Congestion Forecast for Trains with Railroad-Graph-based Semi-Supervised Learning using Sparse Passenger Reports

Soto Anno*
Tokyo Institute of Technology
Tokyo, Japan
anno@miubiq.cs.titech.ac.jp

Kota Tsubouchi
LY Corporation
Tokyo, Japan
ktsubouc@lycorp.co.jp

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

ABSTRACT

Forecasting rail congestion is crucial for efficient mobility in transport systems. We present rail congestion forecasting using reports from passengers collected through a transit application. Although reports from passengers have received attention from researchers, ensuring a sufficient volume of reports is challenging due to passenger's reluctance. The limited number of reports results in the sparsity of the congestion label, which can be an issue in building a stable prediction model. To address this issue, we propose a **s**emi-s**u**pe**r**vised method for **con**gestion **for**ecasting for **t**rains, or **SURCONFORT**. Our key idea is twofold: firstly, we adopt semi-supervised learning to leverage sparsely labeled data and many unlabeled data. Secondly, in order to complement the unlabeled data from nearby stations, we design a railway network-oriented graph and apply the graph to semi-supervised graph regularization. Empirical experiments with actual reporting data show that SURCONFORT improved the forecasting performance by 14.9% over state-of-the-art methods under the label sparsity.

CCS CONCEPTS

• **Information systems** → *Information systems applications*; • **Human-centered computing** → *Empirical studies in ubiquitous and mobile computing*;

KEYWORDS

Railway Passengers, Congestion Forecasting, Train Congestion, Railroad Graph, Graph Regularization, Sparse User Reports

ACM Reference Format:
Soto Anno*, Kota Tsubouchi, and Masamichi Shimosaka. 2024. Congestion Forecast for Trains with Railroad-Graph-based Semi-Supervised Learning using Sparse Passenger Reports. In *The 32nd ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL '24), October 29-November 1, 2024, Atlanta, GA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3678717.3691239

* This work was performed during a research internship at LY Corporation.

SIGSPATIAL '24, October 29-November 1, 2024, Atlanta, GA, USA

ACM ISBN 979-8-4007-1107-7/24/10.
https://doi.org/10.1145/3678717.3691239

1 INTRODUCTION

Forecasting rail congestion is crucial for transport systems, as congestion can pose risks such as commuters falling from platforms, breaking windows, or operations with doors unclosed. Traditionally, congestion has been monitored through ticket gates [8] and CCTV footage [9]. However, ticket gates could not quantify passengers inside individual trains. Moreover, it is well known that vision-based technique like CCTVs suffers from occlusion [9].

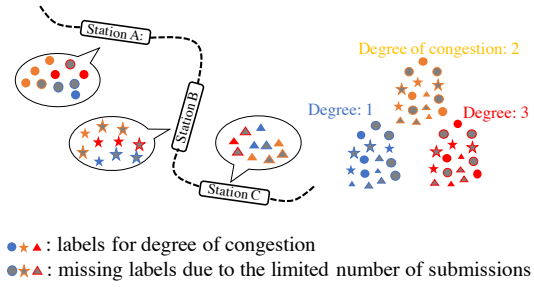
Crowdsourced information provided by passengers has received attention from researchers. Lathia et al. demonstrated the effectiveness of crowdsourced data for timely updates on congestion in transportation systems [6]. In fact, many transit apps, such as Jorudan's Japan Transit Planner, NAVITIME, and LY Corporation's Transit Navigation App, have begun collecting congestion reports from passengers. Our study aims **to forecast rail congestion by leveraging these passenger-submitted reports**.

However, passenger reports are often sparse, as passengers may hesitate to submit reports on heavily crowded trains. As a result, many railways, stations, and time slots lack congestion labels, making it challenging to forecast congestion stably, especially for dates and times with no past reports.

