Settings

Datasets. We crawled Mobike data from three cities, Shanghai, Beijing and Ningbo in China during 01/07/2017 and 31/07/2017. Table 2 provides the details of Mobike data and multi-source data for each city. We realize the average distance between road intersections is around $500m$. Hence, we divide long roads into $< 500m$ segments in the experiments. We computed the average bike density value for each road in the cities with 1-day time window as target labels.

Compared Methods. Our ConvCDAN framework is flexible as many involving components can be removed or enhanced independently. In addition to compare ConvCDAN with its variants, we consider the following baselines:

Table 2: Details of the datasets

Data Type	City		
	Shanghai	Beijing	Ningbo
# of Mobikes	591,295	656,437	35,591
# of POIs	694,898	532,094	85,613
# of Satellite Light	28, 954	23,021	7,482
# of Roads	12,916	8,913	1,242
# of Subway Stations	366	334	53
# of Business Centers	28	26	17

- *Lasso Regression*. Linear regression method with $L1$ norm, performing both feature selection and regularization to enhance prediction accuracy.
- *GBRT*. It is a gradient boosting method to optimize the regression metric and effectively handle data in mixed types.
- *MLP*. A feed-forward network with four hidden layers each of which has 32 neurons and uses *RELU* as the activation function. The loss function is the *RMSE* metric.
- *RankNet* [3]. This has the same neural network structure as MLP, but the objective function is pair-wise ranking loss.
- *LambdaMart* [4]. A boosted tree version of LambdaRank [5], integrating the optimization of the ranking metric like *NDCG* into the framework of the gradient boosting.

Evaluation Metrics. For each city, we mark the top k roads with the largest bike density values as the ground truth hotspots set $H_k(R)$. The predicted hotspots set is denoted by $\hat{H}_k(R)$. Regarding the similarity between hotspots detection and document retrieval process, we adopt the MAP and NDCG [25] metrics to evaluate $\hat{H}_k(R)$ against the ground-truth $H_k(R)$. Formally, MAP@K and NDCG@K are defined as follows:

$$MAP@K = \frac{\sum_{i=1}^{K} P(i) \times rel(i)}{K} \tag{21}$$

$$DCG@K = rel(1) + \sum_{i=2}^{K} \frac{rel(i)}{\log_2 i} \tag{22}$$

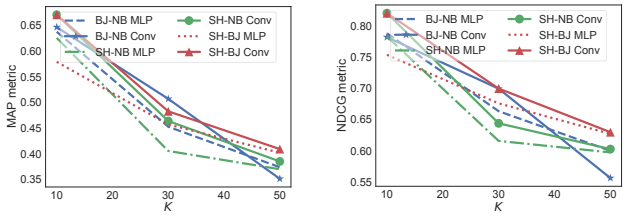
$$NDCG@K = \frac{DCG@K}{IDCG@K} \tag{23}$$

where $P(i)$ is the precision of top i detected roads, $rel(i)$ is an indicator function that equals to 1 if the road ranked at i is a real hotspot, $IDCG@K$ is the value of $DCG@K$ when the predicted bike density values of the roads are ranked perfectly.

5.2 Comparison Results

We mainly consider three detection tasks, namely Beijing → Ningbo, Shanghai → Ningbo, and Shanghai → Beijing, which is consistent with the order of the dockless bike deployment in China. We mark top 5% roads with the largest bike density values as real hotspots.

Table 3 shows the detection performance of various algorithms on the three cases. We find that the performances of all algorithms are consistent over two metrics: MAP@K and NDCG@K. ConvCDAN performs best in all the cases, i.e., reports highest MAP@K and NDCG@K values. Lasso regression reports the worse results



(a) MAP metric (b) NDCG metric

Figure 4: Effects of convolutional operations in FeatureNet

because it cannot capture the nonlinear interactions between features effectively. We observe that all the methods are able to identify at least 30% hotspots according to the MAP metric. This verifies the locations of the dockless shared bike hotspots are tightly related to the geographical information from multi-source data. Two neural methods MLP and RankNet perform better than others baselines, though LambdaMart is the gradient boosting version of the RankNet and integrates the ranking metrics such as NDCG into the optimization process. The reason may be that the neural network methods can capture complex non-linear relationships and are more robust for the distribution difference between city feature spaces. For instance, the introduction of the batch normalization layer helps alleviate the covariance shift problem.

