ABSTRACT
Storm and flood disasters such as typhoons and torrential rains are becoming more intense and frequent. The national government and municipalities must respond to such natural disasters as soon as possible. When the scale of damage is large, however, it takes much time to investigate the severity of damage, and the initial response can be delayed. If we could precisely and rapidly estimate the severity of damage for each city at an early stage, the national government would be able to better support the municipalities, and consequently respond quickly to help citizens. In this paper, we propose a novel approach to estimate the severity of disaster damage within a short time period after a disaster occurs by exploiting real-time population data generated from cellular networks. First, we investigate the relationship between real-time population data and the severity of damage. Then, we design a Graph Convolutional Networks for Disaster Damage Estimation, called D2E-GCN, which fully exploits the directed and weighted characteristics of human mobility graph. We conduct an offline evaluation on real-world datasets including two typhoons that hit Japan. The evaluation results show that the proposed method outperforms baseline methods which do not consider the graph structure of cities, and the proposed method can estimate the severity of damage approximately 48 hours after typhoons passed. Moreover, we find the experimental insight that the estimation performance can be significantly affected by the graph construction method for GCN models.

CCS CONCEPTS
• Information systems → Mobile information processing systems.

KEYWORDS
Mobile Reference-based Population, Disaster Relief, Graph Convolutional Network

1 INTRODUCTION
Weather-related disasters such as typhoons and heavy rains due to climate change are occurring more frequently causing fatal damage all over the world [14]. The Tokyo area experienced two strong typhoons, Faxai and Hagibis, in September and October, 2019, respectively. Typhoon Faxai caused a large-scale power outage1 while 47 rivers breached their banks because of heavy rains brought by Typhoon Hagibis2. In such situations, a speedy, accurate response to the disaster is crucial for national and municipal governments. As the damage scale becomes larger; however, it takes more time to grasp the whole situation, which delays the initial response. Therefore, it is important and necessary to estimate the severity of damage at an early stage. Here, we give an example from the case of Typhoon Faxai. Figure 1(a) shows the trajectory and timeline of the typhoon, which hit the Greater Tokyo area on Sep. 9, 2019. The blue points indicate the center of the typhoon, and the red circles indicate the storm zone. The typhoon made landfall just before 5 a.m. on Sep. 9 over Chiba city in Chiba prefecture, located east of Tokyo. Figures 1(b) and (c) show timelines of the number of partially damaged houses reported to the national government3. Kanagawa prefecture seemed to have the greatest damage in the early stage, at 8 a.m. on Sep. 12, indicated by the red vertical line in Figure 1(c). From the data published one week later, however, we can see that Chiba prefecture actually had the most severe damage as indicated by the dashed red line in the figure. Moreover, the data from a month later, represented by the red line in Figure 1(b), reveal that Chiba prefecture suffered even more serious damage. This example illustrates the difficulty of grasping the whole damage situation immediately after a disaster. Moreover, it is important to detect areas that have received damage serious enough to require the application of disaster recovery laws.

Estimating the severity of damage at an early stage after a disaster is a challenging problem. One possible method is not to estimate but directly observe the earth’s surface by using satellite imagery [26]. Unfortunately, satellite imagery suffers from limited temporal resolution, the presence of clouds etc. Another method is to use large-scale human behavior data such as social-media [27, 30] and GPS data [13, 28]. Several studies have shown the potential of using such data for disaster response. Unfortunately, there are limitations

3https://www.bousai.go.jp/updates/r1typhoon15/index.html

to these approaches. In the case of social-media, the information credibility may be problematic [25, 30], while the coverage is limited in the case of GPS data, especially for rural areas.

In this study, we propose an approach that exploits real-time population data generated from cellular networks, called a Real-Time version of Mobile Spatial Statistics (RT-MSS) [29], to estimate the severity level of a disaster. Cellular-network-based population data is suitable for this task, because it has high reliability, high temporal resolution, and a wide range of coverage. First, we conduct an empirical analysis to investigate the relationship between population dynamics and the damage severity because we have several hypotheses. For example, we expect that the presence of residential population in a city will increase after a natural disaster over that of a normal time, because public transport can be stopped due to a disaster. Conversely, the presence of non-residential population will decrease in comparison to that of a normal time, because people will stay at or close to home. Besides, we assume that the population dynamics before a disaster may not be closely related to the severity of the disaster because it depends on how weather forecasts are reported and how people perceive them, especially in a case of a typhoon. Therefore, we focus on the real-time population after a typhoon passed. Then, we design a model to estimate the severity of disaster damage within a short time period after a disaster occurs by exploiting real-time population data generated from cellular networks. Based on the above hypothesis, we design features to estimate the severity of damage, such as the population ratio at a target time against the normal time. Besides, because population flows can be affected by several factors such as geographical distance and transportation between target cities, we introduce Graph Convolutional Networks for Disaster Damage Estimation, called D2E-GCN, to leverage population flows using a directed city graph. Nodes of the graph represent cities, and an edge is added between two nodes if there is an inter-city population flow. The weight of the edge denotes the number of people who moved between the two cities. Finally, we evaluate the performance of models using actual disaster data (two Typhoons) in Japan. The results show the potential of exploiting real-time population to accurately and quickly estimate the damage level caused by a disaster. Beside, we find that the graph construction method significantly affects the accuracy that can be applied to other GCN-based model.