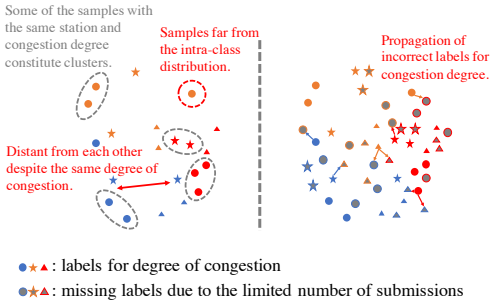
To address this issue, we propose railroad-graph-based **s**emi-s**u**pe**r**vised methodology for **con**gestion **for**ecasting of **t**rain, or **SURCONFORT**, which trains a neural network (NN) to classify the degree of congestion at a given station, date, and time. The core idea of SURCONFORT is twofold: (1) the adoption of semi-supervised learning (SSL) for mitigating the need for labeled data, and (2) railroad network-oriented graph for complementing predictions for unlabeled data by leveraging geospatially nearby labeled stations.

Firstly, we adopt SSL, which has shown promise in computer vision and image classification in recent years [2]. SSL uses sparsely labeled data (e.g., congestion labels for railways, stations, and time slots) and large amounts of unlabeled data (containing only covariates) to improve predictive performance over models trained on labeled data alone. SSL relies on finding relationships between labeled and unlabeled data, often through graph-based methods [7]. However, scarce labeled data makes it difficult to build these graphs or can result in label prediction errors propagating across them.

Secondly, to build an effective graph, we focus on the railroad network, aiming to forecast congestion for unlabeled data using labeled data from nearby stations. Cai et al. found that rail congestion tends to propagate through a network [4], meaning nearby stations often share similar congestion levels, while distant ones differ. To capture this, we design a *railroad graph* where nodes represent stations, and edges reflect station connectivity, direction, and proximity. We then apply graph regularization to the NN, ensuring similar predictions for nearby stations on the railroad graph.



(a) Our problem (left) and the ideal descriptor space (right).



(b) Descriptor space formed by fully-supervised methods (left) and existing semi-supervised methods (right).

Figure 1: (a) Conceptual illustration of our problem (left) and the descriptor space in an ideal state (right). The shape of the data point represents the sample congestion at each adjacent station (circle for station A, star for station B, triangle for station C), and the three colors represent the level of congestion at each station (blue for congestion level 1, orange for 2, red for 3). Samples missing labels in the UGC data are in gray, and the actual congestion level is reflected in the border's color. (b) Conceptual illustration of descriptor space formed by fully-supervised methods (left) and existing SSL methods (right).

The contributions of this work are as follows: (1) we propose SURCONFORT for forecasting train congestion by using sparse passenger reports; (2) we build a railway graph reflecting both line connectivity and geographical proximity to ensure that predictions for proximate stations are similar; (3) we demonstrate the superiority of SURCONFORT over state-of-the-art methodologies by using actual reporting data collected through a transit application.

2 PRELIMINARIES

Problem Formulation. As shown in Fig. 1(a), our goal is to forecast train congestion based on user posts. Users report train congestion levels through a transit app after searching for routes. When submitting, they choose one of four congestion levels: 1 (able to sit), 2 (able to stand comfortably), 3 (shoulders touching), or 4 (unable to move). Each post includes the last departure station, date and time, and the selected congestion level.

We model train congestion using information about stations, dates, and time periods. Let s represent a station, d a date, and t a time period. The station feature is defined as $\boldsymbol{s} \in \mathbb{R}^S$, where S is the number of stations on a railway line. The context feature for date d, $\boldsymbol{c}^{(d)} \in \mathbb{R}^9$, includes the day of the week and holiday status. A day is divided into T segments (e.g., $T = 144$ for 10-minute intervals), and the time feature is $\boldsymbol{t} \in \mathbb{R}^T$. These features are created using 1-of-K encoding. The degree of congestion is the average user-reported level, discretized as $y^{(s,d,t)} \in C := \{0, 1, 2, 3\}$ at station s, date d, and time t. Using the above notation, we can express a collection of n samples $X = (\boldsymbol{x}_1, ..., \boldsymbol{x}_l, \boldsymbol{x}_{l+1}, ..., \boldsymbol{x}_n)$, where each sample $\boldsymbol{x}_i{}^\top = [\boldsymbol{s}_i{}^\top, \boldsymbol{c}_i{}^\top, \boldsymbol{t}_i{}^\top] \in \mathcal{X}$ corresponds to a tuple of station, date, and time (s, d, t) indexed by i. The first l samples ($\boldsymbol{x}_i$ for $i \in L = \{1, ..., l\}$), denoted as X_L, are labeled according to $Y_L = (y_1, ..., y_l)$. The remaining $u = n - l$ samples ($\boldsymbol{x}_i$ for $i \in U = \{l + 1, ..., n\}$), denoted as X_U, are unlabeled due to limited submission data for certain stations, dates, and times.

