

Research Article

ELM Meets Urban Big Data Analysis: Case Studies

Ningyu Zhang,¹ Huajun Chen,² Xi Chen,² and Jiaoyan Chen²

¹College of Computer Science and Technology, Zhejiang University, Zhejiang 310014, China

²Zhejiang University, Zhejiang, China

Correspondence should be addressed to Ningyu Zhang; zhangningyu@zju.edu.cn

Received 10 May 2016; Revised 3 July 2016; Accepted 1 August 2016

Academic Editor: Adel Alimi

Copyright © 2016 Ningyu Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the latest years, the rapid progress of urban computing has engendered big issues, which creates both opportunities and challenges. The heterogeneous and big volume of data and the big difference between physical and virtual worlds have resulted in lots of problems in quickly solving practical problems in urban computing. In this paper, we propose a general application framework of ELM for urban computing. We present several real case studies of the framework like smog-related health hazard prediction and optimal retain store placement. Experiments involving urban data in China show the efficiency, accuracy, and flexibility of our proposed framework.

1. Introduction

Urban computing is a process of acquisition, integration, and analysis of big and heterogeneous data generated by diverse sources in urban spaces, such as sensors, devices, vehicles, buildings, and humans, to tackle the major issues that cities face (e.g., air pollution, increased energy consumption, and traffic congestion) [1]. With the rapid progress of urban computing, lots of applications have appeared to analyze amounts of urban data using big data analysis technologies for real practical problems. Most of them use conventional machine learning and computational intelligence solutions. However, with the exponential growth of data and complexity of systems, those methods have suffered lots of bottlenecks in learning (e.g., intensive human intervention and convergence time).

In fact, the urban data acquired in real situations normally are heterogeneous and in big volumes. Thus, according to studies [2], treating features extracted from different data sources equally does not achieve the best performance. Moreover, most of the models of applications are hard to implement in real situations because of the complexity and sparsity of some data. Therefore, fast machine learning and computational intelligence techniques are needed.

Extreme learning machine (ELM) [3] can be a good solution for the learning of large sample data as it has high

generalization performance and fast training speed. Moreover, the stacked ELMs [4] perform well when facing extremely large and complex training datasets. Indeed, ELM is flexible and easy to implement. In this paper, we propose a general application framework of ELM for urban computing.

To solve challenges in urban computing using ELM, we firstly analyze the types, sources, and structures of urban data. We standardize the form of data from different sources and fuse data across modalities as input of ELM. In fact, the urban data can be obtained from online social media and offline physical sensors. Online social media (twitter, user comments) have opinions about places. These data are normally text (user ratings are numeric) and uncertain. The bad situation is that the amounts of these data are influenced by the populations or type of district to a great extent. For instance, it is quite easy to retrieve amounts of data in a metropolis. However, less-developed regions have small populations and, hence, relatively low social media activity. Meanwhile, offline data reflect the physical condition of this region such as the flow of taxis and buses, traffic congestion index, real estate, POIs, road network, air quality, and meteorological elements. The amounts of these data in different regions are usually similar. Nevertheless, these data are from different sources and are heterogeneous. We divide the city into regions by the road network. We obtain huge amounts

of data in each region. We formalize these data and build standard feature vectors including social view from online social media and physical view from physical sensors through location based feature selection. Social view and physical view are treated as different views on an object. We adopt deep autoencoder method [5] to capture the middle-level feature representation between two modalities (e.g., social view and physical view). Then, we train the data flexibly, which means that the data with different sparse degree will be trained by different kinds of ELM based methods.

Furthermore, we propose several case studies of our framework. For different real applications, we can easily add or remove data sources and adjust parameters of the model. We use the social media, meteorological elements, and air quality to make smog-related health hazard prediction. Truly, these data are heterogeneous, while stacked ELM [4] can flexibly train these data. We use user comments, social media, the flow of taxis and buses, traffic congestion index, real estate, POIs, and road network to help optimally retain store placement. In real situations, the user comments data suffer the data sparsity problem. We handle this by transferring knowledge from regions with huge data by domain adaptation transfer ELM [6].

The major contributions of this paper are as follows:

- (1) We propose a general application framework of ELM for urban computing.
- (2) We propose several case studies for our framework such as smog-related health hazard prediction and optimal retain store placement.
- (3) We evaluate our framework in real urban datasets and reveal the advantages in precision and learning speed.

The remainder of this paper is organized as follows. Section 2 briefly reviews existing studies on ELM and urban computing. Section 3 presents the framework and key challenges. Section 4 describes the case studies. Section 5 presents the results of our experiments. Finally, Section 6 summarizes our findings and concludes the paper with a brief discussion on the scope for future work.

2. Related Work

2.1. ELM. Extreme learning machine (ELM) has recently attracted many researchers' interest due to its very fast learning speed, good generalization ability, and ease of implementation [3]. Zhou et al. [4] proposed stacked ELMs (S-ELMs) which are specially designed for solving large and complex data problems. The S-ELMs serially connect stacked small ELMs divided by a single large ELM network and can approximate a very large ELM network with small memory requirement. Moreover, the S-ELMs support the ELM autoencoder during each iteration which significantly improves the accuracy on big data problems. Therefore, it has great chance to improve efficiency and precision using S-ELMs to train heterogeneous data. However, it is necessary to have a standard data modal for it.

With the rapid growth of cities, huge amounts of data can be obtained, while there still exist data sparsity problems as

a result of sensor coverage, difference of human activities, and so on. L. Zhang and D. Zhang [6] proposed a unified framework, referred to as Domain Adaptation Extreme Learning Machine (DAELM), which learns a robust classifier by leveraging a limited number of labeled data from target domain for drift compensation as well as gases recognition in E-nose systems, without loss of the computational efficiency and learning ability of traditional ELM. Nevertheless, different urban data have different modalities and are difficult to transfer knowledge directly.

