

Article

# Large-Scale, Fine-Grained, Spatial, and Temporal Analysis, and Prediction of Mobile Phone Users' Distributions Based upon a Convolution Long Short-Term Model

Guangyuan Zhang <sup>1</sup>, Xiaoping Rui <sup>2,\*</sup>, Stefan Poslad <sup>3</sup>, Xianfeng Song <sup>1</sup>, Yonglei Fan <sup>1</sup> and Zixiang Ma <sup>3</sup>

<sup>1</sup> College of Resources and Environment, University of Chinese Academy of Sciences, Beijing 100049, China, zhangguangyuan16@mails.ucas.ac.cn (G.Z.); xfsong@ucas.ac.cn (X.S.); fanyonglei17@mails.ucas.ac.cn (Y.F.)

<sup>2</sup> School of Earth Sciences and Engineering; Hohai University; Nanjing 211000, China

<sup>3</sup> Queen Mary University of London, London E1 4NS, UK; stefan.poslad@qmul.ac.uk (S.P.); zixiang.ma@qmul.ac.uk (Z.M.)

\* Correspondence: ruixpsz@163.com

Received: 26 March 2019; Accepted: 6 May 2019; Published: 9 May 2019



**Abstract:** Accurate and timely estimations of large-scale population distributions are a valuable input for social geography and economic research and for policy-making. The most popular large-scale method to calculate such estimations uses mobile phone data. We propose a novel method, firstly based upon using a kernel density estimation (KDE) to estimate dynamic mobile phone users' distributions at a two-hourly scale temporal resolution. Secondly, a convolutional long short-term memory (ConvLSTM) model was used in our study to predict mobile phone users' spatial and temporal distributions for the first time at such a fine-grained temporal resolution. The evaluation results show that the predicted people's mobility derived from the mobile phone users' density correlates much better with the actual density, both temporally and spatially, as compared to traditional methods such as time-series prediction, autoregressive moving average model (ARMA), and LSTM.

**Keywords:** population density; deep learning; mobile phone data; spatial–temporal data analysis and prediction; kernel density estimation (KDE); long short-term memory (LSTM); convolution LSTM (ConvLSTM); autoregressive moving average (ARMA)

## 1. Introduction

Mapping population dynamics is of great significance for city and transport planning [1–3], public safety warning [4–6], disaster impact assessments [7–9], and epidemic modeling [10–12]. The analysis of mobile phone data is frequently used to map the spatial and temporal situation of users [13,14]. However, a more fine-grained temporal resolution of dynamic population distributions is still a challenge when studying human activities [13]. There is a range of motivations and applications for doing this. For instance, the prediction of people density could help drive improved regional authority policy decisions, so as to help avoid stampedes such as the tragic Shanghai Stampede, when 36 people died and 49 were injured during the New Year's Eve celebration on 31 December 2014 in the Shanghai Bund [6]. Another example is that there are a large number of shared bicycles in many big cities, especially in China, where bicycles could be better distributed according to the dynamic movement of the population in order to increase revenue [15]. Takeaway-food delivery could be made more efficient if service providers were aware in advance where and when an area holds a strong potential purchasing power [16].

There is a range of sensors that could be used to profile users' mobility, especially in urban areas. For example, fixed sensors at traffic lights can be used to predict when road congestion at those locations will occur and disappear [17]. Sensors such as global positioning system (GPS) can be used, either fixed in vehicles [18] or in mobile phones that can accompany users as they move about [19]. However, these mobile sensing techniques face several challenges. Firstly, the sensor data have to be acquired via a mobile app and uploaded, and there are no such apps that have good coverage and that freely share their mobility data. There are privacy concerns even if the mobility data are anonymized [20]. The use of mobility sensors and associated apps can be energy-hungry such that a device running such an app can more quickly run out of energy and shut down [21]. Our method to collect data from the mobile phone network overcomes each of these challenges provided that a mobile phone network operator shares the data. Its key characteristics are that (1) many people have mobile phones; (2) only the time and location of the phone within a cell of the network is recorded (thus, we cannot derive the exact address of someone's home location); (3) the method is low-energy because it just relies on a mobile phone cell's wireless access point to periodically send "ping" signals to check if a mobile phone receiver is within its range.

Mobile phone networks, also called cellular networks, are composed of cells, i.e., geographic zones around a phone tower or wireless network access point. Each mobile phone can be located by identifying the geographic coordinates of the transmitting tower it currently communicates with, within its associated cell. Furthermore, when a user makes a call or sends a text message, these will be recorded in the call detail record (CDR) which is located at its nearest phone tower. CDRs are not as structured as traditional travel survey data, which contain location and time information for meaningful activity destinations, or as precise as global navigation satellite system (GNSS) data, which provide location data with higher frequency and accuracy [22]. However, as a by-product for billing purposes carried out routinely by mobile service carriers, CDR data can be obtained at a much lower cost and on a greater scale than GNSS data [23]. Reference [14] highlights how facilitating access to anonymized mobile phone data might enable fast and cheap production of population maps in emergency situations and in data-scarce environments, e.g., in rural or less accessible regions where there is less digital infrastructure such as public transport, local area networks, etc.

Although, the global mobile phone penetration rate (i.e., the percentage of active mobile phone subscriptions within the population) reached 100% in developed countries in 2014. In contrast, in developing countries, it is about 90%; however, this continues to rise [24]. Note that mobile phone density may not be equal to population density because of periods when the phones are not in use such as at nighttime or during daytime because one person may have more than one phone, more than one subscription with the same phone, or have no phone. Deville et al. [14] used active users and active records of mobile phones to estimate the population and to characterize the relationship between users' phone use density and the actual population density. This was used in several applications [25–28]. However, using census data which can be collected by systematically acquiring and recording information about the members of a given population is often not very accurate to study a dynamic population distribution [13].

Some studies (see Section 2) separate daytime, e.g., from 7:00 a.m. to 8:00 p.m. and nighttime as 8:00 p.m. to 7:00 a.m. and then use cell mobile metadata to calculate the users' density at night. Some other studies use the census data, which are infrequently collected by the government, i.e., every 10 years, to estimate the census population density and the actual population density. Therefore, in this study, we use the mobile cell phone data which can directly reflect the phone users' distribution to characterize people's mobility based on the ubiquitous nature of mobile phones in modern society [14]. As machine learning and deep learning techniques are currently popular to predict or estimate output values in many fields [29], in this study, a deep learning approach is also proposed for estimating and predicting mobile users' dynamic distribution that potentially contributes to mapping and predicting a population's dynamics at a more fine-grained spatial scale and temporal scale, i.e., every two hours, compared to two traditional methods which can be used as a baseline,

i.e., the autoregressive moving average (ARMA) model and a popular deep learning method that uses a long short-term memory (LSTM) model for predicting time-series data.

The remainder of this article is organized as follows: Section 2 reviews related work. Section 3 presents the mobile phone dataset from 2 February 2015 to 17 February 2015 in Beijing. Section 4 presents the methods for calculating users' distributions and predicting their density using the ARMA model, LSTM model, and convolution LSTM (ConvLSTM) model. In Section 5, we evaluate their accuracy and compare these three methods' abilities to predict the urban population distribution both in time and space. In Section 6, we present our conclusions and our thoughts for future research.

## 2. Related Work

The focus here is on mapping mobile phone users' distribution which can present the mobility of people and contribute to the estimation of the real dynamic population distribution in space and time. Furthermore, the focus is also on predicting mobile phone users' spatial and temporal distribution in the future, forecasting, at least one whole day in advance, with high accuracy.

In terms of population mapping, most previous studies concentrated on using simple areal-weighting methods (a technique for estimating the values of overlapping but incongruent polygon features) [29–32] and estimating the population data from the census data with respect to the census administrator units or regions [33]. Others used some ancillary data such as remote-sensing images, land-use data, e.g., urban or sub-urban boundaries, to estimate population counts within the census units. However, the census and ancillary data used in these previous studies usually lag behind the time of interest significantly because a census enumeration is recommended about only once every ten years by the United Nations (UN) [34]. It is really is a challenge to acquire population data for such a large urban area at a daily or even higher temporal resolution [13]. In recent years, advances in information communication technology (ICT) and the increased accessibility of location-aware mobile devices provided multi-source location-based big data about human activity [35]. Combined with other ancillary data, mobile data were utilized in modeling human mobility [36–38] and in making dynamic estimates of population distribution more feasible [13,14]. Furthermore, some efforts on nowcasting, i.e., forecasting changes in the near future of the population distribution, using mobile phone data, were undertaken; for example, Reference [39] used CDRs and a machine learning approach to estimate residents' distribution within a territory. Reference [40] proposed an analytical framework based on CDRs to nowcast the population count, well-being, and economic development. Note that there is no standard definition of what near future means. Because of the physical inertia to make detectable changes in the physical environment, nowcast takes the order of hours, rather than minutes, within this context. In some cases, it is not clear how far in the future the estimate is made for (e.g., see References [39,40]).

An autoregressive moving average model (ARMA) is used traditionally to predict future values of a time series in several important fields of interest such as linear prediction, system identification, and spectral analysis [41–43]. More recently, some studies and applications focused on the use of machine learning and deep learning, especially for spatial prediction such as traffic flow prediction [17,44–47], precipitation prediction [48–50], and some prediction of social issues [51,52].

Among the deep learning models, Long Short-Term Memory (LSTM) achieved a good performance within the field of natural language (NL) processing [53] and trajectory prediction [54]. Although the accuracy of the LSTM model is high, its input data are one-dimensional, and it is not suitable for spatial sequence data such as video, satellite, or radar image datasets; hence, the LSTM model is often combined with other models. A CNN–LSTM algorithm combines a convolutional neural network (CNN) with LSTM such that the CNN part of the model processes the data and the one-dimensional result then feeds into an LSTM model. For example, a CNN–LSTM algorithm was proposed and applied by Liu et al. [55] and Rad et al. [56]. The former's experimental results on the MNIST and FashionMNIST datasets show that the algorithm is universal. For example, it was used for image recognition and for classification tasks. The experiments for the latter show that transferring the raw feature space to a dynamic feature

space via the proposed architecture enhances the performance of an automatic stereotypical motor movement (SMM) detection system, especially for skewed training data when using a CNN–LSTM, to model the temporal patterns in the sequence of multi-axis Inertial Measurement Units (IMUs) signals in terms to the spatial and temporal data. Although related to CNN–LSTM, a convolution LSTM (ConvLSTM) network is different in that ConvLSTM replaces the LSTM matrix multiplication with a convolution operation at each gate in the LSTM cell. For example, ConvLSTM was proposed to build an end-to-end trainable model for rain or precipitation when capturing spatiotemporal correlations [50]. Liu et al. used a ConvLSTM module to analyze historical traffic flow data and to validate that it can achieve better prediction accuracy compared with existing approaches [57]. Qiao et al. proposed a time-distributed ConvLSTM model to extract spatiotemporal features of multi-sensor time series for health monitoring [58]. Yuan et al. proposed a Hetero-ConvLSTM framework, where they incorporated spatial graph features and a spatial model ensemble to address the spatial heterogeneity of the data, e.g., in urban versus rural regions. The extensive experiments show that the proposed framework makes reasonably accurate predictions and significantly improves the prediction accuracy over baseline approaches [59].

In conclusion, spatial–temporal population estimates of mobile phone users extracted from CDRs may exhibit unstable time sequences across some neighboring cells. ARMA is a simpler and more efficient model which is suitable for a time series with stationary values that vary at a relatively small scale; however, this suffers from poor accuracy for population estimates. As a popular deep learning model, LSTM is currently being used for a wide range of predictions for time-series data with good accuracy, but it does not take into account the effects of dissimilar neighboring spatial features, which can make the time prediction less accurate. ConvLSTM improves LSTM in that its use of convolutions can improve the impact of spatial autocorrelation (see Section 4).

### 3. Data

In this study, we used the anonymized individual call detail records (CDRs) that contain information about user identifiers (IDs), time, and corresponding base station locations (Table 1). The dataset was collected anonymously for scientific research. Whenever a user calls or sends a text message provided by his or her Chinese mobile phone operator, a new data record is recorded. All personally identifiable information is masked. The dataset for this study includes over 4.8 billion records of more than 300 million users per day from 2 February to 17 February 2015 in Beijing, China. Users are more active during the daytime than at night both on weekdays and weekends (Figure 1).