On average, compared with the most competitive baseline method, ConvCDAN achieves 34%, 16.28% improvements on MAP@10 and NDCG@10, respectively. Similar improvements can be found in the other four metrics. An interesting observation is that ConvCDAN achieves the best performance in Beijing → Ningbo case, followed by Shanghai → Beijing and finally Shanghai → Ningbo. This is counterintuitive because Shanghai and Beijing are both tier-1 cities and should have similar feature distributions. The reason may be that when detecting hotspots in the target city, the road network complexity affects the detection results greatly. We notice the number of roads in Shanghai, Beijing and Ningbo are 12916, 8913 and 1242 respectively, and the sparsity of road networks in Beijing and Shanghai arises more difficulties in hotspots detection.

5.3 Effects of Convolutional Operations in G_f

We evaluate the effects of different designs for FeatureNet. To exclude the influence from ranking loss L_{rank} and DomainNet, we set $\alpha = 0$ and $\lambda = 0$ in the overall loss function (Equation 20). Figure 4(a) and 4(b) provide the results of using a simple two-layer feedforward neural network (MLP) and our convolutional design (Conv) for FeatureNet. In all the three cases, Conv achieves better performance than MLP, i.e., on average it reports 6.25% and 2.27% higher values in $MAP@K$ and $NDCG@K$ metrics, respectively. Due to the short-riding property of bike routes, the bike densities in neighbor roads typically vary smoothly and affect each other. Convolutional operations are effective in capturing such local interactions and lead to better detection results.

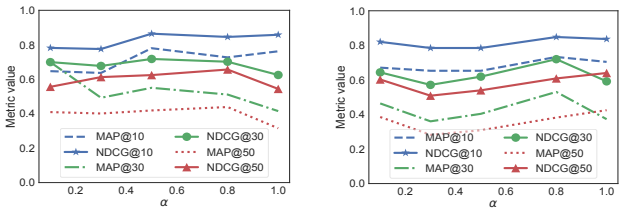
5.4 Effects of Ranking Loss for G_y

We now evaluate the effects of integrating ranking loss into our framework by varying the ranking loss weight α. For this experiment, we follow the proposed design of FeatureNet and remove

Table 3: Comparison results

City Pair	Method	MAP@10	NDCG@10	MAP@30	NDCG@30	MAP@50	NDCG@50
BJ->NB	Lasso	0.3393	0.5548	0.4326	0.6564	0.3433	0.5677
	GBRT	0.4422	0.6523	0.3438	0.5922	0.3497	0.5958
	MLP	0.6378	0.7878	0.4528	0.663	0.3739	0.5996
	RankNet	0.5975	0.7663	0.4247	0.6326	0.3839	0.6081
	LambdaMart	0.5	0.5701	0.4321	0.6031	0.4217	0.6432
	ConvCDAN	**0.8664**	**0.9266**	**0.58**	**0.756**	**0.4674**	**0.6683**
SH->NB	Lasso	0.3544	0.5739	0.3985	0.6375	0.3431	0.5821
	GBRT	0.5408	0.7314	0.4211	0.6420	0.3862	0.6163
	MLP	0.6259	0.7827	0.4056	0.6154	0.37	0.597
	RankNet	0.5809	0.7511	0.4299	0.6429	0.3579	0.5846
	LambdaMart	0.4655	0.6683	0.4269	0.6441	0.4361	0.6509
	ConvCDAN	**0.8**	**0.8701**	**0.5012**	**0.6912**	**0.4389**	**0.6615**
SH->BJ	Lasso	0.3272	0.4950	0.3251	0.4823	0.3041	0.4712
	GBRT	0.3972	0.5983	0.3028	0.4445	0.2979	0.4438
	MLP	0.5792	0.7534	0.4572	0.6756	0.4021	0.6267
	RankNet	0.5911	0.753	0.4578	0.6739	0.4115	0.6382
	LambdaMart	0.4342	0.5774	0.3948	0.5692	0.3183	0.5073
	ConvCDAN	**0.8154**	**0.9052**	**0.5329**	**0.7352**	**0.4407**	**0.6539**