2.2. Cross-Domain Data Fusion. The data from different domains consist of multiple modalities. According to [7], there are three categories of methods that can fuse multiple datasets: (1) stage-based fusion methods [8], (2) feature-level based methods [5, 9], and (3) semantic meaning-based methods [10–12]. Ngiam et al. [5] proposed deep autoencoder method to capture the middle-level feature representation between two modalities (e.g., audio and video).

However, there is a lack of research which utilized methods to fuse both social media data and physical sensor data for a general application framework for urban computing. This is mainly due to the lack of (1) systematic approaches for collecting, modeling, and analyzing such information and (2) efficient machine learning framework which can combine features from both social and physical views.

2.3. Urban Computing. The urban data of a city (e.g., human mobility and the number of changes in a POI category) may indicate real conditions and trends in the cities [13].

For instance, the tweets from social media and meteorological elements may indicate the existence of smog-related health hazards. Chen et al. [14] modeled smog-related health hazards and smog severity through mining raw microblogging text and network information diffusion data and developed an artificial neural network- (ANN-) based model to forecast smog-related health hazard with the current health hazard and smog severity observations. This paper is a follow-up work and as social media and physical sensors are different modalities, we adopt urban knowledge fusion and propose a general framework.

Furthermore, human mobility with POIs may have contributions to the placement of some businesses. Scellato et al. [15] used the location based social network to study the optimal retail store problem. They collected and analyzed data from Foursquare to understand the popularity of the three major retail stores in New York. They evaluated a diverse set of data mining features and analyzed the spatial and semantic information about location and mode of user action in the surrounding area. The problem arises when we cannot obtain sufficient online data in some small cities. This paper is also follow-up work of [16].

3. Approach

3.1. Key Challenges. According to [1], urban computing has three main challenges: urban sensing and data acquisition, computing with heterogeneous data, and hybrid systems blending the physical and virtual worlds. In real situations,

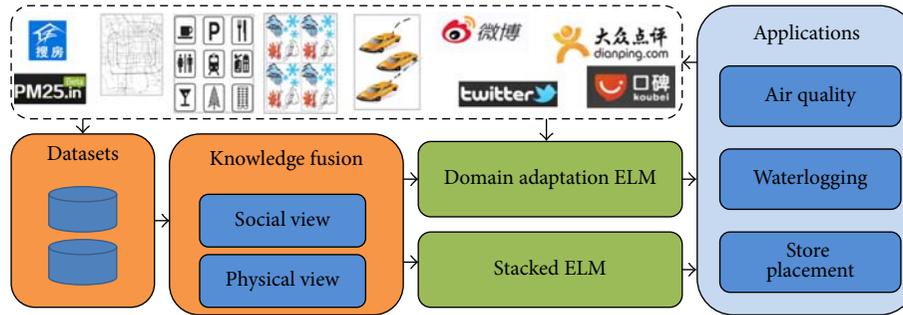


FIGURE 1: General application framework of ELM for urban computing.

many conventional machine learning methods suffer key challenges summarized as follows.

(1) *Poor Learning Efficiency.* As there are huge urban data generated from social media and physical sensors, it is great challenge for the training efficiency of machine learning. Seriously, many applications of urban computing have timeliness requirements such as disaster and traffic monitor. It is very urgent to train a model as fast as possible and get the final result.

(2) *Complex Model to Handle Heterogeneous Data.* Data from different sources consist of multiple modalities, each of which has different representation, distribution, scale, and density. For example, text is usually represented as discrete sparse word count vectors, whereas an image is represented by pixel intensities or outputs of feature extractors which are real-valued and dense. POIs are represented by spatial points associated with a static category, whereas air quality is represented using a geotagged time series. Human mobility data is represented by trajectories, whereas a road network is denoted as a spatial graph. Treating different datasets equally or simply concatenating the features from disparate datasets cannot achieve a good performance in data mining tasks. As a result, fusing data across modalities becomes a new challenge in big data research, calling for advanced data fusion technology.

(3) *Sparsity of Data and Hard for Implementation.* It is quite easy to retrieve kinds of data in a metropolis. Nevertheless, some relatively small cities have small populations and, hence, relatively low social media activity. Specifically, we face the following two challenges: (1) the label scarcity problem and (2) the data insufficiency problem. Therefore, it is difficult to train and implement in real situations.

3.2. *General Framework.* As Figure 1 shows, our framework consists of two parts: urban knowledge fusion and training which is based on ELM. We firstly fuse the data obtained from both social media and physical sensors using deep autoencoder [5]. For those with abundant data, we use stacked ELM [4] to train. For those with sparse data, we use DAELM [6] and transfer knowledge from regions with abundant data.



FIGURE 2: Segmented regions.

3.3. Typical Technologies

3.3.1. *Map Segmentation.* We divide a city into disjointed blocks (<https://github.com/zxlzr/Segment-Maps>), assuming that placement in a block g is uniform. Road network is usually composed of a number of major roads, such as the ring road; the city is divided into areas [13]. We map the projection of the vector-based road network on a plane. Then, the road network is converted into a raster model [17]. Actually, each pixel of a projected map image can be viewed as a block element of a raster map. Consequently, the road network is converted into a binary image. Then, we extract the skeleton of the road, while retaining the original two-value image topology. Figure 2 shows the result of the procedure described above for Beijing's road network. Finally, we get the blocks g of cities.

3.4. *Location Based Feature Selection.* For each block, we can access the features over its neighbourhood and simply calculate the average value. However, there exist problems if ignoring the location information of the surrounding features. There are two reasons we need to consider. (1) From the distance perspective, if the selected location is far from the target location, it may have few impacts on the target location and vice versa. (2) From the density perspective, if one of the neighbour blocks has a lot of stores while the others have few, maybe this block has greater influence on the target location. So we propose a location based feature selection (LBFS)

method to calculate features considering neighbourhood's impact. Suppose we want to calculate a feature of target block g ; g has t neighbours N_1, N_2, \dots, N_t , f_i is the feature vector of each block N_i , d_i is the distance between N_i and g , n_i is the number of feature points (which means there are n stores in block i , and m_i is the measure of N_i).