Our goal is to build a classifier using labeled samples X_L with Y_L and unlabeled samples X_U. The classifier is a model that takes an input from $\mathcal{X}$ and outputs a vector of class confidence scores for congestion, denoted as $f_{\boldsymbol{\theta}} : \mathcal{X} \mapsto \mathbb{R}^4$, where $\boldsymbol{\theta}$ represents the model parameters. The predicted congestion level is the one with the highest confidence score, given by $\hat{y}_i = \arg\max_j f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)_j$, where j is the j-th dimension of the vector.

Challenges in Modeling Congestion with Sparse Passenger Reports. As shown in Fig. 1(b), when the dataset contains limited labeled samples, such as user-generated content (UGC) missing key congestion indicators, neural networks trained in a fully-supervised manner struggle to capture meaningful patterns. This can lead to dispersed descriptors within the same congestion level, making accurate predictions difficult. Semi-supervised methods, like LP-DeepSSL [3], attempt to mitigate this by using pseudo-labels and label propagation, but they are prone to compounding errors due to inaccurate affinity matrices. As a result, label mispredictions can propagate through the model, reducing overall performance.

3 PROPOSED METHOD: SURCONFORT

The previous section highlights that the key challenge in predicting sparse data is creating optimal descriptor spaces within the network. Given our assumption of similar congestion dynamics, the descriptor space should capture the adjacency or spatial proximity between stations. Our approach leverages the concept of graph regularization [1] to refine descriptor spaces by ensuring that feature representations of neighboring points on the graph are mapped close to one another so the model assigns similar labels to neighboring samples. To achieve this, we employ a neural graph machine (NGM) [1], a type of semi-supervised learning that integrates neural models with graph regularization. We build the railroad graph and define a graph regularization term based on it. This enables the model to ensure that descriptors from adjacent stations share similar representations if they exhibit the same congestion levels. A conceptual illustration is provided in Fig. 2.

We leverage graph regularization to deal with the dispersion issue discussed in Section 2. The model is based on graph theory, where each node represents a station, and each edge indicates the similarity between two stations. We define the weighted difference between two descriptors, $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$, as $\omega_{\mathrm{G}}(\boldsymbol{v}_i, \boldsymbol{v}_j) = \frac{1}{2}||\boldsymbol{v}_i - \boldsymbol{v}_j||_2^2 W_{i,j}$,

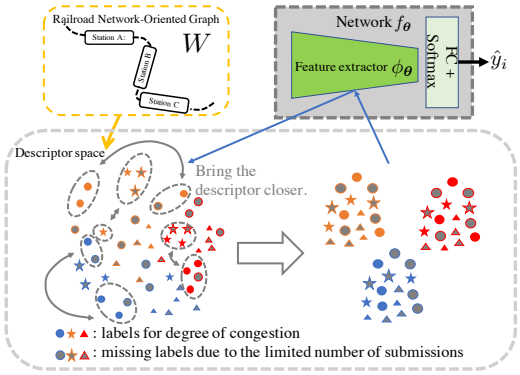


Figure 2: Conceptual illustration of SURCONFORT.

where $W_{i,j}$ denotes a railroad network-oriented adjacency matrix reflecting the similarity between two stations s_i and s_j of the descriptors $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$. This regularization aligns the descriptor space within the model with the structure of the railroad network.

