MOIRE: Mixed-Order Poisson Regression towards Fine-grained Urban Anomaly Detection at Nationwide Scale

Masamichi Shimosaka
Department of Computer Science
Tokyo Institute of Technology
Tokyo, Japan
simosaka@miubiq.cs.titech.ac.jp

Kota Tsubouchi
Yahoo Japan Corporation
Tokyo, Japan
ktsubouc@yahoo-corp.jp

Yanru Chen
Department of Computer Science
Tokyo Institute of Technology
Tokyo, Japan
chen@miubiq.cs.titech.ac.jp

Yoshiaki Ishihara
Department of Computer Science
Tokyo Institute of Technology
Tokyo, Japan
yishihara@miubiq.cs.titech.ac.jp

Junichi Sato
Yahoo Japan Corporation
Tokyo, Japan
jsato@yahoo-corp.jp

***Abstract*—The analysis of crowd flow in urban regions (urban dynamics) from GPS traces has been actively explored over the last decade. However, the existing prediction models assume that the population density in the analysis area is almost uniform, making it difficult to analyze fine-grained urban dynamics on a nationwide scale, where urban and rural areas coexist. In this paper, we propose a predictive model, called *mixed-order Poisson regression (MOIRE)*, to capture changes in active populations nationwide by combining lower-order patterns and higher-order interaction effects. The proposed method utilizes multiple pieces of contextual information that greatly affect crowd flows (e.g., time-of-day, day-of-the-week, weather situation, holiday calendar information). We evaluated MOIRE on two massive GPS datasets gathered in urban regions at different scales. The results show that it has better predictive performance than the state-of-the-art method. Moreover, we implemented an anomaly detection system in urban dynamics for the whole nation of Japan in accordance with MOIRE specifications. This application enabled us to confirm MOIRE's performance intuitively.**

***Index Terms*—Spatio-temporal prediction; Urban computing; Anomaly detection; Application and systems;**

I. Introduction

[1] Forecasting population density changes in urban regions, that is, urban dynamics, is playing an increasingly important role in urban planning [1], emergency management [2], public services [3], and commercial activities. In the past, urban dynamics studies relied on collecting data from questionnaire-based surveys [4]. However, such surveys usually require laborious efforts. At the same time, the rapid popularization of smart devices has led to the accumulation of a large amount of GPS data, and many recent studies on urban dynamics analysis have utilized mobility logs containing GPS information, instead of conducting expensive surveys.

[1]K. Tsubouchi equally contributed as M. Shimosaka.

Data obtained from GPS logs have unique spatio-temporal properties [5], and approaches based on tensor factorization have been used in many studies to extract urban dynamics patterns from them [6] [7] [8]. Moreover, another prominent means of analyzing human activity patterns is mixture modeling, which has high explanatory power with non-linear distributions [9] [10] [11]. However, approaches based on mixture modeling and factorization are not capable of providing accurate urban dynamics predictions; this problem is similar to the well-known cold-start problem in recommendation systems [10]. In contrast to previous work based on pattern extraction, the great majority of studies on urban dynamics have integrated contextual information into regression models and reported accurate prediction results [12] [13] [14].

However, as yet, there are no urban dynamics pattern prediction systems that use regression models at a nationwide level, because the existing methods cannot deal with large differences in population between grid cells. For instance, the population in a 1-km-square grid cell in New York City is rather different from the population in a 1-km-square grid cell in San Diego. In fact, there is always a trade-off between simple and complex model representations; that is, while a simple model representation might perform stably in both urban and rural areas, it is usually not capable of representing complicated patterns in metropolitan areas; and while a complex model might have higher explanatory power with a large population, it might become unstable in low-population areas. Similar trade-offs can be found in various fields of machine learning [15] [16], and no single method has enough flexibility and stability at the same time in urban dynamics. This trade-off issue has hitherto prevented us from developing urban dynamics prediction systems that can function at the nationwide level.

This trade-off issue is also critical when we consider the construction of event or anomaly detection systems using the predictions of urban dynamics, for example where crowd-level congestion or disaster is to be detected. Richer representation of explanatory variables tends to be suitable for providing precise predictions of everyday situations in areas with large populations; however, it also risks leading to overreaction to anomaly in some special situations, due to the zero-frequency problem. For example, we may consider whether both condition and day-of-the-week information together would improve prediction performance for regular situations, such as sunny weather conditions for holidays in amusement areas, while the combination of snowy and day-of-the-week information may lead to over-fitting in regions where it does not snow and that information is not applicable.

To address this issue, this paper proposes **M**ixed-**O**rder Poi**s**son **Re**gression (**MOIRE**) aiming for flexibility and stability over a wider range of regions. Basically, simple models with lower-order terms (e.g., linear Poisson regression) are unable to capture higher-order coupling effects, and although models with complex higher-order terms (e.g., multilinear Poisson regression) have greater flexibility, they may severely worsen the zero-frequency problem and not obtain stable results. The basic idea of MOIRE is to combine lower-order patterns and higher-order interaction effects simultaneously. We rigorously evaluated the performance of MOIRE on large grid cells with high populations and on finer-grained grid cells with fewer people in them. Finally, we implemented MOIRE in a country-wide anomaly detection system that yielded intuitive predictions across the nation of Japan.

Our contributions can be summarized as follows:

- **Mixed-Order Poisson Regression Model.** We propose the mixed-order Poisson regression (MOIRE) model for urban dynamics prediction. Moreover, we design a practical feature encoding approach to produce MOIRE models with high predictive power for urban dynamics.
- **Comprehensive experiments to validate the effectiveness of the proposed method.** We conducted experiments on two massive datasets compiled from smartphone mobility logs with GPS information, including crowd flow data in large urban regions and in finer-grained grid cells. The results showed that our method is capable of providing better predictions than the state-of-the-art method on both scales of urban region.
- **Implementation of MOIRE in a national-scale urban anomaly detection system.** We implemented an anomaly detection system for Japan in accordance with the MOIRE methodology, and found that the system can extract accurate patterns regardless of the size of the population in each grid cell. We conducted two case studies and found that the system provided intuitive anomaly detection when MOIRE was used to extract urban dynamics patterns.

II. Related Work

This section highlights the relevant work for our project in terms of accurate modeling of urban dynamics and large-scale event detection using GPS traces.

Accurate Urban Dynamics Prediction: Due to the properties of the spatio-temporal data, methods based on tensor factorization have gained remarkable attentions recently. Fan et al. modeled people flow with a three-dimensional tensor and decomposed it into basic life-pattern tensors by non-negative tensor factorization [7]. Zheng et al. modeled the noisy situation of NYC with a three-dimensional tensor and recovered the noisy situation through a context-aware tensor-decomposition approach [17]. However, it is not trivial to apply this model for event detection. Mixture modeling is also employed to capture basic spatio-temporal population patterns [11] [18]; however, these methods have a well-known cold-start problem [10] in the recommendation system, and the estimation may be unstable depending on the combination of included features . In contrast to the generative modeling of urban dynamics, recent advances in urban dynamics prediction that include convolutional neural networks [19] provide highly accurate prediction results. This approach is inspired by the success of neural-based computer vision, and assumes that the layout of the target grid cells is like an image. However, this is not always true when we consider urban dynamics prediction at the nationwide level, because the population density per cell is highly diverse at that level. Though some work based on generalized regression discusses the importance of higher-order interaction among the explanatory variables [14], no work discusses the importance of mixed order of interaction to achieve both stability and flexibility.

