Research Article

A Privacy-Preserving Crowd Flow Prediction Framework Based on Federated Learning during Epidemics

Weiya Wang,1,2 Geng Yang,1,3,4 Lin Bao,2 Ke Ma,1 and Hao Zhou1

1School of Computer Science, Nanjing University of Post and Telecommunication, Nanjing, China
2China Telecom Jiangsu Branch, Nanjing, China
3Key Laboratory of Broadband Wireless Communication and Sensor Networks Technology of Ministry of Education, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210003, China
4Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing, Jiangsu 210003, China

Correspondence should be addressed to Geng Yang; yangg@njupt.edu.cn

Received 28 February 2022; Accepted 4 October 2022; Published 26 October 2022

Academic Editor: Mohamed Amine Ferrag

Copyright © 2022 Weiya Wang 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.

Predicting and managing the movement of people in a region during epidemics’ outbreak is an important step in preventing outbreaks. The protection of user privacy during the outbreak has become a matter of public concern in recent years, yet deep learning models based on datasets collected from mobile devices may pose privacy and security issues. Therefore, how to develop an accurate crowd flow prediction while preserving privacy is a significant problem to be solved, and there is a tradeoff between these two objectives. In this paper, we propose a privacy-preserving mobility prediction framework via federated learning (CFPF) to solve this problem without significantly sacrificing the prediction performance. In this framework, we designed a deep and embedding learning approach called “Multi-Factors CNN-LSTM” (MFCL) that can help to explicitly learn from human trajectory data (weather, holidays, temperature, and POI) during epidemics. Furthermore, we improve the existing federated learning framework by introducing a clustering algorithm to classify clients with similar spatio-temporal characteristics into the same cluster, and select servers at the center of the cluster as edge central servers to integrate the optimal model for each cluster and improve the prediction accuracy. To address the privacy concerns, we introduce local differential privacy into the FL framework which can facilitate collaborative learning with uploaded gradients from users instead of sharing users’ raw data. Finally, we conduct extensive experiments on a realistic crowd flow dataset to evaluate the performance of our CFPF and make a comparison with other existing models. The experimental results demonstrate that our solution can not only achieve accurate crowd flow prediction but also provide a strong privacy guarantee at the same time.

1. Introduction

The gathering and movement of people is an important factor in infectious disease outbreaks. How to predict the crowd flow while protecting the privacy of users will help in the prevention and control of outbreaks. Recently, the coronavirus disease 2019 (COVID-19) outbreak has quickly spread globally and has shown a significant impact on public health and the economy [1, 2]. Preliminary evidence clearly proved that the virus could cluster with outdoor particulate matter (PM) under certain circumstances [3, 4]. The coronavirus disease 2019 (COVID-19) has many cases of human-to-human transmission. A more effective way to reduce transmission and control the outbreak is to impose lockdown and isolation of persons [5]. A better understanding of human mobility can help to more easily control the spreading of contagious diseases by limiting contact among individuals [6], since the transmission of infected people from one place to another is an important way to infect the susceptible ones, either in a small scale area [7, 8] or from a worldwide viewpoint [9–12]. Therefore, during an epidemic, predicting the trajectory of the population allows for more efficient surveillance and management of the risk area, reducing the spread of the epidemic.

The prevalence of mobile Internet has provided massive amounts of continuous data for crowd trajectory prediction...
during epidemics. In this paper, we focus on the travel areas of people in the city during the epidemic at different times of the day. To predict the crowd flow of urban region, some state-of-the-art deep learning models have been proposed [13–17]. Nevertheless, the existing approaches are still not well suited for instant flow prediction especially for rapid flow variation scenarios usually due to social emergency incidents/accidents. In the past, most user trajectory prediction approaches have the following limitations: (1) ignored the influence of other factors, such as weather conditions and functional attributes of the area, resulting in a single data source structure and compromised prediction accuracy. The weather has proven to be useful as features for a crowd flow prediction tasks such as bike-sharing demand prediction [18] and metro passenger flow forecasting [19]. In many application scenarios, we tend to focus more on functional regions such as business area, residential area, administrative area, and scenic spot, which are more connected with human behavior. Dividing the regions by functional attributes is more suitable for the accurate expression of human dynamics and population movement patterns. (2) Lack of spatio-temporal adaptive models for fine-grained feature processing of crowd flow. The human trajectory during the epidemic is affected by the epidemic prevention policy, time, weather, and other factors, which will show certain regularity. Some recent work has used deep neural networks to solve this problem [14]. However, the accuracy of a deep neural network approach relies heavily on its architecture and hyper-parameters [20]. In addition, these models [21–26] are designed for a specific type of data and are difficult to be generalized to handle other types of spatio-temporal data.

To tackle the aforementioned challenges, we propose a deep learning approach with fusion of multiple feature sources (MFCL) to predict crowd flows. In particular, our deep learning approach combines a convolutional neural network (CNN) component and a long short-term memory (LSTM) component. Since human mobility is closely related to the functional attributes of regions, this method divides the city into $M$ grids and determines the functional attributes of regions according to the POI attributes of each region. Firstly, we extract the feature factors that affect the human mobility prediction, such as weather and time of day, as the feature vectors for model training. Then we use CNN for local feature extraction and LSTM for model training and prediction.

In the field of crowd mobility prediction, current prediction models are mainly divided into two categories: (1) personal model with self-information; and (2) joint model with population information. Personal model includes Markov model, hidden Markov model, and decision tree. These works try to model the mobility behavior of individuals with various methods by only utilizing their own mobility data records. The single personal machine learning model performs poorly in real-world scenarios with sparse and limited data records. We consider use a joint training model which uses multi-party mobile data to train the federated model to solve the problem of sparsity of data records. However, the dataset used in a joint training model may contain a large amount of user’s private data and there are substantial user privacy and security issues in the process of data upload and centralized model training. How to develop accurate crowd flow prediction while preserving privacy is a major problem that needs to be addressed. Madan and Goswami [27] introduced a privacy control mechanism based on “k-anonymous diffusion,” which can complete taxi-order scheduling without leaking user privacy. Le Ny et al. [28] proposed a differentially private real-time traffic state estimator system to predict traffic flow. In order to improve the prediction accuracy of crowd movement and to address the balance between data availability and privacy protection, we introduce a privacy-preserving crowd flow prediction framework based on Federated Learning (FL) [29, 30] during epidemics. In our framework, what is uploaded to the server is not data, but only the intermediate weight gradient of the prediction model. First, we introduce a differential privacy protection mechanism for local data based on the risk level of different regions during the epidemic. Secondly, we combine the idea of clustering to design a three-layer Federated Learning framework with practicality. Finally, we propose an efficient aggregation strategy that aggregates model parameters from different regional devices to achieve robust convergence. In FL, private data is stored and analyzed in the local devices, so our approach protects the security of the user’s private data while ensuring the accuracy of the prediction. The major contributions of this paper are summarized as follows:

1. To ensure the accuracy of crowd movement prediction in different areas of the city during the epidemic, we propose a novel deep prediction model called “Multi-Factors CNN-LSTM” (MFCL) that incorporates multi-factors related to human trajectory data (weather, holidays, temperature, and POI) during an epidemic. The MFCL consists of an embedding component, an urban regional classification model (URCM), a convolutional neural network (CNN) component, and a long short-term memory (LSTM) component. The embedding component can capture the categorical feature information and identify correlated features. The URCM identifies and classifies areas of the city according to their functional attributes. In theory, the CNN component can extract features of local trajectory while the LSTM component has the benefits of maintaining a long-term memory of historical data.

2. In order to solve the problem of the balance between the utility and security of user location data privacy, we propose CFPF, a novel Privacy-preserving Crowd Flow Prediction framework via federated learning, to protect the user privacy and get accurate crowd flow forecasts. The framework balances the data availability and security in a real environment by introducing a differential privacy protection mechanism for local data based on the risk level of different regions during the epidemic. Privacy-preserving methods used in CFPF satisfy the differential privacy-preserving property.
(3) We simulate the crowd flow environment during the COVID-19 epidemic on real desensitized mobile 4G data and validate the effectiveness of our crowd flow prediction model and privacy-preserving framework after extensive experiments. The experimental results show that the CFPF achieves a better balance of prediction accuracy and data privacy protection.