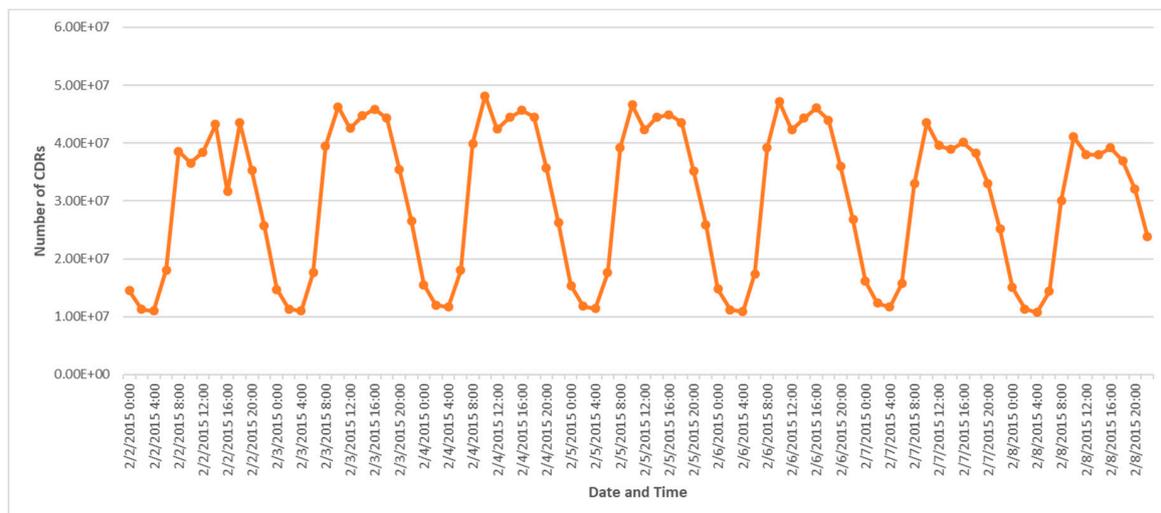
**Table 1.** Call detail record (CDR) data structure input for analysis. FID—feature identifier; ID—identifier.

FID	Name	Description
1	Time	Interactive time of users and base station
2	CI	Corresponding base station ID
3	Tmsi	Encrypted ID of users

We used the locations of all 51,216 mobile base stations based upon their unique index (Table 2). The coverage area of each mobile base station can be approximated as the Voronoi polygon that surrounds it. When a phone is used to make a call or send a text message, its location is found through verifying the range of the specific mobile base station to which the phone is connected.

**Table 2.** Base station index data structure.

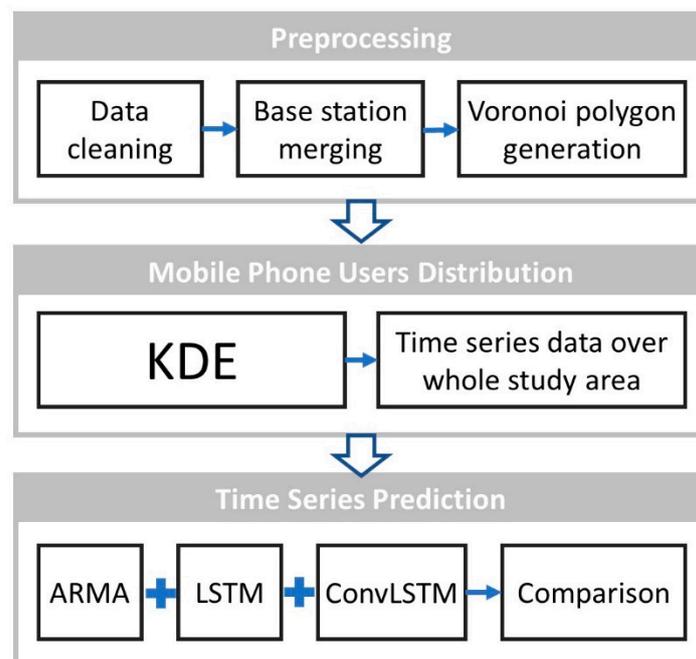
FID	Name	Description
1	CI	Unique ID of base station
2	Lon, Lat	Latitude and longitude of base station location



**Figure 1.** The number of call detail records (CDRs) distributed every 2 h over one week.

#### 4. Method

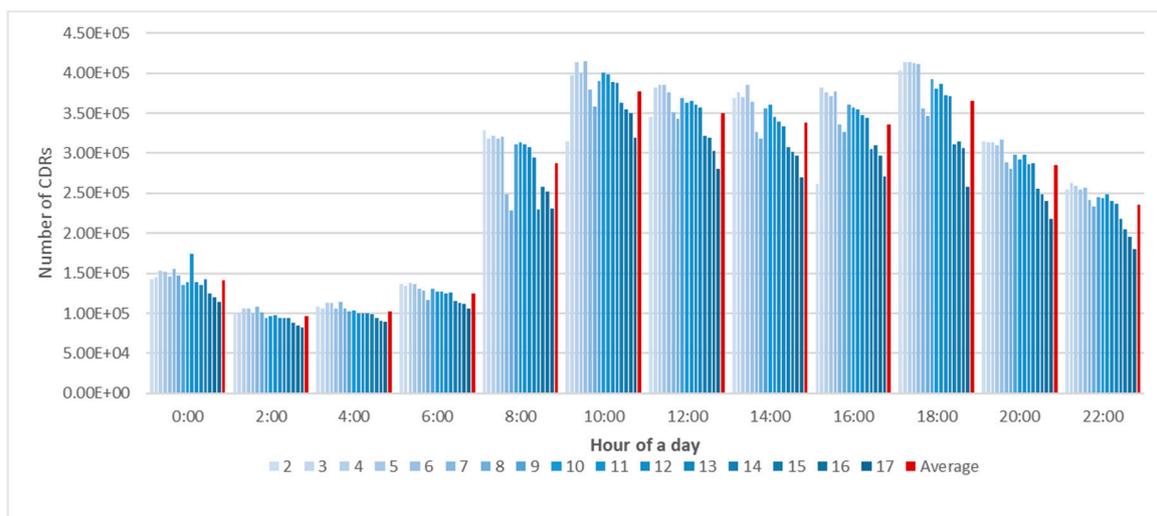
Our method overview is as follows: we firstly extracted the CDRs of mobile users every two hours (this time window was selected as a tradeoff between the higher computation cost needed for a shorter time window and the greater spatial variance of CDRs if a longer time window is used). In addition, only the first 30 s of CDRs for each 2-h time window per base station was analyzed as being a representative sample of user mobility in order to further reduce the data to make it more manageable for data analysis, yielding 160 MB of data (for 30 s) instead of 42 GB (for 2 h). There are two justifications for analysis of 30-s data segments every 2 h: (1) to reduce the data analysis to make computation more accessible in poorer world regions where they may lack a more costly data processing infrastructure to perform such an analysis; (2) most users use their smart phone for less than 30 s [60]; hence, 30 s of CDR encompasses the start and end of most calls and texting. The user density within the whole study region was generated from a kernel density analysis, sequencing the period of 16 days and structuring this as a raster image that consisted of a matrix of cells (or pixels) organized into rows and columns (or a grid), where each cell contained a value representing the user density as Voronoi polygons. Finally, a traditional ARMA model, a deep learning long short-term memory (LSTM) model, and an LSTM model combined with convolution were used to train and predict the temporal and spatial density distribution of mobile phone users, and their accuracy was calculated and compared. An overview of the method to determine the temporal spatial distribution of users is given in Figure 2.



**Figure 2.** Method overview to perform a temporal and spatial prediction of mobile phone users.

#### 4.1. Data Preprocessing

Firstly, we cleaned the data using three steps as follows: (1) to simplify the amount of data analysis, we considered 2-h time windows for analysis starting from 12:00 a.m. during each of the 16 days in the dataset, generating 192 time-series data files or records; (2) for every 2 h, the first 30 s of data were analyzed as a representative sample for each record phone call or text message, with each user being regarded as being stationary within a Voronoi polygon region (Figure 3); (3) there were a few no-value data fields for some records such as latitude and longitude (which could, for example, have been caused by power outages and data loss during data communication). These were regarded as outliers and deleted.

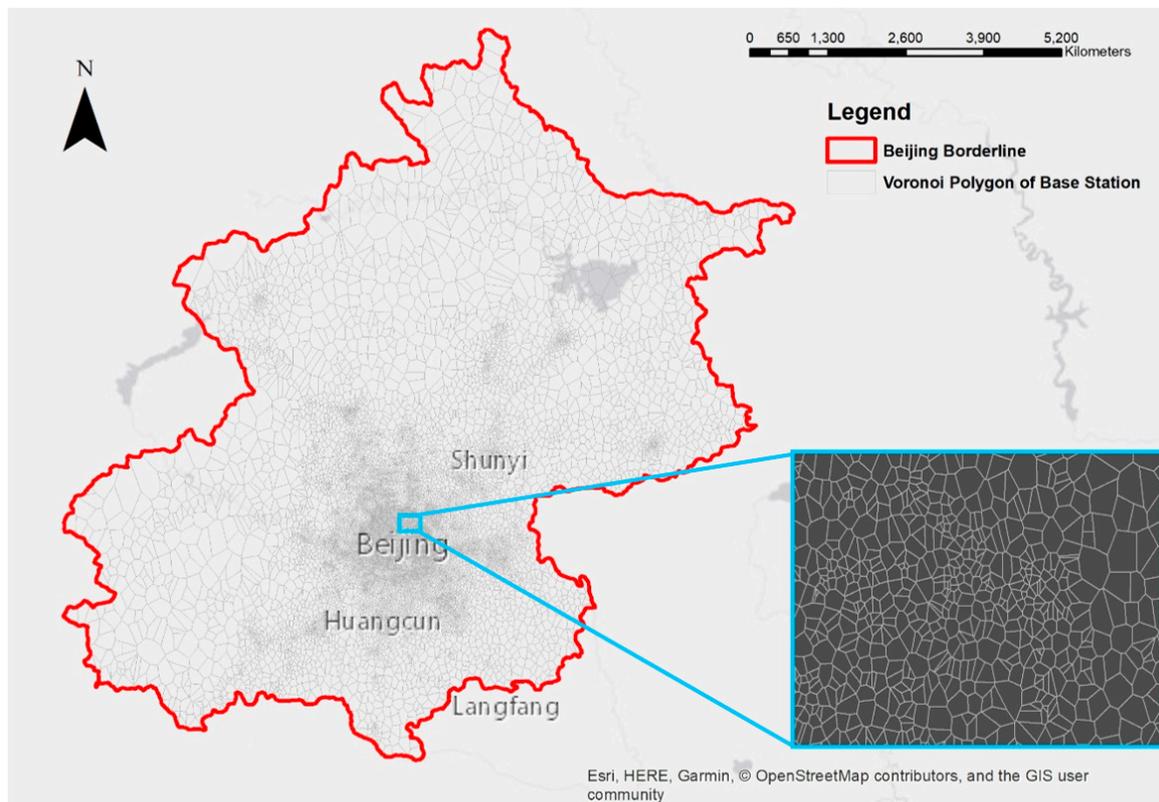


**Figure 3.** Variation in the number of CDRs for the first 30 s every 2 h over 16 days (2–17 February).

Secondly, because the accuracy of the latitude and longitude of mobile phone base stations is not enough when some base stations are located on top of buildings or otherwise close to each other (this varies depending on the phone cell range), we combined such base stations which had the same

latitude and longitude. Hence, after combining 17,447 base stations, we matched the records to the corresponding base stations over the whole time sequence.

Finally, a Thiessen polygon algorithm [61] was utilized to create a Voronoi polygon for each mobile phone base station in order to define the location of phone users within a district (Figure 4). Then, for each polygon corresponding to each base station, we generated random points to represent the distribution of phone users as one point per user.