Large-Scale Anomaly Detection System: Anomaly detection / event detection is expected to be one prominent application of large-scale GPS traces. Typical social events to be detected include damaged/congested areas due to disasters, and congested areas thanks to festivals or similar events. Recent advances in the literature shows that the system is able to detect congested railway stations [20]; however, it is not feasible to apply this technique to find events other than railway events at a national scale. Zhang et al. [19] showed accurate crowd flow prediction and event detection results; however, it is not trivial to apply this technique across any or all regions (as the target area is not always laid out in a rectangle). Jing et al. also focused on event detection [21]; however, their model is also inspired by the convolution operation like images, and therefore is not applicable to nationwide event detection. In contrast to these models, our model is individually optimized by the data obtained in the corresponding cell, and therefore, we could ensure the scalability of the model towards nationwide anomaly detection. The generalized regression approach [14] could be scalable since their parameters in each cell are optimized individually across cells. However, the performance and stability of their model are not so good as those of our model (see experimental results below).

III. Problem-Setting and Baselines

A. Problem-Setting as Poisson Regression

In this paper, daily changes in the active population within a target urban region are modeled as urban dynamics. We divide one day into several time segments and define the active population in the target area of a certain time segment based on the total number of access logs in that segment. It can be assumed that the active population of each time segment follows a Poisson distribution.

We denote the number of time segments in one day as S. In the target urban region, h_s represents the active population of the s-th time segment, which is a non-negative integer. Therefore, $H = [h_1, h_2, \ldots, h_S]$ can represent a one-day active population transition. Following a prior study [14], we assume that each h_s follows a Poisson distribution, and thus that the likelihood of h_s can be written as $p(h_s) = \mathrm{Pois}(h_s|\lambda_s) = \frac{\lambda_s^{h_s}\mathrm{e}^{-\lambda_s}}{h_s!}$, where λ_s is the mean parameter of the Poisson distribution. Here, to predict the true active population h_s, our task turns into one of estimating the mean parameter λ_s.

B. SOIRE: Single-Order Poisson Regression Models (Baseline)

As shown in Section II, many Poisson regression models have been researched in attempts to capture urban dynamics patterns. Unlike MOIRE, all of the previous methods calculate only single-order models; that is, there are many varieties of single-order Poisson regression models (SOIRE). Here, we treat them individually as baselines.

1) Time-Only Poisson Regression Model:
We have observed that the active population is greatly affected by the time-of-day, so we can assume the mean parameter λ is regressed by a time factor with weight parameters as follows:

$$\lambda(\boldsymbol{t}) = \exp(\boldsymbol{t}^\mathsf{T}\boldsymbol{W}) \tag{1}$$

where $\boldsymbol{W}$ is a weight-parameter vector and $\boldsymbol{t}$ is an S-dimensional vector of which the s-th component corresponds to the s-th time segment.

This time feature vector $\boldsymbol{t}$ can be formulated as an indicator function of a normal distribution:

$$\boldsymbol{t} = \{t_s | t_s = \mathcal{N}(s|\tau, \sigma), s = 1, \ldots, S\}, \tag{2}$$

where $\mathcal{N}(\tau, \sigma)$ is a normal distribution with mean parameter τ, corresponding to the target time segment.

However, this time-only Poisson regression model does not utilize external information such as weather situation or calendar information, and thus it cannot capture the effect of external factors on the active population. Figure 1(a) illustrates the prediction of a time-only Poisson regression model compared with the ground truth during a weekday and a weekend in Shinjuku, Tokyo. It shows that time-only Poisson regression provides the same prediction regardless of the significant differences in population dynamics between the weekday and weekend.

2) Linear Poisson Regression Model:
Prior studies show that active population transition is closely related to many external factors, such as day-of-the-week, weather situation, and calendar [14] [22]. To model urban dynamics in a way that incorporates these external factors, a naive approach is to simply concatenate all the factors into features of the model:

$$\lambda(\boldsymbol{t}, \boldsymbol{d}) = \exp([\boldsymbol{t}^\mathsf{T}, \boldsymbol{d}_1^\mathsf{T}, \boldsymbol{d}_2^\mathsf{T}, \ldots, \boldsymbol{d}_N^\mathsf{T}]\boldsymbol{W}) \tag{3}$$

where $\boldsymbol{d} = \{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_N\}$ are external factor vectors with one-hot encoding.

However, such a model does not work properly, because it is not capable of handling the interaction effect between factors. For example, Figure 1(b) shows the predictions made by a linear Poisson regression model of population changes during a weekday and during a weekend in Shinjuku. We can see that although this prediction model captures the decrease in the active population on the weekend, it does not combine the effects of the time and day-of-the-week factors: the predicted crowd flow also has two commute peaks for the weekend.

3) Bilinear Poisson Regression Model:
In order to handle the coupling effect between the time factor and external factors on active population, a bilinear Poisson regression model is proposed [14]:

$$\lambda(\boldsymbol{t}, \boldsymbol{d}) = \exp([\boldsymbol{d}_1^\mathsf{T}, \boldsymbol{d}_2^\mathsf{T}, \ldots, \boldsymbol{d}_N^\mathsf{T}]\boldsymbol{W}\boldsymbol{t}). \tag{4}$$

Here, the weight parameter $\boldsymbol{W}$ is in the matrix representation. This model has already shown advantages for handling the interaction effect between the time factor and external factors.

However, since all the external factors are simply concatenated in this model, it cannot capture coupling effects among external factors $\{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_N\}$.

4) Multilinear Poisson Regression Model:
The coupling effect among all crowd flow factors can be handled by the following multilinear Poisson regression model:

$$\lambda(\boldsymbol{t}, \boldsymbol{d}) = \exp(\mathrm{Vec}(\boldsymbol{t} \otimes \boldsymbol{d}_1 \otimes \boldsymbol{d}_2 \otimes \cdots \otimes \boldsymbol{d}_N)^\mathsf{T}\boldsymbol{W}) \tag{5}$$

where $\otimes$ is the Kronecker product operator, $\mathrm{Vec}(\cdot)$ denotes a vectorization operation to expand the tensor-form feature into a vector, and $\boldsymbol{W}$ is a weight-parameter vector. The idea of this model was proposed in [22], which adopted a tensor form for the weight parameter.

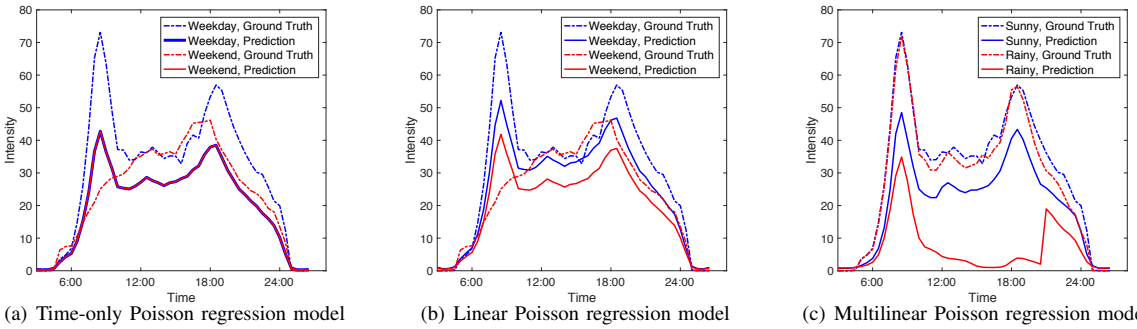
On the one hand, the multilinear Poisson regression model hopefully captures the coupling effects between all factors, but on the other hand, combining all the factors by the Kronecker product would increase the feature dimension and greatly worsen the zero-frequency problem. Figure 1(c) illustrates the prediction result for a rainy day and a sunny day in Shinjuku. The performance of this model exhibits a dramatic decline during the period from 10:00 to 21:00 on the rainy day, which results from the worsened zero-frequency problem. When we try to calculate the ratio between current observed crowd flow and predicted crowd flow and then employ it as an anomaly score for event detection, the precision of event detection tends to be deteriorate due to this improper prediction.

IV. MOIRE: Mixed-Order Poisson Regression Model

A. Basic Idea of MOIRE

The discussion of single-order Poisson regression models in the previous section has indicated that a trade-off exists between simple and complex feature representations: simple models with lower-order terms are unable to capture higher-order coupling effects; while models with complex higher-order terms have greater flexibility, but may significantly worsen the zero-frequency problem and not obtain stable results.