The rest of this paper is organized as follows: We first formulate the related work in Section 2. Then, we introduce details of the crowd flow model in epidemic and differential privacy and geo-indistinguishability in Section 3. In Section 4, we proposed a privacy-preserving crowd flow prediction framework based on federated learning. Specifically, select key features, introduce crowd flow prediction models, and build a federal learning framework. After the framework description, we apply CFPF on real-world mobility datasets with multi-user environment and conduct extensive analysis on the prediction performance and the effectiveness of the privacy-preserving mechanism in Section 5. After systematically reviewing the related works in Section 6, we conclude our paper.

2. Related Work

2.1. Crowd Flow Prediction. Crowd flow prediction, instead, is the task of forecasting the incoming and outgoing flows of people on a geographic region, which has an impact on public safety, the definition of on-demand services [31, 32], and traffic optimization [33–38]. Predicting urban crowd flow helps urban management, especially in the alert phase of infectious disease outbreaks, and knowing in advance which areas of the city will be the gathering areas for crowd flow is essential for city managers to take measures in advance. Modeling citywide crowd flows has been studied by several recent works. The prediction of crowd flow is mainly the prediction of spatio-temporal series that can generally be divided into two categories: parametric model-based approaches and nonparametric model-based approaches. Traditionally, parametric time series models include ARIMA such as Kohonen-ARIMA (KARIMA) [39], subset ARIMA [40], seasonal ARIMA ([41] and [39]), and regression models with spatio-temporal regularization [42]. Williams [43] used ARIMA model to predict the short-term traffic flow. The advantage of the parametric model is that the model is highly transparent and easy to understand. These solutions usually take less time than nonparametric models. However, the complex temporal and spatial correlations of the crowd flow data are hard to be captured due to their limited learning ability.

The nonparametric model-based approaches assume that the data distribution cannot be defined in terms of a finite set of parameters. Aude Hofleitner [44] proposed a dynamic Bayesian network to predict the traffic flow of an entire arterial road network with hundreds of road links based on sparse probe data. With the development of data storage and computing power, neural network-based models have become popular in crowd flow prediction. Davis and Nihan et al. in [45] proposed k-NN model for short-term traffic flow prediction. Reference [46] proposed a Spatio-Temporal Dynamic Network (STDN) model which combines CNN model and RNN model to capture both spatial and temporal correlations for traffic prediction based on road network. Reference [47] proposed a deep learning framework for traffic flow forecasting named the Diffusion Convolutional Recurrent Neural Network (DCRNN) to model the traffic flow as a diffusion process on a directed road graph. The authors of [48] introduced Long Short-Term Recurrent Neural Networks (LSTMs) which is capable of learning long or short-term dependencies for traffic flow prediction. However, LSTM is not good at mining spatial correlation relations. Some of the research work proposes solutions with a combination of CNN and LSTM for both spatial and temporal dependent flow prediction. Zhang et al. [49, 50] proposed a deep learning model ST-ResNet to collectively forecast the inflow and outflow of crowds in each region of a city based on DeepST. ST-ResNet contains four major components: closeness, period, trend, and external unit. ST-ResNet provides a fusion mechanism which allows assigning different weights to each output of the closeness, period, and trend unit. This method also takes into account external factors such as weather, holiday event, and metadata. The ST-ResNet can only predict one step ahead and cannot be used for predicting multi-step ahead according to the complex spatio-temporal history. Yao [46] proposed a Spatial-Temporal Dynamic Network (STDN) model for traffic prediction. Reference [14] proposed a multi-task deep learning framework to simultaneously predict the node flow and edge flow in a constructed urban spatio-temporal network. Reference [51] proposes DeepSTN+ architecture that improves the accuracy of prediction and stability of the model compared to ST-ResNet. However, most of these models are supervised and require a large amount of training data. Due to the large number of parameters in deep learning models, scarcity of training data can lead to overfitting. Crowd flow prediction can also be treated as a spatio-temporal graph (STG) prediction problem, where each node represents a region with time-varying flow. For example, Sun et al. [52] used spatial graph convolution to construct a multi-view graph convolutional network (MVGCN) for crowd flow prediction. It uses each graph convolution network to predict the inflow and outflow of people in a region integrating geospatial locations through spatial graph convolution.

2.2. Privacy Issues for Crowd Flow Prediction Systems. In the research of crowd flow prediction systems, a large amount of user data needs to be trained, which may result in the leakage of user privacy [53]. Brain et al. in [54] proposed a data-sharing algorithm based on information-theoretic k-anonymity principle. However, this algorithm may leak privacy during data-sharing operations. Zhou [55] proposed a privacy-preserving transportation traffic measurement scheme for cyber-physical road systems by using maximum-likelihood estimation (MLE) to obtain the prediction result. Reference [56] proposes a scheme for protecting the location of vehicles using encryption methods. These privacy-
preserving methods mentioned above sacrifice the accuracy of the prediction results and cannot handle large amounts of data in a short period of time.

FL is a new machine learning paradigm that encourages participants to collaboratively train a globally shared model released by the central server. FL effectively eliminates privacy threats to individuals by performing prediction tasks locally. However, traditional FL relies heavily on the central server which manages local model updates and thus cannot avoid the security vulnerability of a single point of failure [57]. Xu [58] proposes the first privacy-preserving federated learning framework VerifyNet. The [59, 60] applied edge computing to a federated learning framework to obtain faster training time and better privacy preservation. In order to improve the convergence efficiency of FL, the literature [61] used a momentum gradient descent method in the local update process of FL. The FL framework alone has the following security threats: (1) Privacy leakage after a single point of attack on the central server. (2) Malicious reporting of false data from the local server affects the overall prediction model. Li in [62] combines blockchain with FL and introduces a crowdsourcing framework to improve the overall security of the framework. Reference [63] proposes a novel privacy-preserving Mobility prediction framework that introduces differential privacy into the optimization of the local models to obtain the controlled and privacy-preserving embedding table for secure model sharing via federated learning to protect the user’s privacy. There are two main schemes to add DP noise to FL: (1) The global DP scheme that adds noise to the aggregated gradient information on the server side before the gradient information is distributed to clients [64, 65]. (2) The local DP scheme adds noise to the gradient information on users before it is uploaded to the server. Additionally, DP can be applied to various FL algorithms such as FedSGD [66] and FedAvg [67]. Reference [68] proposes a hybrid approach that integrates federated learning with local differential privacy techniques. They deploy LDP mechanisms to gradients in FedSGD algorithm. Hu et al. [69] propose a privacy-preserving FL approach for learning effective personalized models. They use Gaussian mechanism, a centralized DP mechanism, to protect the privacy of the model.

2.3. Limitations of Previous Studies. In summary, previous studies failed to address the following issues: First, these privacy-preserving approaches based on federated learning fail to achieve the tradeoff between accuracy and privacy, leading to significant performance reduction in the mobility prediction task. Second, although some previous studies consider the impacts of weather conditions and other features on crowd flow prediction, most of them lack spatio-temporal adaptive model for fine-grained feature processing of crowd flow. In contrast to the methods mentioned above, we focus on the task of moving predictions for different risk populations during epidemics. We introduce differential privacy into the optimization of local models and propose a privacy-preserving framework that incorporates spatio-temporal adaptive model during epidemics, which balances data utility with secure privacy to overcome the above limitations of previous studies.

3. Preliminaries

3.1. Crowd Flow Model in Epidemics

Definition 1. Cell region: The city is divided into a $m \times n$ grid map based on the longitude and latitude. Each grid is an equal-sized cell region. We denote all the cell regions as $R = \{r_{1,1}, \ldots, r_{i,j}, \ldots, r_{m,n}\}$, where $r_{i,j}$ is the $i$-th row and $j$-th column cell region of the grid map.

Definition 2. Region: Based on the a priori knowledge and POIs of the city, the city is divided into different nonintersecting parts with basic functional attributes (such as business area, residential area, and administrative area) and road characteristics, which is so called regions. Each region has a specific ID and contains a number of cell region as shown in Figure 1.