**Figure 4.** Mobile base stations are represented by Voronoi polygons in Beijing.

#### 4.2. Modeling Mobile Users' Population Distribution Using Kernel Density Estimation (KDE)

We cannot simply input the locations of phone users as the counts in each Voronoi polygon into the neural network to do a time-series data prediction, e.g., using ConvLSTM (see Section 4.3), because they are impossible to input into ConvLSTM because of their irregular shape. Instead, we firstly performed a nonparametric estimation of the distribution of mobile phones to convert them into a set of counts for regular resized grid cells. To calculate this distribution, an appropriate search radius was calculated by geographic information system (GIS) software, e.g., ArcGIS (Available from <http://desktop.arcgis.com/en/arcmap/>, retrieved 01/11/2018), to produce an estimation of the density at different spatial resolutions, such as 800 m, before producing a raster grid consisting of square 800 m  $\times$  800 m cells for input into ConvLSTM. This could also be transformed to other spatial resolutions, such as a grid of square cells 5 km in size in order to decrease the calculation cost of the model allowing it to be performed on a regular personal computer (PC) without any special hardware acceleration such as a graphics processing unit (GPU).

Kernel density estimation (KDE) was used here to perform an automatic search to explore the hotspots of the event distribution, and this method uses complex distance attenuation to measure changes in event density [62–64]. A GIS-based KDE estimation method mainly uses a moving window to calculate and output the point or line density of each grid cell. Given the sample  $(x_1, x_2, \dots, x_n)$  is an independent identically distributed (iid) sample extracted from the population with a distribution

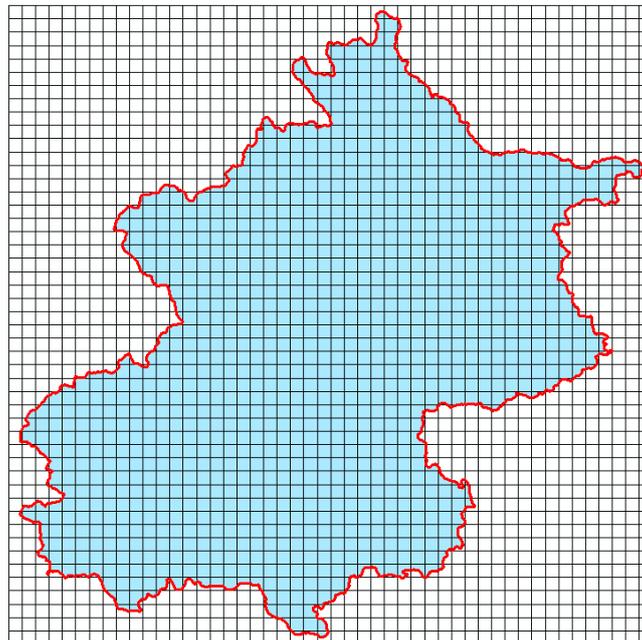
density function  $f$  at a point  $x$ ,  $f(x)$ , calculated using a Rosenblatt–Parzen kernel estimate, we get the following:

$$f_n(x) = \frac{1}{nh} \sum_{i=1}^n k\left(\frac{x-x_i}{h}\right), \quad (1)$$

where  $k$  is the kernel function,  $h > 0$  is a smoothing parameter for the kernel called the bandwidth, and  $(x - x_i)$  is the distance from the estimated point  $x$  to a sample point  $x_i$ .

When KDE is processed, the determination or selection of the bandwidth  $h$  has great influence on the calculation result. As  $h$  increases, the change of the point density in space is smoother, but the structure of the density is masked. When  $h$  is reduced, the estimated dot density change can change very abruptly between Voronoi polygons. In specific applications, it is necessary to test different  $h$  values according to the different land use in physical environments, e.g., presence of banks or train stations, in order to explore the nuclear density surface that can match the actual situation.

The specific steps for KDE estimation are as follows: (1) define a search radius to count the number of events that fall within the circle using a sliding circle; (2) determine the output raster size based on the density accuracy requirements; (3) calculate the density contribution of each event to each grid in the circular domain using the kernel function; (4) assign the density value of each raster to the value of the density contribution of each event in the raster search radius; (5) output the density values for each raster. In this study, we compared the different bandwidths (Figure 5) and finally determined the characteristics of mobile phone users' distribution for the Beijing area within a 150-km default radius, using the ArcGIS10.5 kernel density estimation tool (See <http://desktop.arcgis.com/en/arcmap/10.5/tools/spatial-analyst-toolbox/how-kernel-density-works.htm>).



**Figure 5.** Training grids for the three population density prediction methods: autoregressive moving average (ARMA), long short-term memory (LSTM), and convolution LSTM (ConvLSTM).

#### 4.3. Prediction Models for Time-Series Data

Having created the time-series user density distributions, we introduce an improved prediction model to forecast future user density distributions based on ConvLSTM and then compare its predictions to two baseline models: ARMA and LSTM. Firstly, the baseline systems are described and then the new method is explained.

#### 4.3.1. ARMA Model

The ARMA model consists of two parts, the autoregressive (AR) part and the moving average (MA) part; it is developed using the following equation:

$$S(t) = \sum_{i=1}^p \alpha_i S(t-i) + \sum_{j=1}^q \beta_j e(t-j), \quad (2)$$

where  $S(t)$  is the predicted mobile phone user density at time  $t$ . In the AR part,  $p$  is the order of the AR process, and  $\alpha_i$  is the AR coefficient. In the MA part,  $q$  is the order of the MA error term,  $\beta_j$  is the MA coefficient, and  $e(t)$  is the white noise that produces random uncorrelated variables with zero mean and constant variance [59]. The future values can be predicted using the realized ARMA model. For example, Equation (3) is applied to predict the hour-ahead forecasting ( $h = 1, 2, 3, \dots$ , hours).

$$S(T) = \sum_{i=1}^p \alpha_i S(T-i) + \sum_{j=1}^q \beta_j e(T-j), \quad (3)$$

where  $S(T)(T = t + h)$  is the predicted mobile phone user density at time  $t + h$ .

In this study, to decrease the amount of computation and usage of the device memory, we separated the whole Beijing area into  $48 \times 48$  rectangle grid cells, in which every cell was a square grid of  $5 \text{ km} \times 5 \text{ km}$ . Then, the average density value was extracted into every cell to represent the local user density. There was a total of 192 cells generated from the corresponding raster images. Using this method, we separately predicted the values of 12:00 a.m., 2:00 a.m.,  $\dots$ , 10:00 p.m. on 17 February 2015 using the history data between 2 February 2015 and 16 February 2015, calculating cells one by one for a total of 2304 times.

#### 4.3.2. LSTM and Convolutional LSTM (ConvLSTM) Models

Long short-term memory (LSTM) is a type of recurrent neural network (RNN) node structure known to have good performance when handling time-series data with temporal autocorrelations [59]. It was used to successfully learn and generalize the properties of time sequences such as traffic flow [65] and financial stock option return [66]. The core concept of LSTM is the cell state affected by various interlinked gates. The cell state acts as a transport highway that transfers relative information all the way down the sequence chain as the “memory” of the network. The cell state can carry relevant information throughout the processing of the sequence. Thus, even information from earlier time steps can make its way to later time steps, reducing the effects of short-term memory. As the cell state evolves, information gets added or removed via gates, acting as a type of neural network that decides which information is allowed to exist for the cell state by learning what information is relevant (during training) [67]. In an LSTM network (Figure 6), at each time step  $t$ , the hidden state  $h_t$  is updated by the current data, i.e., at the same time step  $X_t$ , the hidden states at the previous time step  $h_{t-1}$ , the input gate  $i_t$ , the forget gate  $f_t$ , the output gate  $o_t$ , and a memory cell  $C_t$  are updated as well [53]. The inner principle of the model is similar to that of ConvLSTM; thus, its equations will not be repeated here, as they are given in the introduction of the ConvLSTM model below.

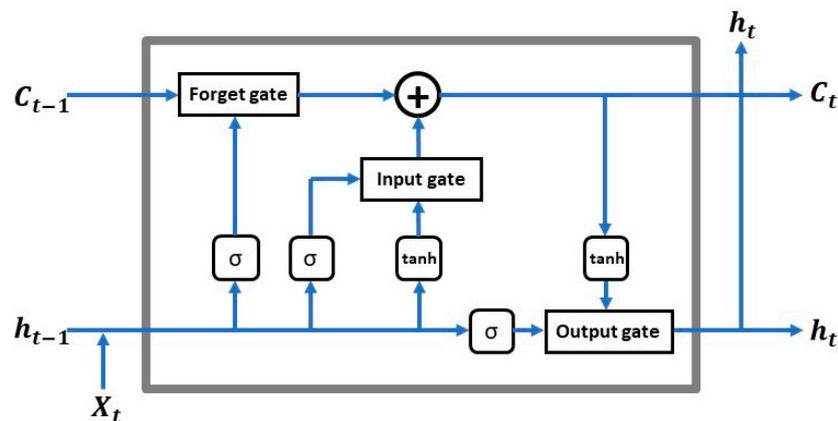


Figure 6. The inner structure of an LSTM cell.

The ConvLSTM model is a variation of LSTM to handle spatiotemporal prediction problems, which were firstly introduced by Shi et al. [50] for precipitation nowcasting, where nowcasting is a technique for very short-range forecasting of the current state using an estimate of speed and direction of movement. In this paper, we follow the formulation of ConvLSTM as in Reference [50], which includes inputs  $X_1, \dots, X_t$ , cell outputs  $C_1, \dots, C_t$ , hidden states  $h_1, \dots, h_t$ , and gates  $i_t, f_t, t$ , and uses a three-dimensional (3D) tensor structure. The first two dimensions of the three-dimensional spatial-temporal tensor of each input feature of a ConvLSTM network are the spatial dimensions and the third dimension is time. The input-to-state and state-to-state transitions of the ConvLSTM cell involve convolutional operations that output three-dimensional tensors, as with the original LSTM model (Figure 7). This model can be further formulated using the following equations, where ‘ $*$ ’ denotes the convolution operation and ‘ $\circ$ ’ denotes the Hadamard product.

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * h_{t-1} + W_{ci} \circ h_{t-1} + b_i), \quad (4)$$

$$f_t = \sigma(W_{xf} * X_t + W_{hf} * h_{t-1} + W_{cf} \circ h_{t-1} + b_f), \quad (5)$$

$$t = \sigma(W_x * X_t + W_h * h_{t-1} + W_{co} \circ h_{t-1} + b), \quad (6)$$

$$C_t = f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * X_t + W_{hc} * h_{t-1} + b_c), \quad (7)$$

$$h_t = t \circ \tanh(C_t). \quad (8)$$

In the above equations,  $i_t, f_t$ , and  $t$  are the outputs of the input gate, forget gate, and output gate for time step  $t$ .  $C_t$  is the cell output at time step  $t$ .  $h_t$  is the hidden state of a cell at time step  $t$ . Sigmoid ( $\sigma$ ) is used as the gating function for the three gates, since it outputs a value between 0 and 1. It can either let no flow or a complete flow of information through the gates. On the other hand, to overcome the vanishing gradient problem, which is a difficulty found in training artificial neural networks with gradient-based learning methods and backpropagation, a function is needed ( $\tanh$ ) whose second derivative can be sustained for a longer range before going to zero.  $W$  and  $b$  are weight matrices and bias vector parameters which need to be learned during training. These equations are illustrated in Figure 6 and described above.

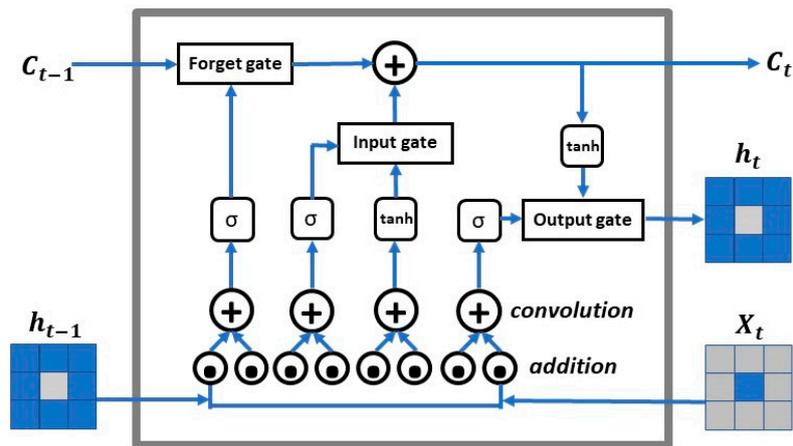


Figure 7. The inner structure of an example ConvLSTM cell.