The basic idea of the mixed-order Poisson regression (MOIRE) is to represent the urban dynamics parameter using lower-order patterns and higher-order coupling patterns simultaneously. MOIRE represents the urban dynamics of grid cells with a fluent population by using higher-order coupling terms so that it can represent the complicated urban dynamics of metropolitan areas. At the same time, MOIRE can represent grid cells with a small population by using only the lower-order patterns, thereby avoiding the zero-frequency (or low-frequency) problem.



(a) Time-only Poisson regression model (b) Linear Poisson regression model (c) Multilinear Poisson regression model

Fig. 1. Prediction results of three single-order Poisson regression models

B. Generalized Representation of MOIRE

Analogous to multiple linear regression, all of the single-order Poisson regression models discussed previously that have independent variables $\{\boldsymbol{d}_0, \boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_N\}$ that can be extended into an enriched and generalized representation:

$$\ln \lambda(\boldsymbol{t}, \boldsymbol{d}) = W_0 + \sum_{i=0}^{N} \boldsymbol{d}_i^{\mathsf{T}} \boldsymbol{W}_1^{(i)} + \sum_{i=0}^{N} \sum_{j=i}^{N} \mathrm{Vec}(\boldsymbol{d}_i \otimes \boldsymbol{d}_j)^{\mathsf{T}} \boldsymbol{W}_2^{(i,j)} + \sum_{i=0}^{N} \sum_{j=i}^{N} \sum_{k=j}^{N} \mathrm{Vec}(\boldsymbol{d}_i \otimes \boldsymbol{d}_j \otimes \boldsymbol{d}_k)^{\mathsf{T}} \boldsymbol{W}_3^{(i,j,k)} + \ldots \tag{6}$$

where W_0 is a constant and $\boldsymbol{W}_1^{(i)}, \boldsymbol{W}_2^{(i,j)}, \boldsymbol{W}_3^{(i,j,k)}$ are weight-parameter vectors. Specifically, $\boldsymbol{W}_1$ is composed of weight-parameter vectors corresponding to the first-order terms, while the components of $\boldsymbol{W}_2$ are weight-parameter vectors of the second-order terms, and so forth.

Therefore, this generalized model can include an infinite number of terms consisting of independent variables with different orders. Actually, each of the single-order Poisson regression models discussed above can be regarded as a special case of this generalized model with a certain combination of terms having the same order.

In practice, we do not use all the terms in the generalized model due to concern over computational cost. We explore a great variety of combinations of different-order terms for feature encoding, from which new predictive models are obtained, here called mixed-order Poisson regression (MOIRE) models.

Theoretically, the advantage of our model is that it captures the higher-order coupling effect for higher explanatory power and compensates for zero-frequency situations by utilizing lower-order patterns, thereby achieving flexibility and stability at the same time.

C. Mixing Up Multiple Orders for MOIRE

As mentioned above, the features for MOIRE can be produced using various polynomial combinations of factors (including the time factor and external factors). In practice, we first determine several fundamental models and then create combinations and derivatives based on them.

The fundamental models used in the experiments included the four single-order Poisson regression models mentioned above and one external-only Poisson regression model:

1) **Time-only:** Time-only Poisson regression model, defined by Eq. (1)
2) **Linear:** Linear Poisson regression model, defined by Eq. (3)
3) **Bilinear:** Bilinear Poisson regression model, defined by Eq. (4)
4) **Multilinear:** Multilinear Poisson regression model, defined by Eq. (5)
5) **External-only:** The external-only Poisson regression model includes only external factors simply concatenated into a feature, which implies that it does not consider a time factor. Although this model alone may not make satisfactory predictions, it can be used as a benchmark for comparison as well as a fundamental model for combination. It is defined as $\lambda(\boldsymbol{d}) = \exp([\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_N]^{\mathsf{T}} \boldsymbol{W})$.

A variety of new MOIRE models can be obtained as derivatives based on the fundamental models above by employing some of the following operations:

- **Simple Combination:** Simply combine features from some of the fundamental models to make a feature of the new predictive model. For example, a **Linear** and **Bilinear** model combination, denoted as **Linear+Bilinear**, can be written as $\ln \lambda(\boldsymbol{t}, \boldsymbol{d}) = [\boldsymbol{t}^{\mathsf{T}}, \boldsymbol{d}_1^{\mathsf{T}}, \ldots, \boldsymbol{d}_N^{\mathsf{T}}, \mathrm{Vec}^{\mathsf{T}}([\boldsymbol{d}_1^{\mathsf{T}}, \ldots, \boldsymbol{d}_N^{\mathsf{T}}] \otimes \boldsymbol{t})]\boldsymbol{W}$.
- **Adding a Constant Feature:** Add a constant feature to the model. For example, a **Linear+Bilinear** model with a constant feature, denoted as **Linear+Bilinear+C**, can be written as $\ln \lambda(\boldsymbol{t}, \boldsymbol{d}) = [1, \boldsymbol{t}^{\mathsf{T}}, \boldsymbol{d}_1^{\mathsf{T}}, \ldots, \boldsymbol{d}_N^{\mathsf{T}}, \mathrm{Vec}^{\mathsf{T}}([\boldsymbol{d}_1^{\mathsf{T}}, \ldots, \boldsymbol{d}_N^{\mathsf{T}}] \otimes \boldsymbol{t})]\boldsymbol{W}$. In fact, a model with only one constant feature degenerates to the simple Poisson regression model. Therefore, we expect that a constant feature will be helpful to capture the basic rate of population change.
- **Low-Rank Decomposition:** The weight-parameter vectors of all five fundamental models can be rewritten in matrix form. Assuming that the rank of the weight-parameter matrix is prone to decrease, we adopt a low-rank decomposition to achieve a rank reduction. (Note that since the intention of this operation is to reduce the rank and make parameter learning more stable, we did not conduct this decomposition on models with a quasi-diagonal weight matrix.)
  For example, a **Linear+Bilinear+C** model with a low-rank decomposition can be written as follows:

$$\begin{aligned} \ln \lambda(\boldsymbol{t}, \boldsymbol{d}) &= [1, \boldsymbol{t}^{\mathsf{T}}, \boldsymbol{d}_1^{\mathsf{T}}, \ldots, \boldsymbol{d}_N^{\mathsf{T}}, \mathrm{Vec}^{\mathsf{T}}([\boldsymbol{d}_1^{\mathsf{T}}, \ldots, \boldsymbol{d}_N^{\mathsf{T}}] \otimes \boldsymbol{t})]\boldsymbol{W} \\ &= [1, \boldsymbol{d}_1^{\mathsf{T}}, \ldots, \boldsymbol{d}_N^{\mathsf{T}}]\boldsymbol{W}[1, \boldsymbol{t}^{\mathsf{T}}]^{\mathsf{T}}. \end{aligned} \tag{7}$$

For the low-rank decomposition, the weight matrix $\boldsymbol{W} \in \mathbb{R}^{(M+1)\times(S+1)}$ shown above can be assumed to be a product of two low-rank matrices, $\boldsymbol{U} \in \mathbb{R}^{(M+1)\times K}$ and $\boldsymbol{V} \in \mathbb{R}^{(S+1)\times K}$. Note that K, M and S satisfy $K \ll M$ and $K \ll S$. The above equation becomes

$$\begin{aligned} \ln \lambda(\boldsymbol{t}, \boldsymbol{d}) &= [1, \boldsymbol{d}_1^{\mathsf{T}}, \ldots, \boldsymbol{d}_N^{\mathsf{T}}]\boldsymbol{W}[1, \boldsymbol{t}^{\mathsf{T}}]^{\mathsf{T}} \\ &= [1, \boldsymbol{d}_1^{\mathsf{T}}, \ldots, \boldsymbol{d}_N^{\mathsf{T}}]\boldsymbol{U}\boldsymbol{V}^{\mathsf{T}}[1, \boldsymbol{t}^{\mathsf{T}}]^{\mathsf{T}}. \end{aligned} \tag{8}$$

This shrinkage does not only makes the model more stable in learning the parameters; it also makes the result more interpretable, which is helpful for understanding the characteristics of urban dynamics.

