Received: 7 September 2023 | Accepted: 30 March 2024

DOI: 10.1111/mice.13207

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

RESEARCH ARTICLE

Traffic prediction via clustering and deep transfer learning with limited data

Xiexin Zou | **Edward Chung**

Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hong Kong, China

Correspondence
Edward Chung, Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hong Kong, China.
Email: edward.cs.chung@polyu.edu.hk

Funding information
Research Impact Fund; Research Grants Council of the Hong Kong Special Administrative Region, China, Grant/Award Number: R5029-18

Abstract

This paper proposes a method based on the clustering algorithm, deep learning, and transfer learning (TL) for short-term traffic prediction with limited data. To address the challenges posed by limited data and the complex and diverse traffic patterns observed in traffic networks, we propose a profile model based on few-shot learning to extract each detector's unique profiles. These profiles are then used to cluster detectors with similar patterns into distinct clusters, facilitating effective learning with limited data. A Convolutional Neural Network - Long Short-Term Memory (CNN-LSTM)-based predictive model is proposed to learn and predict traffic volumes for each detector within a cluster. The proposed method demonstrates resilience to detector failures and has been validated using the Performance Measurement System dataset. In scenarios with less than 2 months of training data and 10% failed detectors, the mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE) for station-level traffic volume prediction increase from 12.7 vehs/5 min, 20.9 vehs/5 min, and 10.5% to 13.9 vehs/5 min, 24.2 vehs/5 min, and 11.7%, respectively. For lane-level traffic volume prediction, the average MAE, RMSE, and MAPE increase from 4.7 vehs/5 min, 7.7 vehs/5 min, and 15% to 5.6 vehs/5 min, 9.6 vehs/5 min, and 16.8%. Furthermore, the proposed method extends its applicability to traffic speed and occupancy prediction tasks. TL is integrated to improve speed/occupancy prediction accuracy by leveraging knowledge obtained from traffic volume, considering the correlation between traffic flow, speed, and occupancy. When less than 1 month of traffic speed/occupancy data is available for learning, the proposed method achieves an MAE, RMSE, and MAPE of 0.7 km/h, 1.3 km/h, and 1.3% for station-level traffic speed prediction and 0.5%, 1.1%, and 11% for station-level traffic occupancy.

1 | INTRODUCTION

Short-term traffic volume prediction is an important part of intelligent transportation applications and the key to supporting traffic management. Accurate prediction is essential for freeway management, highway management, ramp control, and variable speed limit systems. It enables traffic flow optimization, enhances road efficiency, reduces

congestion, and provides precise traffic information for informed traffic management decision-making. A large amount of available traffic data and the rapid development of deep learning (DL) have inspired researchers to apply neural networks in the field of transportation engineering (Adeli, 2001; Haghighat et al., 2020; Nguyen et al., 2018) and develop many models to infer traffic states from historical observations (Haghighat et al., 2020; Nguyen et al., 2018).

Various DL models, including Multilayer Perceptron (MLP) (Innamaa, 2000), autoencoders (Sen Zhang et al., 2019), Convolution Neural Network (CNN) (Khajeh Hosseini & Talebpour, 2019; Q. Liu et al., 2018; Ma et al., 2017), and Recurrent Neural Network (RNN) (Abduljabbar et al., 2021; Cui et al., 2020; Gu et al., 2019) and their variants, as well as ensemble models (Cao et al., 2020; Lu et al., 2020; Rajalakshmi & Ganesh Vaidyanathan, 2022) have been used to capture traffic trends for traffic volume prediction, travel time estimation (Dharia & Adeli, 2003), and incident detection (Samant & Adeli, 2000). Early DL models for traffic prediction primarily focused on forecasting the time series of individual detectors, training separate models for each detector. This approach offers the advantage of personalized modeling based on detector-specific characteristics. However, it requires substantial hyperparameter tuning and necessitates a large amount of training data and computational resources (Duan et al., 2016). Currently, LSTM is the most popular and effective DL model for capturing temporal variations in traffic data (Abduljabbar et al., 2021).

The reliability of DL models relies on the availability of a sufficient number of trainable data samples. One approach to address situations where training data are limited is to employ clustering methods to group the investigated objects into multiple clusters (Doğan, 2021). Each cluster corresponds to a unique DL model. This method also reduces the number of models that need to be trained, thereby saving resources. In traffic prediction research, clustering techniques can be employed based on spatial relationships between detectors, functional similarity, or analysis of traffic flow similarity. Han et al. (2019) proposed DeepCluster, which utilizes k-means clustering to group all detectors and develop a single detector prediction model for each cluster. Song et al. (2018) divided historical daily traffic flow into multiple groups and trained prediction models using data from individual detectors. During prediction, the model corresponding to the cluster selected based on context matching is used for prediction. Ghosh-Dastidar and Adeli (2003) applied wavelet-based signal processing, statistical clustering analysis, and neural network pattern recognition for highway event detection based on individual detectors' speed, flow, and occupancy. Jiang and Adeli (2004) estimated the work zone capacity of a highway work zone with limited training data by utilizing a subtraction clustering method combined with a radial basis function.

The methods above overlook spatial relationships and the issue of detector reliability. In cases where detectors fail, algorithms often resort to historical empirical data, such as Average Annual Daily Traffic. Some studies have proposed methods to address detector reliability, such as using data from reliable neighboring detectors for missing data imputation in the Performance Measurement System (PeMS) dataset (Chen, 2002). In the work by X. Yu et al. (2023), sensor deployment and sensor fault assessment are integrated to tackle sensor localization. These approaches are not suitable for real-time imputation and do not provide predictions.

It has been demonstrated that incorporating spatial features in traffic data can significantly enhance the performance of traffic prediction models (Do et al., 2019; Shen et al., 2021). Some studies have developed multi-detector prediction models for specific roads or small areas. Abduljabbar et al. (2021) employed LSTM to predict traffic volume for detectors on a single road. Cui et al. (2020) predicted traffic speed for detectors on four adjacent highways. B. Yao et al. (2017) proposed a support vector machine model incorporating spatial and temporal parameters for short-term single-step traffic speed prediction. By formatting traffic data into regular tensor forms, Shuaichao Zhang et al. (2020) utilized 3D convolutional networks to capture correlations between neighboring detectors on road segments to enhance predictions, and Y. Liu et al. (2021) proposed U-Net for traffic state prediction. For small-scale traffic networks, Tarunesh and Chung (2020) utilized autoencoders to select strongly correlated detectors and employed artificial neural networks for prediction.

These works are applicable to scenarios involving several roads or small areas with strongly correlated detectors or detectors that can be represented in regular formats. However, the irregular installation positions of detectors make it challenging to format them into regular inputs. Moreover, considering the spatial relationships of all detectors would require learning a large number of parameters, which is infeasible given the limited training data and computational resources available. To address the issue of irregular detector arrangements in a traffic network, some researchers have turned to graph neural networks (GNNs) as an alternative to CNN. GNNs extend convolution operations to graphs, enabling the capture of spatial relationships (G. Li et al., 2021; Zhao et al., 2019; Zhou et al., 2022). Various approaches have been explored, such as utilizing geographic distance between detectors (Y. Li et al., 2018; B. Yu et al., 2018; Zhao et al., 2019), correlation coefficients (Lv et al.,

TABLE 1 Related work on traffic prediction of multi detectors.