ConvLSTM has some useful properties for mobile users' distribution prediction, as the LSTM part may capture the temporal autocorrelation in the data, and the convolution operator may capture the local spatial features, which are caused by the spatial autocorrelation [68].

We input the same time-series sets as input data into the realized LSTM model for training and prediction. To validate the accuracy of the ConvLSTM model training, a cross-validation method is presented in Section 5.1, explaining how to use ConvLSTM to predict the mobile phone user density. We selected the first 180 combined grids or cells (over the whole 15 days from 2 February to 15 February 2015) to input to the ConvLSTM module for training, and then tested the prediction (17 February 2015) on the last 12 grids to compare its accuracy. This is the same training data versus test data separation that we also used to test the prediction of spatial density for the ARMA and LSTM models.

## 5. Results and Discussion

This section is organized as follows: we firstly show the process and results of the distribution of mobile phone users using the CDRs from a specific telecom network (China Mobile) operator over the period. Then, we test the predictions using the ConvLSTM model and discuss the results and their accuracy using a cross-validation strategy suitable for time-series data. Finally, the predicted results of ARMA, LSTM, and ConvLSTM models are presented, and their accuracies, both at a temporal and spatial scale, are analyzed and compared.

### 5.1. Determination of Mobile Users' Population Distribution Using KDE

Figure 8 gives an example of a random sample of mobile users' distribution at two time points on 2 February 2015, e.g., 2:00 a.m. and 10:00 a.m., based on the statistic number of records over a 30-s interval. A random sample was used because showing all users would flood the figure making it difficult to see patterns. As Figure 8a shows, at midnight (12:00 a.m.), random users have a relatively high density in core city areas such as the Dongcheng and Chaoyang districts of Beijing, while suburban areas such as the Shunyi and Huairou districts always have a lower density of people using mobile phones, with a lower density of base stations. In contrast, in the daytime (10:00 a.m.), the activity of people using phones increases in both city areas and suburban districts; however, the activity in city areas still shows a much higher level than that in the suburban districts.

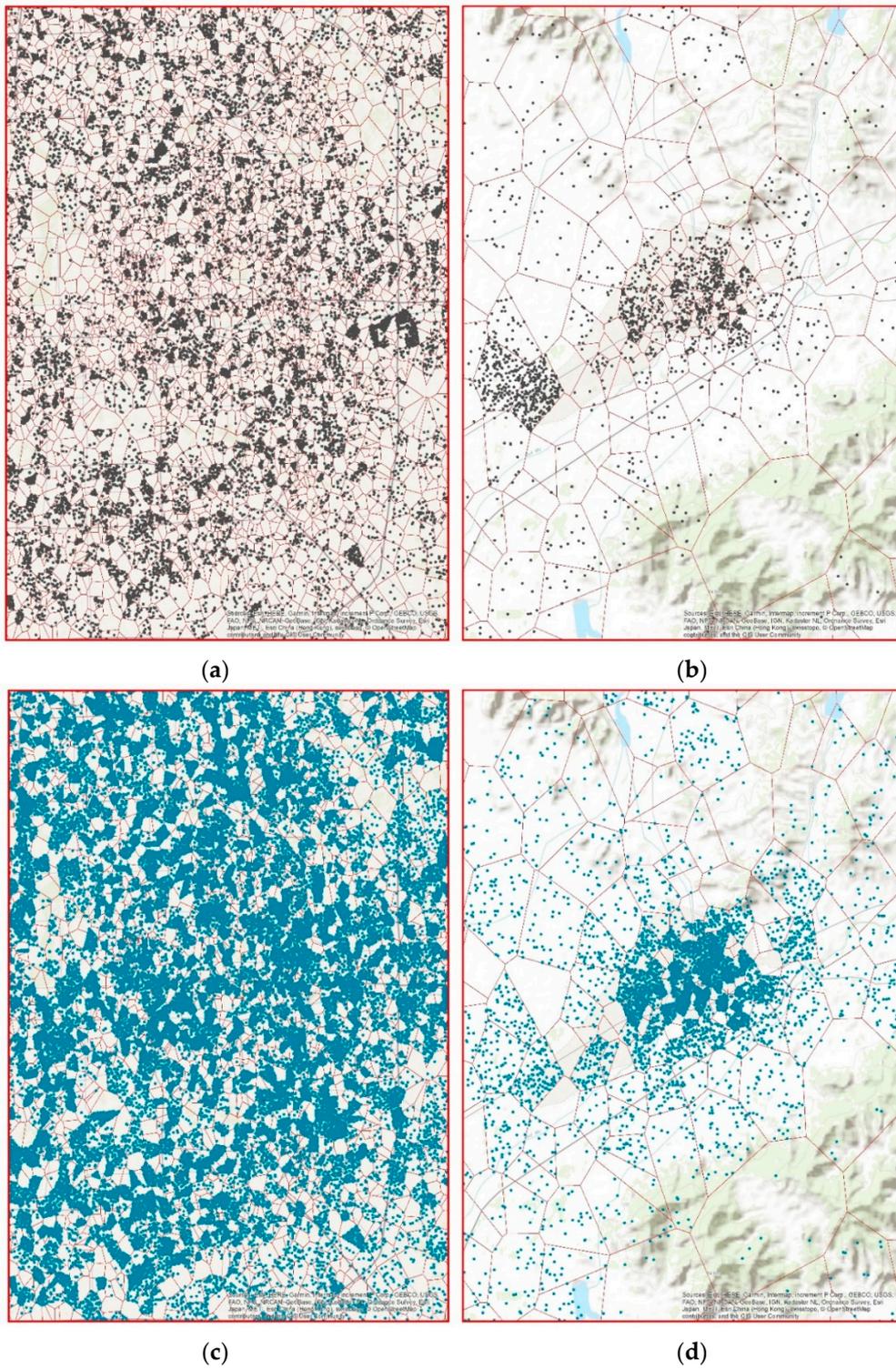
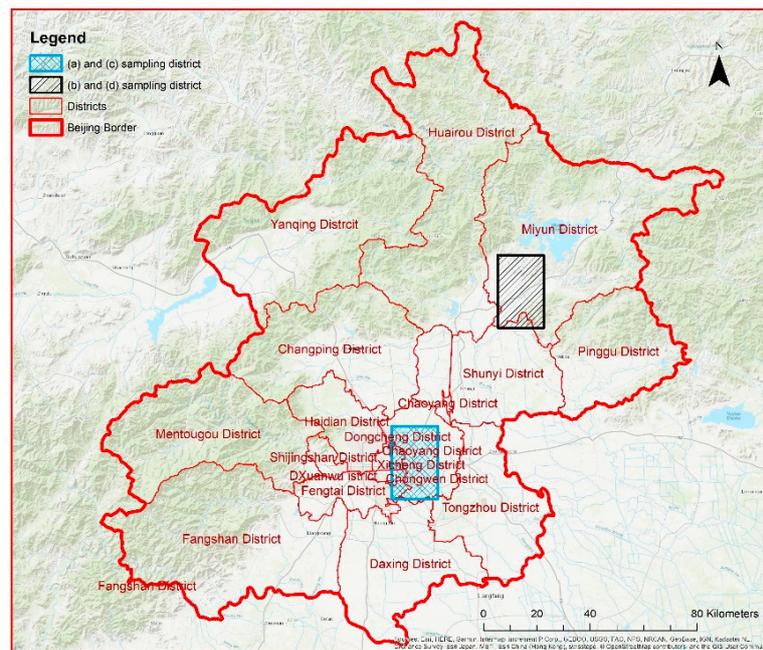


Figure 8. Cont.

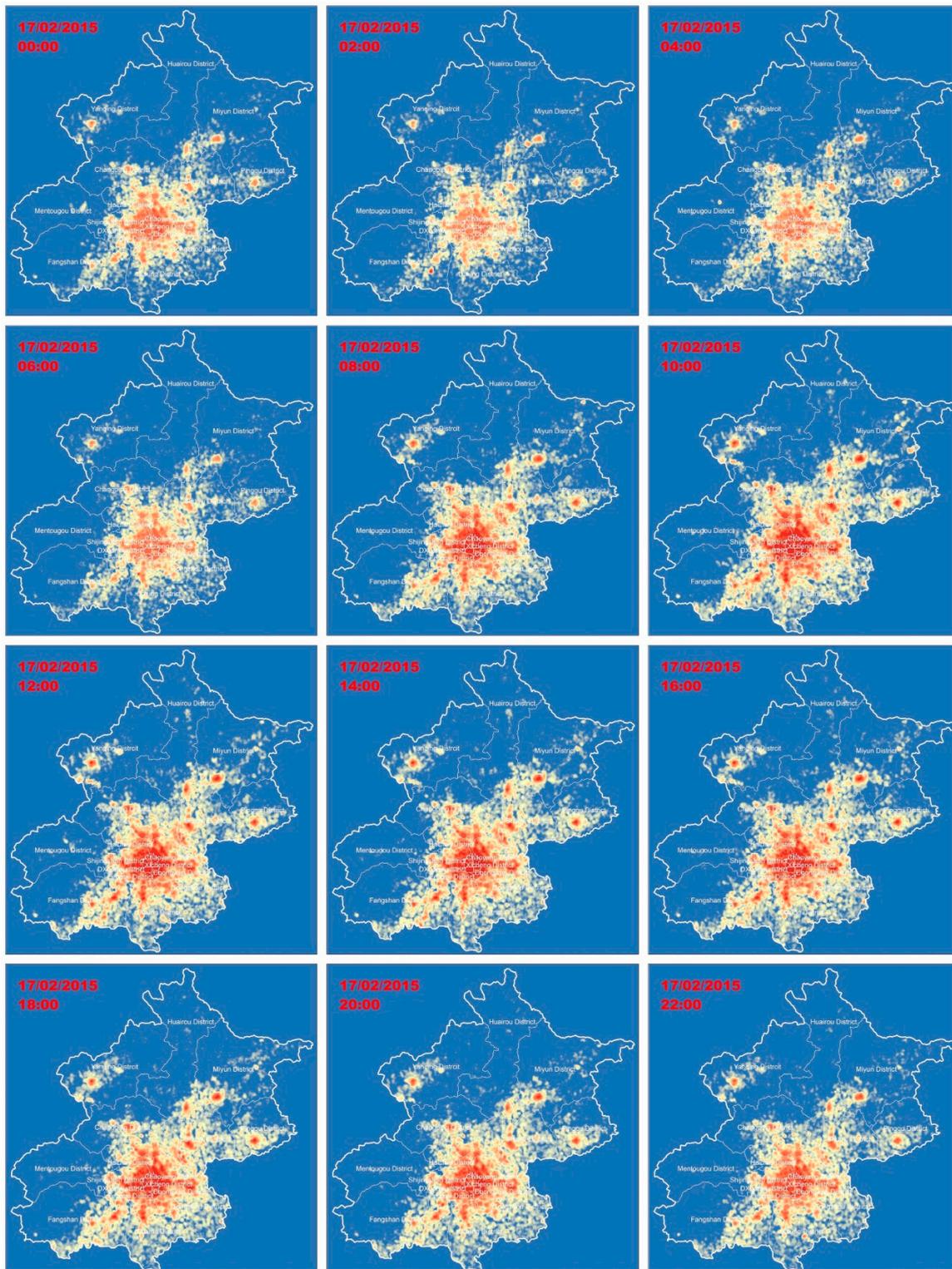


(e)