Definition 3. Crowd Flow: The crowd flow of region is the number of individuals (e.g., people or vehicles) moving into/out of the corresponding region during a time interval $t$, as shown in Figure 1. $C_{Tr} \subseteq \Delta$ denotes inflows and outflows of all the cell regions during $t$. In a real geographical context, a crowd flow consists of multiple ($\geq 2$) travelers whose trajectory is similar.

$$C_{Tr} = \{\text{position \rightarrow t} \in \Delta t \wedge \text{position \rightarrow (x, y)} \in r_{i,j} | r_{i,j}\},$$

(1)

where the $r_{i,j}$ indicates a cell region and position is the current position of the individual. Let $C$ be the set of locations of all the individual trajectories then the inflow to a cell region $r_{i,j}$ is the number of people that stay in $r_{i,j}$ at time $t$ but not in $r_{i,j}$ at time $t - 1$.

$$\text{inflow} = \sum_{\text{Tr} \in C} \lceil t > 1: r_{i,j} \notin C_{Tr-1} \wedge r_{i,j} \in C_{Tr} \rceil.$$  

(2)

The out flow from a cell region $r_{i,j}$ is the number of people that stay in $r_{i,j}$ at time $t$ but no longer in $r_{i,j}$ at time $t + 1$.

$$\text{outflow} = \sum_{\text{Tr} \in C} \lceil t > 1: r_{i,j} \in C_{Tr} \wedge r_{i,j} \notin C_{Tr+1} \rceil.$$  

(3)

Definition 4. Trajectory: A location record $p = \langle id, l, t \rangle$ can be denoted as a tuple of three elements: user id, location identification $l$, and timestamp $t$. A trajectory is defined as an ordered set of location records $\{p_1, p_2, \ldots, p_n\}$, i.e., $\text{tr}_{id} = \{\langle l_1, t_1 \rangle, \langle l_2, t_2 \rangle, \ldots, \langle l_n, t_n \rangle\}$. Generally, $l$ is the latitude and longitude of the user or the ID of the region. Considering the mobility characteristics of human, we choose 5 minutes as the default trajectory time sampling point based on the time interval of the 4G signal reporting to the base station. The time sampling point can be easily adapted to the requirement.
Definition 5. Person Trajectory Similarity: The person trajectory similarity $\Phi(i, j)$ is a measure of the overlap of the trajectory owned by the user $i$ and $j$. The higher the $\Phi(i, j)$, the more likely two users own the same trajectory. In this paper, we define the $\Phi(i, j)$ as the same cell region through which the trajectories of users $i$ and $j$ pass in time order during the same time period.

Definition 6. Crowd flow prediction problem: By analyzing the user’s trajectory database $DT$ and the spatial and temporal similarity between the user’s trajectory data, the moving position of the next flow of people can be predicted. Given the users’ trajectory database $DT = \{Tr_1, Tr_2, \ldots, Tr_n\}$, the crowd flow prediction task is to obtain the probability of the location $(l_{n+1}, t_{n+1})$ of crowd flow in the next time $t_{n+1}$.

Definition 7. User Risk Classification: Take the COVID-19 pandemic as an example. China’s national government service platform launched the “Epidemic Prevention and Health Information Code” and set up a personal health code for dynamic management in three colors: “green” represents low risk, “yellow” represents medium risk, and “red” represents high-risk. Based on the health code, we create a risk classification for humans.

3.2. Differential Privacy and Geo-Indistinguishability

3.2.1. Differential Privacy. Differential privacy is a recent privacy model which provides a strong privacy guarantee. The main idea of differential privacy is that after adding or deleting a record to the database, there is almost no difference in the output of the same algorithm applied to the database. Formally, differential privacy is defined as follows:

Definition 8. $\varepsilon$-differential privacy: A private algorithm $A$ gives $\varepsilon$-differential privacy if for any neighboring databases $D_1$ and $D_2$, and for any possible output $S \in \text{Range}(A)$, we have

$$\Pr[A(D_1) = S] \leq e^\varepsilon \times \Pr[A(D_2) = S],$$

where $\Pr$ represents the probability, and $\text{Range}(A)$ represents the value range of the output result of algorithm $A$.

Definition 9. Sensitivity: For any neighboring databases $D_1$ and $D_2$, the sensitivity of $f$ is

$$\Delta f = \max_{D_1, D_2} \|f(D_1) - f(D_2)\|.$$

Currently, differential privacy protection has two main methods: the Laplace’s mechanism and the exponential mechanism.

Definition 10. $\varepsilon$-local differential privacy: Traditional DP requires a central trusted party which is often not realistic. To remove that limitation, local differential privacy (LDP) has been proposed. A randomized mechanism $M$ satisfies $\varepsilon - \text{LDP}$ for any pair input $x$ and $x'$ in $D$, and any output $Y$ of $M$.

$$\Pr[M(x) = Y] \leq e^\varepsilon \cdot \Pr[M(x') = Y].$$

The privacy guarantee of mechanism $M$ is controlled by privacy budget. To combat inference attacks against shared data values, companies including Google, Apple, and Microsoft employ local differential privacy (LDP), the state-of-the-art in privacy-preserving data collection.

Theorem 1. Laplace’s mechanism: The Laplace mechanism is to add noise following a Laplace distribution to the output of the algorithm, so that the algorithm meets the differential privacy protection. For any function $f: D \rightarrow \mathbb{R}^n$ with sensitivity $\Delta f$, the algorithm $A$

$$A(D) = f(D) + \text{Lap}(\Delta f/\varepsilon).$$

The Laplace distribution with magnitude $\lambda$, i.e., $\text{Lap}(\lambda)$, follows the probability density function as $\text{Lap}(\lambda) = 1/2\lambda e^{-|a|/\lambda}$, where $\lambda = \Delta f/\varepsilon$ is determined by the sensitivity $\Delta f$ and the privacy budget $\varepsilon - \text{LDP}$, for any pair of inputs.

Theorem 2. Sequential composition: For the same data set, if the whole privacy protection process is divided into the different privacy protection algorithms $(A_1, A_2, \ldots, A_n)$ whose privacy protection levels are $(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)$, then the privacy protection level $\sum_{i=1}^n \varepsilon_i$ of the whole process needs to satisfy differential privacy protection.

Theorem 3. Parallel composition: For the disjoint data set, if the whole privacy protection process is divided to the different privacy protection algorithms $(A_1, A_2, \ldots, A_n)$ whose privacy protection levels are $(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)$, then the privacy protection level $\max(\varepsilon_i)$ of the whole process needs to satisfy differential privacy protection.

3.3. Geo-Indistinguishability. By generalizing the definition of differential privacy, [70, 71] proposed a generalization of
differential privacy under the metric. Let $\chi$ be a set of locations and let $Z$ be a set of query outputs, while $M$ is an obfuscation mechanism which ensures that $M(x)$ and $M(x')$ are similar to a certain degree for any two locations $x, x' \in \chi$, thus making it impossible to distinguish the true position. The distance $d_\rho(x, x')$ that expresses the distinguishability level between $x$ and $x'$. A small value denotes that the locations should remain indistinguishable, while a large value means that we allow the adversary to distinguish them. Let $\rho(Z)$ denotes the set of probability measures over $Z$. The multiplicative distance $d_\rho$ on $\rho(Z)$ is defined as $d_\rho(\sigma_1, \sigma_2) = \sup_{z \in Z} |\ln \sigma_1(z)/\sigma_2(z)|$ with $|\ln \sigma_1(z)/\sigma_2(z)| = 0$ if both $\sigma_1$ and $\sigma_2$ are zero and $\infty$ if only one of them is zero.

In this paper, we use the Euclidean distance between $x$ and $x'$ to represent the $d_\rho(x, x')$. The $\varepsilon$-geo-indistinguishability is defined as follows:

**Definition 11.** $\varepsilon$-geo-indistinguishability. A mechanism $M: \chi \rightarrow \rho(Z)$ satisfies $\varepsilon$-geo-indistinguishability if $\forall x, x' \in \chi, z \subseteq Z$,

$$d_\rho(Pr(M(x) \in z), Pr(M(x') \in z)) \leq \varepsilon d(x, x').$$

Has been proposed a technique that achieves $\varepsilon$-geo-indistinguishability named Planar Laplacian mechanism (PLM) which generalizes the Laplace Mechanism to two dimensions as follows:

$$Pr(PLM_\varepsilon(x) = z) = \frac{\varepsilon^2}{2\pi} e^{-\varepsilon d(x, z)}.$$  

4. Methodology

In this section, we proposed a privacy-preserving crowd flow prediction framework based on federated learning as shown in Figure 2. In the proposed scheme, we first select the spatio-temporal feature of the prediction model to quantify the factors that influence the trajectory of people during epidemics. We select the weather and temperature feature, temporal feature, and regional feature as the main features of the crowd flow prediction model. In particular, we construct the urban regional classification model to classify different regions according to their regional functional attributes and quantify the functional attributes of the regions.