The contributions of this study are the following:

- We formulate a disaster damage-level estimation problem as a node classification problem, and propose an approach to estimate the disaster damage-level by exploiting real-time population data generated from cellular networks.
- We design effective features and D2E-GCN model to estimate the disaster damage-level based on our hypotheses that (1) there is a difference of the population dynamics between a damaged city and a less damaged city regarding residential and non-residential population, and (2) the estimation accuracy can be improved by incorporating a directed city graph which reflects human mobility flows.
- Offline evaluation on actual disaster data including two Typhoons that hit Japan show the potential of the proposed approach, which outperforms baseline methods and rapidly (approximately 48 hours after the typhoon had passed) estimated the damage level. Moreover, we find the experimental insight that the estimation performance can be significantly affected by the graph construction method for GCN models. This experimental insight can be applicable to a large number of GCN models such as GraphSAGE.

The rest of this paper is organized as follows. The next section describes preliminaries for this study, consisting of GCN model, definitions and the problem formulation. Section 3 describes the data used in this study. In Section 4, we conduct an empirical analysis to investigate the relationship between real-time population and the severity level of damage. In Section 5, we describe our approach for disaster damage estimation. We explain an evaluation to validate our approach and discuss limitations and future directions in Section 6. In Section 7, we review related work on disaster response and GCN. Finally, we conclude the paper in Section 8.

2 PRELIMINARIES

2.1 Graph Convolutional Networks

Let $G = (V, E)$ be a graph with node features $x_v$ (e.g., population of a city) where $v \in V$ is each city, and $e \in E$ represents an inter-city population flow. GCN model iteratively updates the node feature by aggregating (e.g., mean) the node features of its neighbors. After $k$ iteration of aggregation, GCN model aggregates the features of

Figure 1: (a) The trajectory and timeline of Typhoon Faxai, (b) the number of partially damaged houses in each prefecture from Sep. 9 to Oct. 10, and (c) the number of partially damaged houses three days after the typhoon hit, which corresponds to the red rectangle in (b). These graphs illustrate the difficulty of grasping the whole situation at an early stage after a disaster.
where $h_0^k = x_v$, $h_0^k$ denotes a representation of node $v$ of the $k$-th layer, and $N(v)$ denotes the neighborhood node of node $v$. This notation follows that in [15] and [34]. AGGREGATE and COMBINE are abstract operation, and several functions can be used for AGGREGATE such as the element-wise mean and max-pooling etc. COMBINE$_k$ combines the representation of target node $h_k^{k-1}$ and the representation of neighborhood nodes $a_v^k$. A concatenation operation can be used as COMBINE$_k$. In GCN model of [21], AGGREGATE and COMBINE are integrated as follows:

$$h_k^t = \text{ReLU}(W \cdot \text{M}

\text{AN}[h_k^{k-1}, u \in N(v)]) \quad (3)$$

where $W$ is a learnable weight matrix, and MEAN represents element-wise mean.

\subsection{2.2 Definitions}

\hspace{1em} \textbf{Definition 1 (Real-Time Population Data)} A record of the real-time population consists of a timestamp $t$, grid code $g$, residential (habitation) city $c$, and estimated real-time population $p_{g,c,t}$. That is, $(t, g, r, pop)$. pop is also denoted as $p_{g,c,t}(t, g, r)$. The grid code indicates the location of a square area that does not overlap with any others. We can use various grid-code implementations, such as Geohash [1]. In this study, we use the grid codes defined by the Japanese Statistics Bureau [6]. We use the half grid square, which has 500 meters on each side. By aggregating the $p_{g,c,t}$ within a specific city $c$, we define three types of population $p_{g,c,t}$: (1) residential population ($a = \text{residential}$), (2) non-residential population ($a = \text{non-residential}$), and (3) total population ($a = \text{total}$) as follows:

$$p_{g,c,t} = \left\{ \begin{array}{ll}
\sum_{g,c,t \in G_c} p_{g,c,t} & \text{if } a = \text{residential} \\
\sum_{g,c,t \in G_c} p_{g,c,t} & \text{if } a = \text{non-residential} \\
\sum_{g,c,t \in G_c} p_{g,c,t} & \text{if } a = \text{total} \\
\end{array} \right. \quad (4)$$

where $G_c$ is the set of grid codes included in city $c$ and $a \in \{\text{residential}, \text{non-residential}, \text{total}\}$. To distinguish the real-time population and the population reported by a census, we call the latter population the census-based population.

\hspace{1em} \textbf{Definition 2 (Real-Time Population History)} We define a real-time population history $x_c$ as a sequence of $p_{g,c,t}$ for a city $c$ and they are sorted by the timestamp $t$. That is, $x_c = \{p_{g,c,t_0}, p_{g,c,t_1}, \cdots, p_{g,c,t_s}\}$, where $s$ is the length of $x_c$.

\hspace{1em} \textbf{Definition 3 (Disaster Damage Data)} Disaster damage data consists of a city and the number of demolished houses. There are several degrees and types of damage, such as completely destroyed, half destroyed, partially destroyed, and flooded above floor level. These degree and types are used in the disaster recovery laws in Japan (the Disaster Relief Act (DRA) [23], and the Act on Support for Livelihood Recovery of Disaster Victims (ASLRDV) [24]). Then, we count the number of demolished houses, which is used in the decision process for the disaster damage level defined in the laws.

\begin{table}[h]
\centering
\caption{Criterion of damage level}
\begin{tabular}{|c|c|}
\hline
Census-based pop. of a city & Criteria number \\
\hline
100,000 $\leq p_{\text{census}}$ & 10 \\
50,000 $< p_{\text{census}} < 100,000$ & 5 \\
$p_{\text{census}} < 50,000$ & 2 \\
\hline
\end{tabular}
\end{table}

by the following rule: one demolished house = one completely destroyed house = two half destroyed houses = three houses flooded above floor level. Thus, we calculate the number of demolished houses, total damage, for a city as the following:

$$\text{total damage}_c = \text{numcomplete} + \frac{1}{2}\text{numhalf} + \frac{1}{3}\text{numflood}$$

where $\text{numcomplete}$ is the number of completely destroyed houses, $\text{numhalf}$ is the number of half destroyed houses, and $\text{numflood}$ is the number of houses flooded above floor level.