The model is optimized by maximizing the log likelihood against training data. Given L days of training data over L cells

for S time segments, the log likelihood of the data is written as

$$\begin{aligned}\ln L(\boldsymbol{U},\boldsymbol{V}) &= \sum_l \sum_s \ln \mathrm{Pois}(h^{(l,s)}|\lambda(\boldsymbol{t},\boldsymbol{d})) \\ &\propto \sum_l \sum_s \{h^{(l,s)}\boldsymbol{d}^{(l,s)}\boldsymbol{U}\boldsymbol{V}^{\mathsf{T}}[1,\boldsymbol{t}^{(l,s)\mathsf{T}}]^{\mathsf{T}} \\ &\quad - \exp(\boldsymbol{d}^{(l,s)}\boldsymbol{U}\boldsymbol{V}^{\mathsf{T}}[1,\boldsymbol{t}^{(l,s)\mathsf{T}}]^{\mathsf{T}})\}.\end{aligned} \tag{9}$$

V. Experiments

We conducted experiments on two massive datasets to demonstrate that the proposed method achieves both stability in coarse-grained grid cells and expressiveness in finer-grained ones. The datasets were collected from smartphone mobility logs containing GPS information; one dataset contains crowd flow data in 300 large urban areas (square regions 900 m in length), while the other contains data in finer-grained mesh regions 200 m in length.

A. Experiment on Large Urban Areas

1) Dataset: The dataset for this experiment was gathered from the mobility logs provided by a disaster alert mobile application [2]. A mobility log was used to record data only when the mobile device was moving, to make it suitable for analysis of the active population.

Three-hundred urban areas were selected as target regions. The size of each target area was 900 m × 900 m, and as the active population, the number of access logs was counted within each target region and each time segment of 365 days (from July 1, 2013 to June 30, 2014). The time interval was set to 30 minutes.

2) Evaluated Models: We evaluated 28 predictive models, including the single-order Poisson regression models and MOIRE models produced by our method. The models are listed in Table I.

TABLE I
Evaluated Models

Models	Formulation of λ
Time-only:	$\ln \lambda_1(\boldsymbol{t}) = \boldsymbol{t}^{\mathsf{T}}\boldsymbol{W}_1$
Linear:	$\ln \lambda_2(\boldsymbol{d},\boldsymbol{t}) = [\boldsymbol{d}_1^{\mathsf{T}},\boldsymbol{d}_2^{\mathsf{T}},\boldsymbol{d}_3^{\mathsf{T}},\boldsymbol{t}^{\mathsf{T}}]\boldsymbol{W}_2$
Bilinear:	$\ln \lambda_3(\boldsymbol{d},\boldsymbol{t}) = Vec^{\mathsf{T}}([\boldsymbol{d}_1^{\mathsf{T}},\boldsymbol{d}_2^{\mathsf{T}},\boldsymbol{d}_3^{\mathsf{T}}]\otimes\boldsymbol{t})\boldsymbol{W}_3$
Multilinear:	$\ln \lambda_4(\boldsymbol{d},\boldsymbol{t}) = Vec^{\mathsf{T}}(\boldsymbol{d}_1\otimes\boldsymbol{d}_2\otimes\boldsymbol{d}_3\otimes\boldsymbol{t})\boldsymbol{W}_4$
External-only:	$\ln \lambda_5(\boldsymbol{d}) = [\boldsymbol{d}_1^{\mathsf{T}},\boldsymbol{d}_2^{\mathsf{T}},\boldsymbol{d}_3^{\mathsf{T}}]W_5$
Linear+Bilinear:	$\ln \lambda_6(\boldsymbol{d},\boldsymbol{t}) = \ln \lambda_2(\boldsymbol{d},\boldsymbol{t}) + \ln \lambda_3(\boldsymbol{d},\boldsymbol{t})$
Linear+Multilinear:	$\ln \lambda_7(\boldsymbol{d},\boldsymbol{t}) = \ln \lambda_2(\boldsymbol{d},\boldsymbol{t}) + \ln \lambda_4(\boldsymbol{d},\boldsymbol{t})$
Bilinear+Multilinear:	$\ln \lambda_8(\boldsymbol{d},\boldsymbol{t}) = \ln \lambda_3(\boldsymbol{d},\boldsymbol{t}) + \ln \lambda_4(\boldsymbol{d},\boldsymbol{t})$
Linear+Bilinear+Multilinear:	$\ln \lambda_9(\boldsymbol{d},\boldsymbol{t}) = \ln \lambda_2(\boldsymbol{d},\boldsymbol{t}) + \ln \lambda_3(\boldsymbol{d},\boldsymbol{t}) + \ln \lambda_4(\boldsymbol{d},\boldsymbol{t})$
Time-only+C:	$\ln \lambda_{10}(\boldsymbol{t}) = [1,\boldsymbol{t}^{\mathsf{T}}]\boldsymbol{W}_{10}$
Linear+C:	$\ln \lambda_{11}(\boldsymbol{d},\boldsymbol{t}) = [1,\boldsymbol{d}_1^{\mathsf{T}},\boldsymbol{d}_2^{\mathsf{T}},\boldsymbol{d}_3^{\mathsf{T}},\boldsymbol{t}^{\mathsf{T}}]\boldsymbol{W}_{11}$
Bilinear+C:	$\ln \lambda_{12}(\boldsymbol{d},\boldsymbol{t}) = [1,Vec^{\mathsf{T}}([\boldsymbol{d}_1^{\mathsf{T}},\boldsymbol{d}_2^{\mathsf{T}},\boldsymbol{d}_3^{\mathsf{T}}]\otimes\boldsymbol{t})]\boldsymbol{W}_{12}$
Multilinear+C:	$\ln \lambda_{13}(\boldsymbol{d},\boldsymbol{t}) = [1,Vec^{\mathsf{T}}(\boldsymbol{d}_1\otimes\boldsymbol{d}_2\otimes\boldsymbol{d}_3\otimes\boldsymbol{t})]\boldsymbol{W}_{13}$
External-only+C:	$\ln \lambda_{14}(\boldsymbol{d}) = [1,\boldsymbol{d}_1^{\mathsf{T}},\boldsymbol{d}_2^{\mathsf{T}},\boldsymbol{d}_3^{\mathsf{T}}]\boldsymbol{W}_{14}$
Linear+Bilinear+C:	$\ln \lambda_{15}(\boldsymbol{d},\boldsymbol{t}) = \ln \lambda_2(\boldsymbol{d},\boldsymbol{t}) + \ln \lambda_{12}(\boldsymbol{d},\boldsymbol{t})$
Linear+Multilinear+C:	$\ln \lambda_{16}(\boldsymbol{d},\boldsymbol{t}) = \ln \lambda_2(\boldsymbol{d},\boldsymbol{t}) + \ln \lambda_{13}(\boldsymbol{d},\boldsymbol{t})$
Bilinear+Multilinear+C:	$\ln \lambda_{17}(\boldsymbol{d},\boldsymbol{t}) = \ln \lambda_3(\boldsymbol{d},\boldsymbol{t}) + \ln \lambda_{13}(\boldsymbol{d},\boldsymbol{t})$
Linear+Bilinear+Multilinear+C:	$\ln \lambda_{18}(\boldsymbol{d},\boldsymbol{t}) = \ln \lambda_2(\boldsymbol{d},\boldsymbol{t}) + \ln \lambda_3(\boldsymbol{d},\boldsymbol{t}) + \ln \lambda_{13}(\boldsymbol{d},\boldsymbol{t})$
Bilinear+Time-only:	$\ln \lambda_{19}(\boldsymbol{d},\boldsymbol{t}) = \ln \lambda_3(\boldsymbol{d},\boldsymbol{t}) + \ln \lambda_1(\boldsymbol{t})$
Bilinear+Time-only+C:	$\ln \lambda_{20}(\boldsymbol{d},\boldsymbol{t}) = \ln \lambda_3(\boldsymbol{d},\boldsymbol{t}) + \ln \lambda_{10}(\boldsymbol{t})$
Bilinear+External-only:	$\ln \lambda_{21}(\boldsymbol{d},\boldsymbol{t}) = \ln \lambda_3(\boldsymbol{d},\boldsymbol{t}) + \ln \lambda_5(\boldsymbol{d})$
Bilinear+External-only+C:	$\ln \lambda_{22}(\boldsymbol{d},\boldsymbol{t}) = \ln \lambda_3(\boldsymbol{d},\boldsymbol{t}) + \ln \lambda_{14}(\boldsymbol{d})$
Bilinear (LR):	$\ln \lambda_{23}(\boldsymbol{d},\boldsymbol{t}) = [\boldsymbol{d}_1^{\mathsf{T}},\boldsymbol{d}_2^{\mathsf{T}},\boldsymbol{d}_3^{\mathsf{T}}]\boldsymbol{U}_{23}\boldsymbol{V}_{23}\boldsymbol{t}$
Multilinear (LR):	$\ln \lambda_{24}(\boldsymbol{d},\boldsymbol{t}) = Vec^{\mathsf{T}}(\boldsymbol{d}_1\otimes\boldsymbol{d}_2\otimes\boldsymbol{d}_3)\boldsymbol{U}_{24}\boldsymbol{V}_{24}\boldsymbol{t}$
Bilinear+Multilinear (LR):	$\ln \lambda_{25}(\boldsymbol{d},\boldsymbol{t}) = [\boldsymbol{d}_1^{\mathsf{T}},\boldsymbol{d}_2^{\mathsf{T}},\boldsymbol{d}_3^{\mathsf{T}},Vec^{\mathsf{T}}(\boldsymbol{d}_1\otimes\boldsymbol{d}_2\otimes\boldsymbol{d}_3)]\boldsymbol{U}_{25}\boldsymbol{V}_{25}\boldsymbol{t}$
Bilinear+Time-only (LR):	$\ln \lambda_{26}(\boldsymbol{d},\boldsymbol{t}) = [1,\boldsymbol{d}_1^{\mathsf{T}},\boldsymbol{d}_2^{\mathsf{T}},\boldsymbol{d}_3^{\mathsf{T}}]\boldsymbol{U}_{26}\boldsymbol{V}_{26}\boldsymbol{t}$
Bilinear+External-only (LR):	$\ln \lambda_{27}(\boldsymbol{d},\boldsymbol{t}) = [\boldsymbol{d}_1^{\mathsf{T}},\boldsymbol{d}_2^{\mathsf{T}},\boldsymbol{d}_3^{\mathsf{T}}]\boldsymbol{U}_{27}\boldsymbol{V}_{27}[1,\boldsymbol{t}]$
Bilinear+Linear+C (LR):	$\ln \lambda_{28}(\boldsymbol{d},\boldsymbol{t}) = [1,\boldsymbol{d}_1^{\mathsf{T}},\boldsymbol{d}_2^{\mathsf{T}},\boldsymbol{d}_3^{\mathsf{T}}]\boldsymbol{U}_{28}\boldsymbol{V}_{28}[1,\boldsymbol{t}]$