Work	Method	Advantages	Limitations
Doğan (2021); Han et al. (2019); Jiang and Adeli (2004); Song et al. (2018)	Clustering techniques and single-detector prediction models	More available training data for each deep learning (DL) model, suitable for limited training data scenarios	Overlook spatial relationships and the issue of detector reliability
Abduljabbar et al. (2021); Cui et al. (2020); Do et al. (2019); Y. Liu et al. (2021); Shen et al. (2021); Tarunesh and Chung (2020); B. Yao et al. (2017); Shuaichao Zhang et al. (2020)	Single prediction model inputs and outputs prediction for all investigated detectors	Consider spatial relationships, suitable for small regions with strongly correlated detectors or cases where all detectors can be arranged as regular inputs	Unsuitable for large traffic networks with irregular topologies, require sufficient training data
Fang et al. (2021); G. Li et al. (2021); M. Li and Zhu (2021); Y. Li et al. (2018); Lv et al. (2021); B. Yu et al. (2018); Zhao et al. (2019); Zhou et al. (2022).	Graph neural networks	Can capture spatial relationships in irregular traffic topologies	Constructing traffic graphs is challenging for large traffic networks, with high demands for training data and computational resources
Zou et al. (2024)	Clustering, DL models	Suitable for traffic networks with multiple detectors, consider the issue of detector reliability	Unsuitable for scenarios with limited training data
This paper	Clustering based on few-shot learning, DL models, and transfer learning	Suitable for traffic networks with multiple detectors with limited training data, consider the issue of detector reliability	Unsuitable for unseen scenarios

2021), and dynamic time-warping (DTW) distance of observed volume/speed time series (Fang et al., 2021; M. Li & Zhu, 2021) to construct the graph for learning. These studies have achieved promising results for small regions. However, constructing a traffic graph for large traffic networks remains challenging. Furthermore, learning graphs requires substantial computational resources and sufficient data.

The above studies apply to a single detector, intersection, or road in the case of sufficient data. However, they cannot be applied to the whole network with limited training data. Our previous research utilized clustering algorithms and LSTM models for network-level traffic volume prediction. However, it was designed explicitly for scenarios with ample training data (Zou et al., 2024). Training those deep models becomes particularly challenging when data are limited. This study aims to address the prediction of traffic volumes for all detectors using limited training data. The related work on traffic prediction involving multiple detectors is presented in Table 1.

Transfer learning (TL) is applying the knowledge or pattern learned in a task or dataset (source) to other datasets or similar tasks (target), considered a general method to resolve the insufficient amount of data effectively. Research has applied TL to the transportation domain (Tan et al., 2018; Zhuang et al., 2021). The TL method used in most studies is model or parameter transfer (Mallick et al., 2021; Manibardo et al., 2020; B. Wang et al., 2018; J. Wang et al., 2016). J. Wang et al. (2016) divided all the road segments into groups based on their speed time series correlation. Data from all segments are used to train an initial model, which is then transferred and fine-tuned for each road segments' speed prediction. Mallick et al. (2021) cut the target and source city into subgraphs with the same number of nodes separately. Models trained on source subgraphs are transferred to the graph of the target city. Some studies are on feature transferring. For flow prediction of grids, L. Wang et al. (2019) proposed to find a data-sufficient city (source) with a similar traffic pattern to a data-deficient city (target) based on the time series correlation between different cities. The features learned in the source city are transferred to the target city using a region-matching function to assist prediction. H. Yao et al. (2019) learned a global common fixed feature from multiple source cities and transferred it to the target city. In addition, there is research to transfer other external data to assist. Ren et al. (2019) introduced data from other locations to assist in learning the passenger flow prediction model for the target location. Another type is to transfer knowledge after projecting the source and target data into a new feature space. He et al. (2020) project the features of source and target origin–destination (OD) pairs into a

latent space where the distribution of OD features is similar for all cities. Then, an adaptive function is applied to transfer the model trained on new source features to the target city. J. Li et al. (2020) proposed to train a model in data from source and target cities. An adaptive loss layer is introduced to filter inconsistent source data while learning relevant information. Hence, the target data can benefit from the parameters and features learned from the source data.

This paper presents a method that combines clustering and DL to accurately predict the traffic volume of all investigated detectors across the entire network when limited training data are available. In practical scenarios, different detectors collect different types of traffic data, resulting in varying sample availability for different traffic variables. To address this issue, we propose a TL-based approach that leverages the learned knowledge gained from traffic volumes to enhance the prediction of other traffic variables, such as speed and occupancy, with fewer training samples. Experiments and validations are conducted using the PeMS dataset to assess the proposed method's effectiveness and ability to handle detector failures.

Specifically, we develop a deep profile model for data-deficient scenarios to obtain each detector's profile (deep features) from its daily data with missing values or outliers. Detectors are then grouped into multiple clusters according to the similarity of their profiles. A multi-detector prediction model is trained for a cluster. Clustering reduces the complexity required by the prediction model, thereby reducing the number of parameters that need to be learned and resolving the mismatch between data volume and model complexity. Furthermore, the proposed approach considers the detector's reliability. Since the traffic trends of detectors with similar profiles have certain similarities or correlations, each detector can obtain cluster information from homogeneous detectors to help prediction. The proposed approach can provide predictions for failed detectors that do not report data since references can be obtained from detectors in the same cluster.

The rest of this paper is organized as follows. Section 2 presents the proposed method and model, and Section 3 discusses the experimental results. Finally, the paper is concluded in Section 4.

2 | METHODOLOGY

We aim to utilize several models to provide predictions for all detectors in the investigated traffic network based on their historical observations, considering the detectors' reliability. While most traffic sensors count vehicles, not all measure vehicle speed. In situations where there is a minimal amount of learnable data available for traffic speed/occupancy and a traffic volume prediction model has already been learned, our proposed method utilizes TL to enhance the performance of the speed/occupancy prediction models by leveraging the knowledge acquired from the traffic volume predictions.

Our approach consists of two main steps, first clustering the detectors and then building a predictive model for each cluster, given in Sections 2.1 and 2.2, respectively. The proposed TL approach is presented in Section 2.3.

2.1 | Clustering

In many studies, the similarity of detectors is defined based on factors such as their lane position, point of interest information, and geographical distance. Research focusing solely on traffic data has utilized techniques like time series correlation coefficients or DTW distance to group detectors. The grouping results for weekdays and holidays also tend to differ. However, measurement data inevitably contain outliers and missing values. Since the DL model has a strong feature extraction ability, we extract deep features from the raw data using the DL model and utilize these deep features for detector grouping. Each detector should have distinctive deep features that differentiate it from other detectors and can be extracted across all similar types of dates, which we refer to as profiles.

Generally, treating each detector identity (ID) as a category, a multiclassification DL model can be trained for this purpose. Specifically, one daily data of a detector is input, and a one-hot encoding is an output indicating which category (detector) the daily data is associated with (Zou et al., 2024). Suppose the profile model can distinguish any input data's detector ID, then it can extract the representative signatures of detectors. However, such a model requires sufficient data to be trained. In our scenario, we face the challenge of limited data availability, with only a few tens of samples (daily data) for each of the thousands of detectors. When dealing with limited data and a large number of categories, multiclassification models often encounter severe overfitting issues, resulting in relatively low classification accuracy, around 50%. However, when ample data are available, the accuracy can reach as high as 90%.