(1) *Distance Related Features*. Some kind of features may reduce the impact of the store's score with the increase of distance. So we weigh the features with distances. Formally, we have

$$f_{g1} = \sum \frac{f_i}{d_i}. \quad (1)$$

(2) *Density Related Features*. Several features may have fewer impacts on the store's score with the increase of density of surrounding shops. So we weigh the features with density. Formally, we have

$$f_{g2} = \sum \frac{f_i}{n_i}. \quad (2)$$

(3) *Measure Related Features*. There may exist some features that may have fewer impacts on the store's score with the increase of measure of surrounding regions. So we weigh the features with measure. Formally, we have

$$f_{g3} = \sum \frac{f_i}{m_i}. \quad (3)$$

In the real situation, we use all three kinds of feature selection results. Normally each kind of feature in a grid has three final feature values as a vector. Formally, we have

$$f_g = \text{LBFS}(f, d, n, m), \quad (4)$$

where f are the feature vectors of the neighbours, d are the distances between the middle of neighbours and middle of the target block, n are the numbers of feature points in neighbours, and m are the measures of neighbours.

3.4.1. Urban Knowledge Fusion. For each block in cities we obtain the social view and physical view separately. Each view is represented as a feature vector. Social view is composed of social media text, user comment texts, user ratios, and so on according to different application requirements. Physical view is composed of physical sensor values like the flow of taxis and buses, traffic congestion index, real estate, POIs, road network, air quality, meteorological elements, and so on regarding different application requirements. We adopt the deep autoencoder [5] to capture the middle-level feature representation between social view and physical view. As Figure 3 depicts, deep learning effectively learns (1) a better single modality representation with the help of other modalities and (2) the shared representations capturing the correlations across multiple modalities.

3.4.2. Training. Practically, we obtain the middle-level feature representation of each block. For those with abundant

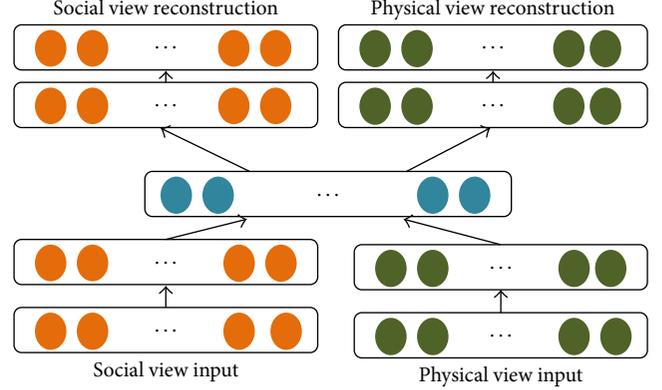


FIGURE 3: Deep autoencoder of social view and physical view.

data, we use stacked ELM [4] to train. It can be used directly to solve regression, binary, and multiclass classification problems regarding different applications. For those with sparse data, we use DAELM [6] to transfer knowledge from regions with abundant data. Actually, we can treat different cities as different domains because data from different cities may have different distributions in feature and label spaces [18]. We use cities with abundant data as sources and transfer knowledge to target ones with sparse data.

4. Case Studies

4.1. Smog-Related Health Hazard Prediction. In fact, smog is a terrible health hazard that affect people's health according to recent research [19]. It is necessary to analyze, monitor, and forecast smog-related health hazards in a timely manner. In recent times, social media has become an increasingly important channel to observe sentiment, trends, and events. Furthermore, there are various physical sensors monitoring smog status, such as air quality stations, weather stations, and earth observation satellites, generating amounts of data about severity of smog. In this paper, we model the smog-related health hazards and smog severity by two indexes (PHI and SSI) using social media. The urban smog-related health hazard prediction problem is a classification problem to assign appropriate class labels (Public Health Index) to the blocks of cities.

Public Health Index (PHI) is the sum of total relative frequencies of smog-related health hazard phrases in the current tweets. D-PHI is an enhanced Public Health Index that includes consideration of diffusion in social networks.

Definition. Smog Severity Index (SSI) is the weighted sum of total relative frequencies of smog severity phrases in the current tweets. D-SSI is an enhanced Smog Severity Index that includes consideration of diffusion in social networks.

Firstly, we extract both smog-related health hazard phrases and smog severity phrases. Secondly, we gather raw tweets with time and location tags from Weibo (a twitter-like

website in China). Thirdly, we calculate the daily relative frequency rf of each phrase:

$$\begin{aligned} rf(p) &= af(p, T_C) \times idf(p, T_H), \\ af(p, T_C) &= \frac{\sum_{d \in T_C} f(p, d)}{T_C}, \\ idf(p, T_H) &= \log \frac{|T_H|}{|\{d \in T_H : p \in d\}|}, \end{aligned} \quad (5)$$

where T_H and T_C represent historical and current tweet sets, respectively, p represents a phrase, d represents a tweet, $f(p, d)$ represents the frequency of phrase p in tweet d , $af(p, T_C)$ represents the average frequency of phrase p in the current tweet set T_C , and $idf(p, T_H)$ represents the inversed document frequency of the tweets with phrase p in the historical tweet set T_H . The logarithm function is to scale up the fraction of rare tweets. The above algorithm is derived from the typical tf-idf algorithm [20]. The difference lies in the replacement of the largest word frequency in current tweet set with the size of current tweet set, which aims at eliminating the influence of other heat phrases. Then, PHI and SSI are calculated with the relative frequencies of all the phrases:

$$\begin{aligned} PHI &= \sum_{p \in P_1} rf(p), \\ SSI &= \sum_{p \in P_2} rf(p) \times order(p), \end{aligned} \quad (6)$$

where P_1 stands for the set of smog-related health hazard phrases, P_2 stands for the set of smog severity phrases. Then, social network diffusion is considered to calculate D-PHI and D-SSI. We calculate the network diffusion-based average frequency:

$$daf(p, T_C) = \frac{\sum_{d \in T_C} (f(p, d) \times (g(d) + 1))}{|T_C| + \sum_{d \in T_C} g(d)}, \quad (7)$$

where $g(d)$ represents a tweet's total number of retweets and likes. Once daf is calculated, we use it to replace the average frequency af to calculate the relative frequency rf and further compute the value of D-PHI and D-SSI. Finally, the PHI, SSI, D-PHI, and D-SSI make up the feature vectors of social view in this problem.