We set up two kinds of characteristics: an hourly feature and a weekday feature. Crowd flow patterns are different for each day in a week. During the COVID-19 outbreak, people mainly stayed at home on weekends, but some still went to work on weekdays. The trajectory of crowd flow showed obvious changes in characteristics. The desensitization data of 4G base stations in a certain area of Nanjing was selected for analysis. During the epidemic period, 7:00 am to 9:00 am was the peak time for the flow of people to work, and 18:00 was the peak time for the flow of people to work. The flow of people was relatively low on Saturday and Sunday. The range of hour features from 1 to 12. The value range of a weekday is between Monday and Sunday. The temporal feature can be transformed into one-hot encoding.
Regional feature (urban regional classification model). Considering that the spread of the epidemic is mostly concentrated in functional areas with high population density, such as airports and shopping malls, the model divides the areas of the city into functional areas, and adds a feature mark to each area (representing the functional attributes of the area). The user’s trajectory through different functional areas reflects changes in user behavior patterns. For example, when a user leaves a residential area, he is more likely to choose an office or shopping mall as his destination. The accuracy of the prediction results will be improved by introducing the regional functional model into the prediction of crowd flow trajectory.

Definition 12. Urban Zoning. Streets and road networks in cities can divide urban areas into different geographical areas as is shown in Figure 3.

Definition 13. Composite arrival matrix: Divide the 24 hours of the day into 12 intervals starting at 0 o’clock. N is the number of partitions in the region, w is the weather value of the time period in the region, and temp is the temperature range of this region in the time period. The composite arrival matrix is an R×N-dimensional matrix. Each row of matrix A represents the number of users arriving in each region at different time intervals, temperature temp, and weather w.

\[
A_t = \{a_{1,1}, a_{1,2}, \ldots, a_{1,12}, a_{2,1}, a_{2,2}, \ldots, a_{2,12}, \ldots, a_{N,1}, a_{N,2}, \ldots, a_{N,12}\},
\]

Definition 14. Composite departure matrix. Similar to the arrival matrix, the composite arrival matrix is an R×N-dimensional matrix. Each row of matrix D represents the number of users leaving in each region at different time intervals, temperature temp, and weather w.

\[
D_t = \{d_{1,1}, d_{1,2}, \ldots, d_{1,12}, d_{2,1}, d_{2,2}, \ldots, d_{2,12}, \ldots, d_{N,1}, d_{N,2}, \ldots, d_{N,12}\},
\]

Definition 15. POI matrix. POI is point of interest on a map, POI points may be shopping malls, schools, hospitals, and office buildings which represent the most intuitive functional attributes of the region. A location may have multiple POI attributes as shown in Figure 3, where there are both shopping malls and office buildings. The POI matrix is an n×m-dimensional matrix where m represents the number of categories of points of interest. The value of \(p_i\) is the proportion of the number of different POI of businesses in the area. For example, in Figure 3, area i has 4 POI attributes, such as shopping mall, fitness, restaurant, and office, so the value of \(p_i\) is \(p_i = \{0.1, 0.2, 0.2, 0.4\}\) as Table 1 shows.

Chen et al. [73] used the NMF model to divide urban areas into three categories: “work,” “residential,” and “other,” and the user’s trajectory can correspond to the movement between these three regions. However, the original NMF model cannot use one type of information for decomposition, resulting in biased results. In order to integrate weather, temperature, arrival data, departure data, and POI matrix, we use a modified NMF to find more accurate mobility patterns named Joint Nonnegative Matrix Factorization [74, 75]. The objective function of the region functional model is as follows:

\[
\min O(W, H_s, H_r) = \|P - WH_s\|_F^2 + \alpha X - WH_r\|_F^2 + \lambda (\|W\|_F^2 + \|H_s\|_F^2 + \|H_r\|_F^2), X \text{ is the flow change matrix of the user arriving and leaving a certain area, which represent the integration of matrices Composite Arrival Matrix A and Composite Departure Matrix D, } X_k = \{a_{1,1}, a_{1,2}, \ldots, a_{1,12}, a_{2,1}, a_{2,2}, \ldots, a_{2,12}, \ldots, a_{N,1}, a_{N,2}, \ldots, a_{N,12}\}, P \text{ denotes POI matrix. } W \in \mathbb{R}^{N \times K} \text{ is the functional weight matrix of the region, each row of } W \text{ can be denoted as the intensity of each function. } H_s \in \mathbb{R}^{K \times L}, \text{ each row of matrix } H_s \text{ can be denoted as a basic mobility pattern which indicates the moving habits for certain function. } H_r \in \mathbb{R}^{K \times M} \text{ is the distribution pattern matrix of interest points. } \alpha \text{ controls the importance of matrix } P \text{ and matrix } X \text{ in the decomposition process. When optimizing the objective function } \min O(W, H_s, H_r), \text{ the algorithm updates one set of parameters at a time and fix the others. From the above formula, the update formula for } W_{ij}, H_{sij}, \text{ and } H_{rij} \text{ is:}
\]

\[
W_{ij} \leftarrow W_{ij} - \frac{P \cdot H_{ij}^{12} + \alpha X \cdot H_{ij}^{12}}{H_s \cdot H_r^{12} + \alpha H_{s} \cdot H_{r}^{12} + \lambda I}
\]

\[
H_{sij} \leftarrow H_{sij} - \frac{\alpha W_{ij}^{12} \cdot H_{ij}}{\alpha W_{ij}^{12} \cdot H_{s} + \lambda H_{s} + \lambda H_{r}}
\]

\[
H_{rij} \leftarrow H_{rij} - \frac{(1 - \alpha)W_{ij}^{12} \cdot H_{ij}}{(1 - \alpha)W_{ij}^{12} \cdot H_{r} + \lambda H_{r}}
\]
In order to obtain the region functional attribute feature values, we propose the JNMF-based regional function discovery model. The algorithm calculates the composite arrival matrix and composite departure matrix and POI matrix of the region, combined with the JNMF algorithm, and finally obtains the regional feature values of the region. The specific algorithm is shown in Algorithm 1.

4.2. Crowd Flow Prediction Model. The purpose of flow prediction is to predict the flow of people in different regions in the next stage based on the historical trajectory data of people combined with weather, temperature, and other factors. CNN (Convolutional Neural Networks) is suitable for local feature extraction, while LSTM (Long Short-Term Memory) is suitable for time series processing, so the two are complementary in function. In order to improve the prediction accuracy of the prediction model, a prediction model based on CNN and LSTM is proposed in this paper. The model takes the crowd flow trajectory data as the input of CNN model, combines embedding components for feature extraction, and uses LSTM for training and prediction. The framework of our basic mobility prediction model is presented in Figure 4. As Figure 4 shows, the crowd flow prediction model consists of four parts: input module with multi-feature, embedding component, CNN component, and prediction component. Details of these modules are introduced in the following sections:

- Input module with multi-feature: The time information of trajectory data exists in the form of time stamps and cannot be directly input into the prediction model as eigenvalues. We set the time interval \( \text{timesize} \) and divide one day into several time periods to obtain discrete time feature sequences and add them into feature sequences. Formula (10) maps the time of each trajectory point to the time segment, where \( t_i \) is the time when the user appears in the \( P_i \) region (0:00–24:00), and the time segment \( ts \) is defined as

\[
  ts = \left\lfloor \frac{t_i}{\text{timesize}} \right\rfloor.
\]

In order to unify the time-length of trajectory and obtain better prediction results, we divides each trajectory data into fixed-length trajectory based on a sliding window [76–79]. Each trajectory sequence is scanned by a sliding window with \( \text{maxlength} \), The trajectory data are combined with the weather and temperature features. The weather \( w_{it} \) and temperature \( temp_{it} \) conditions are added into the feature vector as 1-dimensional feature vector. Finally, we introduce the weekday features. The eigenvectors of trajectories are shown in formula (11).

\[
  Tr_i = \left\{ p_1, p_2, \ldots, p_n \mid userid = i, time = t_i, weather = w_{it}, temperature = temp_{it}, weekday, areaID, ts = \left\lfloor \frac{t_i}{\text{timesize}} \right\rfloor \right\}.
\]
Input: Composite Arrival Matrix, Composite Departure Matrix, and POI matrix

Output: Regional feature

1. The city is divided into regions based on roads, and \( n \) regions \( R = \{r_1, \ldots, r_n\} \) are obtained.
2. For \( i = 1 \) to \( n \) do.
3. Divide \( r_i \) into \( m \) cell regions according to the latitude and longitude grid.
4. Calculate the Composite Arrival Matrix and Composite Departure Matrix of region \( r_i \).
5. Get the crowd flow change matrix \( X \) of \( r_i \).
6. Query the POI points of the \( r_i \) and calculate the POI matrix \( P \).
7. Based on NMF, the \( X \) matrix is decomposed into \( W, H_u \), and \( H_s \).
8. The optimization objective function needed for \( W \).
9. Each parameter is updated using the gradient descent method.
10. Each regional feature.
11. Return regional feature.