To address these challenges, we propose our profile model, drawing inspiration from prototypical networks (Snell et al., 2017), which have been successfully applied to few-shot multiclassification tasks in the image domain. The proposed profile model extracts deep features from each daily data. It is expected that a detector's query feature will closely resemble its corresponding detector's profile as illustrated in Figure 1. A detailed explanation of the query feature and profile is provided below.

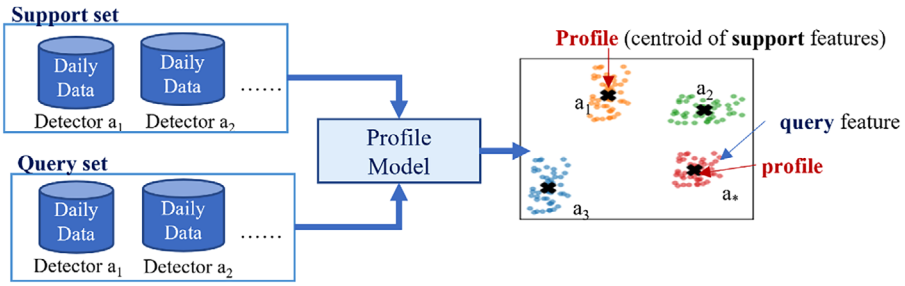

FIGURE 1 Explanation of the profiles and query features.

The training process of the profile model is as follows:

1. Get the daily volume (a 288-dimensional vector). Each element represents a 5-min interval, 288 = 24 h × 12) of each working day of each detector to form a dataset. Divide the dataset into disjoint training and testing sets.
2. Design a profile model to map each daily traffic data to a new space and become a new feature. The features obtained from the daily data in the support set are referred to as support features, while those extracted from the query set are called query features. The centroid of all support features is the profile of the detector.
3. N_c detectors are randomly selected in each training epoch to learn the N_c classification task. Details are as follows:
   - For each selected detector, randomly select $Num_{support}$ days of data from its training set as the support samples and the remaining Num_{query} days as the query samples.
   - Use the profile model to obtain each detector's profile and query features.
   - The profile model is trained using the support data first. Each detector's profile is the centroid of its corresponding support features.
   - Next, train the profile model with query data and calculate the loss of each query sample. The sum of the losses of all query samples is used for backpropagation to update the parameters in the profile model. Figure 2 shows the loss calculation for a query sample assuming three detectors (classification). Specifically, the Euclidean distance between the query feature and all profiles is calculated to form a N_c-d vector, where each element represents the distance between that query feature and a particular detector's profile. Next, the Softmax function is applied to convert this distance vector into a multi-classification probability. The loss is the negative of the probability that the query sample belongs to its correct detector. Suppose that among the profiles of all detectors, the query feature is closest to the profile of its related detector. In that case, the model can judge that it belongs to the correct class, and the resulting loss is also small. The loss used for backpropagation to update the parameters is the sum of losses of overall query samples.
4. Every time the parameters are updated 100 times through backpropagation, the classification accuracy of the test samples is calculated; that is, to use the current model parameters to get the test features. Then, calculate the distance between each test feature and all stored profiles and further calculate the classification accuracy.
5. After training for 100 epochs, N_c detectors are randomly selected again.
6. Repeat Steps 3–5 until satisfactory test accuracy is obtained.

The proposed profile model is shown in Figure 3. The input data are the volume data measured from the detector on a specific day, a $(1, 1, 288)$ tensor. We first fuse information from adjacent time slices using four stacked convolutional layers with $(1, 3)$ kernels. The padding of the feature map is set to $(0, 1)$ to ensure that the size remains unchanged after each convolution. After four convolutional layers, a $(1, 1, 288)$ tensor is obtained. Each element integrates information from adjacent time slices and is a local feature with short-term traffic characteristics. After that, two stacked fully connected (FC) layers are applied to obtain global features. The concatenation of global and local features is sent to a convolutional layer to get a $(1, 1, 288)$ tensor. Finally, the tensor is reshaped into a $288 - d$ feature, the representative information extracted from the input daily volume data.

This method achieves 95% accuracy in identifying detectors on the query set and 87.22% on never-before-seen samples (test set). This method can extract representative profiles in the case of a limited sample size.

As the number of training epochs increases, there is a high possibility that any two detectors may appear in the same training epoch, despite only a small number of detectors being randomly selected for multi-classification

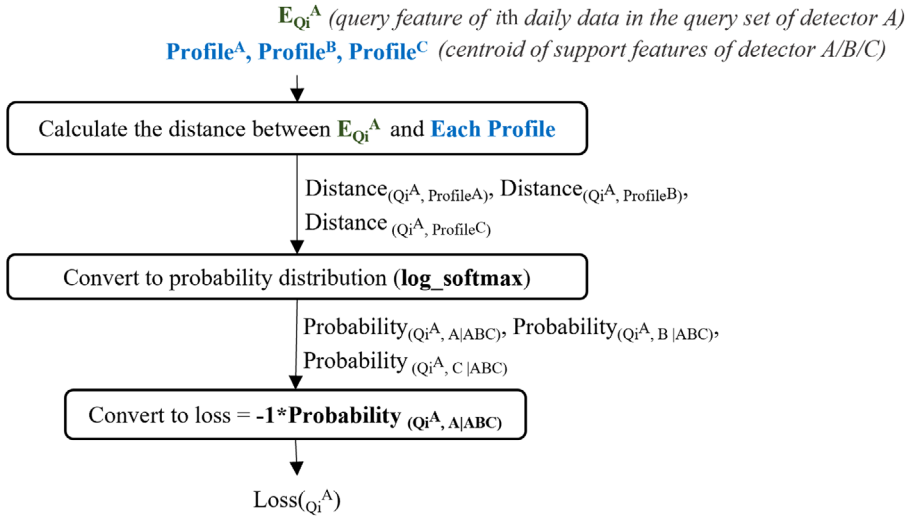

FIGURE 2 Illustration of loss calculation.

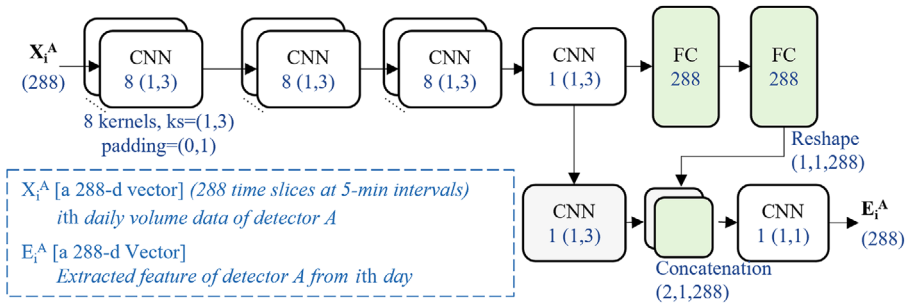

FIGURE 3 Profile model. FC, fully connected.

in each iteration. The learned profile model ensures that the query features of each detector are close to its corresponding support features.