Moreover, we also extract features from air quality, including both air pollution concentrations (CO, NO₂, SO₂, O₃, PM_{2.5}, and PM₁₀) and air quality index (AQI) which comprehensively evaluates the air quality. We extract records from various meteorological elements, including humidity, cloud value, pressure, temperature, and wind speed, all of which have been proven to affect smog disasters greatly. For example, high wind speed and low cloud value usually leads smog pollution to decrease in the next day.

4.2. Optimal Retain Store Placement. The optimal placement of a retail store has been of prime importance. For example,

a new restaurant set up in a street corner may attract lots of customers, but it may close months later if it is located in a few hundred meters down the road. In this paper, the optimal retain store placement problem is a rank problem. We calculate scores for each of the candidate areas and rank them. The top-ranked areas will be the optimal region for placing. We get the label data from Dianping score (<http://www.dianping.com/>) and assume that the data observed can evaluate the popularity of a place. For this problem, we analyze the data obtained and build social and physical view and then build a classifier according to our framework.

A strong regional economy usually indicates high demand according to recent studies [15, 21]. Therefore, we mined the block's neighbour's user reviews from <http://www.dianping.com/>.

(1) Dianping Score. For each region g , we will retrieve service quality, overall satisfaction, environment class, and consumption level by mining the reviews of business venues neighbourhood regions. Region g has t neighbourhoods. We access the users' opinions over the neighbourhood region N_i to form features.

Overall Satisfaction. Since the overall rating of a business venue in block g represents the satisfaction of users, we use the LBFS to get the overall ratings of all business venues located in N_i as a numeric score of overall satisfaction. Formally, we have

$$F_s = \text{LBFS}(\text{overall satisfaction}, d, n, m). \quad (8)$$

Service Quality. Similarly, we use the LBFS to compute the service rating of business venues in N_i and express the service quality of the neighbourhood as

$$F_q = \text{LBFS}(\text{Service Rating}, d, n, m). \quad (9)$$

Environment Class. The level of the surrounding environment can reflect the current environment class of the area. Therefore, we use LBFS to get environment rating as

$$F_e = \text{LBFS}(\text{environment rating}, d, n, m). \quad (10)$$

Consumption Cost. The average value of the consumer's behavior of the surrounding regions can reflect the current consumption of the area. So we use LBFS to get the consumption cost of business venues of neighbourhood as a feature:

$$F_c = \text{LBFS}(\text{average cost}, d, n, m). \quad (11)$$

Bus transits are slow and cheap and are mainly distributed in areas having a large number of IT and educational establishments. The price of real estate and the traffic congestion index indicate whether the facility planning is balanced. We exploit these features to uncover the implicit preferences for a neighbourhood.

Bus-Related Features. Medium income residents choose bus. Because most of the city's residents are from the background of the middle class, bus traffic may represent the bulk of

the city's flow. We try to measure the arrival, departure, and the volume of the transition on the streets of each block. For a region g , we try to use BT as the set of bus trajectories of a city, each of which is denoted by a tuple $\langle p, d \rangle$, where p is a pickup bus stop and d is a drop-off bus stop [21].

Bus Arriving, Departing, and Transition Volume. We extract the arriving, departing, and transition volumes of buses from smart card transactions. Formally,

$$\begin{aligned} F_b^{\text{BAV}} &= |\langle p, d \rangle \in \text{BT} : p \notin g, d \in g|, \\ F_b^{\text{BLV}} &= |\langle p, d \rangle \in \text{BT} : p \in g, d \notin g|, \\ F_b^{\text{BTV}} &= |\langle p, d \rangle \in \text{BT} : p \in g, d \in g|. \end{aligned} \quad (12)$$

Density. Recent studies have reported that price premiums of up to 10% are estimated for retail stores within 400 m of a large number of bus stops. The density of the bus station is positively correlated with the value of the retail store. Here, we use smart card transactions and propose alternative methods and strategies for density estimation of bus stop. In fact, the number of bus stops in a trip can be approximated by fare. Therefore, we calculate the distance to the ratio of the estimated density in the vicinity of the bus station.

$$F_i^{\text{BSD}} = \frac{\sum_{p \in g, d \in g} \text{dist}(p, d) / \text{fare}(p, d)}{|\langle p, d \rangle \in \text{BT} : p \in g, d \in g|}. \quad (13)$$

Balance. The balance of smart cards can show the pattern of consumption and supply behavior. If some residents always maintain a high balance on the smart card, the huge cost of bus travel may mean (1) these residents are more dependent on the bus, which shows the lack of subway and taxi nearby, and (2) these residents have to go to work in places far away. In other words, these places may be far away. So, we try to use the smart card balance as a feature:

$$F_i^{\text{SCB}} = \frac{\sum_{p \in g, d \in g} \text{balance}(p, d)}{|\langle p, d \rangle \in \text{BT} : p \in g, d \in g|}. \quad (14)$$

Real Estate Features. The real estate prices may reflect the purchasing power and economic index of this region. First, we collect the historical prices of each estate, and we use LBFS to calculate estate price of the neighbourhood of each block. Formally, we have

$$F_i^{\text{RE}} = \text{LBFS}(\text{estate price}, d, n, m). \quad (15)$$

Traffic Index Features. The flexibility of a region like convenient traffic may contribute to the popularity of a region. We obtain the traffic index from <http://nitrafficindex.com/>. Formally, we have

$$F_i^{\text{TI}} = \text{LBFS}(\text{traffic index}, d, n, m). \quad (16)$$

Intraclass Competitiveness. We measure the proportion of neighbouring places of the same type γ_l with respect to the total number of nearby places. Then, we rank areas in reverse

order, assuming that the least competitive area is the most promising one:

$$F_i^{\text{Intra}} = -\frac{N_{\gamma_l}(l, r)}{N(l, r)}. \quad (17)$$

However, it is worth noting that the retail industry's competitive stores and marketing can have a positive or negative impact. For example, one would expect to place a bar in a region with plenty of nightlife locations, because there already exists a service system, and there are a lot of people attracted to the area. However, being surrounded by competitors also means to share customers.