DomainNet to exclude its influence on the performance. Figure 5(a) and 5(b) provide the results for Beijing→Ningbo and Shanghai→Ningbo cases, respectively. It is easy to see that better hotspots detection results can always be achieved by combining regression loss and ranking loss for model optimization, i.e., α is neither 0 nor 1. The best setting for α varies on different hotspots detection tasks. Similar results are observed for Shanghai → Beijing, and we omit the figure due to redundancy. The reasons may be: 1) combining the ranking loss into the objective function controls the importance of the regression task in hotspots detection; 2) optimizing the pair-wise ranking loss can reduce the number of reversely ordered pairs and improve the overall performance for hotspots detection.



(a) Varying α (BJ→NB) (b) Varying α (SH→NB)

Figure 5: Effects of ranking loss for DensityNet

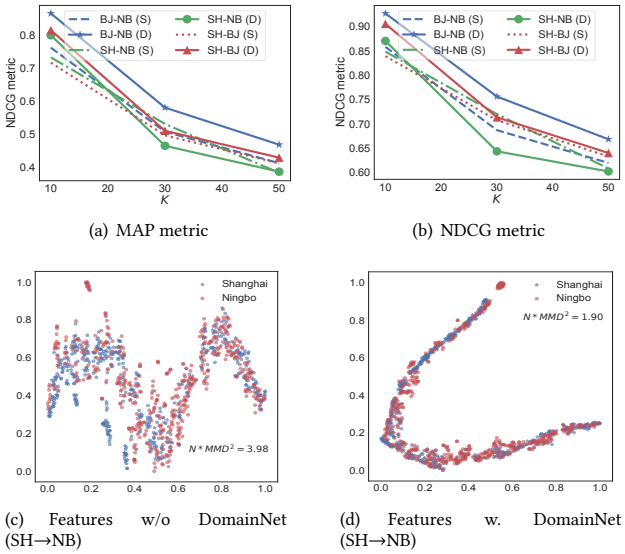
5.5 Effects of Integrating DomainNet G_d

At last, we discuss the effectiveness of DomainNet G_d in our framework. We adopt the proposed FeatureNet and select the best α for each detection task through the validation process. Figure 6(a) and 6(b) show that integrating DomainNet is effective to improve the detection performance in all cases. We can see the obvious performance gaps between ConvCDAN with DomainNet ("D") and ConvCDAN without DomainNet ("S"). To verify whether DomainNet can supervise FeatureNet to learn domain-invariant feature representations, we provide feature visualization results using the TSNE algorithm in Figure 6(c) and 6(d). Consider the detection task Shanghai → Ningbo. We observe that feature distributions across domains become more similar with DomainNet. We also measure feature distribution difference by the *MMD* metric and obtain 3.98 and 1.90 for w/o and with DomainNet, respectively. The visualization results are promising and verify the effectiveness of DomainNet in transferring hotspots knowledge learned from a city with deployed dockless shared bikes to a new city.

6 RELATED WORK

Researches on Bike Sharing Systems. Bike sharing systems have received increasing attention these years. A number of researches focused on the systems with docking stations, and studied the problems such as bike traffic prediction, station optimization, shared bike rebalancing, bike sharing business expansion and so on. For bike traffic prediction, many works organized the bike docking stations in a city into hierarchies via clustering methods and considered the temporal and weather factors to predict bike check-out/check-in traffic in different granularities. Among them, Yang et al. [37] proposed a spatio-temporal bicycle mobility model based on the historical data and predicted the station traffic with sub-hour granularity. Hoang et al. [14] decomposed the traffic flows into seasonal, trend and residual components and modeled each part as intrinsic Gaussian Markov random fields. Some works aim at solving the shared bike rebalancing problem that optimizes the route to move bikes among stations in order to match dynamic riding demands [22, 32]. A recent research [1] on dockless bike sharing system studied bike trajectories to learn people's daily riding patterns for planning bike lanes in the city. However, none of the existing works address the parking hotspots detection problem for dockless shared bikes, or discuss how to transfer knowledge learned from one city to another for bike sharing business expansion.