After training, the obtained profiles are stored and used for subsequent classification. During our model training, we calculate the loss based on the Euclidean distance of each query feature from all profiles. For consistency, the Euclidean distance of profiles is calculated to represent the similarity of corresponding detectors. The smaller the distance of the profiles, the higher the similarity of the detectors. The hierarchical clustering algorithm is applied for classification based on that similarity.

The hierarchical clustering algorithm treats all detectors as independent clusters at the beginning. It then calculates the distance, denoted as $d(C_I, C_J)$, between any two clusters C_I and C_J, using the following formula:

$$d(C_I, C_J) = \sum_{a \in C_I \cup C_J} \left\| profile_a - \mu_{C_I \cup U_J} \right\|$$

where $profile_a$ represents the profiles of each detector belonging to either Cluster C_I or C_J, and μ represents the mean of all profiles in the two clusters. Subsequently, the algorithm selects the two clusters with the smallest distance and combines them in each subsequent step until the desired number of clusters is achieved.

2.2 | Predictive model

After obtaining the clustering results, the proposed method involves training a predictive model assigned to each cluster. All predictive models share the same architecture but are individually trained and fine-tuned on their respective training sets.

As illustrated in Figure 4, the proposed predictive model consists of an LSTM layer, three Refer Modules, and several CNN layers. The Refer Module, a novel component introduced in the proposed predictive model, operates alongside the LSTM module. The LSTM layer generates features for prediction by utilizing the time series features, disregarding the detector ID. In contrast, the output feature of the Refer Module combines the local features with the detector ID and the global cluster features.

To predict the traffic volume for each detector in the cluster for the next time interval (5 min), the model utilizes

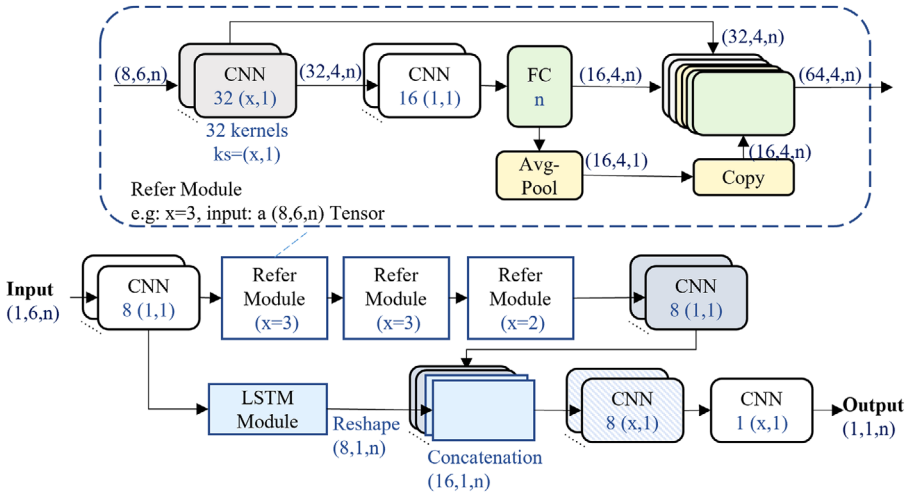

FIGURE 4 Predictive model. Avg-Pool, average pooling; FC, fully connected.

the observations of six historical time intervals (30 min). The input data are represented as a (1, 6, n) tensor, where six denotes the most recent six-time intervals observed, and n represents the number of detectors in the cluster. The output is a (1, 1, n) tensor, predicting each detector's volume for the next time slice.

First, a convolutional layer is applied to create an (8, 6, n) tensor from the input, providing each detector with an 8-d feature at each time slice. Since similar patterns are assumed among detectors in the cluster, it is believed that the features can be shared among them. Since LSTM is well-suited for extracting features from time series data, the (8, 6, n) tensor is reshaped into a (6, 8n) tensor, resulting in 8n-d features per time slice, excluding the detector IDs. The LSTM layer takes this tensor as input and generates 8n-d new features for the subsequent time slice, disregarding the detector information.

The Refer Module, depicted in Figure 4, enhances the model's performance and consists of several components. For instance, the second Refer Module takes a (64, 4, n) tensor as input, representing 64 features per time slice per detector. Using a CNN layer with (3, 1) convolution kernels, this module fuses each detector's adjacent time slice features. This results in a (32, 2, n) tensor, capturing local temporal features while mitigating the impact of outlier values within time slices. Next, a convolution layer reduces the output channel to 16, yielding a (16, 2, n) tensor, equipping each detector with 16 features per time slice. To obtain the global cluster information for each feature on each time slice, a FC layer and an average pooling (Avg-Pool) operation are employed. The FC layer learns global cluster information, while Avg-Pool calculates the mean value of all elements on each channel without learning. The resulting (16, 2, 1) tensor from Avg-Pool is duplicated n times to match the number of detectors, forming a (16, 2, n) tensor that retains global cluster information for each feature across all detectors. Subsequently, all global and local features are concatenated, creating a (64, 2, n) tensor.

To achieve a desired time dimension of 1 for the output feature and incorporate information from all historical time slices of the input, we stack three Refer Modules. The first two modules combine information from the adjacent three-time slices to generate a (64, 2, n) tensor. The third module combines information from two-time slices to produce the desired output (64, 1, n) tensor. After the three stacked Refer Modules, the spatiotemporal features of the six-time slices are fused.

Finally, all the obtained features are combined and passed through two convolutional layers with (1, 1) kernel to generate the output (1, 1, n) tensor.

2.3 | TL for predicting speed and occupancy

Speed and occupancy exhibit correlations with traffic flow according to traffic flow theory (Ke et al., 2016). By changing the input of the prediction model from volume to speed/occupancy and retraining it, the proposed approach can be used to predict traffic speed/occupancy.

The trend in traffic volume provides valuable insights into the corresponding changes in average vehicle speed and occupancy rate. For instance, it exhibits periodic patterns such as consistent peak and off-peak periods on weekdays. During peak periods, recurrent congestion is prevalent, characterized by high traffic volume and lower average vehicle speed. The occupancy rate is typically high

during these periods. When two detectors display similar traffic volume variation trends, their corresponding speed or occupancy variation trends will likely be similar. Hence, the learned global cluster information from the traffic volume prediction model, capturing the correlations between homogeneous detectors, is also helpful in generating accurate speed and occupancy forecasts. This highlights the potential of transferring learned knowledge from traffic volume to enhance speed and occupancy predictions.

TL is a powerful technique in tasks with limited data availability. Thus, when a learned traffic volume prediction model and limited learnable speed and occupancy data are available, we leverage the knowledge acquired from traffic volume via TL to enhance the prediction capabilities of traffic speed/occupancy.

In the volume model, the features learned and utilized for output traffic volume predictions are extracted from historical traffic volume data. These features contain information about the cluster and the temporal dynamics of traffic volume within the cluster, which can provide valuable insights into changes in speed and occupancy. For instance, if the learned features indicate an excessive traffic volume in the subsequent time slice, it implies a potential decrease in speed and an increase in occupancy. Hence, the extracted volume features are transferred and utilized to enhance the accuracy of speed and occupancy predictions.