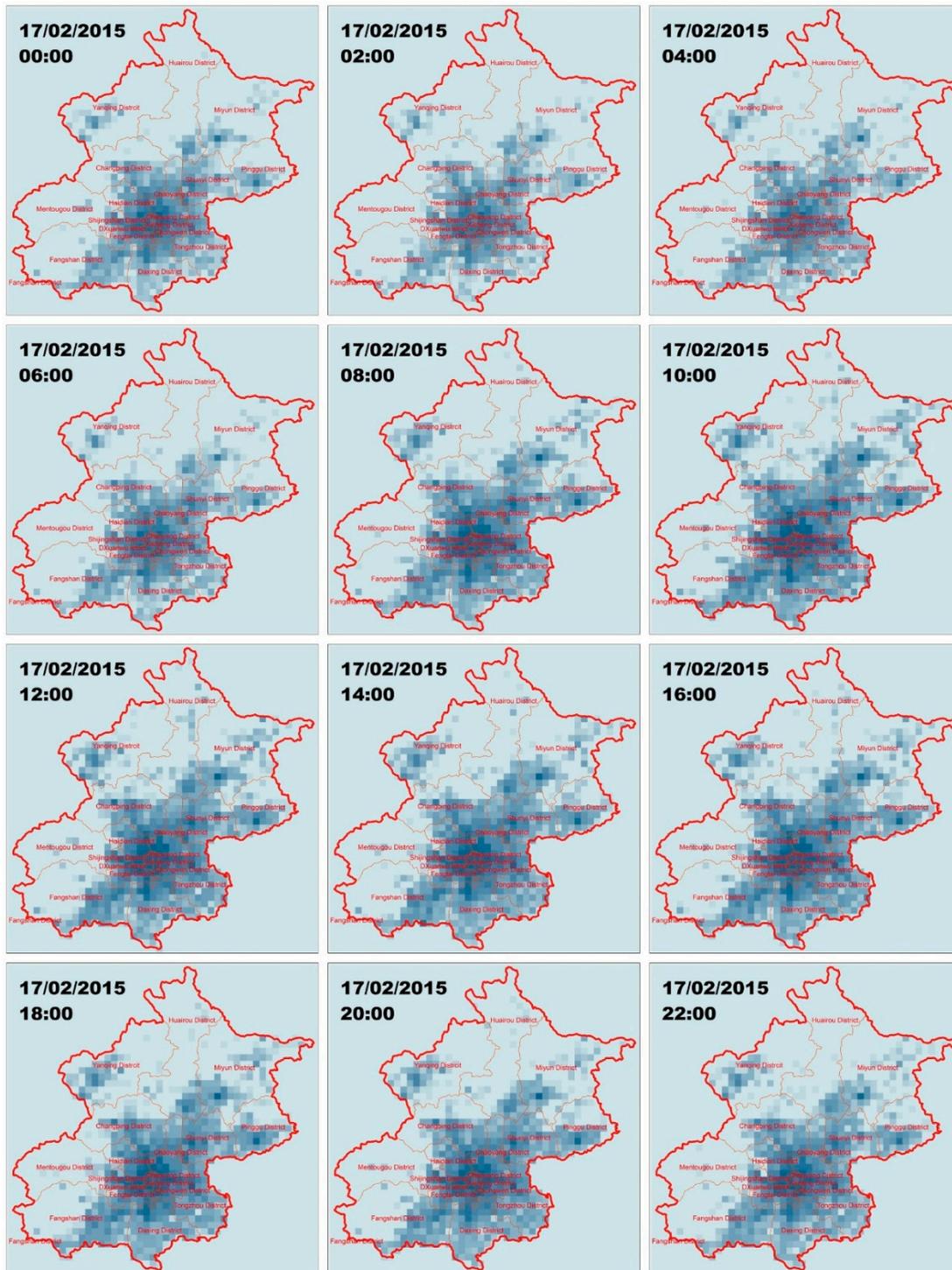
**Figure 8.** The spatial distribution of a random selection of users within each Voronoi polygon for different areas and times: (a) random users within a high-density area at 2:00 a.m.; (b) random users within a low-density area at 2:00 a.m.; (c) random users within a high-density area at 10:00 a.m.; (d) random users within a low-density area at 10:00 a.m.; (e) Illustration showing the location of the areas with high and low population density used for the analysis of the time prediction.

The spatial mobile users' density distribution on 17 February 2015, the predicted targeted day, was depicted using the kernel density method (see Figure 9). Density was visualized using a geometrical interval to classify this distribution into 15 levels, so as to show the differences in user density in space and time. Mobile phone users mainly gather in the center of the city. For example, the total trend of mobile phone users' distribution at midnight is reflected using a distribution scattered around the center of the city, while, during daytime, users gather around the city center.

We extracted the corresponding mean density value into 2034 cells to represent the mobile phone distribution situation of every grid for a total of 192 times (the number of 2-h periods we examined for our study). The results of the decreased resolution had the same characteristics as the original graphs. Then, 180 matrices of 192 were used for training, while the other 12 matrices were used as the test data. The visualization of the last 12 matrices is shown in Figure 10.



**Figure 9.** Mobile users' distribution generated using the kernel density estimation (KDE) method on 17 February 2015 at a higher spatial resolution.

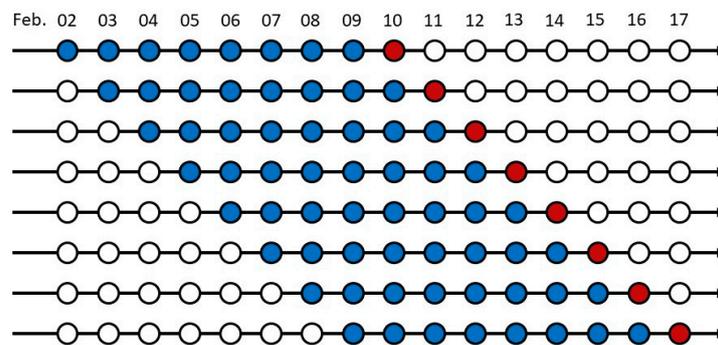


**Figure 10.** Mobile users' distributions generated using kernel density estimation method on 17 February 2015 at a higher spatial resolution.

### 5.2. Prediction Results for the ConvLSTM Model

When testing the accuracy of ConvLSTM model, a cross-validation strategy is needed to assess the predictive performance of the models and to judge how they perform outside the training sample on a new dataset. There are many classical methods of cross-validation that can be used for machine learning and deep learning models, for example, K-fold cross-validation and leave-one-out cross-validation [69]. However, they cannot be used on time-series data because of the timing

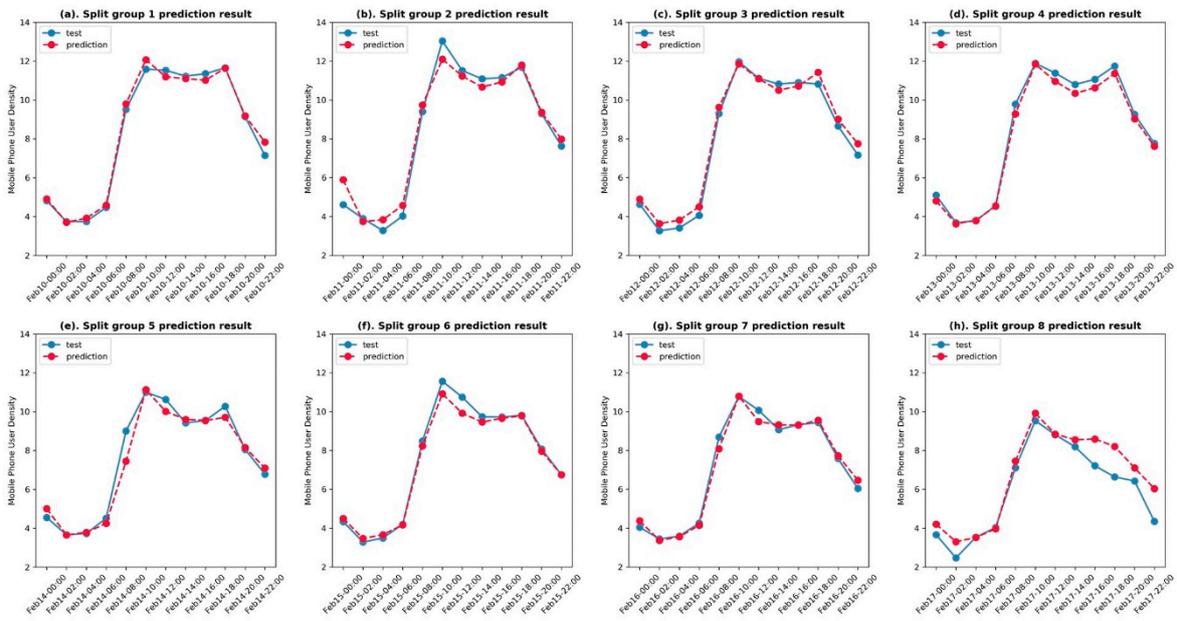
dependence when training and predicting [70]. To cater to the prediction of time sequences, an eight-split time-series cross-validation technique was used in our study, which is illustrated in Figure 11. The data were split into eight groups arranged in chronological order, with the blue circles indicating the test data (one circle presents one day, including a total 12 frames every two hours), the red circles representing the tested or predicted days (12 frames), and the hollow circles representing unused data. The number of blue circles in each split group was eight, whilst the number of red circles was one, which means we trained 96 frames to predict 12 frames using the ConvLSTM model eight times.



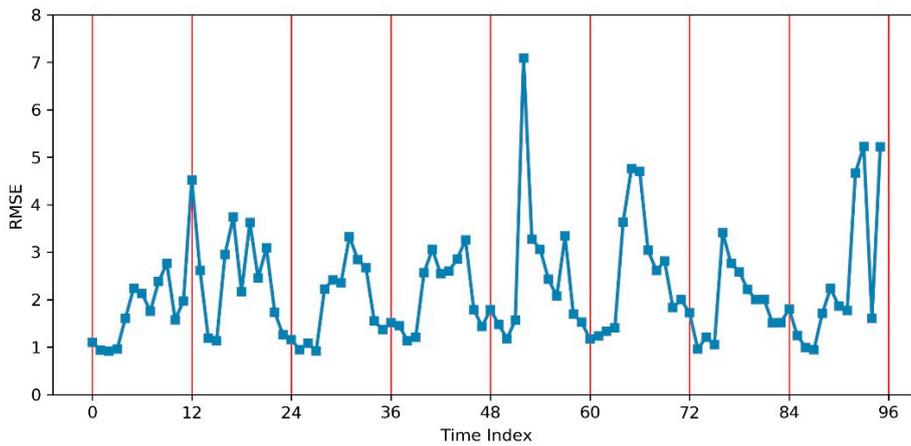
**Figure 11.** A cross-validation method for time-series data.

For each round of the prediction, the parameters of the model included the kernel size, which we proposed to be  $3 \times 3$ , with 40 convolutional filters that can extract important features from the convolution layers, with five units for each. In order to improve the generalization ability and to prevent over-fitting (which is the production of an analysis that corresponds too closely or exactly to a particular set of data, and may, therefore, fail to fit additional data or predict future observations reliably in machine learning or deep learning models), the recurrent weight dropout was set to 0.2 in the model; the number of training times (epochs) was set as 500, whilst the Adam optimizer [71] was used with a learning rate of  $10^{-3}$  and a decay rate of 0.9.

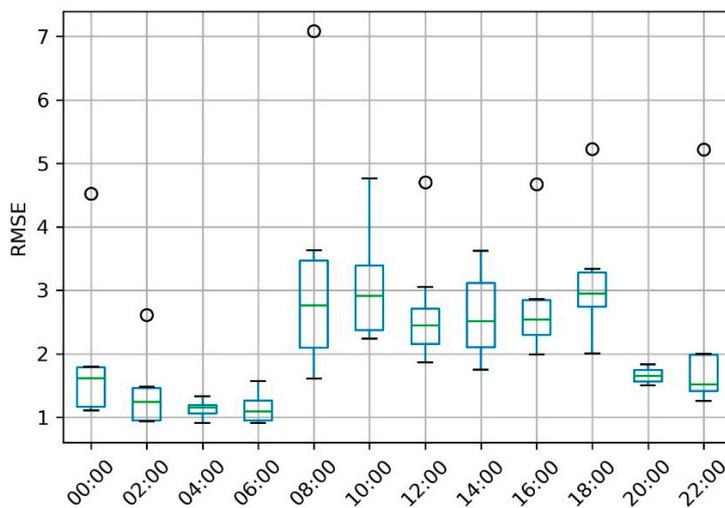
Figure 12 shows the prediction result compared with the test data for eight cross-validation rounds. The accuracy of the predicted values using ConvLSTM appeared good in almost every round. Then, we calculated the root-mean-square error (RMSE) of every predicted and tested  $48 \times 48$  frame. The RMSE results are plotted in Figure 13 over a continuous time sequence, showing that all eight rounds had an RMSE fluctuating between 0.5 and 7, while the values in round 5 were more unstable and had values above 7 (which were then treated as outliers), since the others were no larger than 6. Because we predicted 12 values in one whole day eight times, all RMSE results were extracted in the 12 time nodes from 12:00 a.m. to 10:00 p.m. to draw a boxplot, as shown in Figure 14. This box chart illustrates that, in the first four predicted time nodes, the accuracy reflected by the average RMSEs, which were lower than 2, was much higher, while the next six ones had values between 2 and 3. The RMSEs in the last two nodes were not higher than 2. However, the only outlier in this part was much higher than 5, which was caused by the prediction in round 8 (i.e., when the time node is equal to 96 in Figure 13).