Transfer Learning on Domain Adaptation. Domain adaptation-[29] is transfer learning under the shift between training and test distributions. A rich line of approaches to domain adaptation aims



(a) MAP metric (b) NDCG metric

(c) Features w/o DomainNet (SH→NB) (d) Features w. DomainNet (SH→NB)

Figure 6: Effects of integrating DomainNet

to bridge the gap between the source and target domains by learning domain-invariant feature representations. Such deep representations allow the classifier learned from the source domain to be applied to the target domain effectively. Most works [23, 28, 36] focused on minimizing the maximum mean discrepancy (MMD) metric in the new feature space by matrix factorization or adding the regularization into the neural networks. Long et al. [24] argued that the source classifier and target classifier cannot be shared even if the distribution gap is reduced greatly with the new feature representations. Hence, instead of minimizing the distribution gap by MMD, they introduced the residual function to learn the difference between the source and target classifiers explicitly. Inspired by the idea of the adversarial learning [12], some latest works [6, 10] proposed discriminator network to supervise the learning of the domain-invariant features. However, all these methods are typically applied to the computer version and natural language processing applications [7, 15]. To the best of our knowledge, we are the first to apply the domain adaptation method for transferring hotspots knowledge from one city to another for dockless shared bikes. We also exploit multi-source data for discriminative feature learning.

7 CONCLUSION

In this paper, we study an important problem of detecting parking hotspots for dockless shared bikes in new cities. We propose to quantify road hotness based on a citywide bike probability density function learned by the Kernel Density Estimation. We conduct an empirical analysis on real shared bike data and summarize temporal and spatial properties of hotspots. Hotspots detection in a new city is challenging as no dockless shared bike has been deployed yet and transferring hotspots knowledge from one city to another suffers from the domain shift issue. To address the challenges, we identify the close relationships between the features extracted from multi-source urban data and bike density values of roads. We further introduce a novel Convolutional City Domain Adaptation Network named ConvCDAN to hotspots detection in the target city. ConvCDAN incorporates three sub-nets for feature learning, density prediction and ranking, and domain classification. Extensive experiments are performed using real data from Mobike and the results verify the effectiveness of ConvCDAN in hotspots detection in a new city, compared with various baseline methods.

ACKNOWLEDGMENTS

We thank anonymous reviewers for their insightful and helpful comments, which improve the paper. This work was supported in part by 973 Program (No. 2014CB340303), NSFC (No. 61772341, 61472254, 61572324, 61170238, 61602297 and 61472241) and the Shanghai Municipal Commission of Economy and Informatization (No. 201701052). This work was also supported by the Program for Changjiang Young Scholars in University of China, the Program for China Top Young Talents, and the Program for Shanghai Top Young Talents.

REFERENCES

[1] Jie Bao, Tianfu He, Sijie Ruan, Yanhua Li, and Yu Zheng. 2017. Planning Bike Lanes based on Sharing-Bikes' Trajectories. (2017).
[2] Glen Van Brummelen. 2012. *Heavenly Mathematics:The Forgotten Art of Spherical Trigonometry*. Princeton University Press. 1–2 pages.
[3] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd ICML*. ACM, 89–96.
[4] Christopher JC Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. *Learning* 11, 23-581 (2010), 81.
[5] Christopher J Burges, Robert Ragno, and Quoc V Le. 2007. Learning to rank with nonsmooth cost functions. In *NIPS*. 193–200.
[6] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Michael I Jordan. 2017. Partial Transfer Learning with Selective Adversarial Networks. (2017).
[7] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12, Aug (2011), 2493–2537.
[8] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *KDD*, Vol. 96. 226–231.
[9] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*. 1180–1189.
[10] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research* 17, 59 (2016), 1–35.
[11] Nicolas Gast, Guillaume Massonnet, Daniël Reijsbergen, and Mirco Tribastone. 2015. Probabilistic forecasts of bike-sharing systems for journey planning. In *Proceedings of the 24th CIKM*. ACM, 703–712.
[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.
[13] G. E. Hinton. 2008. Visualizing High-Dimensional Data Using t-SNE. *Vigiliae Christianae* 9, 2 (2008), 2579–2605.
[14] Minh X Hoang, Yu Zheng, and Ambuj K Singh. 2016. FCCF: forecasting citywide crowd flows based on big data. In *Proceedings of the 24th SIGSPATIAL*. ACM, 6.
[15] Judy Hoffman, Sergio Guadarrama, Eric S Tzeng, Ronghang Hu, Jeff Donahue, Ross Girshick, Trevor Darrell, and Kate Saenko. 2014. LSDA: Large scale detection through adaptation. In *NIPS*. 3536–3544.
[16] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*. 448–456.
[17] Neal Jean, Marshall Burke, Michael Xie, W Matthew Davis, David B Lobell, and Stefano Ermon. 2016. Combining satellite imagery and machine learning to predict poverty. *Science* 353, 6301 (2016), 790–794.
[18] KhosrowDehnad. 2012. Density Estimation for Statistics and Data Analysis. *Technometrics* 29, 4 (2012), 495–495.