Additionally, the initial weights of the DL model significantly impact the training process. By transferring well-trained model parameters, the target model can bypass the need to start from scratch, leading to an accelerated optimization process. As a result, the parameters obtained from the trained volume model are transferred to initialize the speed/occupancy model.

As shown in Figure 5, the speed/occupancy predictive model is similar to the volume predictive model. However, a convolutional layer is added at the end to fuse the transferred volume features. The data used for training the prediction model have been changed from traffic volume to traffic speed/occupancy data. For speed and occupancy, the clustering results obtained from the volume data are directly applied. First, copy the parameters of the trained traffic predictive model ("Transferred Module of Volume Model" in Figure 5) to initialize the new model better. Then, the learned volume features on the corresponding time slice of the corresponding detector are introduced to the new model. Specifically, the volume feature is the output of the penultimate layer of the volume predictive model, an $(8, 1, n)$ tensor, that is, an 8−d feature for each detector. Finally, retrain to get new model parameters based on the initialization. The transfer methods used in this paper are model transfer and feature transfer.

The knowledge learned from volume data is used to improve the prediction performance of speed/occupancy.

3 | EXPERIMENTS

3.1 | Dataset

Traffic data from detectors in the San Francisco Bay Area are used to verify the effectiveness of our method. These data are from the Caltrans PeMS, which collects traffic volume, speed, and occupancy from the lane detector every 30 s and aggregates at 5-min intervals.

The effectiveness of the proposed method in predicting station-level volume, speed, occupancy, and lane-level traffic volume is validated using traffic data from all three-lane stations, which corresponded to 527 station detectors and 1581 lane detectors. The results of station-level and lane-level predictions are presented in Sections 3.3 and 3.4, respectively. Section 3.4 also compares the differences between direct station-level predictions and aggregated lane-level predictions to station-level predictions. Additionally, to demonstrate the robustness of the proposed method, results of predictions with 1%−10% faulty detectors are provided.

A limited amount of traffic volume data covering 8 weeks, from January 1, 2022, to February 26, 2022, were selected for training and validation. The data were split into 80% for training and 20% for validation. Additionally, Section 3.3.3 assesses the effectiveness of the proposed TL approach specifically for the station-level speed/occupancy prediction model. For this evaluation, we utilize a subset of the data consisting of 4 weeks, from January 29, 2022, to February 26, 2022, for training the model, and the remaining data are used for testing from February 27, 2022, to March 12, 2022.

The raw volume/speed/occupancy data are scaled to the [0,1] range according to their respective maximum and minimum values for all inputs to the deep model. Model outputs are then rescaled to the original scale and compared to actual observations for testing model performance.

3.2 | Experimental setup

The proposed model is implemented based on PyTorch on the University Research Facility in Big Data Analytics (UBDA) platform of the Hong Kong Polytechnic University. The Central Processing Unit (CPU) and Graphic Processing Unit (GPU) information are Intel(R) Xeon(R) Gold 6130 CPU @ 2.10 GHz, ASPEED Technology, Inc. ASPEED Graphics Family (rev 41). The loss functions of predictive models are set to mean squared error. For all

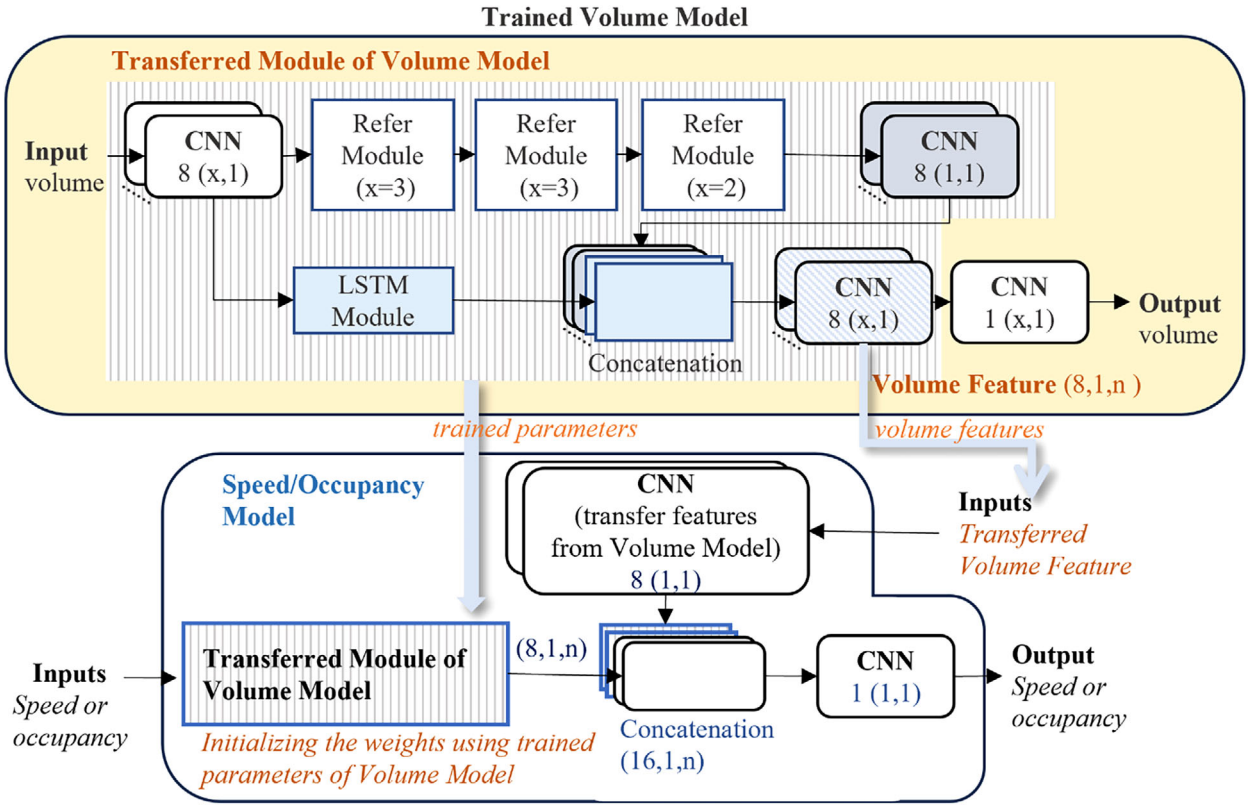

FIGURE 5 Transfer method.

deep models, Adam is used as the optimizer. The dynamic learning rate is applied: If the evaluation indicators of the model do not improve after 50 training epochs, the learning rate is reduced. Furthermore, the early stopping strategy and cross-validation method are also used to prevent overfitting.

The mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE) are applied for metrics in this paper.

3.3 | Results of station-level traffic prediction

The focus of this study is to predict station-level traffic volume for the next 5 min. Results for multi-step predictions up to 1 h are provided in Section 3.3.1. In addition, the model's robustness in the presence of faulty detectors is demonstrated in Section 3.3.2. To validate the effectiveness of TL, Section 3.3.3 examines the impact of incorporating knowledge transfer from traffic volume on speed/occupancy prediction. In such cases, the clustering results obtained from traffic volume are directly utilized, and the prediction model is retrained using traffic speed/occupancy data.