To define the similarity between two stations, $W_{i,j}$, we account for the heterogeneous properties of the railroad, as train congestion dynamics can spread to adjacent stations due to the network's track connections. A simple strategy to incorporate this intuition into $W_{i,j}$ is to use a spatial proximity measure, such as cosine similarity between the locations of two stations: $W_{i,j} = \cos(\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_j)$, where $\boldsymbol{\sigma}_i$ is the spatial embedding vector for station s_i (e.g., latitude and longitude), and $\cos(\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_j) = \frac{\boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_j}{||\boldsymbol{\sigma}_i|| ||\boldsymbol{\sigma}_j||}$. However, this approach does not account for actual railroad connections or the direction (up/down) of travel at a station.

Therefore, we assume that the similarity between two stations is determined by both their connections and spatial proximity. To integrate this domain knowledge into graph regularization, we define a railroad network-oriented adjacency matrix by applying a graph cut method based on train up/down lines, as follows:

$$W_{i,j} = \begin{cases} 1, & \text{if } s_i \in \phi(s_j) \text{ or } s_j \in \phi(s_i) \\ 1 - d/d_{\max}, & \text{if } s_i \notin \phi(s_j), s_j \notin \phi(s_i), d < d_{\max} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where d is the distance between station s_i and s_j, $d_{\max}$ is a predefined maxima of d to ensure the sparseness of the affinity matrix, and $\phi(s)$ is the set of stations connecting to station s.

To perform semi-supervised learning with the graph-regularized objective, we train the model using NGM [1]. The loss function can be defined as follows:

$$L'_{\mathrm{G}}(X_L, Y_L; \boldsymbol{\theta}) = \sum_{i=1}^{l} l_s(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i) + \zeta_{\mathrm{G}} \sum_{(i,j)\in\mathcal{D}_{\mathrm{LL}}} \omega_{\mathrm{G}}(\boldsymbol{v}_i, \boldsymbol{v}_j) + \zeta_{\mathrm{G}} \sum_{(i,j)\in\mathcal{D}_{\mathrm{LU}}} \omega_{\mathrm{G}}(\boldsymbol{v}_i, \boldsymbol{v}_j) + \zeta_{\mathrm{G}} \sum_{(i,j)\in\mathcal{D}_{\mathrm{UU}}} \omega_{\mathrm{G}}(\boldsymbol{v}_i, \boldsymbol{v}_j) \quad (2)$$

where $l_s(\cdot,\cdot)$ is the sample-wise loss function (e.g., cross-entropy loss), ζ_{G} is a hyperparameter controlling the strength of the graph regularization, and $\mathcal{D}_{\mathrm{LL}}$, $\mathcal{D}_{\mathrm{LU}}$, and $\mathcal{D}_{\mathrm{UU}}$ are sets of pairs of labeled-labeled, labeled-unlabeled, and unlabeled-unlabeled samples, respectively.

4 EXPERIMENTS

4.1 Experimental Settings

Datasets and Experimental Setups. We evaluated the models using a dataset of actual user-submitted congestion posts, collected via the transit search engine provided by LY Corporation. The dataset consists of six months' worth of records from November 1st, 2020, to May 20th, 2021. Each post was made after a route search was done and while trains were running on that route. Each record contains the last departure station, the date and time of posting, and an anonymized user ID, which was deleted in the data preprocessing. We aggregated the raw posts from each station and date and time segment to calculate the average degree of congestion. We did not use any personally identifiable information in this experiment.

We selected the JR Yamanote line for our experiments, one of Tokyo's busiest lines with 31.81 million passengers weekly, highlighting the importance of congestion forecasting. The proposed method uses no regional parameters, making it applicable to other areas or lines. We treated one day as 24 hours, dividing it into 144 time segments (10 minutes each). Data from 1:20 A.M. to 4:30 A.M., during out-of-service hours, were excluded from model training and testing. The preprocessed dataset contained 2,034,779 samples, including 10,373 labeled and 2,024,406 unlabeled data points. We varied the labeled training data by 10%, 25%, 50%, 75%, and 100% of the 10,373 labeled samples to assess model robustness when labeled data are sparse. Performance was evaluated using 5-fold cross-validation, with four subsets for training and one for testing.