Feature design: The formulation for the time factor was introduced in Eq. (2). We used day-of-the-week, is-holiday-or-not, and weather information as external factors. Day-of-the-week was a 7-dimensional vector $\boldsymbol{d}_1$ with one-hot encoding; is-holiday-or-not was a 2-dimensional vector $\boldsymbol{d}_2$; and weather data were collected from the Japan Meteorological Agency's website[3]. The weather was divided into four categories: {sunny (1), cloudy (2), rainy (3), or severe (4)}. Therefore, the weather factor was a 4-dimensional vector $\boldsymbol{d}_3$ with one-hot encoding.

[2]A reference to the dataset is not included in the reference list due to the double blind review policy. It will be made public if this paper is accepted.

[3]http://www.data.jma.go.jp/obd/stats/etrn

3) Performance Measurement: Mean absolute error (MAE) rankings were used for performance measurement. Given the ground truth data $h^{(l,s)}$ and prediction value $\hat{h}^{(l,s)}$ in l-th location, s-th time segment in the test dataset, **MAE:** $= \frac{1}{LS}\sum_{l=1}^{L}\sum_{s=1}^{S}|h^{(l,s)} - \hat{h}^{(l,s)}|$, and **Rank:** A five-fold cross validation (CV) was conducted for each model in each target region. All the models were ranked by the average value of MAE overall CVs in each target region.

4) Results:
Table II lists the predictive performance and Figure 2 shows the distribution of rankings for each model. Note that in Table II, MAE_Mean is the average MAE value across all 300 target regions, and rank_Mean is the average rank. The first column shows the five fundamental models and MOIRE models after combination. In the second column: "Plain" means the plain model shown in the first column, "+C" means the plain model combined with a constant feature, "LR" means the model conducting low-rank decomposition, "+C & LR" means the plain model with a constant feature and conducting low-rank decomposition. We can conclude from the results that

- the three proposed low-rank decomposition MOIRE models: **Linear+Bilinear+C** (LR), **Bilinear+Time-only** (LR) and **Bilinear+External-only** (LR) all outperformed **Bilinear** (LR) proposed by [14];
- the MOIRE model **Linear+Bilinear+C** (LR) ranked first in 82% (246 out of 300) of the regions, while **Bilinear** (LR) ranked 4th in 240 out of 300 regions (Figure 2);
- adding a constant term in the feature encoding helped to improve the prediction performance of most of the models in this experiment, probably because models including a constant term can capture the basic, constant behavior of population changes and thus make more accurate predictions;
- those models using low-rank decomposition outperformed the plain model.

TABLE II
Predictive Performance of all 28 Models

Models		MAE_Mean	Rank_Mean
Time-only	Plain	**166.29**	**21.9**
	+ C	167.25	23.3
Linear	Plain	134.95	21.7
	+ C	**134.71**	**20.8**
Bilinear	Plain	118.37	15.8
	+ C	116.95	10.4
	LR	**115.60**	**4.2**
Multilinear	Plain	250.18	25.8
	+ C	156.65	24.4
	LR	**130.87**	**21.4**
External-only	Plain	331.78	27.9
	+ C	**331.58**	**27.0**
Linear+Bilinear	Plain	116.30	7.9
	+ C	116.07	6.5
	+ C & LR	**115.07**	**1.3**
Linear+Multilinear	Plain	119.87	17.4
	+ C	**119.61**	**16.5**
Bilinear+Multilinear	Plain	122.52	19.1
	+ C	118.17	14.9
	LR	**117.64**	**13.6**
Linear+Bilinear+Multilinear	Plain	117.26	11.7
	+ C	**116.97**	**9.9**
Bilinear+Time-only	Plain	116.16	7.5
	+ C	115.80	5.1
	LR	**115.16**	**2.0**
Bilinear+External-only	Plain	117.41	13.0
	+ C	117.21	11.7
	LR	**115.46**	**3.2**

B. Experiment on Fine-Grained Mesh Regions

1) Dataset: We carefully chose a large urban region and divided it into grid cells as targets to be analyzed. The whole area covered 3 km × 3 km in the center of Tokyo. The size of each grid cell was