\hspace{1em} \textbf{Definition 4 (Disaster Damage Level)} We define a disaster damage level (or simply the damage level) as a binary variable, where $y_c = 1$ indicates severe damage, and $y_c = 0$ indicates non-severe damage for a city $c$. We focus on the fact that the national government applies disaster recovery laws to support municipalities and citizens in the event of a serious disaster based on the number of demolished houses. These laws are mainly applied when the number of demolished houses exceeds the criterion defined in each law. Thus, we create a ground truth label $y_c$ for each city based on the criteria which depend on the census-based population $p_{\text{census}}$ shown in Table 1 used in [23] and [24]. If the number of demolished houses exceeds the criterion, then $y_c = 1$; otherwise, $y_c = 0$.

\subsection{2.3 Problem Formulation}

Now, we formally define the disaster damage-level estimation problem as the following:

\hspace{1em} \textbf{Definition 5 (Disaster Damage-Level Estimation Problem)} Given the real-time population history $x_c$ of a city within a specific time duration $h_{dur}$ after a disaster occurs (e.g., after $T$ hours), our goal is to estimate the city’s disaster damage level $y_c$ from the population dynamics features $x'_c$ calculated from $x_c$. A supervised machine learning (ML) model can be trained and evaluated using a dataset $D_{aill} = \{D_1, D_2, \cdots, D_N\}$. Here, $N$ is the number of disaster cases, $D_n$ is a dataset of a disaster, $D_n = \{x'_c, y'_c \mid c \in C\}$, and $C$ is a set of target cities.

\section{3 DATA}

We use two datasets, i.e., the real-time population data and the disaster damage data. We analyze the effects of Typhoons Faxai and Hagibis in 2019 as case studies. Because the areas damaged by these two Typhoons were mainly around Tokyo, we target Tokyo Metropolis (excluding all of the Izu Islands except for Oshima), Chiba, and Kanagawa prefectures. The total number of cities is 171.

\textbf{Real-Time Population Data:} We use RT-MSS data, which provides estimated population for all of Japan [29]. The RT-MSS data is calculated at 10-minute intervals for each of the 500-m grid squares based on NTT DOCOMO’s cellular network data to provide estimated population data up to an hour before of the current time.
RT-MSS data is commercially available by DOCOMO Insight Marketing, INC. The basic idea of RT-MSS is as follows. Suppose that the population of Japan is 100 million, and 50 millions of which subscribe to a mobile service operator (i.e., the adoption rate is 0.5). The inverse of the adoption rate is called the extrapolation coefficient. Then, a real-time population can be estimated by dividing the number of mobile terminals in a base station area by the adoption rate of 0.5 (or multiplying the number by the extrapolation coefficient of 2.0). The actual adoption rate varies, however, with attributes such as regions, age groups, and gender, and mobile terminals are not always turned on. Thus, it is necessary to consider these factors in calculating the extrapolation coefficient. For more details, refer to [29]. To protect users’ privacy, NTT DOCOMO has published guidelines for creating population data from the operational data of the cellular network [12] and we also follow those guidelines in processing the data. As described in the guidelines, RT-MSS data only provides aggregated population data, not the trajectories of individuals. Table 2 gives an example of RT-MSS data. Here, the grid code 533946113 represents part of Chiyoda Ward, Tokyo. The data indicates that the estimated number of residents of Chiyoda Ward who were in the grid code’s area at 9 a.m. was 20,000, while the estimated number of residents of Chiba city who were in the area at that time was 2,000, and so on. Therefore, we can calculate that the residential population at the time was 20,000, the non-residential population was 7,000, and the total population was 27,000. Because the RT-MSS data gives a grid-level population, while the damage data is at a city-level, we need to match the geographical scales. Thus, we compute a summation of the grid-level RT-MSS data to generate city-level data. Figure 2 shows an example of the population transition aggregated at the city-level for Chiba city from Aug. 26 to Sep. 12, 2019.

Because real-time population data is long-term time series data, the population data can be decomposed into trend, seasonal and event components, using Kalman filter. Trend and seasonal components can be regarded as the stationary component, and the population which cannot be explained as the stationary component can be regarded as the event component. Thus, we decompose the real-time population into the static component, which is the sum of trend and seasonal components, and the event component, using the method proposed in [2].

### Disaster Damage Data
We collected data on the damaged houses in each prefecture from reports published on each prefecture’s official web page. Then, the city-level real-time population data and the ground truth label for each city is concatenated based on a city. The dataset that contains city, pop\_c,a,t, and the ground truth label for each city is concatenated based on a city. The dataset that contains city, pop\_c,a,t, and the ground truth label for each city is concatenated based on a city.

### 4 EMPIRICAL ANALYSIS
We investigate the population transitions in the case of Typhoon Faxai, which hit around Tokyo in September, 2019, to find insights that can be exploited for creating features.