**Figure 12.** The relationship between the test data (blue circle) and prediction results (red circle) for the eight split groups (a–h), following the order from Figure 11, from top to bottom.

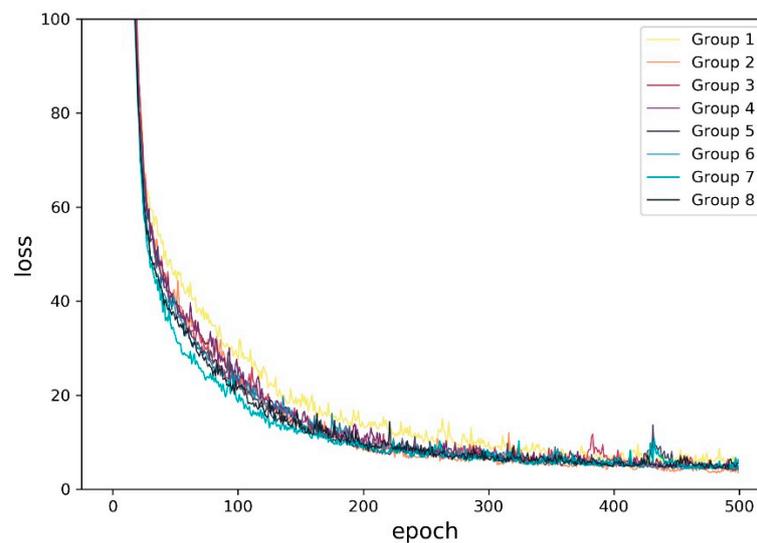


**Figure 13.** The root-mean-square error (RMSE) for every prediction result over a continuous time sequence.



**Figure 14.** The results of eight split groups overlapping 12 time nodes in the predicted day.

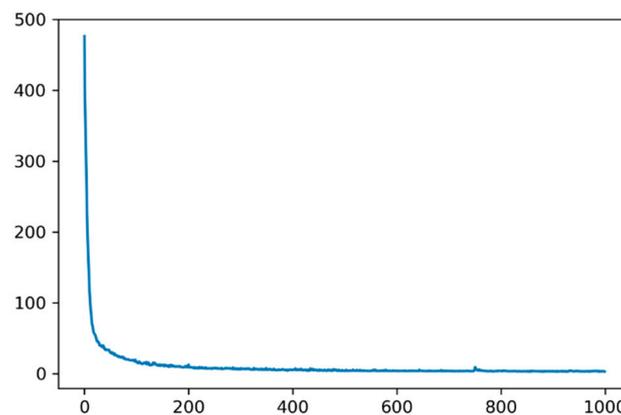
The relationship between the mean absolute error (MAE) and the training epoch of the results, which is called a loss function in machine learning or deep learning, is a non-linear function, as shown in Figure 15. It is illustrated that, when the epoch was greater than 200, the average value of the loss of the eight groups was stable around 5, and, when the epoch was 500, the average loss achieved the lowest value of about 4.4. A similar loss function was exhibited by the ConvLSTM model when we used it to predict the mobile phone user density distribution. In the next step, we also did this for the other baseline methods in order to compare them with the ConvLSTM model.



**Figure 15.** Predicted population density loss versus epoch number for the eight groups.

### 5.3. Prediction Results for the ConvLSTM Model versus the Two Baselines

In order to maximize the potential of the ConvLSTM and baseline prediction models, when using ConvLSTM in this section, all data instances were 192 frames long, where 180 frames were used for the input and 12 frames were used for the prediction. The parameters of the model included the kernel size, which we proposed as  $3 \times 3$ , and the 40 convolutional filters, which can extract important features of the convolution layers in all three ConvLSTM layers with five units each. In order to improve the generalization ability and to prevent over-fitting of the model, the recurrent weight dropout was set to 0.2 in the model; the number of training times (epochs) was 1000, whilst the Adam optimizer was used with a learning rate of  $10^{-3}$  and a decay rate of 0.9. The relationship between the loss of the results and the epoch was a non-linear function, as shown in Figure 16. It is illustrated that, when the epoch was greater than 200, the value of loss was stable around 4, and, when the epoch was 1000, loss achieved the lowest value of 3.23477. The assessment of the predicted results, both in time and space, is mentioned below.



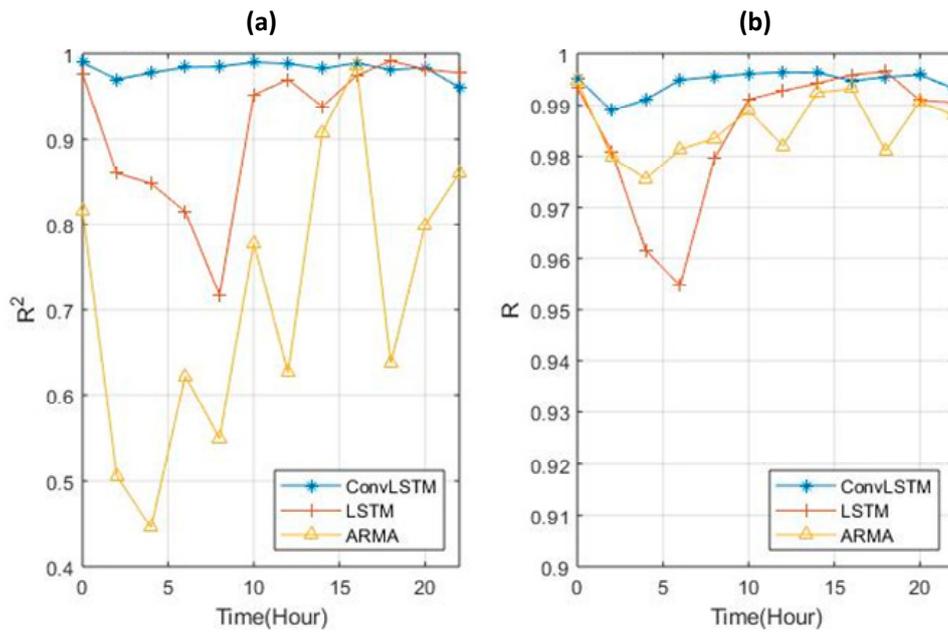
**Figure 16.** Predicted population density loss as a function of the epoch number or training time.

### 5.3.1. Results—Assessment of the Prediction Accuracy in Time

The correlation analysis between the predicted and actual results was studied every two hours. It is obvious that, when using ConvLSTM to predict both the coefficient determination and correlation coefficient of the 12 2-h slots on 17 February, the values were much higher than the results predicted by ARMA and LSTM (0.99 as illustrated in Figure 17a,b). The result in Figure 17a starts at 0.99 at 12:00 a.m. and then decreases to the lowest value at time 2:00 a.m. (0.977). Then, it increases rapidly to 0.993 and keeps stable over the remaining period. The correlation coefficient has a very similar trend to the previous one, with its lowest value being 0.989. The values of both indices were not lower than 90%, which shows that the results of the prediction had a high accuracy in the temporal scale.

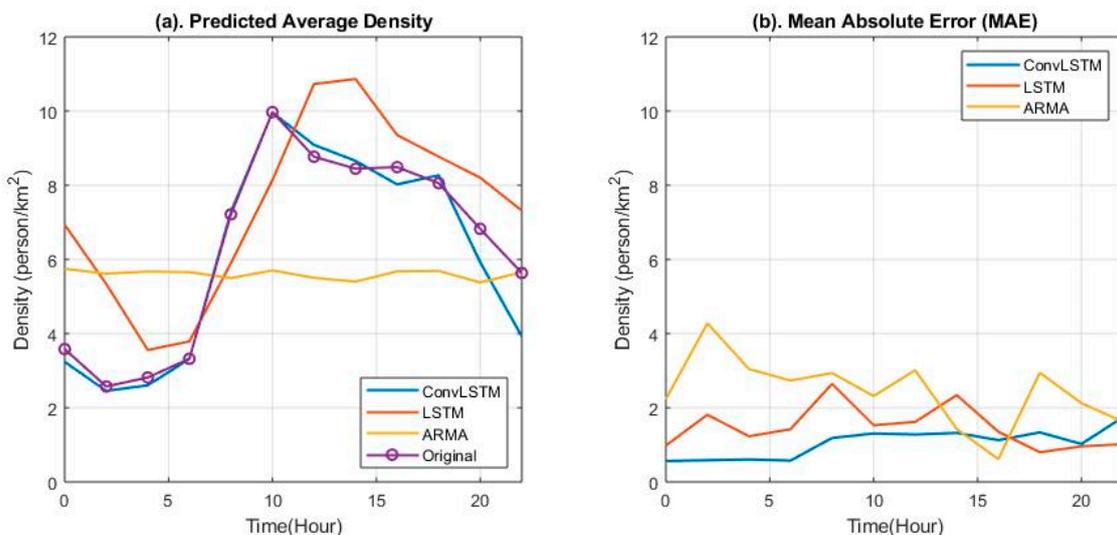
For the LSTM, the trends for the  $R^2$  coefficient determination and the Pearson correlation coefficient ( $R$ ) were similar to those of ConvLSTM, as these both decreased at the start and increased at the end. However, the fluctuation was much bigger for the former than the latter, which means that the prediction by LSTM was worse, and it was dependent on the size of the datasets. Furthermore, ARMA showed even more fluctuation compared to ConvLSTM, where it peaked at 4:00 p.m. and had its lowest point at 4:00 a.m. for the  $R^2$  coefficient determination. Although, in Figure 17b, this got even higher than LSTM from 4:00 am to 8:00 am, ARMA still showed a lower accuracy than ConvLSTM.

It is worth mentioning that both  $R^2$  and  $R$  in the ARMA and LSTM prediction processes fluctuated dramatically over the time. For the ARMA model, this was because it was a linear system model, which by default recognizes the input data as a Gaussian white noise sequence; hence, a relatively stationary sequence would be predicted by a linear system model with a sequence of white noise input, resulting in a larger error when the test value is fluctuating. In contrast, although LSTM was much more accurate than ARMA, even without convolution, a higher total error was generated from the accumulated error from each single cell in the grid. In contrast, because ConvLSTM takes the spatial autocorrelation into consideration, it created less accumulated error and the results of the correlation coefficient and coefficient determination were much better compared to the two baseline models.



**Figure 17.** The change in  $R^2$  coefficient (a, left) and  $R$  correlation coefficient (b, right) for the three methods.