Model	MAE_Mean	Rank_Mean	Distribution of Rank																									
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
LowRank(Linear + Bilinear + Constant)	**115.07**	**1.29**	246	43	3	3	0	3	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LowRank(Bilinear + Time-only)	115.16	2.01	36	242	14	4	2	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LowRank(Bilinear + External-only)	115.46	3.22	2	4	252	25	8	7	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LowRank(Bilinear)	115.60	4.22	4	1	9	240	25	8	8	4	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bilinear + Time-only + Constant	115.80	5.14	7	3	14	17	200	34	12	4	4	1	2	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Linear + Bilinear + Constant	116.07	6.49	0	1	3	4	24	161	71	18	8	3	2	3	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
Bilinear + Time-only	116.16	7.50	0	1	3	6	24	54	63	61	50	29	8	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Linear + Bilinear	116.30	7.94	0	0	0	0	0	4	87	164	30	10	1	0	3	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Bilinear + Constant	116.95	10.38	0	0	0	0	0	0	0	0	100	71	81	29	8	9	1	0	0	0	0	0	1	0	0	0	0	0	0	0
Linear + Bilinear + Multilinear + Constant	116.97	9.86	0	0	0	0	10	18	41	31	50	44	33	30	18	6	3	3	5	3	2	1	0	2	0	0	0	0	0	0
Bilinear + External-only + Constant	117.21	11.74	0	0	0	0	0	1	0	0	9	56	61	99	43	20	8	2	0	0	0	0	0	0	0	1	0	0	0	0
Linear + Bilinear + Multilinear	117.26	11.69	0	0	0	0	0	1	8	11	22	68	41	42	56	25	8	2	4	6	3	1	2	0	0	0	0	0	0	0
Bilinear + External-only	117.41	13.04	0	0	0	0	0	0	0	0	1	4	35	59	105	51	37	5	1	1	0	0	0	1	0	0	0	0	0	0
LowRank(Bilinear + Multilinear)	117.64	13.56	0	0	1	0	1	2	1	3	23	11	27	21	35	46	73	31	9	5	2	9	0	0	0	0	0	0	0	0
Bilinear + Multilinear + Constant	118.17	14.91	0	0	0	0	0	0	1	0	0	0	1	7	22	104	77	68	3	2	6	6	1	2	0	0	0	0	0	0
Bilinear	118.37	15.81	0	0	0	0	1	2	5	1	1	0	3	8	5	20	58	84	29	83	0	0	0	0	0	0	0	0	0	0
Linear + Multilinear + Constant	119.61	16.47	0	0	0	0	1	0	0	1	0	0	1	0	1	10	24	90	139	29	3	0	1	0	0	0	0	0	0	0
Linear + Multilinear	119.87	17.43	0	0	0	1	0	0	0	0	0	1	0	1	0	0	7	11	108	168	0	2	1	0	0	0	0	0	0	0
Bilinear + Multilinear	122.52	19.13	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	270	5	14	1	7	0	0	0	0	0
LowRank(Multilinear)	130.87	21.43	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	155	15	39	6	84	0	0	0	0
Linear + Constant	134.71	20.77	0	3	1	0	1	2	0	1	1	1	0	1	1	5	0	0	0	1	1	32	124	79	41	5	0	0	0	0
Linear	134.95	21.65	3	1	0	0	2	1	1	0	0	0	2	0	2	2	3	0	1	0	1	4	33	123	74	46	1	0	0	0
Multilinear + Constant	156.65	24.42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	80	6	211	1	0	0
Time-only	166.29	21.91	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	11	79	48	13	61	76	9	0	0	0
Time-only + Constant	167.25	23.26	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	6	58	40	29	78	79	9	0	0
Multilinear	250.18	25.79	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	2	4	0	282	1	7
External-only + Constant	331.58	27.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	283	9
External-only	331.78	27.95	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	284	

Fig. 2. Distribution of ranking of each model

200 m $\times$ 200 m, so there were 225 grid cells in this experiment. Similar to the previous experiment, we used data collected over 365 days and divided one day into 48 time intervals.

2) Performance Measurement: Given the ground truth data $h^{[l,s]}$ and prediction value $\hat{h}^{[l,s]}$, we employ

$$\text{MNLL} = -\frac{1}{LS}\sum_{l=1}^{L}\sum_{s=1}^{S}\ln p(h^{(l,s)}|\hat{h}^{(l,s)}),$$

as well as MAE used in the previous experiment.

3) Evaluated Models: Since the low-rank decomposition models showed advantages in the previous experiment, we evaluated four of them with regularization using the Frobenius norm in this experiment. The baseline model was the low-rank decomposition bilinear model proposed in a prior study [14]. We also evaluated three low-rank decomposition MOIRE models. The formulations of λ for these models are listed in Table III.

TABLE III
EVALUATED MODELS

Model	Formulation of λ
Bilinear (LR)	$\ln\lambda(\boldsymbol{d},\boldsymbol{t}) = [\boldsymbol{d}_1^\top,\boldsymbol{d}_2^\top,\boldsymbol{d}_3^\top]\boldsymbol{U}\boldsymbol{V}\boldsymbol{t}$
Bilinear+Time-only (LR)	$\ln\lambda(\boldsymbol{d},\boldsymbol{t}) = [1,\boldsymbol{d}_1^\top,\boldsymbol{d}_2^\top,\boldsymbol{d}_3^\top]\boldsymbol{U}\boldsymbol{V}\boldsymbol{t}$
Bilinear+External-only (LR)	$\ln\lambda(\boldsymbol{d},\boldsymbol{t}) = [\boldsymbol{d}_1^\top,\boldsymbol{d}_2^\top,\boldsymbol{d}_3^\top]\boldsymbol{U}\boldsymbol{V}[1,\boldsymbol{t}]$
Bilinear+Linear+C (LR)	$\ln\lambda(\boldsymbol{d},\boldsymbol{t}) = [1,\boldsymbol{d}_1^\top,\boldsymbol{d}_2^\top,\boldsymbol{d}_3^\top]\boldsymbol{U}\boldsymbol{V}[1,\boldsymbol{t}]$

4) Results: Table IV shows the mean and median of MAE and MNLL of the evaluated models.[4] Figure 3 is the box plot for MNLL. We can learn from these results that

- our method provides up to a 5.43% reduction in MAE and 12.58% reduction in MNLL relative to the baseline model **Bilinear** (LR) proposed by [14];
- as shown in the box plots, the proposed model **Bilinear+Time-Only** (LR) has not only a smaller mean and median MNLL, but also a much smaller variance, compared with the baseline model.

TABLE IV
PERFORMANCE OF FOUR LOW-RANK DECOMPOSITION MODELS

Model	MAE Mean	MAE Median	MNLL Mean	MNLL Median
Bilinear (LR) **(Baseline)**	4.121	2.349	3.791	2.722
Bilinear+External-only (LR)	4.232	2.458	3.528	2.640
Bilinear+Time-only (LR)	**3.897**	**2.247**	3.327	**2.474**
Bilinear+Linear+C (LR)	4.104	2.406	**3.314**	2.541

[4]The significance of the differences among the MNLL & MAE medians was validated with a Mann–Whitney U test.

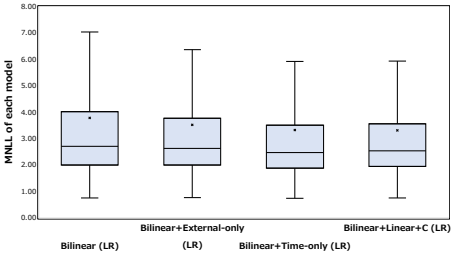


Fig. 3. MNLL of four low-rank decomposition models

VI. NATIONWIDE ANOMALY DETECTION SYSTEM

Urban dynamics prediction can be used for anomaly detection. Previous research has attempted to visualize the degree of an anomaly [14], but as yet there is no system for visualizing the degree of huge anomalies, such as a large-scale disaster, at the national level. In this section, we present an innovative large-scale anomaly detection system in terms of urban dynamics and exemplify the visualization results for anomalies in the case of nationwide disasters in Japan.