**Method:** First, we calculate the population at a normal (non-disaster) time normal\_pop\_c,a,d,h by using the long-term real-time population data for a city c and a population type a:

$$normal\_pop\_c,a,d,h = \sum_{m,M} pop\_c,a,t$$ (6)

where $M = \{m | t_m \in t_{long}, d(t) = d, d \in D_{day\_type}, h(t) = h\}$, $t_{long} \in [t_s,t_e]$ is the learning period ($t_s$ is start time and $t_e$ is end time), $d(t)$ and $h(t)$ denote day type $d$ and the time of day $h$, $D_{day\_type}$ is a set of days in weekdays or weekend ($day\_type = \{weekday, weekend\}$). Although we could calculate the normal

---


---

### Table 2: Example of real-time population data. Grid code 533946113 indicates a location within Chiyoda Ward, Tokyo. Thus, we can calculate the residential and non-residential populations by using the columns for the grid code and residential city.

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Grid code</th>
<th>Residential city</th>
<th>Estimated Population</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-09-09 00:00</td>
<td>533946113</td>
<td>Chiyoda Ward</td>
<td>20,000</td>
</tr>
<tr>
<td>2019-09-09 00:00</td>
<td>533946113</td>
<td>Chiba</td>
<td>2,000</td>
</tr>
<tr>
<td>2019-09-09 00:00</td>
<td>533946113</td>
<td>Yokohama</td>
<td>5,000</td>
</tr>
</tbody>
</table>

---

### Figure 2: Example of time-series data for the real-time population aggregate by city-level in Chiba city.

### Figure 3: Examples of population ratios in different cities before and after Typhoon Faxai passed.
where shows an overview of the proposed method. In this section, we design a GCN-based approach to estimate the damage level from the real-time population dynamics. Figure 4 shows an overview of the proposed method.

population by using a more sophisticated method (e.g., ARIMA model [10]), we simply calculate an average population from the long-term data. In this analysis, we set $D_{\text{weekday}}$ to the weekdays from Aug. 1 ($\tau_1$) to Sep. 6 ($\tau_6$) and set $D_{\text{weekend}}$ to Saturdays and Sundays, except for the period from Aug. 12 to 16, because mid-August period is a common vacation time in Japan. Then, the population ratio at a target time $t'$ is computed as follows:

$$\text{pop}_{\text{ratio},a,t'} = \frac{\text{pop}_{\text{ratio},a,t'}}{\text{normal}_{\text{pop},a,d',t'}}$$  \hspace{1cm} (7)

where $d'$ and $h'$ are the day type and the time of day of $t'$.

Figure 3 shows examples of the normalized population (the population ratio) transition around Tokyo from Aug. 26 to Sep. 12, 2019. The red dashed line indicates the time the typhoon completely passed. Chiba, Ichihara, and Tateyama received serious damage (two disaster recovery laws was applied in those cities), Yokohama received relatively moderate damage (one disaster recovery law was applied), and Tokyo’s Koto ward and Mitaka city received relatively less damage (no disaster recovery law).  

**Observations:** From Figure 3, we can see the tendencies as follows.

1. The population in each city was clearly affected by the typhoon just after it passed regardless of the severity of the damage each city suffered. The difference among them is more pronounced in the population the day after or two days after the typhoon passed.
2. The change ratio of the real-time population against the normal time for residential and non-residential populations varies greatly from city to city. Although both Mitaka city and Koto ward received relatively less damage, the population ratio transition was quite different between them. This difference may be attributable to the type of each city. Mitaka city is a commuter town (residential area), whereas Koto ward has a manufacturing district. Because it is very difficult to untangle all factors and we cannot make explicit rules, it is effective to adopt a data-driven approach.

## 5 PROPOSED APPROACH

In this section, we design a GCN-based approach to estimate the damage level from the real-time population dynamics. Figure 4 shows an overview of the proposed method.

**5.1 Features for ML Model**

From the empirical analysis described in the previous subsection and our hypothesis (Section 1), we design three types of population ratio as features: the ratio of relative real-time population to normal time $x_{\text{pop ratio}}$ (Feature 1); the ratio of real-time population to stationary population $x_{\text{static ratio}}$ (Feature 2); and the ratio of residential population to non-residential population $x_{\text{res ratio}}$ (Feature 3). We also design two types of calculating methods: one for calculating the actual population ratio; and the other for anomaly score. For feature 1, the population ratio is calculated by Eq. (7), which means the population ratio is calculated at a target time against the normal time. For feature 2, we use stationary factors of a real-time population as described in Section 3. We calculate the ratio of the real-time population to static component as follows:

$$x_{\text{static ratio}} = \frac{\text{pop}_{\text{ratio},a,t'}}{\text{pop}_{\text{ratio},a,d'}}$$ \hspace{1cm} (8)

where $\text{static}_{\text{pop},a,d',t'}$ represents the static component of the decomposed population for population type $a$, day type $d'$ and the time of a day $h'$. For feature 3, the ratio of a residential population and a non-residential population is calculated as follows:

$$x_{\text{res ratio}} = \frac{\text{pop}_{\text{ratio},a,t'}}{\text{pop}_{\text{ratio},a,d'}}$$ \hspace{1cm} (9)

If we use the population ratio during $h_{\text{dur}}$ hours at target day $t_{\text{target}}$ (e.g., the day after a Typhoon passed) for city $c$ as features, and we set $h_{\text{dur}} = 3$ (i.e., 00:00, 01:00, 02:00, and 03:00), then $x_{\text{pop ratio}} = [x_{\text{pop ratio},a,0}, x_{\text{pop ratio},a,1}, x_{\text{pop ratio},a,2}, x_{\text{pop ratio},a,3}]$. In addition to the population ratio, we calculate basic statistics (i.e., max, min, average, median and std) in $x_{\text{pop ratio}}$ for each population type $a$. These basic statistics are denoted as $x_{\text{pop ratio}}$. We also calculate basic statistics for $x_{\text{static ratio}}$ and $x_{\text{res ratio}}$. These features are denoted by $x_{\text{static ratio}}$ and $x_{\text{res ratio}}$.