Interclass Competitiveness. In order to consider the iterations between different place categories, we adopt the metrics defined by Jensen [22]. In practice, we use the intercategory coefficients described to weigh the desirability of the places observed in the area around the object; that is, the greater the number of the places in the area that attracts the object, the better the quality of the location. More formally, we define the quality of location for a venue of type γ_l as

$$F_i^{\text{Inter}} = \sum_{\gamma_p \in \Gamma} \log(\chi_{\gamma_p, \gamma_l}) \times (N_{\gamma_p}(l, r) - \overline{N_{\gamma_p}(l, r)}), \quad (18)$$

where $\overline{N_{\gamma_p}(l, r)}$ means how many venues of type γ_p are observed on average around the places of type γ_l , Γ is the set of place types, and $\chi_{\gamma_p, \gamma_l}$ are the intertype attractiveness coefficients. Formally, we get

$$\chi_{\gamma_p, \gamma_l} = \frac{N - N_{\gamma_p}}{N_{\gamma_p} \times N_{\gamma_l}} \sum_p \frac{N_{\gamma_l}(p, r)}{N(p, r) - N_{\gamma_p}}. \quad (19)$$

POIs. The POIs indicate the latent patterns of this region which may have contributed to the placement. Moreover, the category of a POI may have a causal relation to it. Let $\#(i, c)$ denote the number of POIs of category $c \in C$ located in g_i , and let $\#(i)$ be the total number of POIs of all categories located in g_i . The entropy is defined as

$$F_i^{\text{POI}} = -\sum_{c \in C} \frac{\#(i, c)}{\#(i)} \times \log \frac{\#(i, c)}{\#(i)}. \quad (20)$$

Business Areas. Business areas have important influence on optimal placement. The locations will get more customers and resources in the business areas. We retrieve the geographic information of business areas from Baidu map api. Finally we build a feature related to business areas. Formally, we have

$$F_{d1} = \text{LBFS}(\text{is neighbour business areas}, d, n, m),$$

$$F_{d2} = \begin{cases} 1 & \text{if } g_i \in \text{Business Area} \\ 0 & \text{if } g_i \notin \text{Business Area}, \end{cases} \quad (21)$$

$$F_d = [F_{d1}, F_{d2}],$$

where "is_neighbour_business_areas" is a Boolean value which means whether the block's neighbour is a business area.

TABLE 1: Details of the datasets.

Datasets	Size (M)	Sources
Comments	2,523	http://dianping.com/
Tweets	11,023	http://weibo.com/ , http://twitter.com/
Buses	254	http://chelaile.net.cn/
Traffic	119	http://www.nitrafficindex.com/
Real estate	35	http://www.soufun.com/
Air	534	http://www.pm25.in/
POI, business areas	10	http://map.baidu.com/
Road network	9	http://openstreetmap.org/
Meteorological	98	http://forecast.io/ , http://noaa.gov/

5. Experiments

5.1. Urban Data. Table 1 lists the data sources. For social view, we crawl online business reviews from <http://www.dianping.com/>, which is a site for reviewing business establishments in China. Each review includes the shop ID, name, address, latitude, longitude, consumption cost, star rating, POI category, city, environment, service, overall ratings, and comments. We also crawl huge data from Weibo (a twitter-like website in China) and Twitter with geotags. For physical view, we extract features from smart card transactions from open web API. Moreover, we crawl the traffic index from <http://www.nitrafficindex.com/>, which is open to the public. Finally, we crawl the estate data from <http://www.soufun.com/>, which is the largest real estate online system in China. Moreover, we crawl huge air quality data from <http://www.pm25.in/>. We collected POI and business areas data from Baidu maps for each city. The road network data was gathered from OpenStreetMaps. We collected meteorological data from the National Oceanic and Atmospheric Administration’s (NOAA) web service and <http://forecast.io/> every hour. All the experiments can be obtained from open web service. The source code of this paper can be obtained from <https://github.com/zxlzr/ELM-for-Urban-Computing>.

5.2. Metrics. In total, we have two main metrics for our framework, the precision and efficiency. For specific case studies, for smog-related health hazard prediction, we have the following.

Precision and Recall. The precision and recall are given as follows:

$$\begin{aligned} \text{Precision} &= \frac{|\cap(\text{prediction set}, \text{reference set})|}{|\text{prediction set}|}, \\ \text{Recall} &= \frac{|\cap(\text{prediction set}, \text{reference set})|}{|\text{reference set}|}. \end{aligned} \quad (22)$$

For optimal retain store placement, we have the following.

Normalized Discounted Cumulative Gain. The discounted cumulative gain (DCG@N) is given by

$$\text{DCG}[n] = \begin{cases} \text{rel}_1 & \text{if } n = 1 \\ \text{DCG}[n-1] + \frac{\text{rel}_n}{\log_2 n} & \text{if } n \geq 2. \end{cases} \quad (23)$$

Later, given the ideal discounted cumulative gain DCG' , NDCG at the n th position can be computed as $\text{NDCG}[n] = \text{DCG}[n]/\text{DCG}'[n]$. The larger the value of $\text{NDCG}@N$, the higher the top- N ranking accuracy.