3.3.1 | Multi-step station-level traffic volume prediction

The multi-step prediction results for station-level traffic volume are shown in Figure 6a. Among them, "MC" refers to the proposed method, which first utilizes all the daily data to train a profile extraction model and perform clustering. Then, a prediction model is trained for each cluster. The stations are divided into five clusters in the experiment. Compared to the proposed approach, "Any MC" denotes randomly dividing the stations into five clusters based on their station ID. "SC" refers to training a single predictive model for all station detectors without clustering. "Naïve" represents the case of traffic prediction using recent historical observations as the prediction.

It can be seen from Figure 6a that the prediction performance of the proposed method can be expressed as MAE, RMSE, and MAPE as 12.7 vehs/5 min, 20.9 vehs/5 min, and 10%, respectively. For one-step prediction, the proposed method is only 4 vehs/5 min away from Naïve's MAE result. However, when the time window is 1 h, the difference can reach 13 vehs/5 min. The performance of the SC method is very close to Naïve. SC did not benefit from DL techniques. When checking the

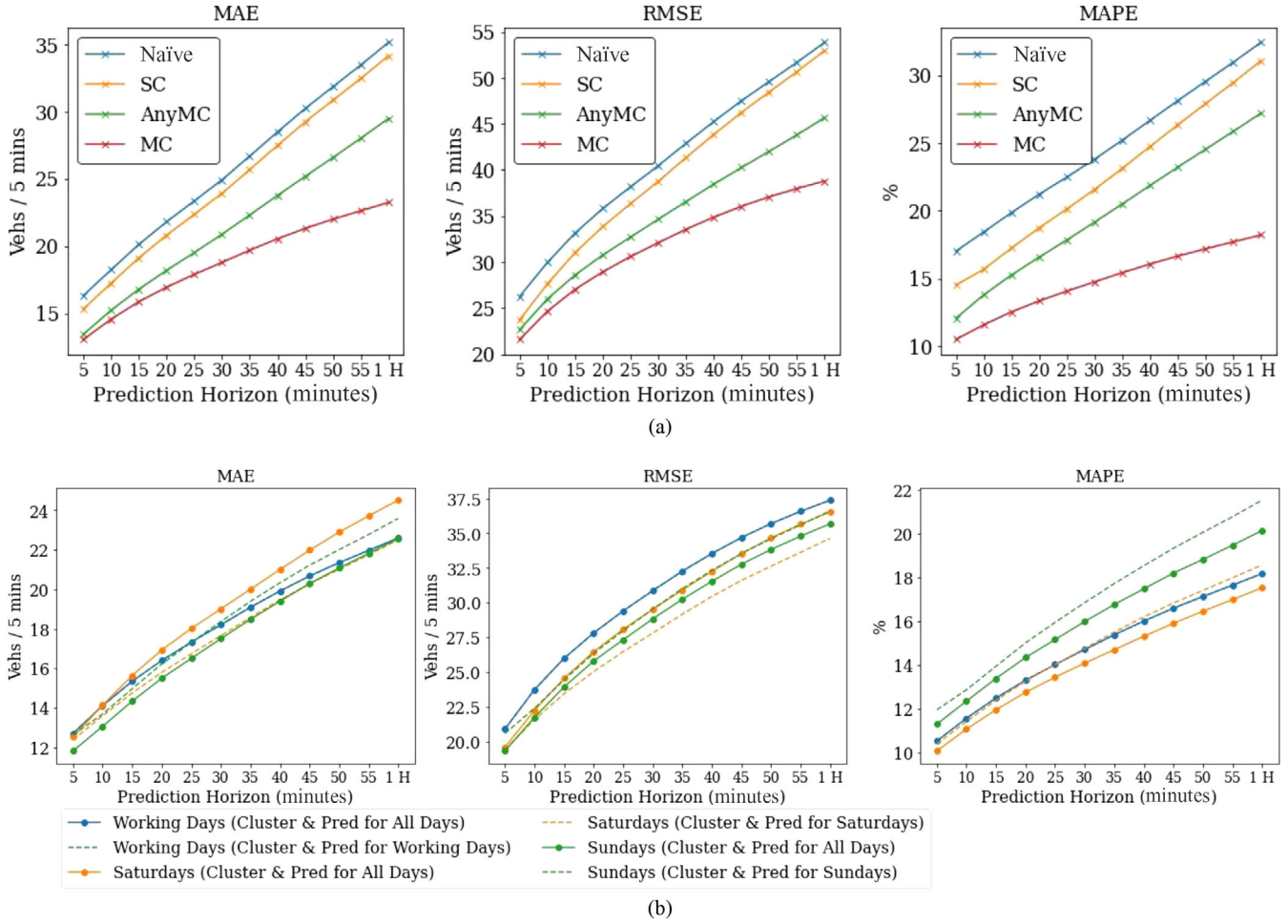

FIGURE 6 Station-level multi-step traffic volume prediction without failures. (a) Station-level multi-step traffic volume prediction and (b) station-level multi-step traffic volume prediction based on different clustering and training. MAPE, mean absolute percentage error; RMSE, root mean square error.

loss comparison during training, it can be found that the SC method performs well on the training set, even better than the MC method, but poorly on the test set. The reason is that many complex heterogeneous features in the entire network require complex models to learn. However, complex models also mean that the number of parameters that need to be learned is enormous. When the amount of data is limited, there can be severe overfitting, causing the model to perform poorly on data it has never seen.

It can be seen that clustering can effectively alleviate the overfitting problem. After clustering, the number of parameters to learn is significantly reduced without reducing the amount of data per cluster. This is also why the AnyMC method outperforms SC. The proposed MC method further outperforms the AnyMC method because the similarity of detectors is considered. The difference between AnyMC and the proposed MC gradually increases as the multi-step prediction is performed. The proposed method ensures a homogeneous pattern within each cluster after grouping; they can benefit from each other instead of being noisy. The proposed method can provide effective volume prediction for stations.

However, it is expected that traffic patterns on weekdays and weekends would exhibit variations, leading to distinct profiles and clustering outcomes. The approach above generates a classification result encompassing all days. To conduct a more comprehensive evaluation, we trained separate profile models for clustering and the prediction model using only working days, Saturdays, or Sundays and make comparisons. In Figure 6b, the same color represents predictions for dates of the same type. The solid line indicates the results obtained using the clustering result based on all days. It can be observed that the difference in prediction performance between the two approaches is minimal. However, we believe that utilizing distinct classification results for different types of dates could potentially yield improved results when a sufficient amount of training data is available. Nevertheless,