**Algorithm 1:** NMF-based regional function discovery model.

**Embedding Component.** During the epidemic period, attention should be paid to the region and the time where people gather. Using latitude and longitude as location data will result in higher data dimensions. Therefore, we use the area ID to represent the area embedding and the time segment ID to represent the time embedding. The embedding vector can represent the correlation between different categories. Through the output of the previous period and the input of the current period, \( h_{t-1} \) is the output data of the previous period, then the states of each unit can be denoted as follows:

\[
i_t = \sigma(W_f \cdot \left[ h_{t-1}, x_t \right] + b_f),
\]

input gate

\[
\text{forget gate } f_t = \sigma(W_f \cdot \left[ h_{t-1}, x_t \right] + b_f).
\]

Through the output of the previous period and the input of the current period, a prestate is firstly calculated for the model, and then the useful information in the prestate is finally added into the hidden layer state by the input gate.

\[
\tilde{C}_t = \tanh(W_c \cdot \left[ h_{t-1}, x_t \right] + b_c),
\]

\[
C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t.
\]

The output gate determines the output information in the current period according to \( h_{t-1} \) and \( x_t \) and the current hidden layer state.

\[
\text{Output gate } O_t = \sigma(W_o \cdot \left[ h_{t-1}, x_t \right] + b_o),
\]

\[
h_t = O_t \cdot \tanh(C_t).
\]

In this paper, the prediction ability is improved by increasing the model depth. Three layers of LSTM cell are set...
up and the hidden layers number is 128. The prediction algorithm is as the Algorithm 2 shows.

Computational Complexity. The computational complexity of crowd flow prediction model can be estimated by summarizing up the computational costs of the embedding CNN and LSTM components, respectively.

1. **Computational Cost of Embedding Component**: The computational cost of embedding component mainly depends on the dimension of input categorical vector and the embedding dimension. In particular, we denote the computational cost of the embedding component by \( N_{\text{Embedding}} \), which can be calculated by

\[
N_{\text{Embedding}} = N_v \times N_c \times D_i \times D_o,
\]

where \( N_v \) is the number of time and area categorical features and \( D_i \) is the embedding dimension. The calculation is mainly based on the concatenation of categorical feature vectors according to the embedding dimension.

2. **Computational Cost of CNN Component**: The standard convolution computational cost per time step in CNN component is \( O(N_{\text{CNN}}) \), where \( N_{\text{CNN}} \) is the total number of parameters in a CNN. \( N_{\text{CNN}} \) can be calculated as

\[
N_{\text{CNN}} = C_i \times C_o \times D_i^2 \times D_f^2,
\]

where \( C_i \) is the number of input channels, \( C_o \) is the number of output channels, \( D_i^2 \) is the kernel size, and \( D_f^2 \) is the feature map size.

3. **Computational Cost of LSTM Prediction Model Component**: It is shown in [80] that the learning computational complexity per time step in LSTM prediction model is \( O(N_{\text{LSTM}}) \), where \( N_{\text{LSTM}} \) is the total number of parameters in a standard LSTM network. In particular, \( N_{\text{LSTM}} \) can be calculated by

\[
N_{\text{LSTM}} = \sum_{i=1}^{N_i} N_c \times 4 + N_i \times N_c \times 4 + N_c \times N_o + N_i \times 3,
\]

where \( N_i \) is the number of input units, \( N_c \) is the number of memory cells, and \( N_o \) is the number of output units.

4.3. Federated Learning Framework. In this section, we propose a novel federated learning-based privacy-preserving crowd flow prediction framework. As presented in Figure 5, the whole process of proposed framework can be divided into five steps:

(i) The central server (cloud server) divides the region into different client groups based on the K-Means algorithm and the geographical proximity. According to the definition of the constrained K-Means clustering algorithm, our goal is to determine the cluster center that minimizes the Sum of Squared Error (SSE). The SSE is defined as follows: As the location is defined by the longitude and latitude, the distance between locations can be measured by Euclidean Metric Distance Squared.

\[
d(x, y)^2 = \sum_{i=1}^{n} (x_i - y_i)^2,
\]

\[
\text{SSE} = \sum_{i=1}^{j} \sum_{m=1}^{k} \frac{1}{y_{im}} d(x, C_m^t)^2,
\]

where \( C_m^t \) denotes the \( m \)th cluster center in round \( t \) iterations, \( y_{im} \) is a solution for cluster assignment and update. If and only if \( y_{im} = 1 \), \( m = 1, \ldots, k \), we can obtain the optimal clustering center.

(ii) The cloud server selects the edge central server in client group, and the edge server is the closest server to the center of client group. The design of edge central server reduces the frequency of communications between cloud server and clients in the original FL architecture, and because the edge central server is the center of client group, the prediction model trained in client group can better
reflect the characteristics of the region where it is located.

(iii) The cloud server distributes the copy of the global model to all clients via the edge central server, and each client group trains its copy on local data. Considering the different level of privacy protection suitable for users in different risk level areas during the epidemic compared with traditional data privacy protection, our local privacy-preserving method is to train different parts of the whole model with different data. We divide the clients with user data into different groups by their privacy leakage risk level. Lines 20–30 in Algorithm 3 illustrate the local privacy protection method. If there is a high-risk user in client group, we should pay more attention to the accuracy of the model prediction, and thus directly train the clients in the risky client group with normal data. If there is no high-risk user, we should pay more attention to user privacy protection, so we need to train the client in the normal group with protected data (e.g., noisy data protected with the differential privacy mechanism). The noisy data satisfied the differential privacy requirement is generated based on the planar Laplace mechanism introduced in [81]. We map each true location $l$ to a randomly drawn location point $p$ in the infinite continuous space $P$ according to the probability density function, which is formulated as follows:

$$D_\varepsilon (l, p) = \frac{\varepsilon^2}{2\pi} e^{-\varepsilon d_\varepsilon(l, p)} d_\varepsilon(l, p) = d_{\text{euclidean}}(l, p).$$

where $l$ is the original real location, $p$ is the noisy location, $\varepsilon$ denotes the privacy budget of differential privacy. Based on this formulation, we can generate obfuscated location data (noisy data) from the normal data to train the risky client group.

(iv) Each client uploads the trained crowd flow prediction model to the edge central server. The edge central server generates a new local model by aggregating the received various local models from the clients in clients group. Every edge central server uploads the local model to cloud server. Shown in Algorithm 3, the clients upload the trained local model weights, and the edge central server needs to
aggregate them to obtain a local optimal model in each client group. We choose a simple global optimization method FedAvg [82] which is utilized for federated learning settings as the default optimization method of deep learning based crowd flow prediction model in our problem.

$$w_{G_m}^{t} \leftarrow w_{G_m}^{t-1} - \alpha \sum_{i=1}^{|G_i|} \beta_i^{t} \nabla w_i^{t} - \sum_{i=1}^{|G_i|} \beta_i^{t} w_i^{t},$$

(19)

If client group has $G_V$ clients participate in the local training, the edge central server will obtain $|G_V|$ updated local models. Based on the local prediction model, the local optimal parameters $w_{G_m}^{t}$ in each client group is updated by $w_{G_m}^{t} \leftarrow w_{G_m}^{t-1} - \alpha \sum_{i=1}^{|G_i|} \beta_i^{t} \nabla w_i^{t} - \sum_{i=1}^{|G_i|} \beta_i^{t} w_i^{t}$, where $\alpha$ denotes the learning rate, $\beta$ denotes the scale factor which is the sample size with the normalized loss of each local model during the training.