From the empirical analysis described in Section 4, we suppose that the real-time population at a specific time is almost the same, as seen in Figure 3. Therefore, we can assume that the real-time population at the same time of day follows a normal distribution, and we can then detect abnormal population changes by using a classical statistical test. Thus, we use anomaly scores of population ratio as features. We use Hotelling’s t-squared test [17] to compute anomaly scores for real-time populations. The basic procedure of Hotelling’s t-squared test is as follows.

**Step 1:** Calculate the mean population $\mu_{c,a,d,h}$ and population variance $\sigma_{c,a,d,h}^2$ for each city $c$, population type $a$, day type $d$, and time of a day $h$.

**Step 2:** Calculate the anomaly score $a$ for a real-time population $p_{c,a}$ as $a(p_{c,a}) = \frac{(p_{c,a} - \hat{\mu}_{c,a,d,h})^2}{\sigma_{c,a,d,h}^2}$. We calculate anomaly scores for all three types of features. We also calculate basic statistics for anomaly scores.

Finally, all features (the population ratios and their statistics, and anomaly scores and their statistics) are concatenated, and then used as input to ML model.

$$x_c = \text{CONCAT}[x_f | f \in F]$$  \hspace{1cm} (10)
We design a model called D2E-GCN, which exploits the population flow to estimate the severity of damage. Our method consists of two steps: (1) building a directed city graph from real-time population data; and (2) edge-weighted directed graph convolutional network.

5.2 D2E-GCN Model

We design a model called D2E-GCN, which exploits the population flow to estimate the severity of damage. Our method consists of two steps: (1) building a directed city graph from real-time population data; and (2) edge-weighted directed graph convolutional network.

5.2.1 Building City Graph. In the example of real-time population data in Table 2, a grid code can be associated with cities. Thus, we can consider that the real-time population as people’s movement flows from residential cities (source cities) to the city indicated by the grid code (destination city). In the case of Table 2, as the grid code 553946113 represents part of Chiyoda Ward, Tokyo, people from Chiba to Chiyoda Ward is 20,000, and people from Yokohama to Chiyoda Ward is 5,000 and so on. Based on these real-time population data, we built a directed city graph whose nodes are cities and edges are inter-city population flows. The weight of each edge represents the estimated number of people who moved between the two cities. We consider two variants of normalization for the weight of edge: (1) normalization by the maximum number for each city (Max-norm); and (2) normalization by exponential decay function (Exp-norm) defined as follows:

\[
\mathcal{w}_{ij} = 1 - \exp(-\lambda \cdot \text{inter-city_pop}_{ci, cj})
\]

where \( \lambda \) is decay parameter, \( \mathcal{w}_{ij} \) denotes the weight for a city pair (\( ci, cj \)), \( cj \) is target city (destination city), \( ci \) is source city, \( \text{inter-city_pop}_{ci, cj} \) is real-time population from city \( j \) to city \( i \) defined in the following equation:

\[
\text{inter-city_pop}_{ci, cj} = \sum_{g \in G_i; r \in \mathcal{G}} \text{pop}(t, g, r)
\]

where \( G_i \) is the set of grid codes included in city \( ci \). We use the normalization method (2) because the weights of self-loops are large compared to the number of people who moved between different cities. The unique characteristics of directed city graph compared to those in the existing studies are directed and weighted edges.

5.2.2 Edge-weighted Directed Graph Convolutional Network. We design a GCN model which can exploit the characteristics of directed city graph. To incorporate directed characteristics, we filter \( N(v) \) by the direction of an edge. Specifically, we only use neighbor nodes if the direction of nodes indicates from the neighbor nodes to a target node. Moreover, to consider the weighted edge characteristics, we replace the element-wise mean with the weighted element-wise mean. The weight for each node is calculated as follows:

\[
\tilde{\mathcal{w}}_{ij} = \frac{\mathcal{w}_{ij}}{\sum_{k \in \mathcal{N}(i)} \mathcal{w}_{ik}}
\]

Finally, our GCN model can be calculated as follows:

\[
\mathbf{h}^2_{0} = \text{ReLU}(\mathbf{W} \cdot \text{WeightedMEAN}(\mathbf{h}^{\mathcal{W}}_{0}^{-1}, u \in N(v) \cup \{v\}))
\]

where WeightedMEAN represents the weighted element-wise mean using the weight defined in Eq. (13). By uniformly sampling a fixed size of neighbors from \( N(v) \), our model can be extended to GraphSAGE [15] based model. Although Graph Attention Networks (GAT) [31] can be used to exploit the edge weight, exploiting GAT model requires more learnable parameters for the attention layer. Because we would like not to increase learnable parameters due to limited number of disaster data, we explicitly leverage the population flow from the real-time population as the edge weights. After that, \( \mathbf{h}^2_{0} \) is fed to the output layer (softmax function) for classification. To train the model, we use binary cross entropy as the loss function defined as follows:

\[
L_{CLF}(w) = -\frac{1}{n} \sum_{i=1}^{n} y_i \log(p_i) + (1 - y_i) \cdot \log(1 - \log(p_i))
\]

where \( n \) is the sample size, \( p_i \) is the output of the classifier of the \( i \)th sample and \( y_i \) is an annotated label of the \( i \)th sample.