Model Setting. For the context denoted by $\boldsymbol{c}_d$, we used day-of-the-week and holiday features. The day-of-the-week feature was a seven-dimensional vector, and the holiday feature was a two-dimensional vector, both one-hot encoded. We concatenated these vectors as ${\boldsymbol{c}_d}^\top = [{\boldsymbol{c}_d^{(1)}}^\top, {\boldsymbol{c}_d^{(2)}}^\top] \in \mathbb{R}^9$. For hyperparameters, we set $\delta = 0.9$, $k = 50$, and $\gamma = 3$, following [3]. For the graph regularization term, we used $\zeta_{\mathrm{G}} = 0.7$ in the evaluation (Section 4.2.1).

Comparison Methods. We compare SURCONFORT with the following baselines: (1) **Random**, which randomly predicts the labels; (2) **MODE**, which returns the most frequent labels from the training data for the same day and time. If no such data exists, it randomly selects a label; (3) **SNN**, a simple neural network trained in a fully-supervised manner. The model consists of four fully connected layers with ReLU activation for the first three layers and Softmax for the final layer. The output dimensions of the layers are 128, 256, 128, and 4, respectively. Batch normalization layers were added before each fully connected layer, except the first one; (4) **LP** (Label Propagation) [11] and (5) **LS** (Label Spreading) [10], pioneering methods in graph-based semi-supervised learning, which perform label induction using the "natural graph," where the similarity between two samples is defined by the L2 distance in the input space; (6) **LP-DSSL** [3], a state-of-the-art method for graph-based semi-supervised learning, which uses the descriptor-based graph. All models were optimized using Adam [5] with a learning rate 0.0001.

4.2 Experimental Results

Table 1: Performance comparison for railroad congestion prediction. The second column lists learning protocols: "stats." refers to statistical methods, while SL and SSL indicate supervised and semi-supervised learning, respectively.

			Ratio of labeled data (%)				
Model	Protocol	Graph	10%	25%	50%	75%	100%
Random	-	-	24.60	25.05	25.81	24.34	25.46
MODE	stats.	-	28.03	31.66	37.34	41.42	44.61
SNN	SL	-	54.94	56.15	56.23	57.42	58.75
LP [11]	**SSL**	natural	51.99	52.92	54.42	56.01	56.84
LS [10]	**SSL**	natural	51.99	52.92	54.42	56.02	56.83
LP-DSSL [3]	**SSL**	descriptor	49.38	52.97	55.35	57.10	58.88
SURCONFORT	**SSL**	**rail**	**56.76**	**58.08**	**59.41**	**60.52**	**60.35**

4.2.1 Performance Comparison. The experimental results are presented in Table 1, using classification accuracy as the evaluation metric. SURCONFORT outperformed all other graph-based approaches across all percentages of labeled data.

We observed a decline in prediction performance for all models as the amount of labeled data decreased, highlighting the severe impact of label sparsity in UGC data on prediction accuracy. Despite this, the proposed method successfully minimized performance loss. Specifically, SURCONFORT improved forecasting performance by 9.2% compared to LP and LS, 14.9% compared to LP-DSSL, and 3.3% compared to SNN when trained on just 10% of the data. Notably, it achieved the highest accuracy across all rounds of 5-fold cross-validation, confirming the significance of the improvement. Conversely, LP-DSSL performed the worst among machine learning methods, excluding Random and MODE, with 10% labeled data, but outperformed LP, LS, and SNN when all data was labeled. This suggests LP-DSSL struggles when labeled data is very sparse. Overall, these results indicate that SURCONFORT is highly effective for predicting train congestion, even with limited labeled data.

Table 2: Performance comparison for ablation study. SSL stands for semi-supervised learning.