A. Dataset

We used crowd flow data collected from individual location data sent from a smartphone installed with an app and agreeing to GPS transmission. The dataset was composed from GPS data acquired all over Japan and canceled personal information. The degree of anomaly was calculated for each grid cell measuring 500 m x 500 m. In the system, we selected 82,842 cells among all the cells in Japan island as targets of anomaly detection, where there are at least 3 million GPS records for 90 days recording. In other words, we eliminate areas where quite small number of GPS data. In the training phase, the GPS data obtained for 90 days before the target date of anomaly detection is used.

B. Model Used for Visualization

In constructing an anomaly state detection system, it is important to accurately predict the steady state, or else the deviation from the steady state cannot be obtained accurately. We selected **Bilinear+Linear+C** (LR) for this experiment, on the basis of the results of previous experiments. This model is one of the MOIRE models. We used day-of-the-week and is-holiday-or-not as features of this model. Each day was divided into 48 intervals.

C. Metric for Anomaly Measurement

The degree of the anomaly $\mathcal{I}^{(l,s)}$ is defined as $\mathcal{I}^{(l,s)} = \frac{h^{(l,s)}-\hat{h}^{(l,s)}}{\hat{h}^{(l,s)}}$, and this score is used for anomaly visualization. Here, $h^{(l,s)}$ indicates the observed value at each interval. Whereas $\hat{h}^{(l,s)}$ denotes the predicted one. The score increases when the area is congested, and it approaches -1 as the observed one becomes smaller than the prediction. Note that we remove the anomaly calculation when $\hat{h}^{(l,s)}$ approaches 0 due to instability.

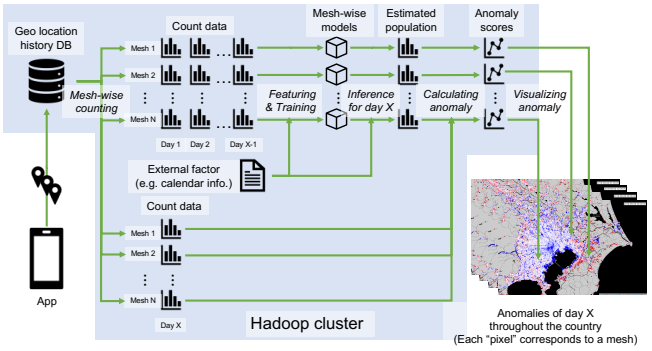


Fig. 4. Large anomaly detection system

D. Architecture of Anomaly Detection System

Figure 4 shows the overall system diagram. This system is roughly divided into conduct two functions. The first is a steady-state estimation using count data. Since this calculation can be performed independently for each grid cell, it is performed using simultaneous multi-expansion by a Spark application running on a Hadoop cluster. The second function is to calculate the anomaly score using the estimated value and the actual measurement for the day on which the user wants to calculate the anomaly degree, and plot it on a map of each grid cell and each interval.

E. Visualization Examples of Anomaly

This section exemplifies the visualization results on the degree of anomaly via the system in order to verify the effectiveness of MOIRE. As an intuitive confirmation of the effectiveness of the anomaly visualization, we consider two cases of ultra-large disasters (typhoons) in Japan, in September and October 2019 respectively.

1) Visualization of Anomalies after Typhoon Strikes in Tokyo: A very large typhoon, "Faxai," born on September 5 at distance 1,800 km south of Japan, was landing in Tokyo region in early morning of September 9, 2019. Due to the strong rain and wind caused by Faxai, all transportation systems in the region were stopped through morning commuter time, then gradually recovered by noon that day. In Tokyo-region transportation systems, very high congestion was observed around late morning, because the crowd had to stay at home until the recovery of transportation systems, then immediately go out to their workplaces after that recovery.

In Figure 5, a timeline of the anomalies on the day after the typhoon struck, and the anomalies on another (usual) day (the bottom right) are illustrated. In this figure, the blue part indicates an anomaly cell where fewer people stay than usual, while the red part indicates more people than usual. The white part indicates a regular cell, where the number of the people stays is expected to match the prediction results. In contrast, the dark gray areas are out of anomaly detection, where average population is less than some threshold. If we employ such areas for visualization, the result tends to be noisy and interferes without understanding of crowd behavior through the systems. Therefore, we eliminate such areas from the visualization result.

The timeline shows a snapshot of anomalies in Tokyo at 30 minutes intervals, indicating that 1) trains were stopped in a wide area (7:00 AM), and so anomaly few people stayed at railway stations 2) some trains began operating, and so more people started to commute, that is, anomaly congestion at railways in suburban regions was observed (9:00 AM), 3) the congestion gradually shifted into the central part of metropolitan Tokyo (11:00 AM), 4) the congestion was eliminated and the operation of trains recovered to normal level (2:00 PM). This sequence implies that our anomaly detection systems could capture local transition of congestion as well as global trend of congestion behavior in the Tokyo region. It should be noted that we found anomaly areas with red parts in the east side of Tokyo region (Chiba prefecture) and the south side (Kanagawa prefecture) even when congestion in the central region of Tokyo disappeared. This may be because people in this region stayed at home due to the out-of-service railways and interruption of electric service. In contrast, for crowd behavior on the day the typhoon struck, the rightmost part at the bottom in Figure 5 shows the degree of the anomaly at 7:00 AM on another day, where the crowd commuted regularly. This implies that the proposed anomaly detection system is able to suppress false positives.

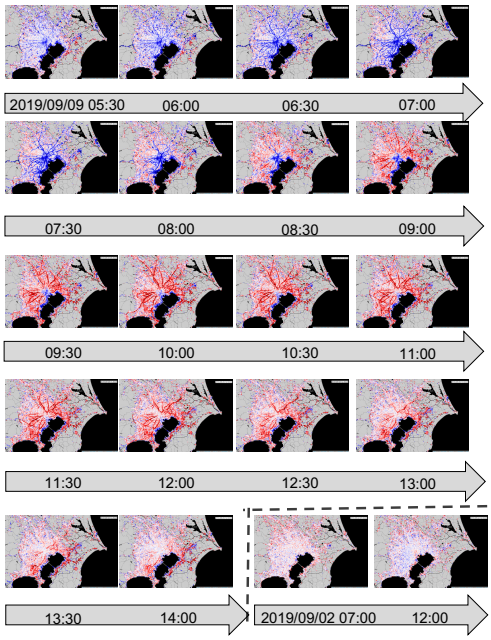


Fig. 5. Sequence of anomalies in transportation systems due to Typhoon Faxai

2) Large-Area Anomaly Visualization: Next, we exemplify the visualization result of the anomaly detection system for another typhoon, "Hagibis," which struck Japan in October 2019. Hagibis, born October 6, 2019, in the Mariana Islands, landed in Japan on October 12. It was notorious as one of the largest typhoons ever to hit Japan; as this implies, its effects were not merely local, but spanned the national level, where 64 people in Japan were killed by Hagibis. The case of this nationwide disaster is used for the verification of our anomaly detection system. Throughout the visualization of anomalies caused by this typhoon, we also illustrate the corresponding rainfall map in Japan in order to ascertain the relationship between the damage done by Hagibis and anomalies in terms of urban dynamics.

Figure 6 depicts the anomaly visualization at nationwide scale when Hagibis struck Japan. The weather maps are provided by JAXA[5]. The snapshots are taken from October 11 at 6:00 to October 14 at 0:00 per every 6 hours. At 12:00 and 18:00 on October 12 th, it is obvious that there are wide areas of anomaly in the east of Japan due to the strong rain and wind when Hagibis was landing Japan. This sequence implies that the anomaly trend in terms of urban dynamics is highly ascertained with the movement of the gigantic typhoon Hagibis. It should be noted that our anomaly detection system is able to capture anomalies on a finer-grained map in rural areas. The bottom right figures show the anomalies in Nagano prefecture where flooding occurred in this area [6]. As this visualization result shows, our anomaly detection system is able to help us understand anomalies at both the nationwide scale and the fine-grained local scale.