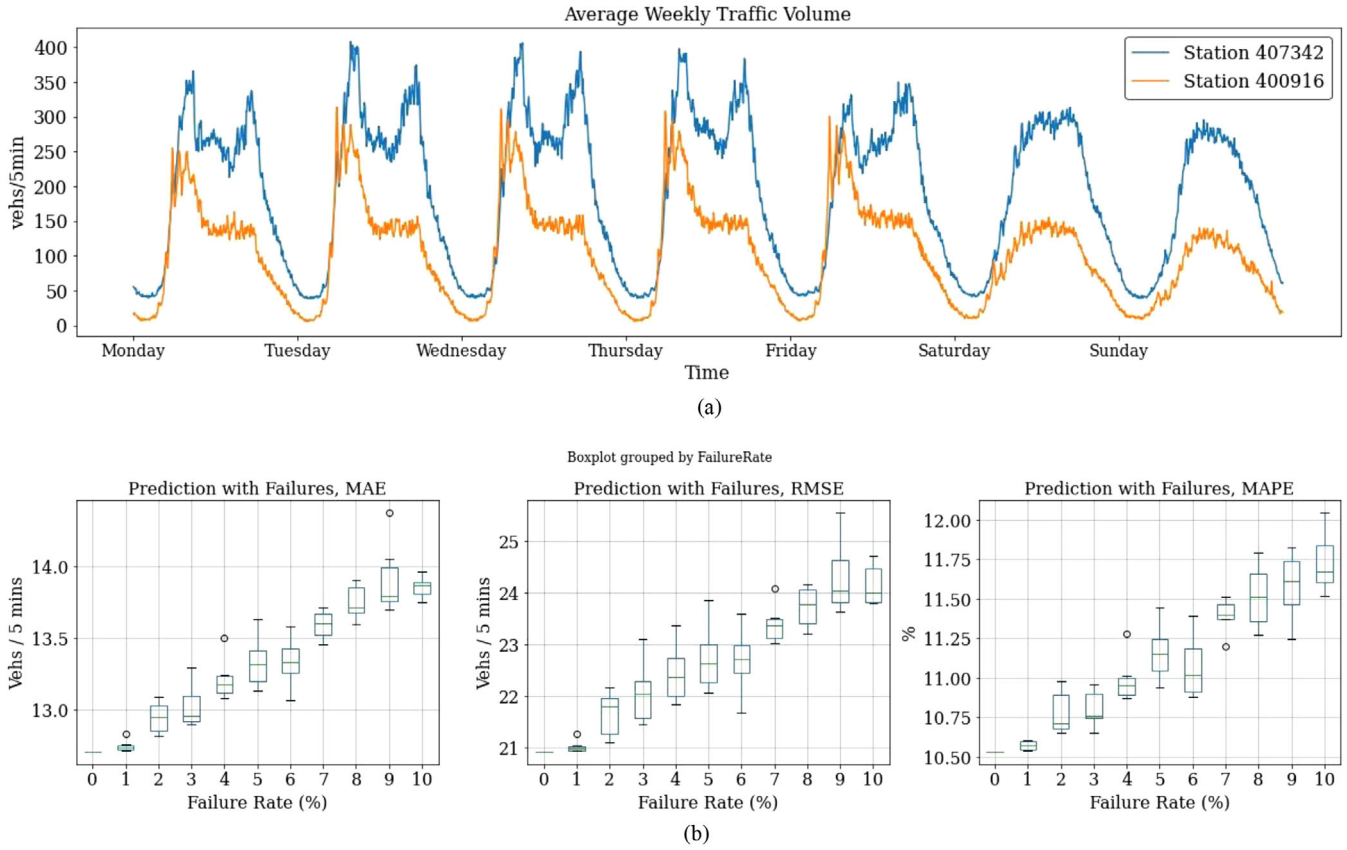

FIGURE 7 Station-level traffic volume prediction with failures. (a) Average weekly traffic volume and (b) station-level volume prediction with failures. MAPE, mean absolute percentage error; RMSE, root mean square error.

implementing such an approach would entail increased maintenance costs. Hence, our classification is based on all days for the subsequent results.

3.3.2 | Resilience of station-level traffic volume prediction against failures

Detector failures are a common issue caused by equipment malfunctions or network problems, resulting in a lack of up-to-date observations of faulty detectors in the input of the prediction model. Real-time data imputation is performed using the average weekly traffic volume obtained from the historical data of failed detectors and the information of the current cluster. The imputed data are then fed into the prediction model to generate subsequent predictions.

The average traffic volume for each 5-min interval from Monday to the weekend is calculated for each detector using historical data as shown in Figure 7a. For a detector i in Cluster C_I, if it fails at time τ (corresponding to Day of Week D_a, Time of Day T_b), the online imputation method is as follows:

$$Missing_i^{\tau} = AWT_i^{D_a,T_b} \times Avg_{j \in C_I}\left(\frac{X_j^{\tau}}{AWT_j^{D_a,T_b}}\right)$$

where X_j represents the real-time measured traffic volume of working detector j in cluster I; 1%–10% of detectors are randomly selected as "failed detectors" and set not to provide any input to test the resilience of the proposed method to failures. When a detector is set to fail, its historical observations are assumed to be unknown. Six random cases are tested and the result is shown in Figure 7b. As the failure rate increases, the performance of the model gradually deteriorates. It should be noted that when all detectors in a cluster fail, the cluster's prediction model cannot provide forecasts based on real-time information.

However, the probability of such a scenario occurring is extremely rare. With faulty detectors, the model exhibits prediction accuracy with corresponding MAE, RMSE, and MAPE values ranging from 12.5 to 14.5 vehs/5 min, 21 to 25.5 vehs/5 min, and 10.5% to 2.2%, respectively.