Precision and Recall. We choose to use a four-level rating system as $(3 > 2 > 1 > 0)$. To simplify our evaluation task, we treat the ratings less than 2 as low values and ratings of 3 as high values. In a top- N block list E_N sorted in descending order of the prediction values, we define the precision and recall as $\text{precision}@N = (E_N \cap E_{\geq 2})/N$ and $\text{recall}@N = (E_N \cap E_{\geq 2})/E_{\geq 2}$, where $E_{\geq 2}$ are blocks whose ratings are greater than or equal to 2. For efficiency, we recorded the training time and compare with baselines.

5.3. Results. For smog-related health hazard prediction, the stacked ELM is trained hidden layer ANN with 10 to 40 hidden nodes; the BP is trained ANN with 2 to 3 hidden layers and 8 to 15 nodes in each hidden layer. Two classic SVM regression methods, nu-SVR and epsilon-SVR, are provided by LIBSVM [23]. Random forest regression method is provided by sklearn. Meanwhile, The accuracies of the health hazard prediction models using our framework, nu-SVR, epsilon-SVR, and random forest, are shown in Table 2. $p(\text{nf})$ means the precision of methods without knowledge fusion. We can find that two ANNs’ methods outperform the SVM regression methods and the random forest regression method in forecasting the next day PHI, and the ELM achieves slightly higher prediction accuracy than the multiple hidden layers ANNs trained by BP.

For optimal retain store placement, the models trained with a single city’s data are used as baselines. Our datasets have the data obtained from five cities in China. The blocks in each city have a label $\{0, 1, 2, 3\}$ from <http://www.dianping.com/> based on the values observed by users. For example, we have “2,31.” “2” means the id of city, “13” means the id of block, and the final “2” is the label. We treat each city as a single domain, containing hundreds of blocks. For example, if we want to do optimal retain store placement in Hangzhou, we use 75% of data in Hangzhou and all the other cities’ data to train. The remaining 25% of data in Hangzhou are for testing. Actually, Hangzhou is treated as target domain, while the other cities are source domains.

Figure 4 shows the results of concatenation, knowledge fusion, and sampling methods for Starbucks. The concatenation method only concatenates social view and physical view into one single view to adapt to the learning setting. The results show that knowledge fusion works better. The sampling method means we manually filter some negative samples based on knowledge fusion, which will make better results as figure shows.

TABLE 2: The results of smog-related health hazard prediction.

Cities	Stacked ELM			BP			nu-SVR			Epsilon-SVR			Random forest		
	p (nf)	p	Time	p (nf)	p	Time	p (nf)	p	Time	p (nf)	p	Time	p (nf)	p	Time
Beijing	0.65	0.80	3	0.66	0.79	15	0.78	0.69	9	0.56	0.53	9	0.10	0.15	7
Tianjin	0.92	0.89	3	0.56	0.59	13	0.58	0.73	10	0.66	0.73	11	0.15	0.14	8
Shanghai	0.53	0.88	4	0.55	0.69	16	0.55	0.64	11	0.53	0.53	10	0.12	0.30	8
Hangzhou	0.31	0.86	3	0.63	0.65	10	0.54	0.55	9	0.66	0.83	9	0.15	0.34	7
Guangzhou	0.53	0.54	5	0.65	0.69	15	0.53	0.66	11	0.48	0.53	21	0.10	0.34	8
Average	0.75	0.73	4	0.56	0.56	15	0.59	0.65	10	0.59	0.73	10	0.20	0.34	7