(v) The cloud server generates a global model by aggregating the received local models from all edge central servers. The cloud server distributes the updated global model to repeat the aforementioned iii) and iv) steps until meeting the stop criteria.

5. Performance Analysis

5.1. Datasets. The experiments are conducted on big data platform which is IBM X3650M4 with Intel e5 CPU 16 Core (2.2 GHZ) 256 GB RAM, and 3 TB ROM. All the proposed algorithms are coded by the Python3.6 programming language in CentOS7.5. All experiments are conducted using TensorFlow and PyTorch. We selected the desensitized 4G base station data from an anonymous user group in Nanjing from March 1 to March 22 as the traffic flow data. The data fields are as follows: anonymized user ID, timestamp, longitude, latitude, city, and grid number. We also introduce some external features include weather, temperature, and POIs, etc. The data description is shown in Table 2. The data is the observed crowd flow, and $\hat{Y}_p$ is the predicted crowd flow.

We compare the prediction performance of different methods in terms of Mean Absolute Error (MAE), mean absolute percentage error (MAPE)) and Root Mean Square Error (RMSE). We adopt MAE, MAPE, and RMSE as the evaluation metrics defined as follows:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |Y_i - \hat{Y}_p|,$$

$$MAPE = \frac{100}{n} \sum_{i=1}^{n} \frac{|Y_p - Y_i|}{Y_i},$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_p)^2},$$

(20)

where $Y_i$ is the observed crowd flow, and $\hat{Y}_p$ is the predicted crowd flow.

5.3. Overall Performance. In this section, we first introduce the model parameters in the prediction framework of this paper and then analyze the accuracy of the prediction performance of different methods in terms of Mean Absolute Error (MAE), mean absolute percentage error (MAPE)) and Root Mean Square Error (RMSE). We adopt MAE, MAPE, and RMSE as the evaluation metrics defined as follows:

(1) ARIMA: Auto Regressive Integrated Moving Average is a classic statistics-based method for time series prediction. This method is a common model in the field of time series prediction., which is widely used in the early research period.

(2) LSTM: Long Short-Term Memory network is an improved version of RNN. It has the advantages of capturing the long temporal feature of input time series. Recently, LSTM has been applied in traffic flow forecasting. We used the implementation in TensorFlow.

(3) ConvLSTM: ConvLSTM is a variant of LSTM which contains a convolution operation inside the LSTM cell. ConvLSTM considers both geographical spatial and temporal dependency of the spatial-temporal data, and is widely used in many spatial-temporal prediction tasks.

(4) ST-ResNet: Spatio-Temporal Residual Network (ST-ResNet) is a state-of-the-art neural network-based single-task learning model for urban crowd flow prediction. It stacks convolutional layers and residual units to capture the spatial and short/long-term temporal dependencies.

(5) DeepST: It is a deep spatio-temporal neural network designed for urban crowd flow prediction. It uses a temporal dependent instances to capture the temporal closeness, period seasonal trend, and convolutional neural network to capture near and far spatial dependencies.

(6) GEML: Grid Embedding-based Multi-Task Learning (GEML) is a multi-task learning framework that predicts the flow OD matrix and crowd flows simultaneously. It uses grid embedding and multi-task LSTM to capture the spatial-temporal representations of the crowd flow data.

5.2. Baselines and Evaluation Metrics. We compared the proposed model, labeled as “CFPF,” with the following 6 baseline methods.

(1) ARIMA: Auto Regressive Integrated Moving Average is a classic statistics-based method for time series prediction. This method is a common model in the field of time series prediction, which is widely used in the early research period.
algorithm in this paper, present the comparison results with baseline models under the constrain of privacy-preserving settings, the efficiency of the algorithm, the analysis of privacy protection and analyze the effects of key parameters in the training procedure of the proposed system.

In the federal learning architecture of this paper, the model size is about 50 M, the local training epoch is about 100 seconds, the transfer rate size is 12 MB/s, and the max communication overhead is about 4% of the overall time. Using the extranet environment, the transfer rate drops to 4 M/s, and the communication overhead accounts for 12.5% of the overall duration. The main parameters of our privacy-preserving crowd flow prediction are set as follows: (1) The parameters for the crowd flow prediction model: the size of area embedding = 64, the size of time embedding = 24; (2) hyper-parameters for the local training of prediction model: learning rate = 0.02, mini-batch size = 128, dropout rate = 0.5, and 107 base station locations (client = 107) in the experimental area were statistically analyzed, 10 central servers were obtained using K-means clustering algorithm.; (3) hyper-parameters for the global aggregation: local epoch = 1.

5.3.1. Prediction Accuracy (RE). During the COVID-19 outbreak, we focus on the movement of crowd between functional areas. For example, the movement of crowds from an office building to a shopping mall. Therefore, the number of crowds in office buildings and shopping malls in a specific area was selected as the experimental observation area in the experiment. Among the baseline algorithms, the ConvLSTM and ST-ResNet algorithms have higher prediction accuracy, so we compare the CFPF algorithm with the ConvLSTM and ST-ResNet algorithms in the prediction accuracy session. We compared the performance of our crowd flow prediction framework with the baseline methods in two days of weekdays and weekends for prediction, as shown in Figures 6 and 7. The experimental results show that the CFPF and ST-ResNet models are with much better accuracy when compared with ConvLSTM. The prediction performance is given in Table 3 for all baselines performance comparison. From the results, it can be observed that MAE of ST-ResNet is lower than those of ARIMA, LSTM, ConvLSTM, DeepST, and GEML. Figure 8 illustrates the loss of the ST-ResNet model and the CFPF model. From the results, the loss of the CFPF model is not significantly different from the ST-ResNet model. This proves that the CFPF has good convergence and stability. Compared with ST-ResNet, the CFPF model predicts with higher accuracy. Our model can perfectly capture the proposed spatio-temporal characteristics and temporal queue. The result also shows that the designed crowd flow prediction framework is effective to complex crowded environment from the large area data.

5.4. Performance Comparison and Effects of System Parameters. In this section, we present the comparison results with baseline models under the constrain of privacy-preserving settings and analyze the effects of system parameters in the training procedure of the proposed system.

From Table 3, we can draw three key conclusions: First, federated learning framework performs better than individual machine learning models. The traditional statistics-based method ARIMA performs poorly because it uses only time series data ignoring the geographical correlation between regions. The model performance is improved by introducing ConvLSTM and LSTM learning models with spatio-temporal data features. This indicates that combining spatio-temporal feature values will be beneficial to improve the model accuracy. Second, the deep learning model ST-ResNet and the multi-task learning model DeepST reflect the superiority of deep learning models in predicting crowd mobility data. And the introduction of multi-task learning
and federated learning framework has a great improvement on the deep model performance. Third, the prediction accuracy of our CFPF model is affected by the privacy-preserving settings, and this method still achieves a competitive performance after adjustment and optimization.

In this section of experiments, we explore the impact of different number of edge central server (i.e., $K \in \{2, 4, 6, 8, 10, 12\}$), number of clients (i.e., $C \in \{50, 150, 200, 250, 300, 350\}$), and the CNN depth (i.e, $D = 8, 16, 32, 64$) on the overall performance of the algorithm. In this paper, $K$ represents the number of central servers in the geographic space, obtained using the clustering algorithm. In the area of the experiment, we selected different edge central servers (i.e., different values of $K$) to test the performance of the CFPF, as shown in Table 4. In this experiment, the prediction accuracy and performance of the algorithm can reach the optimal case for $K = 8$. This indicates that in our scheme the overall prediction efficiency is improved by classifying similar spatio-temporal feature regions and populations into the same class by clustering. In Figure 9, the number of clients is equal to the number of location data servers (e.g., base stations) involved in training in the experimental area. Figure 10 shows the impact of the number of clients selected to participate in the training on

![Algorithm 2: Multi-factors CNN-LSTM prediction model.](image)

![Figure 8: Loss of the ST-ResNet and CFPF.](image)
the final performance. As the number of clients increases from 100 to 250, the final performance keeps improving. However, we observe that the overall accuracy and performance of the algorithm does not significantly improve anymore when the number of clients increases from 250 to 350. This indicates that the number of clients involved in the training is not as large as possible, but too many clients will affect the performance of the system and increase the system running time. To analyze the effect of the depth of the CNN on the algorithm, we choose RMSE as a measure of the efficiency of the algorithm. As shown in Figure 11, when the depth of CNN increases from 1 to 4, the RMSE decreases rapidly from 17.67 to 9.89; when the depth increases from 4 to 10, the RMSE decreases slowly or even rebound. This indicates that the depth of the CNN has an improvement on the algorithm efficiency within a certain range. Therefore, we believe that choosing an appropriate CNN depth (CNN = 4 in this paper) for the training dataset is beneficial to the overall prediction accuracy of the system. Based on the aforementioned results, by considering the final performance and accuracy, choosing proper clients, edge central server and CNN depth to participate in the optimization becomes important for the system.
5.5. Privacy Analysis. We discuss the privacy-preserving capability of the CFPF proposed in this paper from the following aspects:

(1) **System Architecture**: The CFPF is developed based on the FL framework. In this system, we avoid uploading any private data directly to the cloud server, only the trained model weights and parameters (e.g., gradient) are uploaded, and the private data are only stored and accessed on the local device. In this way, private data seems to be protected naturally.