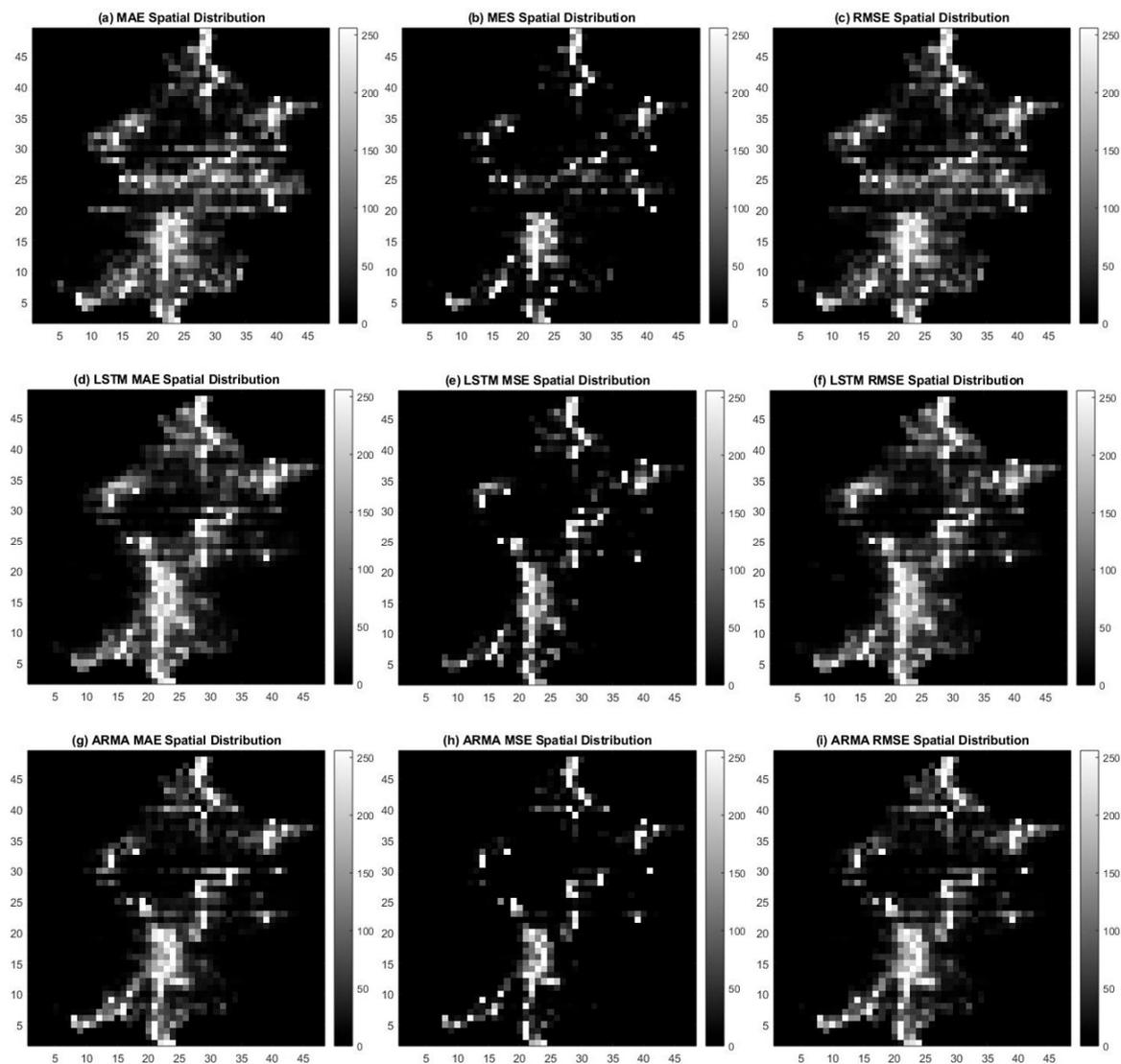
Figure 18a shows the transformation of the mean original and predicted values generated using the three methods over all grids during the whole test day, which can be used to intuitively explain the distance between the original and predicted values in the time series. Figure 18b shows the three mean absolute error (MAE) results generated by Figure 18a. The blue line representing ConvLSTM starts at 0.35, keeps stable to 6:00 a.m. and then increases slightly, and then remains stable to 8:00 p.m., before another increase at 10:00 p.m. The continuous rising trend of the MAE explains that the error of the predicted result processed by the ConvLSTM module grew with the accumulation of error. In contrast, both LSTM and ARMA showed bigger errors during the whole day except at 4:00 p.m. (yellow line) and after 6:00 p.m. (red line). ARMA results were much more stable when the density significantly changed, while LSTM showed a similar trend as time passed, albeit unsynchronized with the original line representing the test data.



**Figure 18.** Transformation of the mean absolute error, and mean tested and predicted values for user density over all grids on 17 February 2015.

### 5.3.2. Results—Assessment of the Prediction Accuracy in Space

In terms of the spatial scale, we extracted the actual results with the corresponding original data and calculated the mean absolute error (MAE), mean squared error (MSE), and root-mean-squared error (RMSE) of every cell for the three approaches. Then, the spatial distribution of the three indices was visualized in the  $48 \times 48$  matrix with 255 grayscale levels (Figure 19). The whiter regions in the graphs correspond to a higher error of prediction for user density. In contrast, the blacker regions mean the error is lower. Intuitively, the error distribution results from the corresponding user density in each cell represent the features in urban areas, which have more errors, while the sub-rural areas have fewer errors. Thus, the distribution of the error scan shows the effect of the prediction models in space.



**Figure 19.** The spatial distribution of errors on 17 February 2015 for the three methods: (a–c) are the result maps for the mean absolute error (MAE), mean squared error (MSE), and root-mean-squared error (RMSE) distributions generated by ConvLSTM; (d–f) are the results generated by the LSTM model in the same order, while (g–i) are those generated by the ARMA model.

Moran’s I analysis of the three indices is shown in Table 3, where the Moran’s indices of MAE for ConvLSTM, LSTM, and ARMA were 0.816785, 0.891316, and 0.859372, respectively. Moran’s indices of MSE were 0.733462, 0.742480, and 0.745777, and the equivalents of RMSE were 0.835418, 0.887711, and 0.858644. All Moran’s index values were between 0.7 to 1, which illustrates that the error of

prediction had a positive spatial autocorrelation. However, the indices of ConvLSTM for MAE, MSE, and RMSE were lower than those for the other models. This means that the spatial relationship's negative influence decreased for ConvLSTM, which considers spatial autocorrelation when using a convolution.

**Table 3.** Moran's I indices of the spatial distribution of the three methods. LSTM—long short-term memory; ConvLSTM—convolution LSTM; ARMA—autoregressive moving average; MAE—mean absolute error; MSE—mean squared error; RMSE—root-mean-squared error.

	MAE	MSE	RMSE
ConvLSTM	0.816785	0.733462	0.835418
LSTM	0.891316	0.742480	0.887711
ARMA	0.859372	0.745777	0.858644

## 6. Conclusions

This study used a convolutional long short-term memory (ConvLSTM) module to predict the activity of mobile phone users' distribution with a traditional time-series predicted model (ARMA) and a popular deep learning method (LSTM) as baselines for comparison. The evaluation results showed that the predicted density correlated much better with the original data at the temporal and spatial scales used when using ConvLSTM as compared to the other two methods, which do not consider the spatial autocorrelation. The MAE of the predicted results of ConvLSTM ranged from 0.6 to 1.8 over 17 February 2015, which means that the model was much more stable and accurate than the other two baseline methods. Moran's I index for the error distribution was still lower than that of the other baseline methods in space, showing the positive effect on the errors caused by the spatial autocorrelation. Our proposed method, ConvLSTM, can help us better understand mobile phone users' population dynamics and help more accurately calculate the population density for at least one day in advance based upon 15 days of history. Utilizing this method, population activity hotspots can be predicted, and this facilitates more dynamic and more efficient city resource and commercial deployment.

In the future, the method of converting mobile phone users into a population distribution and its application for a more dynamic prediction of people distribution using ConvLSTM will be investigated so as to improve the forecasting ability. In addition, we will use the mobile phone signal data and real people density data to analyze the demographics and people flow, so as to detect their residence and mobility paths, for which we can then use ConvLSTM to predict if their distribution can be converted into grids. Furthermore, some occupations, such as bus drivers and bank staff, based on their location and movement patterns, can be detected by mining their mobile phone data, which can then be used to predict the distribution of mobile users in a city via a spatial deep learning model.

**Author Contributions:** G.Z. modelled the ConvLSTM network, analyzed data and drew the main conclusions. X.R. summarized the framework of the article, reviewed the related work and sorted through most of the references. S.P. directed and critically revised the overall methodology and its presentation. X.S. preprocessed the data and built the main dataset for prediction. Y.F. modelled ARMA and LSTM as baseline methods and analyzed and improved their model formulas. Finally, Z.M. provided the cross-validation strategy.

**Funding:** The research reported herein was sponsored by the National Key Research and Development Program of China (Grant No. 2017YFB0503605), the National Natural Science Foundation of China (Grant No. 41771478), the Fundamental Research Funds for the Central Universities (Grant No. 2019B02514), and the Beijing Natural Science Foundation (Grant No. 8172046).

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Becker, R.A.; Caceres, R.; Hanson, K.; Loh, J.M.; Urbanek, S.; Varshavsky, A.; Volinsky, C. A tale of one city: Using cellular network data for urban planning. *IEEE Pervasive Comput.* **2011**, *10*, 18–26. [[CrossRef](#)]
2. De Nadai, M.; Staiano, J.; Larcher, R.; Sebe, N.; Quercia, D.; Lepri, B. The death and life of great Italian cities: A mobile phone data perspective. In Proceedings of the 25th International Conference on World Wide Web, Montréal, QC, Canada, 11–15 April 2016; pp. 413–423.
3. Tao, S.; Corcoran, J.; Mateo-Babiano, I.; Rohde, D. Exploring Bus Rapid Transit passenger travel behaviour using big data. *Appl. Geogr.* **2014**, *53*, 90–104. [[CrossRef](#)]
4. Li, Q.; Xu, B.; Ma, Y.; Chung, T. Real-time monitoring and forecast of active population density using mobile phone data. In Proceedings of the National Conference on Big Data Technology and Applications, Harbin, China, 25–26 December 2015; pp. 116–129.
5. Traag, V.; Browet, A.; Calabrese, F.; Morlot, F. Social event detection in massive mobile phone data using probabilistic location inference. In Proceedings of the 2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing, Boston, MA, USA, 9–11 October 2011.
6. Zhou, J.; Pei, H.; Wu, H. Early Warning of Human Crowds Based on Query Data from Baidu Maps: Analysis Based on Shanghai Stampede. In *Big Data Support of Urban Planning and Management*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 19–41.
7. Bengtsson, L.; Lu, X.; Thorson, A.; Garfield, R.; Von Schreeb, J. Improved response to disasters and outbreaks by tracking population movements with mobile phone network data: A post-earthquake geospatial study in Haiti. *PLoS Med.* **2011**, *8*, e1001083. [[CrossRef](#)] [[PubMed](#)]
8. Min, G.Y.; Jeong, D.H. Research on assessment of impact of big data attributes to disaster response decision-making process. *J. Soc. E-Bus. Stud.* **2013**, *18*. [[CrossRef](#)]
9. Wilson, R.; Erbach-Schoenberg, E.; Albert, M.; Power, D.; Tudge, S.; Gonzalez, M.; Guthrie, S.; Chamberlain, H.; Brooks, C.; Hughes, C. Rapid and near real-time assessments of population displacement using mobile phone data following disasters: The 2015 Nepal Earthquake. *PLoS Curr.* **2016**, *8*. [[CrossRef](#)] [[PubMed](#)]
10. Faria, N.R.; Rambaut, A.; Suchard, M.A.; Baele, G.; Bedford, T.; Ward, M.J.; Tatem, A.J.; Sousa, J.D.; Arinaminpathy, N.; Pépin, J. The early spread and epidemic ignition of HIV-1 in human populations. *Science* **2014**, *346*, 56–61. [[CrossRef](#)]
11. Lopez, D.; Gunasekaran, M.; Murugan, B.S.; Kaur, H.; Abbas, K.M. Spatial big data analytics of influenza epidemic in Vellore, India. In Proceedings of the 2014 IEEE International Conference on Big Data (Big Data), Washington, DC, USA, 27–30 October 2014; pp. 19–24.
12. Vespignani, A. Predicting the behavior of techno-social systems. *Science* **2009**, *325*, 425–428. [[CrossRef](#)]
13. Liu, Z.; Ma, T.; Du, Y.; Pei, T.; Yi, J.; Peng, H. Mapping hourly dynamics of urban population using trajectories reconstructed from mobile phone records. *Trans. GIS* **2018**, *22*, 494–513. [[CrossRef](#)]
14. Deville, P.; Linard, C.; Martin, S.; Gilbert, M.; Stevens, F.R.; Gaughan, A.E.; Blondel, V.D.; Tatem, A.J. Dynamic population mapping using mobile phone data. *Proc. Natl. Acad. Sci. USA* **2014**, *111*, 15888–15893. [[CrossRef](#)] [[PubMed](#)]
15. Deng, L.; Xie, Y.; Huang, D. Bicycle-sharing facility planning base on riding spatio-temporal data. *Planners* **2017**, *10*, 82–88.
16. Davis, B.; Lockwood, A.; Alcott, P.; Pantelidis, I.S. *Food and Beverage Management*; Routledge: London, UK, 2018.
17. Mei, H.; Ma, A.; Poslad, S.; Oshin, T.O. Short-term traffic volume prediction for sustainable transportation in an urban area. *J. Comput. Civ. Eng.* **2013**, *29*, 04014036. [[CrossRef](#)]
18. Kang, L.; Poslad, S.; Wang, W.; Li, X.; Zhang, Y.; Wang, C. A public transport bus as a flexible mobile smart environment sensing platform for IoT. In Proceedings of the 2016 12th International Conference on Intelligent Environments (IE), London, UK, 14–16 September 2016; pp. 1–8.
19. Zhang, Z.; Poslad, S. A new post correction algorithm (PoCoA) for improved transportation mode recognition. In Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics, Manchester, UK, 13–16 October 2013; pp. 1512–1518.
20. Titkov, L.; Poslad, S.; Tan, J.J. An integrated approach to user-centered privacy for mobile information services. *Appl. Artif. Intell.* **2006**, *20*, 159–178. [[CrossRef](#)]