6 EVALUATION

6.1 Evaluation Setting

Data: We evaluated our approach for the cases of Typhoons Faxai and Hagibis with regard to a total of 171 cities. That is, the number of samples is 342. Because natural disasters do not occur frequently, we selected these two Typhoons as examples of large-scale disaster in Japan in 2019. The label ratios were imbalanced: we had 27.2% for label 1 (severe) and 72.8% for label 0 (non-severe) in total. More specifically, we had 31.6% for label 1 in the case of Typhoon Faxai, and 22.8% for label 1 in the case of Typhoon Hagibis. We calculated the normal population for each city by using the real-time population data of the preceding month for each case. For calculating features, we used the real-time population data of the day after the typhoon passed based on observation (1). Specifically, as Typhoon Faxai hit around Tokyo on September 9th, the data of September 10th (=test) was used for Typhoon Faxai. For Typhoon Hagibis, as the typhoon hit around Tokyo on October 12th to the midnight of 13th, the data of October 14th was used. We set the time duration \( h_{dur} = 24 \), which is divided into time slots of three hours each (i.e., 00:00, 03:00, 06:00, 09:00, 12:00, 15:00, 18:00, 21:00, and 24:00) for the sake of reducing computational cost. In the case of Typhoon Hagibis, the last data was the real-time population at 12 a.m. on October 15th. Thus, it is considered that the estimation was carried out approximately 48 hours after the typhoon passed.

Metrics: We used the Precision, Recall, and F1 score as metrics. Recall is important for disaster recovery because it is better to have less oversight which affects human life.

Comparison Methods and Hyperparameters: We compared our GCN model with XGBoost (XGB) [9] and LightGBM (LGBM) [19] as baseline models for a model which does not consider the graph structure of cities. We selected XGB and LGBM as representative models because these models are commonly used in the current ML studies. We also compared the proposed model with the representative graph-based models, i.e. the GCN model of Kipf et al. [21] and GraphSAGE [15]. In these models, the city graph was constructed as undirected and unweighted because the methods of [21] and [15] treated that graph was undirected and the edge weight was binary. The hyperparameters of XGB and LGBM were tuned
were trained on training data, and the hyperparameters of ML models were trained within the ranges of $[1, 5]$, $[1e-8, 1.0]$, and $[1e-8, 1.0]$ respectively. For LGBM, regularization parameters ($\lambda_{l1}, \lambda_{l2}$) were tuned within the ranges of $[1e-8, 10.0]$, $[1e-8, 10.0]$, $[2, 32]$, and $[1, 3]$, respectively. In the proposed method (D2E-GCN), learnable parameters were trained using Adam optimizer [20] with learning rate $1e-3$ and weight decay $1e-4$. Number of epochs was 200, and early stopping was applied to avoid overfitting. Patience parameter for early stopping was 50. We selected the decay parameter $\lambda$ for the exponential decay function $5e-6$ from $\{1e-6, 5e-6, 1e-5, 5e-5, 1e-4, 5e-4\}$ because it was the best performance parameter.

Cross Validation: We conducted five-fold leave-one-group-out cross-validation. Specifically, the 171 cities were first divided into five groups. Four groups out of five were used as training data, and the one remaining group was used as test data. Next, the training data was further divided into training (80%) and validation (20%). Then, the ML models (baseline and the proposed model) were trained on training data, and the hyperparameters of ML models were tuned using validation data. The performance of each model was evaluated on test data.

To increase the robustness of the evaluation or comparing the proposed method with existing methods, we evaluated the prediction performance ten times by changing the random seed because the sample size was limited.

City Graph Construction: The number of nodes was 171 which is equal to the number of cities because the nodes of directed city graph were cities, whereas the number of edges was 28,757. The average degree was 168.2. The actual constructed city graph is shown in Figure 5. The blue points indicate each city and the red lines indicate the connection between cities. The line width represents the edge weight. The number of GCN layers was three, and the dimension of hidden layer was 64. We constructed the directed city graph using the real-time population data in early September, 2019 because no disaster occurred during the period. Because we consider two methods for edge weight normalization (i.e., Max-norm and Exp-norm), we compare these methods in advance. F1-score of the proposed method with Max-norm was 0.7243, and with Exp-norm was 0.7322. Thus, we use Exp-norm in the following evaluation.

### 6.2 Evaluation Results

#### 6.2.1 Quantitative Evaluation

First, we compare the performance of the proposed method with baseline methods. Table 3 shows the results. Bold style indicates the best performance, and bracket indicates standard deviation. In almost all cases except for precision of LGBM and the proposed method, there are statistically significant differences at the 0.999 level. For the case of precision among LGBM and the proposed method, there is a statistically significant difference at the 0.95 level. The performance of the original GCN [21] and GraphSAGE [15] were significantly lower than the proposed method. The proposed method with both GCN-based and GraphSAGE-based methods outperformed baseline methods regarding all metrics by approximately 4%-10%.

Next, we evaluate the effect of graph characteristics of the directed city graph. Specifically, the unique characteristics of our directed city graph compared to those of the existing studies are directed and weighted edge. Thus, to analyze the effect of each component, we perform the ablation study with the following settings:

1. **Undirected/Unweighted**: City graph constructed without directional information and edge weights. That is, the constructed city graph is undirected and the edge information is binary. This method is corresponding to the method of Kipf et al. [21].
2. **Directed/Unweighted**: City graph constructed with directional information, but without edge weights. The constructed city graph is directed and the edge information is binary.
3. **Undirected/Weighted**: City graph constructed without directional information, but with edge weights. The constructed city graph is undirected and the edge has real-valued weight.
4. **Directed/Weighted**: City graph constructed with directional information and edge weights. The constructed city graph is directed and the edge has real-valued weight.