(2) **Data Protection**: For data with high privacy protection level, we propose a privacy-preserving local optimization method which is based on LDP. In this paper, clients are divided into different according to different privacy protection levels. The client without privacy issue is to be trained with the normal data, the client with privacy risk is to be trained with the protected data \( (D_2, (l, p) = e^2/2\pi e^{-\alpha d_e(l, p)}, d_e(l, p) = d_{euclidean}(l, p)) \). The noisy data satisfied the differential privacy requirement is generated based on the planar Laplace mechanism introduced in [72]. On the obfuscated location data (noisy data) from the normal data to train the risky client group to make it robust to the potential attack.

To analyze the impact of privacy budget on performance, we use the accuracy of the algorithm as a measure. In Figure 12, we assign a privacy budget from 0 to 10 to the algorithm CFPF. Experiments show that the algorithm maintains high accuracy when the privacy budget increased.
to 3. This is because more privacy budgets are required for complex prediction tasks. Since the CFPF model is based on a distributed privacy framework, the predictions are trained by encrypting and reaggregating the data from the distributed clients instead of directly accessing the original data. Experiments show that the CFPF model achieves a balance between privacy protection and data availability.

6. Conclusions and the Future Work

In this paper, we study crowd flow prediction model with privacy protection. Based on the concept of federation learning, we propose a practical crowd prediction framework by introducing “Multi-Factors CNN-LSTM” algorithm to analyze the key factors affecting crowd flow during epidemics in order to achieve the desired prediction performance. In terms of privacy protection, the private data are protected by differential privacy mechanism, while the data are only used in the local server and there is no transmission of sensitive data in the system. We compare our method with existing ARIMA, LSTM, ConvLSTM, ST-ResNet, DeepST and GEML methods in 4G base station dataset. The experimental results show that the method achieves a better balance of prediction accuracy and data privacy protection compared with competing methods. The future work can be divided into two directions: (1) optimizing the existing prediction models to better capture the spatio-temporal correlations among traffic flow data in order to further improve the prediction accuracy. (2) In this paper, only simple crowd flow scenarios during epidemics are considered, and we will optimize the whole framework to adapt to more complex and realistic scenarios.

Data Availability

The experiment data required to reproduce these findings cannot be shared at this time as the data also form part of an ongoing study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61872197, 61972209), the Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX180891), the Natural Science Foundation of Jiangsu Province (BK20161516, BK20160916), and the Postdoctoral Science Foundation Project of China (2016M601859), and in part by the Postgraduate Research & Practice Innovation Program of Jiangsu Province KYCX210789.

References

[1] R. Li, S. Pei, B. Chen et al., “Substantial undocumented infection facilitates the rapid dissemination of novel coronavirus (SARS-CoV-2),” Science, vol. 368, no. 6490, pp. 489–493, 2020.

[2] R. Yan, Y. Zhang, Y. Li, L. Xia, Y. Guo, and Q. Zhou, “Structural basis for the recognition of SARS-CoV-2 by full-length human ACE2,” Science, vol. 367, no. 6485, pp. 1444–1448, 2020.

[3] L. Setti, F. Passarini, G. De Gennaro et al., “SARS-CoV-2RNA found on particulate matter of Bergamo in Northern Italy: first evidence,” Environmental Research, vol. 188, p. 109754, 2020.

[4] Y. Jiang, H. Wang, and Y. Chen, “Clinical data on hospital environmental hygiene monitoring and medical staff protection during the coronavirus disease 2019 outbreak,” medRxiv, vol. 23, p. 57, 2020.

[5] D. S. Hui, E. I Azhar, T. A. Madani et al., “The continuing 2019-nCoV epidemic threat of novel coronaviruses to global health — the latest 2019 novel coronavirus outbreak in
Wuhan, China," *International Journal of Infectious Diseases*, vol. 91, pp. 264–266, 2020.

[6] I. M. Longini, A. Nizam, S. Xu et al., “Containing pandemic influenza at the source,” *Science*, vol. 309, no. 5737, pp. 1083–1087, 2005.

[7] S. Eubank, H. Guclu, V. S. Anil Kumar et al., “Modelling disease outbreaks in realistic urban social networks,” *Nature*, vol. 429, no. 6988, pp. 180–184, 2004.

[8] D. Easley and J. Kleinberg, “Networks, crowds, and markets: reasoning about a highly connected world[]" *Significance*, vol. 9, no. 1, pp. 43–44, 2012.

[9] R. M. Anderson, C. Fraser, A. C. Ghani et al., “Epidemiology, transmission dynamics and control of SARS: the 2002–2003 epidemic,” *Philosophical Transactions of the Royal Society of London Series B Biological Sciences*, vol. 359, no. 1447, pp. 1091–1105, 2004.

[10] L. Hufnagel, D. Brockmann, and T. Geisel, “Forecast and control of epidemics in a globalized world,” *Proceedings of the National Academy of Sciences*, vol. 101, no. 42, pp. 15124–15129, 2004.

[11] S. Riley, “Large-scale spatial-transmission models of infectious disease,” *Science*, vol. 316, no. 5829, pp. 1298–1301, 2007.

[12] M. A. Ferrag, L. Shu, and K. K. R. Choo, “Fighting COVID-19 and future pandemics with the Internet of things: security and privacy perspectives,” *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 9, pp. 1477–1499, 2021.

[13] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, “Dnn-based prediction model for spatio-temporal data,” in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 1–4, 2016.

[14] J. Zhang, Y. Zheng, and D. Qi, “Deep spatio-temporal residual networks for citywide crowd flows prediction,” *Thirty-first AAAI conference on artificial intelligence*, vol. 76, p. 98, 2017.

[15] J. Zhang, Y. Zheng, J. Sun, and D. Qi, “Flow prediction in spatio-temporal networks based on multitask deep learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 3, pp. 468–478, 2020.

[16] Z. Lin, J. Feng, Z. Lu, Y. Li, and D. Jin, “DeepSTN+: context-aware spatial-temporal neural network for crowd flow prediction in metropolis,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 1020–1029, 2019.

[17] H. Li, J. Liu, K. Wu, Z. Yang, R. W. Liu, and N. Xiong, “Spatio-temporal vessel trajectory clustering based on data mapping and density,” *IEEE Access*, vol. 6, pp. 58939–58954, 2018.

[18] Y. Li, Y. Zheng, and H. Zhang, “Traffic prediction in a bike-sharing system,” in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 1–10, China, July 2016.

[19] Y. Liu, Z. Liu, and R. Jia, “DeepPF: a deep learning based architecture for metro passenger flow prediction,” *Transportation Research Part C: Emerging Technologies*, vol. 101, pp. 18–34, 2019.

[20] L. Mourad, H. Qi, Y. Shen, and B. Yin, “ASTIR: spatio-temporal data mining for crowd flow prediction,” *IEEE Access*, vol. 7, pp. 175159–175165, 2019.

[21] X. Zhou, Y. Shen, and Y. Zhu, “Predicting multi-step citywide passenger demands using attention-based neural networks,” in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 736–744, 2018.

[22] J. Ye, L. Sun, and B. Du, “Co-prediction of multiple transportation demands based on deep spatio-temporal neural network,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 305–313, USA, August 4 - 8, 2019.