21. Oshin, T.O.; Poslad, S.; Zhang, Z. Energy-efficient real-time human mobility state classification using smartphones. *IEEE Trans. Comput.* **2015**, *64*, 1680–1693. [[CrossRef](#)]
22. Zheng, Y.; Zheng, W.; Xie, X. Collaborative Location and Activity Recommendations. U.S. Patent 8,719,198, 6 May 2014.
23. Jiang, S.; Ferreira, J.; Gonzalez, M.C. Activity-based human mobility patterns inferred from mobile phone data: A case study of Singapore. *IEEE Trans. Big Data* **2017**, *3*, 208–219. [[CrossRef](#)]
24. World Telecommunication Development Conference Dubai, United Arab Emirates. Available online: <https://www.itu.int/en/ITU-D/Conferences/WTDC/Pages/default.aspx> (accessed on 7 May 2019).
25. Blondel, V.D.; Decuyper, A.; Krings, G. A survey of results on mobile phone datasets analysis. *EPJ Data Sci.* **2015**, *4*, 10. [[CrossRef](#)]
26. Kraemer, M.U.; Hay, S.I.; Pigott, D.M.; Smith, D.L.; Wint, G.W.; Golding, N. Progress and challenges in infectious disease cartography. *Trends Parasitol.* **2016**, *32*, 19–29. [[CrossRef](#)] [[PubMed](#)]
27. Toole, J.L.; Lin, Y.-R.; Muehlegger, E.; Shoag, D.; González, M.C.; Lazer, D. Tracking employment shocks using mobile phone data. *J. R. Soc. Interface* **2015**, *12*. [[CrossRef](#)]
28. Lu, X.; Wrathall, D.J.; Sundsøy, P.R.; Nadiruzzaman, M.; Wetter, E.; Iqbal, A.; Qureshi, T.; Tatem, A.; Canright, G.; Engø-Monsen, K. Unveiling hidden migration and mobility patterns in climate stressed regions: A longitudinal study of six million anonymous mobile phone users in Bangladesh. *Glob. Environ. Chang.* **2016**, *38*, 1–7. [[CrossRef](#)]
29. Zhang, G.; Rui, X.; Fan, Y. Critical review of methods to estimate PM<sub>2.5</sub> concentrations within specified research region. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 368. [[CrossRef](#)]
30. Balk, D.; Yetman, G. *The Global Distribution of Population: Evaluating the Gains in Resolution Refinement*; Center for International Earth Science Information Network (CIESIN), Columbia University: New York, NY, USA, 2004.
31. Bhaduri, B.; Bright, E.; Coleman, P.; Urban, M.L. LandScan USA: A high-resolution geospatial and temporal modeling approach for population distribution and dynamics. *GeoJournal* **2007**, *69*, 103–117. [[CrossRef](#)]
32. Doxsey-Whitfield, E.; MacManus, K.; Adamo, S.B.; Pistolesi, L.; Squires, J.; Borkovska, O.; Baptista, S.R. Taking advantage of the improved availability of census data: A first look at the gridded population of the world, version 4. *Pap. Appl. Geogr.* **2015**, *1*, 226–234. [[CrossRef](#)]
33. Lloyd, C.T.; Sorichetta, A.; Tatem, A.J. High resolution global gridded data for use in population studies. *Sci. Data* **2017**, *4*, 170001. [[CrossRef](#)]
34. Department of International Economic and Social Affairs. Statistical Office. *Principles and Recommendations for Population and Housing Censuses*; revision 2; United Nations: New York, NY, USA, 2008.
35. Lwin, K.K.; Sugiura, K.; Zetsu, K. Space-time multiple regression model for grid-based population estimation in urban areas. *Int. J. Geogr. Inf. Sci.* **2016**, *30*, 1579–1593. [[CrossRef](#)]
36. Douglass, R.W.; Meyer, D.A.; Ram, M.; Rideout, D.; Song, D. High resolution population estimates from telecommunications data. *EPJ Data Sci.* **2015**, *4*, 4. [[CrossRef](#)]
37. Kang, C.; Ma, X.; Tong, D.; Liu, Y. Intra-urban human mobility patterns: An urban morphology perspective. *Phys. A: Stat. Mech. Its Appl.* **2012**, *391*, 1702–1717. [[CrossRef](#)]
38. Khodabandelou, G.; Gauthier, V.; El-Yacoubi, M.; Fiore, M. Population estimation from mobile network traffic metadata. In Proceedings of the 2016 IEEE 17th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), Coimbra, Portugal, 21–24 June 2016; pp. 1–9.
39. Gabrielli, L.; Furletti, B.; Trasarti, R.; Giannotti, F.; Pedreschi, D. City users' classification with mobile phone data. In Proceedings of the 2015 IEEE International Conference on Big Data (Big Data), Santa Clara, CA, USA, 29 October–1 November 2015; pp. 1007–1012.
40. Pappalardo, L.; Vanhoof, M.; Gabrielli, L.; Smoreda, Z.; Pedreschi, D.; Giannotti, F. An analytical framework to nowcast well-being using mobile phone data. *Int. J. Data Sci. Anal.* **2016**, *2*, 75–92. [[CrossRef](#)]
41. Rojas, I.; Valenzuela, O.; Rojas, F.; Guillén, A.; Herrera, L.J.; Pomares, H.; Marquez, L.; Pasadas, M. Soft-computing techniques and ARMA model for time series prediction. *Neurocomputing* **2008**, *71*, 519–537. [[CrossRef](#)]
42. Ji, W.; Chee, K.C. Prediction of hourly solar radiation using a novel hybrid model of ARMA and TDNN. *Sol. Energy* **2011**, *85*, 808–817. [[CrossRef](#)]

43. Ling-ling, L.; Li, J.-H.; He, P.-J.; Wang, C.-S. The use of wavelet theory and ARMA model in wind speed prediction. In Proceedings of the 2011 1st International Conference on Electric Power Equipment-Switching Technology, Xi'an, China, 23–27 October 2011; pp. 395–398.
44. Zhang, J.; Zheng, Y.; Qi, D.; Li, R.; Yi, X. DNN-based prediction model for spatio-temporal data. In Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Burlingame, CA, USA, 31 October–3 November 2016; p. 92.
45. Huang, W.; Song, G.; Hong, H.; Xie, K. Deep Architecture for Traffic Flow Prediction: Deep Belief Networks with Multitask Learning. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 2191–2201. [[CrossRef](#)]
46. Polson, N.G.; Sokolov, V.O. Deep learning for short-term traffic flow prediction. *Transp. Res. Part C Emerg. Technol.* **2017**, *79*, 1–17. [[CrossRef](#)]
47. Zhao, Z.; Chen, W.; Wu, X.; Chen, P.C.; Liu, J. LSTM network: A deep learning approach for short-term traffic forecast. *IET Intell. Transp. Syst.* **2017**, *11*, 68–75. [[CrossRef](#)]
48. Qiu, M.; Zhao, P.; Zhang, K.; Huang, J.; Shi, X.; Wang, X.; Chu, W. A Short-Term Rainfall Prediction Model using Multi-Task Convolutional Neural Networks. In Proceedings of the 2017 IEEE International Conference on Data Mining (ICDM), New Orleans, LA, USA, 18–21 November 2017; pp. 395–404.
49. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 3104–3112.
50. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; Woo, W.-C. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 802–810.
51. Blumenstock, J.; Cadamuro, G.; On, R. Predicting poverty and wealth from mobile phone metadata. *Science* **2015**, *350*, 1073–1076. [[CrossRef](#)]
52. Bogomolov, A.; Lepri, B.; Staiano, J.; Oliver, N.; Pianesi, F.; Pentland, A. Once upon a crime: Towards crime prediction from demographics and mobile data. In Proceedings of the 16th International Conference on Multimodal Interaction, Istanbul, Turkey, 12–16 November 2014; pp. 427–434.
53. Zhao, R.; Yan, R.; Wang, J.; Mao, K. Learning to monitor machine health with convolutional bi-directional lstm networks. *Sensors* **2017**, *17*, 273. [[CrossRef](#)] [[PubMed](#)]
54. Althé, F.; de La Fortelle, A. An LSTM network for highway trajectory prediction. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 353–359.
55. Liu, T.; Bao, J.; Wang, J.; Zhang, Y. A Hybrid CNN–LSTM Algorithm for Online Defect Recognition of CO<sub>2</sub> Welding. *Sensors* **2018**, *18*, 4369. [[CrossRef](#)]
56. Rad, N.M.; Kia, S.M.; Zarbo, C.; Jurman, G.; Venuti, P.; Furlanello, C. Stereotypical motor movement detection in dynamic feature space. In Proceedings of the 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), Barcelona, Spain, 12–15 December 2016; pp. 487–494.
57. Liu, Y.; Zheng, H.; Feng, X.; Chen, Z. Short-term traffic flow prediction with Conv-LSTM. In Proceedings of the 2017 9th International Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, China, 11–13 October 2017; pp. 1–6.
58. Qiao, H.; Wang, T.; Wang, P.; Shibin, Q.; Lan, Z. A Time-Distributed Spatiotemporal Feature Learning Method for Machine Health Monitoring with Multi-Sensor Time Series. *Sensors* **2018**, *18*, 2932. [[CrossRef](#)]
59. Yuan, Z.; Zhou, X.; Yang, T. Hetero-convlstm: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 984–992.
60. Andrews, S.; Ellis, A.; Shaw, H.; Lukasz, P. Beyond self-report: Tools to compare estimated and real-world smartphone use. *PLoS ONE* **2015**, *10*, e0139004. [[CrossRef](#)]
61. Brassel, K.E.; Reif, D. A procedure to generate Thiessen polygons. *Geogr. Anal.* **1979**, *11*, 289–303. [[CrossRef](#)]
62. Rosenblatt, M. Remarks on some nonparametric estimates of a density function. *Ann. Math. Stat.* **1956**, 832–837. [[CrossRef](#)]
63. Parzen, E. On estimation of a probability density function and mode. *Ann. Math. Stat.* **1962**, *33*, 1065–1076. [[CrossRef](#)]
64. Dehnad, K. *Density Estimation for Statistics and Data Analysis*; Taylor & Francis Group: Abingdon, UK, 1987.

65. Fu, R.; Zhang, Z.; Li, L. Using LSTM and GRU neural network methods for traffic flow prediction. In Proceedings of the 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), Wuhan, China, 11–13 November 2016; pp. 324–328.
66. Chen, K.; Zhou, Y.; Dai, F. A LSTM-based method for stock returns prediction: A case study of China stock market. In Proceedings of the 2015 IEEE International Conference on Big Data (Big Data), Santa Clara, CA, USA, 29 October–1 November 2015; pp. 2823–2824.
67. Gers, F.A.; Schmidhuber, J.; Cummins, F. Learning to forget: Continual prediction with LSTM. In Proceedings of the 1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470), Edinburgh, UK, 7–10 September 1999.
68. Boots, B.N.; Getis, A. *Point Pattern Analysis*; SAGE Publications, Incorporated: Beverly Hills, CA, USA, 1988; Volume 8.
69. Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. *IJCAI* **1995**, *14*, 1137–1145.
70. Bergmeir, C.; Benítez, M. On the use of cross-validation for time series predictor evaluation. *Inf. Sci.* **2012**, *191*, 192–213. [[CrossRef](#)]
71. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).