[5]https://sharaku.eorc.jaxa.jp/GSMaP/index.htm
[6]https://www.sankei.com/affairs/news/191014/afr1910140011-n1.html

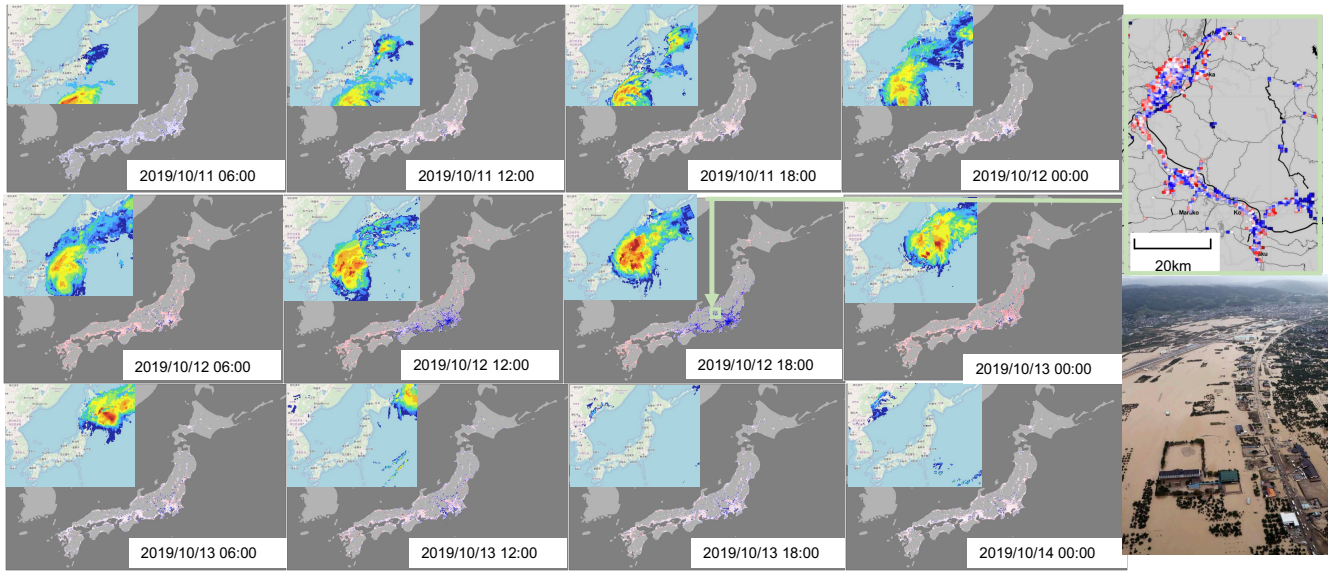


Fig. 6. Anomaly scores and precipitation during Hagibis struck in national scale, and fine grained anomaly scores in flooded area

VII. Conclusion

We proposed the *Mixed-Order Poisson Regression (MOIRE)* model for urban dynamics prediction. MOIRE enables us to combine higher- and lower-order effects of contextual information on crowd flow. Moreover, we also designed a practical approach to produce MOIRE models with high predictive power. We evaluated our method in experiments on two massive datasets composed from smartphone mobility logs. In addition, we constructed a large-scale nationwide anomaly detection system using highly accurate steady-state predictions of MOIRE. We found that the system captured not only anomalies caused by traffic disruption but also the spatio-temporal anomaly distribution along the path of the typhoon.

References

[1] J. Yuan, Y. Zheng, and X. Xie, "Discovering regions of different functions in a city using human mobility and pois," in *Proc. of ACM SIGKDD*, 2012.

[2] X. Song, Q. Zhang, Y. Sekimoto, T. Horanont, S. Ueyama, and R. Shibasaki, "Modeling and probabilistic reasoning of population evacuation during large-scale disaster," in *Proc. of ACM SIGKDD*, 2013.

[3] L. Sun, D.-H. Lee, A. Erath, and X. Huang, "Using smart card data to extract passenger's spatio-temporal density and train's trajectory of MRT system," in *Proc. of the ACM SIGKDD international workshop on urban computing*, 2012.

[4] Y. Sekimoto, R. Shibasaki, H. Kanasugi, T. Usui, and Y. Shimazaki, "Pflow: Reconstructing people flow recycling large-scale social survey data," *IEEE Pervasive Computing*, vol. 10, no. 4, pp. 27–35, 2011.

[5] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "DNN-based prediction model for spatio-temporal data," in *Proc. of ACM SIGSPATIAL*, 2016.

[6] K. Takeuchi, R. Tomioka, K. Ishiguro, A. Kimura, and H. Sawada, "Non-negative multiple tensor factorization," in *Proc. of ICDM*, 2013.

[7] Z. Fan, X. Song, and R. Shibasaki, "Cityspectrum: A non-negative tensor factorization approach," in *Proc. of UbiComp*, 2014.

[8] Z. Yao, Y. Fu, B. Liu, W. Hu, and H. Xiong, "Representing urban functions through zone embedding with human mobility patterns," in *Proc. of IJCAI*, 2018.

[9] M. Shimosaka, T. Ishino, H. Noguchi, T. Sato, and T. Mori, "Detecting human activity profiles with dirichlet enhanced inhomogeneous poisson processes," in *Proc. of ICPR*, 2010.

[10] B. Lika, K. Kolomvatsos, and S. Hadjiefthymiades, "Facing the cold start problem in recommender systems," *Expert Systems with Applications*, vol. 41, no. 4, pp. 2065–2073, 2014.

[11] K. Nishi, K. Tsubouchi, and M. Shimosaka, "Extracting land-use patterns using location data from smartphones," in *Proc. of the 1st International Conference on IoT in Urban Space*, 2014, pp. 38–43.

[12] L. Dai, W. Qin, H. Xu, T. Chen, and C. Qian, "Urban traffic flow prediction: A Map-Reduce based parallel multivariate linear regression approach," in *Proc. of ITSC*, 2014.

[13] L. Ma, X. Yan, and W. Qiao, "A quasi-poisson approach on modeling accident hazard index for urban road segments," *Discrete dynamics in nature and society*, vol. 2014, 2014.

[14] M. Shimosaka, K. Maeda, T. Tsukiji, and K. Tsubouchi, "Forecasting urban dynamics with mobility logs by bilinear poisson regression," in *Proc. of UbiComp*, 2015.

[15] I. Kamwa, S. Samantaray, and G. Joós, "On the accuracy versus transparency trade-off of data-mining models for fast-response pmu-based catastrophe predictors," *IEEE Transactions on Smart Grid*, vol. 3, no. 1, pp. 152–161, 2011.

[16] J. Hainmueller and C. Hazlett, "Kernel regularized least squares: Reducing misspecification bias with a flexible and interpretable machine learning approach," *Political Analysis*, vol. 22, no. 2, pp. 143–168, 2014.

[17] Y. Zheng, T. Liu, Y. Wang, Y. Zhu, Y. Liu, and E. Chang, "Diagnosing New York city's noises with ubiquitous data," in *Proc. of UbiComp*, 2014.

[18] M. Shimosaka, T. Tsukiji, S. Tominaga, and K. Tsubouchi, "Coupled hierarchical Dirichlet process mixtures for simultaneous clustering and topic modeling," in *Proc. of ECML/PKDD*, 2016.

[19] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. of AAAI*, 2017.

[20] T. Konishi, M. Maruyama, K. Tsubouchi, and M. Shimosaka, "CityProphet: City-scale irregularity prediction using transit app logs," in *Proc. of UbiComp*, 2016.

[21] R. Jiang, X. Song, D. Huang, X. Song, T. Xia, Z. Cai, Z. Wang, K.-S. Kim, and R. Shibasaki, "DeepUrbanEvent: A system for predicting citywide crowd dynamics at big events," in *Proc. of ACM SIGKDD*, 2019.

[22] M. Shimosaka, T. Tsukiji, H. Wada, and K. Tsubouchi, "Predictive population behavior analysis from multiple contexts with multilinear poisson regression," in *Proc. of ACM SIGSPATIAL*, 2018.