The results of performance comparison are shown in Figure 6. By comparing (1) with (2), and (3) with (4), we can see the effect of directional information. In both (2) and (4), the directional information contributes to improving the performance. Moreover, by comparing (1) with (3), and (2) with (4), we can see the effect of considering the edge weight. Here, in both (3) and (4), the edge weight highly contributes to improving the performance. From these results, both the directional information and the edge weight are effective for estimating the damage level. The edge weight is more effective than the directional information.

Finally, we further investigate the effect of the edge weight. We verify whether only the edges with a high weight value are effective or all edges are effective with the following settings:

- A directed city graph is built by method (1) which uses the maximum number for normalization.
- Next, the edges are filtered out by thresholding if the edge weight is less than a threshold.

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>XGB</td>
<td>0.6717 (0.0680)</td>
<td>0.6185 (0.0476)</td>
<td>0.6433 (0.0535)</td>
</tr>
<tr>
<td>LGBM</td>
<td>0.7022 (0.0944)</td>
<td>0.6041 (0.0577)</td>
<td>0.6480 (0.0696)</td>
</tr>
<tr>
<td>GCN [21]</td>
<td>0.4066 (0.1340)</td>
<td>0.5110 (0.0367)</td>
<td>0.4402 (0.1074)</td>
</tr>
<tr>
<td>GraphSAGE [15]</td>
<td>0.4220 (0.1314)</td>
<td>0.5094 (0.0419)</td>
<td>0.4495 (0.1055)</td>
</tr>
<tr>
<td>D2E-GCN (GCN-based/proposed)</td>
<td>0.7535 (0.0852)</td>
<td>0.7001 (0.0730)</td>
<td>0.7249 (0.0742)</td>
</tr>
<tr>
<td>D2E-GCN (GraphSAGE-based/proposed)</td>
<td><strong>0.7599 (0.1018)</strong></td>
<td><strong>0.7089 (0.0813)</strong></td>
<td><strong>0.7323 (0.0862)</strong></td>
</tr>
</tbody>
</table>
Then, the classification performance is measured. We set the threshold in 0.1 steps from 0.0 to 0.9 and verified the performance. The results of performance comparison are shown in Figure 7. The classification performance decreases as the threshold increases. From this result, we conclude all edges are effective if the edge weight is small.

6.2.2 Qualitative Evaluation. Apart from the quantitative evaluation of classification performance, we also conducted a qualitative evaluation to analyze the effectiveness features. We investigate the feature importance of XGB and LGBM model for qualitative evaluation because GCN-based model cannot output feature importance. Table 4 shows the result of the importance ranking. Due to the space limit, we only show the result of LGBM but XGB indicates similar tendency. From the top-ranked features, we can see that (1) anomaly score is more effective than population ratio, (2) total population is more effective than other population, and (3) the night-time population (e.g., 00:00, 21:00 and 24:00) is important for classification.

6.3 Discussion, Limitation and Lessons Learned

Effectiveness: From Table 3, it is effective that the representation learning using GCN that considers the graph structure. The improvement of recall is important for disaster response because passing over a city that received serious damage is critical. The proposed method improved recall by approximately 10%, which is nearly equal to 17 cities since a total number of cities was 171. We roughly estimate the census-based population of these 17 cities by averaging census-based population of all 171 cities. The average census-based population was 207,464. then the total population of 17 cities was 3,526,888. Thus, by taking into account of label ratio (27.2%), the improvement of recall may affect $3,526,888 \times 0.272 = 959,314$, which is almost a million, people in a disaster time.

Speediness: In the case of Typhoon Faxai, the ASLRDV was applied to all cities in Chiba prefecture, Oshima town (Tokyo Metropolis), and Yokohama city (Kanagawa prefecture)\(^3\). It was applied first to Oshima, on Sep. 24, and last to Yokohama, on Oct. 9. Our method could have detected severe damage 48 hours after (at 00:00 on Sep. 11) the Typhoon hit around Tokyo. This was two weeks earlier than the first application of the ASLRDV. In addition, the average computational time of train and inference for our evaluation setting was 1193.8 (s) (std. 149.7(s)) on MacBook Pro (CPU: 2.3 GHz quad-core Intel Core i7, Memory: 32GB, no GPU). Even if it is applied to more cities, it is possible to perform high-speed calculations using GPU servers. We believe that this speediness is important for emergency circumstances, e.g., municipalities could save lives thanks to the speediness of our method.

Effect of blackout: In the case of Typhoon Faxai, a large-scale power outage occurred in approximately 930,000 households especially for Chiba prefecture\(^4\). Thus, the real-time population could be affected by power outage. Figure 8 shows the population ratio transition in Minamiboso city and Kyonan town, both of which are located in Chiba prefecture. The red line represents the time

\(^{3}\)http://www.bousai.go.jp/taisaku/eki/katsusaiken/t1typhoon15.html

the Typhoon Faxai passed. This figure indicates that the relative population of both cities are extremely less than the normal time. This is probably attributable to the mobile terminals whose batteries ran out and could not be recharged due to the power outage; because real-time population is estimated based only on mobile terminals whose batteries are live. However, the information on the occurrence of blackout itself may be useful for estimating the disaster damage level, although we need more examples of disaster to conclude the effects of black out.