[19] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[20] Min Lin, Qiang Chen, and Shuicheng Yan. 2013. Network In Network. *Computer Science* (2013).

[21] Junming Liu, Qiao Li, Meng Qu, Weiwei Chen, Jingyuan Yang, Hui Xiong, Hao Zhong, and Yanjie Fu. 2015. Station site optimization in bike sharing systems. In *Data Mining (ICDM), 2015 IEEE International Conference on*. IEEE, 883–888.

[22] Junming Liu, Leilei Sun, Weiwei Chen, and Hui Xiong. 2016. Rebalancing Bike Sharing Systems: A Multi-source Data Smart Optimization. In *Proceedings of the 22nd ACM SIGKDD*. ACM, 1005–1014.

[23] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. 2015. Learning transferable features with deep adaptation networks. In *ICML*. 97–105.

[24] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. 2016. Unsupervised domain adaptation with residual transfer networks. In *NIPS*. 136–144.

[25] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. 2008. *Introduction to information retrieval*. Vol. 1. Cambridge university press Cambridge.

[26] Luis M Martinez, Luís Caetano, Tomás Eiró, and Francisco Cruz. 2012. An optimisation algorithm to establish the location of stations of a mixed fleet biking system: an application to the city of Lisbon. *Procedia-Social and Behavioral Sciences* 54 (2012), 513–524.

[27] Sinno Jialin Pan, James T Kwok, and Qiang Yang. 2008. Transfer Learning via Dimensionality Reduction.. In *AAAI*, Vol. 8. 677–682.

[28] Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, and Qiang Yang. 2011. Domain Adaptation via Transfer Component Analysis. *IEEE Transactions on Neural Networks* 22, 2 (2011), 199–210.

[29] Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2010), 1345–1359.

[30] David Sculley. 2010. Combined regression and ranking. In *Proceedings of the 16th ACM SIGKDD*. ACM, 979–988.

[31] Simon J Sheather and Michael C Jones. 1991. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B (Methodological)* (1991), 683–690.

[32] Adish Singla, Marco Santoni, Gábor Bartók, Pratik Mukerji, Moritz Meenen, and Andreas Krause. 2015. Incentivizing Users for Balancing Bike Sharing Systems.. In *AAAI*. 723–729.

[33] Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. 2007. A Hilbert space embedding for distributions. In *International Conference on Algorithmic Learning Theory*. Springer, 13–31.

[34] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[35] Michelle Toh. 2017. China's bike-sharing companies have hit a roadblock. http://money.cnn.com/2017/12/29/investing/china-bike-sharing-boom-bust/index.html?from=timeline. (2017).

[36] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. 2014. Deep domain confusion: Maximizing for domain invariance. (2014).

[37] Zidong Yang, Ji Hu, Yuanchao Shu, Peng Cheng, Jiming Chen, and Thomas Moscibroda. 2016. Mobility Modeling and Prediction in Bike-Sharing Systems. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 165–178.

[38] Ming Zeng, Tong Yu, Xiao Wang, Vincent Su, Le T Nguyen, et al. 2016. Improving Demand Prediction in Bike Sharing System by Learning Global Features. (2016).