			Ratio of labeled data (%)				
Model	SSL	railroad graph	10%	25%	50%	75%	100%
SURCONFORT	✓	✓	**56.76**	**58.08**	**59.41**	**60.52**	**60.35**
NGM [1]	✓	-	55.96	57.69	58.67	59.49	60.09
SNN	-	-	54.94	56.15	56.23	57.42	58.75

4.2.2 Ablation Study. SURCONFORT is defined by (1) the use of SSL and (2) the inclusion of the railroad graph, as explained in Section 3. To evaluate the impact of these components, we measured forecasting performance using two SURCONFORT variants: (1) without the railroad graph, which aligns with the original NGM [1] trained on a natural graph, and (2) without both SSL and the railroad graph, which corresponds to SNN.

As shown in Table 2, the adoption of SSL and the railroad graph significantly enhanced performance. SURCONFORT outperformed SNN by up to 3.3% (without both SSL and the railroad graph) and NGM by up to 1.7% (without the railroad graph). These findings demonstrate the effectiveness of combining SSL and the railroad graph in the proposed method.

5 DISCUSSION

As highlighted in Section 4.2.1, LP-DSSL, which uses label propagation and model retraining based on SNN, underperformed compared to SURCONFORT, especially with sparse label data. This is due to the learned descriptor space having a poor intra-class distribution, which hinders effective pseudo-labeling.

In the findings of the ablation study presented in Table 2 of Section 4.2.2, we saw a progressive improvement in performance from SNN to NGM, and then SURCONFORT. These results highlight the benefits of both semi-supervised learning and graph regularization, particularly by incorporating station adjacency knowledge.

Although the performance gap between NGM and SURCONFORT is small, it can be explained. NGM, regularized by a natural graph using L2 similarities of input vectors (with station one-hot encodings as edges), is partially regularized by station proximity. In this sense, NGM is a variant of SURCONFORT, which explains its comparable performance.

6 CONCLUSION

We introduced SURCONFORT, a novel approach for forecasting rail congestion by utilizing passenger-submitted congestion reports. To address the challenge of sparse labels, we designed a railway network-oriented graph and applied it to semi-supervised regularization, leveraging data from nearby stations. Experimental results with real-world data demonstrated the effectiveness of SURCONFORT, showing a 14.9% improvement in forecasting performance compared to state-of-the-art graph-based semi-supervised methods under label sparsity. Future work will address the subjectivity in passenger reports by integrating data selection techniques for crowdsourcing. Additionally, factors like platform distance and alternative transport availability will be considered to enhance the robustness of the railroad graph, incorporating this information into the proximity metric.

Acknowledgement

We thank Mikiya Maruyama, Ryota Kitamura, and Rikako Takada for their thoughtful discussions and cooperation in acquiring data.

REFERENCES

[1] T. D. Bui et al. Neural graph learning: Training neural networks using graphs. In *Proc. of WSDM*, 2018.
[2] D. Dai and L. Van Gool. Ensemble projection for semi-supervised image classification. In *Proc. of ICCV*, 2013.
[3] A. Iscen et al. Label propagation for deep semi-supervised learning. In *Proc. of CVPR*, 2019.
[4] C. Jia et al. Analysis of crowded propagation on the metro network. *Sustainability*, 14(16), 2022.
[5] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
[6] N. Lathia and L. Capra. Tube star: Crowd-sourced experiences on public transport. In *Proc. of MobiQuitous*, 2014.
[7] Y. Ouali et al. An overview of deep semi-supervised learning. *arXiv preprint arXiv:2006.05278*, 2020.
[8] Y. Sugiyama et al. An approach for real-time estimation of railway passenger flow. *Quarterly Report of RTRI*, 2010.
[9] A. Tomar et al. Crowd analysis in video surveillance: A review. In *Proc. of DASA*, 2022.
[10] D. Zhou et al. Learning with local and global consistency. *Advances in neural information processing systems*, 16, 2003.
[11] X. Zhu et al. Semi-supervised learning using gaussian fields and harmonic functions. In *Proc. of ICML*, 2003.