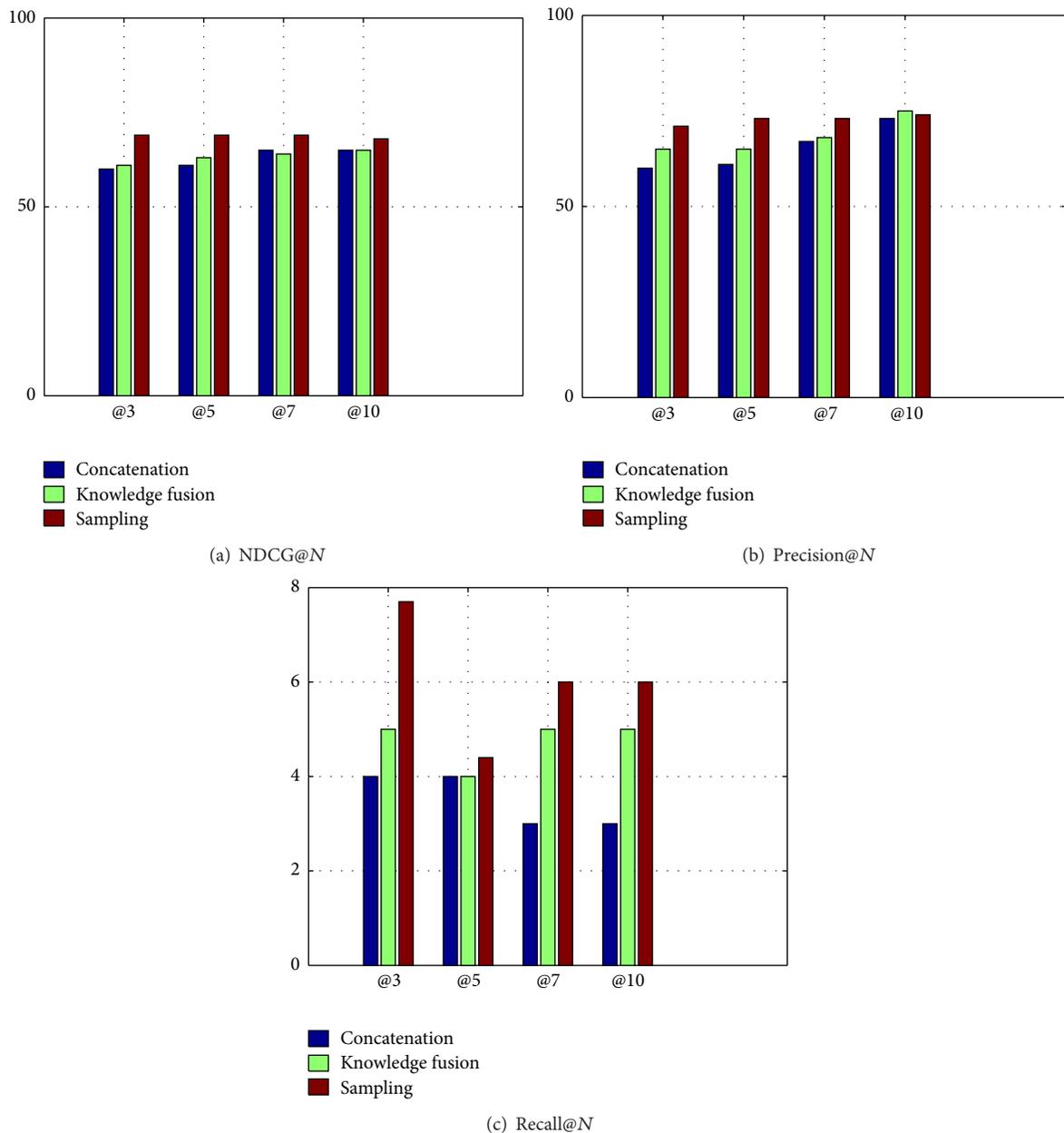


FIGURE 4: NDCG, precision, and recall of @N for Starbucks in Beijing.

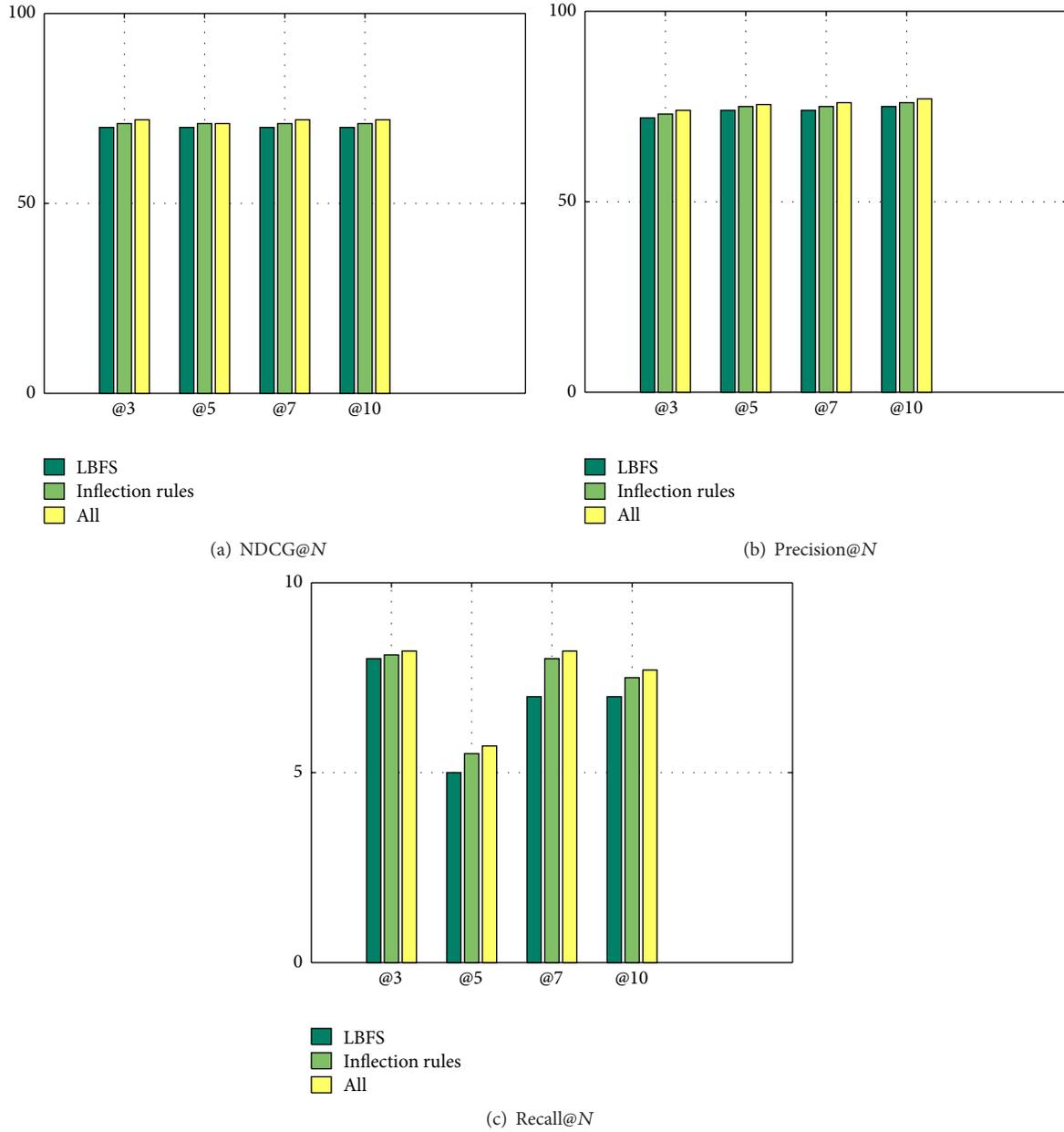


FIGURE 5: NDCG, precision, and recall of @N for Starbucks in Beijing.

Figure 5 shows the results of LBFS, inflection rules, and all methods for Starbucks. The LBFS method means we use location based feature selection to build features while not using the average value based on the methods in the past paragraph. Actually, LBFS works better. The inflection rules method means we use rules to calculate the final score if the area has spatial news such as “new subway, demolition.” The inflection rules consider the situations that our algorithm may not cover and make our method more robust. The all method includes DAELM, inflection rules, LBFS, sampling, and knowledge fusion. There is no doubt that the all method works best. Moreover, our framework performs more efficiently. In Table 3, we present the results obtained for the NDCG@10 metric and training time for all

features across the three chains. The numbers in the brackets are minutes of training. In all cases, we observe a significant improvement in precision and efficiency with respect to the baseline.