[23] B. Du, H. Peng, S. Wang et al., “Deep irregular convolutional residual LSTM for urban traffic passenger flows prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 972–985, 2020.

[24] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, “Revisiting spatial-temporal similarity: a deep learning framework for traffic prediction,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 5668–5675, 2019.

[25] X. Cheng, R. Zhang, and J. Zhou, “Deeptransport: learning spatial-temporal dependency for traffic condition forecasting,” in *Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, Rio de Janeiro, Brazil, 08-13 July 2018.

[26] L. Wang, X. Geng, and X. Ma, “Cross-city transfer learning for deep spatio-temporal prediction,” 2018, http://arxiv.org/abs/1802.00386.

[27] S. Madan and P. Goswami, “A novel technique for privacy preservation using k-anonymization and nature inspired optimization algorithms,” in *Proceedings of the International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM)*, Amity University Rajasthan, Jaipur-India, 2019.

[28] J. Le Ny, A. Touati, and G. J. Pappas, “Real-time privacy-preserving model-based estimation of traffic flows,” in *Proceedings of the 2014 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, pp. 92–102, IEEE, Berlin, Germany, 14-17 April 2014.

[29] J. Konečný, H. B. McMahan, and F. X. Yu, “Federated learning: strategies for improving communication efficiency,” 2016, http://arxiv.org/abs/1610.05492.

[30] M. A. Ferrag, O. Friha, L. Maglaras, H. Janicke, and L. Shu, “Federated deep learning for cyber security in the Internet of things: concepts, applications, and experimental analysis,” *IEEE Access*, vol. 9, pp. 138509–138542, 2021.

[31] M. Wu, L. Tan, and N. Xiong, “A structure fidelity approach for big data collection in wireless sensor networks,” *Sensors*, vol. 15, no. 1, pp. 248–273, 2014.

[32] S. Huang, A. Liu, S. Zhang, T. Wang, and N. N. Xiong, “BD-VTE: a novel baseline data based verifiable trust evaluation scheme for smart network systems,” *IEEE transactions on network science and engineering*, vol. 8, no. 3, pp. 2087–2105, 2021.

[33] Z. Ebrahimpour, W. Wan, and O. Cervantes, “Comparison of main approaches for extracting behavior features from crowd flow analysis,” *ISPRS International Journal of Geo-Information*, vol. 8, no. 10, p. 440, 2019.

[34] K. Jayarajah, A. Tan, and A. Misra, “Understanding the interdependency of land use and mobility for urban planning,” in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, pp. 1079–1087, August 2018.

[35] Y. Shi, H. Feng, X. Geng, X. Tang, and Y. Wang, “A survey of hybrid deep learning methods for traffic flow prediction,” in *Proceedings of the 2019 3rd International Conference on Advances in Image Processing*, pp. 133–138, Association for Computing Machinery, 2019.

[36] P. Xie, T. Li, J. Liu, S. Du, X. Yang, and J. Zhang, “Urban flow prediction from spatiotemporal data using machine learning: a survey,” *Information Fusion*, vol. 59, pp. 1–12, 2020.

[37] X. Yin, G. Wu, and J. Wei, “Deep learning on traffic prediction: methods, analysis and future directions,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 76, p. 754, 2021.
[38] K. Gao, F. Han, P. Dong, N. Xiong, and R. Du, “Connected vehicle as a mobile sensor for real time queue length at signalized intersections,” Sensors, vol. 19, no. 9, p. 2059, 2019.

[39] M. Van Der Voort, M. Dougherty, and S. Watson, “Combining kohonen maps with arima time series models to forecast traffic flow,” Transportation Research Part C: Emerging Technologies, vol. 4, no. 5, pp. 307–318, 1996.

[40] B. M. Williams and L. A. Hoel, “Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results,” Journal of Transportation Engineering, vol. 129, no. 6, pp. 664–672, 2003.

[41] J. Zheng and L. M. Ni, “Time-dependent trajectory regression on road networks via multi-task learning,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 27, no. 1, pp. 1048–1055, 2013.

[42] A. Hofleitner, R. Herring, P. Abbeel, and A. Bayen, “Learning the dynamics of arterial traffic from probe data using a dynamic bayesian network,” IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 4, pp. 1679–1693, 2012.

[43] B. M. Williams, “Multivariate vehicular traffic flow prediction: evaluation of ARIMAX modeling,” Transportation Research Record, vol. 1776, no. 1, pp. 194–200, 2001.

[44] Y. Li, R. Yu, and C. Shahabi, “Diffusion convolutional recurrent neural network: data-driven traffic forecasting,” 2017, http://arxiv.org/abs/1707.01926.

[45] J. Zhang, Y. Zheng, and D. Qi, “DNN-based prediction model for spatio-temporal data,” in Proceedings of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems, pp. 1–4, 2016.

[46] Z. Lin, J. Feng, Z. Lu, Y. Li, and D. Jin, “DeepSTN+: context-aware spatial-temporal neural network for traffic prediction in metropolitan cities,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 1020–1027, 2019.

[47] J. Sun, J. Zhang, and Q. Li, “Predicting citywide crowd flows in irregular regions using multi-view graph convolutional networks,” IEEE Transactions on Knowledge and Data Engineering, vol. 98, p. 45, 2020.

[48] P. Yang, N. Xiong, and J. Ren, “Data security and privacy protection for cloud storage: a survey,” IEEE Access, vol. 8, pp. 131723–131740, 2020.

[49] B. Y. He and J. Y. J. Chow, “Optimal privacy control for transport network data sharing,” Transportation Research Procedia, vol. 38, pp. 792–811, 2019.

[50] Y. Zhou, Z. Mo, Q. Xiao, S. Chen, and Y. Yin, “Privacy-preserving transportation traffic measurement in intelligent cyber-physical road systems,” IEEE Transactions on Vehicular Technology, vol. 65, no. 5, pp. 3749–3759, 2016.
and Digital Content (IC-NIDC 2016), p. 657, Beijing, China, 23–25 September 2016.

[75] S. Gao, L. Denoyer, and P. Gallinari, "Temporal link prediction by integrating content and structure information," in Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, ACM, Glasgow, United Kingdom, October 24–28, 2011.

[76] H. S. Hota, R. Handa, and A. K. Shrivastav, "Time series data prediction using sliding window based RBF neural network [J]," International Journal of Computational Intelligence Research, vol. 13, no. 5, pp. 1145–1156, 2017.

[77] Y. Jiang, G. Tong, H. Yin, and N. Xiong, "A pedestrian detection method based on genetic algorithm for optimize XGBoost training parameters," IEEE Access, vol. 7, pp. 118310–118321, 2019.

[78] R. Wan and N. Xiong, "An energy-efficient sleep scheduling mechanism with similarity measure for wireless sensor networks," Human-centric Computing and Information Sciences, vol. 8, no. 1, pp. 18–22, 2018.

[79] Y. Lu, S. Wu, Z. Fang, N. Xiong, S. Yoon, and D. S. Park, "Exploring finger vein based personal authentication for secure IoT," Future Generation Computer Systems, vol. 77, pp. 149–160, 2017.

[80] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," Proc. 15th Annu. Conf. Int. Speech Commun. Assoc., pp. 1–5, 2014.

[81] M. E. Andrés, E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: differential privacy for location-based systems," 2012, http://arxiv.org/abs/1212.1984.

[82] B. McMahan, E. Moore, D. Ramage, S. Hampson, and A. Blaise, "Communication-efficient learning of deep networks from decentralized data," in Artificial Intelligence and Statistics, pp. 1273–1282, 2017.

[83] J. Zhang, Z. Yu, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," Spatio, vol. 65, p. 8689, 2016.

[84] X. Shi, "Convolutional LSTM network: a machine learning approach for precipitation nowcasting," MIT Press, vol. 123, p. 199, 2015.

[85] M. Khashei and M. Bijari, "An artificial neural network model for timeseries forecasting," Expert Systems with Applications, vol. 37, pp. 479–489, 2010, ARIMA.

[86] Y. Wang, "Origin-destination matrix prediction via graph convolution: a new perspective of passenger demand modeling," GEML, vol. 986, p. 1644, 2019.

[87] J. Zhang, Z. Yu, and Z. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," Deep, vol. 144, p. 65, 2016.