**Limitations and Future Directions:** Our empirical results reveal that the proposed method can achieve a certain level of estimation accuracy in the cases of Typhoons Faxai and Hagibis. There are several limitations and future directions. First, we need to validate the method’s generalizability not only to other typhoons but also to other types of natural disasters, such as earthquakes and heavy rains. In our evaluation, although we split the data of two typhoons into train data and test data, we would like to evaluate our method using other disaster’s data. In addition to the types of a disaster, the effects of time of a day should be investigated in more detail in other cases as well. Moreover, because COVID-19 outbreaks have changed the population flows, we need data of population dynamics under the influence of the pandemic.

Second, the estimation performance has room for improvement. To build a more accurate estimator, we would like to consider the effects of various factors such as blackout, disruption of public transport, characteristics of each city (e.g., administrative, commercial, and residential etc.) because these factors can be considered related to the population dynamics. For example, although a blackout may affect the reliability of RT-MSS data, we can exploit the information itself that a blackout has occurred as a feature to estimate the damage level. Besides, we would like to explore an anomaly detection based on self-supervised learning approach because there are a lot of the data for normal time.

Third, another future direction is to apply our method to estimate the disaster level of a fine-grained location (i.e., at a grid-level), or a fine-grained severity level (e.g., multi-class and regression). Since the severity level is proportional not only to the severity of the damage but also to the population and the number of houses in the city, normalization is necessary. But the normalization method is not obvious. On the other hand, the criterion for binary classification is objectively defined by Japanese law. Thus, we formulated the problem as a binary classification in this study. One possible application of binary classification for a fine-grained severity level is exploiting the probability (not a binary label) of the output of the classifier. If we sort the probability for each city in each prefecture, the priority list for rescue would be generated. This detailed disaster damage information would probably assist rescue efforts.

Finally, we would like to consider a real-world scenario. In the actual rescue, a government will make a decision based not only on our prediction results which have speediness but also on other information sources which have more reliability but costly, e.g., weather and emergency calls. In the case of Japan, the government collects disaster information, e.g., weather information and an aerial image taken by a helicopter [7]. Thus, our prediction can be used as one of a material for decision making. We would like to integrate these information to realize more reliable model by discussing with experts on disaster management.

**Lessons Learned:** (1) Because a graph structure effectively captures the population flow, GCN model can improve the estimation performance compared to a method that does not consider the graph structure of cities. (2) In addition, because the population flows have the direction and the amount, not binary information, the model that leverages these information is valuable for analyzing the population flows. As a result, the proposed method outperformed baseline methods regarding all metrics by up to 10%. This study revealed the potential of rescuing more people in a disaster situation by estimating the severity from population dynamics although several limitations remained as described above.

7 RELATED WORK

**Disaster Management:** According to a survey paper [39], satellite imagery [26] and social-media data [25, 27, 30] provide the most commonly used datasets for disaster management. In addition to those two data sources, mobility data measured by GPS [13, 18, 28, 37] and cellular networks [4, 5, 22] is also widely used. Each data source has advantages and disadvantages. For example, although an advantage of satellite imagery is that we can directly observe disaster damage, it has the disadvantage of limited temporal resolution. Similarly, social-media data has the advantage of a real-time nature but the disadvantage of questionable credibility and less coverage for rural area as well.

Here, we focus on related work using human mobility data. Song et al. analyzed human behavior in the Great East Japan Earthquake and applied a hidden Markov model to propose a prediction model for human mobility under an emergency situation [28]. Yabe et al. used statistical anomaly detection for location coordinates to build label data [36]. Apart from individual mobility prediction, population dynamics in disasters have been investigated [22, 37]. Recently, Yabe et al., also analyzed population recovery pattern for five natural disasters (e.g., Tohoku Tsunami and Hurricane Maria) and found the factors of long-term population displacement[35].

To the best of our knowledge, no prior work has exploited population dynamics to estimate the disaster damage level.

**Graph Convolutional Networks for Geographical Application:** Recently, GCNs have attracted much attention in the ML community [33]. As people’s mobility can be represented as flows in city graphs, there are several applications of the GCN model to the geographical domain such as bike sharing systems [8, 16, 38], passenger demand prediction [32], and influenza spread forecast [11]. In these studies, graphs are constructed by regarding each location as a node, and if there is a people’s flow between two locations, an edge is set between them. Then, variants of GCN model have been applied to the graph. Most of these studies treated the graph as an undirected graph. However, the direction and edge weight in a graph were not fully exploited despite the fact that people’s movement is directional, and the information of the number of people can be leveraged. Yu et al. [40] proposed a method for spatio-temporal app usage representation. Although they exploited the edge weights, they only employed the Max-normalization method for edge weight normalization. Thus, we design a model that exploits the directional information and the edge weight.
8 CONCLUSIONS

In this paper, we have introduced a disaster damage-level estimation problem to support a quick initial response to a natural disaster. First, we conducted an empirical analysis of population transition to investigate the relationship between population dynamics and the severity of a disaster. Then, we have designed a D2E-GCN model to estimate the disaster damage level from the population dynamics. The D2E-GCN model has been designed based on two unique characteristics of directed city graph, i.e., directed and weighted edges. We evaluated our approach by using two real disaster damage datasets from the cases of Typhoons Faxai and Hagibis, which hit the Tokyo area in 2019. Our approach has achieved the best performance in terms of the F1 score (75.2%) compared to baseline methods. The evaluation results showed that the proposed method can detect severe damage quite early from official reports, before a disaster recovery law is applied. Moreover, we found the experimental result that the estimation performance can be significantly affected by the graph construction method for GCN models. The experimental result can be applied to a large number of other GCN models such as GraphSAGE. We believe that this study opens up a novel mobile sensing approach for disaster response, and hope that the approach will be helpful in real world.

REFERENCES