6. Conclusion

In this paper, we propose a general application framework of ELM for urban computing and list three case studies. Experimental results showed that our approach is applicable and efficient compared with baselines.

In the future, we plan to apply our approach to more applications. Moreover, we would like to study the distribution of our framework so it can handle more massive data.

TABLE 3: The best average NDCG@10 results of optimal retain store placement.

Cities	Starbucks	TrueKungFu	YongheKing
<i>MART (single city)</i>			
Beijing	0.743 (15)	0.643 (14)	0.725 (14)
Shanghai	0.712 (15)	0.689 (14)	0.712 (12)
Hangzhou	0.576 (13)	0.611 (12)	0.691 (10)
Guangzhou	0.783 (15)	0.691 (13)	0.721 (12)
Shenzhen	0.781 (16)	0.711 (15)	0.722 (13)
<i>RankBoost (single city)</i>			
Beijing	0.752 (23)	0.678 (20)	0.712 (19)
Shanghai	0.725 (25)	0.667 (20)	0.783 (21)
Hangzhou	0.723 (21)	0.575 (19)	0.724 (15)
Guangzhou	0.812 (22)	0.782 (19)	0.812 (18)
Shenzhen	0.724 (22)	0.784 (17)	0.712 (12)
<i>BP (Single city)</i>			
Beijing	0.753 (45)	0.658 (40)	0.702 (41)
Shanghai	0.724 (44)	0.657 (42)	0.7283 (44)
Hangzhou	0.725 (41)	0.555 (40)	0.714 (45)
Guangzhou	0.832 (45)	0.772 (42)	0.512 (41)
Shenzhen	0.722 (47)	0.754 (42)	0.702 (45)
<i>DAELM</i>			
Beijing	0.755 (3)	0.712 (2)	0.755 (2)
Shanghai	0.745 (5)	0.751 (3)	0.783 (1)
Hangzhou	0.755 (5)	0.711 (2)	0.752 (5)
Guangzhou	0.810 (9)	0.810 (5)	0.823 (4)
Shenzhen	0.780 (9)	0.783 (5)	0.723 (8)

Competing Interests

The authors declare that they have no competing interests.

References

- [1] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: concepts, methodologies, and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 5, no. 3, article 38, 2014.
- [2] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-air: when urban air quality inference meets big data," KDD, 2013.
- [3] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489-501, 2006.
- [4] H. Zhou, G.-B. Huang, Z. Lin, H. Wang, and Y. C. Soh, "Stacked extreme learning machines," *IEEE Transactions on Cybernetics*, vol. 45, no. 9, pp. 2013-2025, 2015.
- [5] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th International Conference on Machine Learning (ICML '11)*, Bellevue, Wash, USA, 2011.
- [6] L. Zhang and D. Zhang, "Domain adaptation extreme learning machines for drift compensation in e-nose systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 7, pp. 1790-1801, 2015.
- [7] Y. Zheng, "Methodologies for cross-domain data fusion: an overview," *IEEE Transactions on Big Data*, vol. 1, no. 1, pp. 16-34, 2015.
- [8] Y. Zheng, Y. Liu, J. Yuan, and X. Xie, "Urban computing with taxicabs," HUC, 2011.
- [9] N. Srivastava and R. Salakhutdinov, "Multimodal learning with deep Boltzmann machines," *Journal of Machine Learning Research*, vol. 15, pp. 2949-2980, 2014.
- [10] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT '98)*, pp. 92-100, Madison, Wis, USA, July 1998.
- [11] M. Gönen and E. Alpaydm, "Multiple kernel learning algorithms," *Journal of Machine Learning Research*, vol. 12, pp. 2211-2268, 2011.
- [12] Y. Zheng, X. Yi, M. Li et al., "Forecasting fine-grained air quality based on big data," in *Proceedings of the 21st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '15)*, pp. 2267-2276, August 2015.
- [13] N. J. Yuan, Y. Zheng, X. Xie, Y. Wang, K. Zheng, and H. Xiong, "Discovering urban functional zones using latent activity trajectories," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 3, pp. 712-725, 2015.
- [14] J. Chen, H. Chen, Z. Wu, D. Hu, and J. Z. Pan, "Forecasting smog-related health hazard based on social media and physical sensor," *Information Systems*, 2016.
- [15] S. Scellato, A. Noulas, and C. Mascolo, "Exploiting place features in link prediction on location-based social networks," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*, pp. 1046-1054, August 2011.
- [16] N. Zhang, H. Chen, X. Chen, and J. Chen, "ELM meets urban computing: ensemble urban data for smart city application," in *Proceedings of ELM-2015 Volume 1*, pp. 51-63, Springer, 2016.
- [17] J. Yuan, Y. Zheng, and X. Xie, "Discovering regions of different functions in a city using human mobility and POIs," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*, pp. 186-194, ACM, Beijing, China, August 2012.
- [18] Y. Wei, Y. Zheng, and Q. Yang, "Transfer knowledge between cities," KDD, 2016.
- [19] V. Hughes, "Where there's smoke," *Nature*, vol. 489, no. 7417, pp. S18-S20, 2012.
- [20] J. Ramos, "Using tf-idf to determine word relevance in document queries," in *Proceedings of the 1st Instructional Conference on Machine Learning*, 2003, <https://www.semanticscholar.org/paper/Using-Tf-idf-to-Determine-Word-Relevance-in-Ramos/b3bf6373ff41a115197cb5b30e57830c16130c2c>.
- [21] Y. Fu, Y. Ge, Y. Zheng et al., "Sparse real estate ranking with online user reviews and offline moving behaviors," in *Proceedings of the 14th IEEE International Conference on Data Mining (ICDM '14)*, pp. 120-129, Shenzhen, China, December 2014.
- [22] P. Jensen, "Network-based predictions of retail store commercial categories and optimal locations," *Physical Review E*, vol. 74, no. 3, Article ID 035101, 2006.
- [23] C.-C. Chang and C.-J. Lin, "LIBSVM: a Library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, article 27, 2011.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