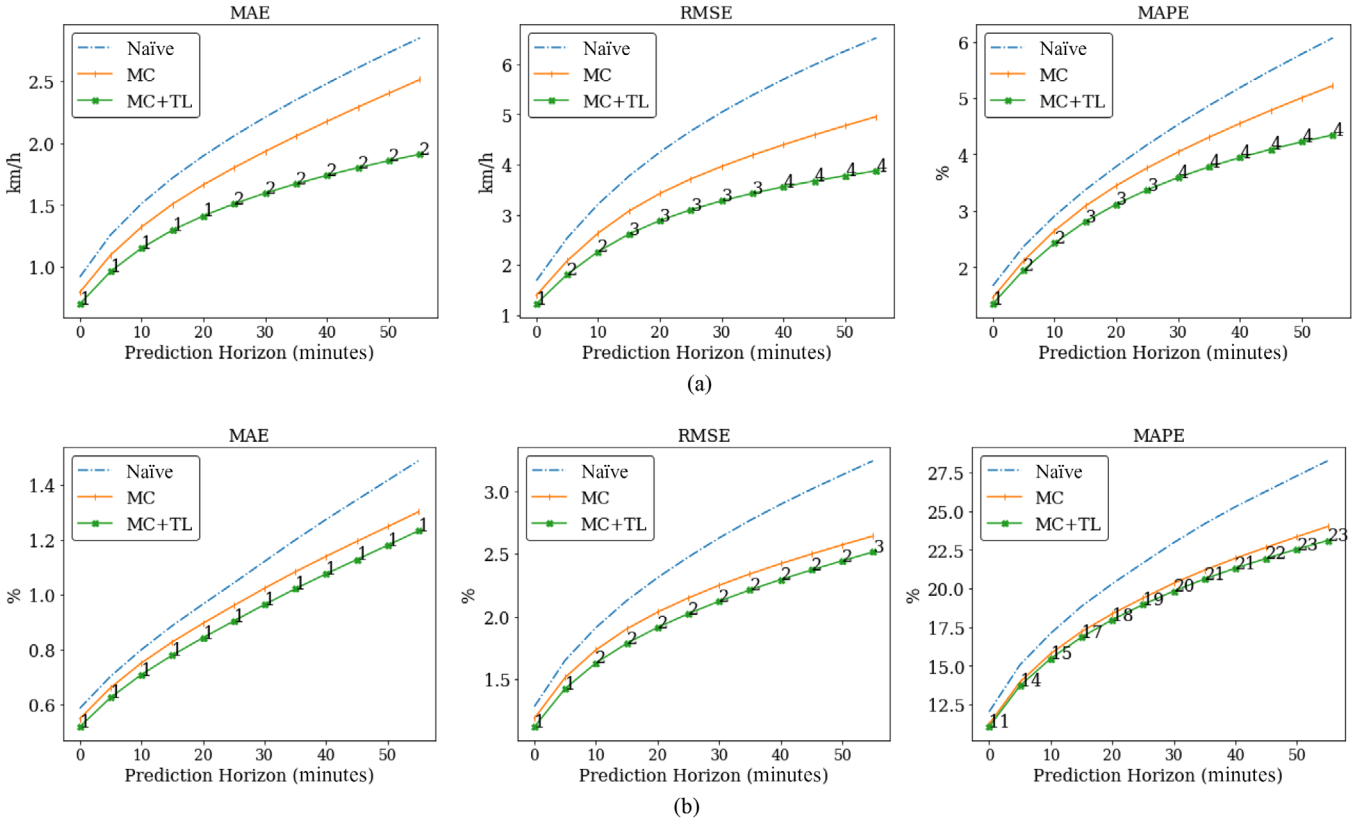

FIGURE 8 Station-level prediction with transfer learning (TL). (a) Station-level speed prediction (TL) and (b) station-level occupancy prediction (TL). MAPE, mean absolute percentage error; RMSE, root mean square error.

3.3.3 | Station-level traffic speed/occupancy prediction with/without TL

After obtaining the traffic volume prediction model, if available training data are scarce for traffic speed/occupancy prediction (only 4 weeks), TL, as mentioned in Section 2.3, can assist in training the traffic speed/occupancy prediction model and improve its performance. This traffic volume prediction model is trained using an 8-week dataset. Subsequently, the acquired knowledge and model parameters are transferred to predict traffic speed/occupancy. The corresponding prediction model is retrained by leveraging the transferred knowledge and the available traffic speed/occupancy data.

Figure 8a,b shows the prediction performance for occupancy and speed with and without transfer training. Compared to the Naïve method, the MC method does not improve performance much due to an obvious limitation: Only 4 weeks of speed/occupancy data are available for training and validation. After introducing the model and volume feature transferring, the prediction performance of speed and occupancy has improved. Although very subtle, it still shows the effectiveness of transfer when both the source and target tasks have limited data. And due to the assistance of the additional volume feature, the result of multi-step prediction with TL is better than without.

3.4 | Results of lane-level traffic volume prediction

The proposed method is extended to predict lane-level traffic volumes, as lane-level forecasting offers more detailed traffic information than station-level forecasting. The prediction performance of lane-level traffic volume is presented in Section 3.4.1, while the robustness of the model with failed detectors is discussed in Section 3.4.2. Additionally, a comparison is provided in Section 3.4.3 between station-level predictions using station-level data and the aggregation of lane-level predictions to obtain station-level results. This comparison allows for an evaluation of the effectiveness of the different approaches in predicting traffic volumes at the station level.

3.4.1 | Multi-step lane-level traffic volume prediction

The multi-step prediction results are shown in Figure 9a. MAE, RMSE, and MAPE results of the proposed method

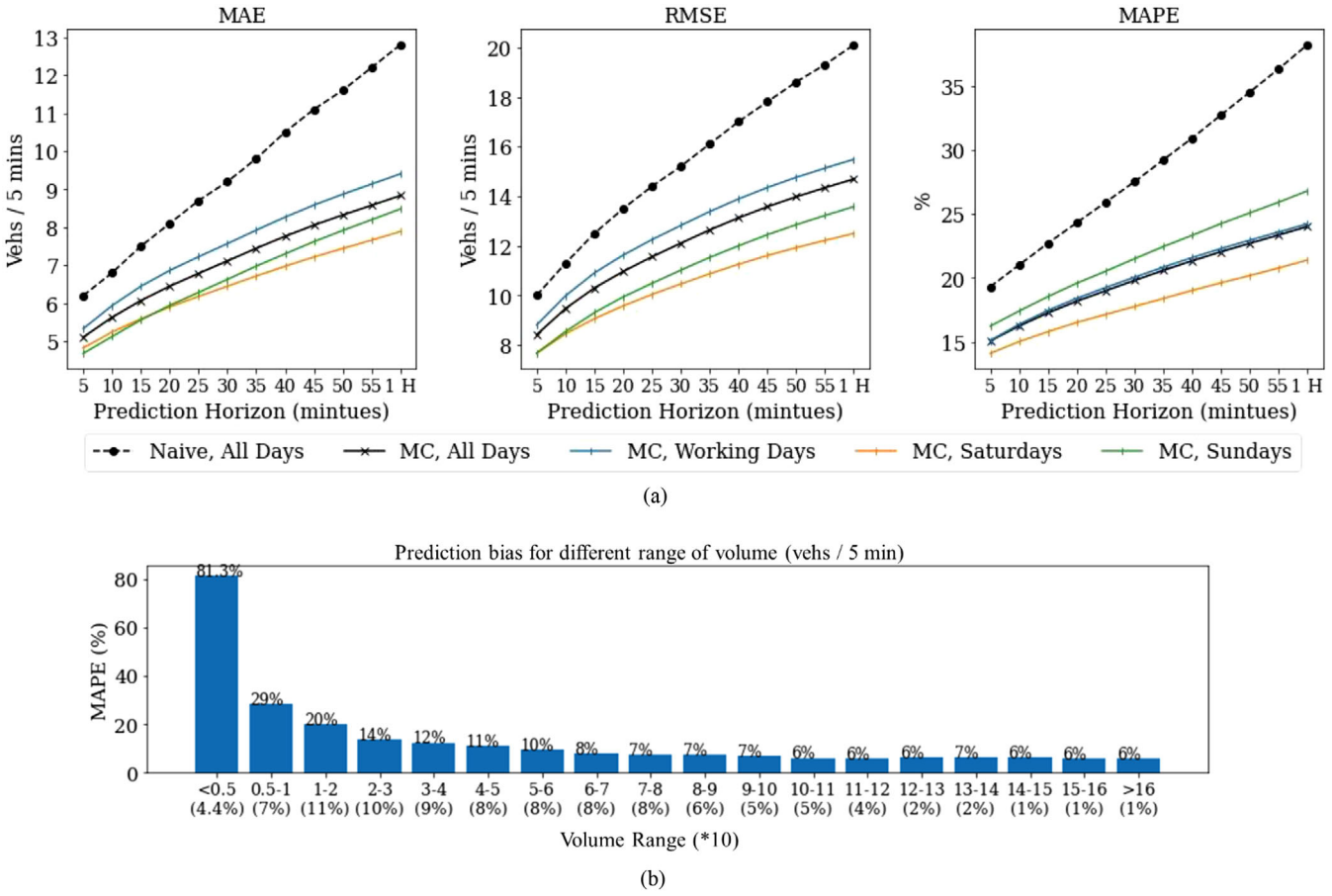

FIGURE 9 Lane-level traffic volume prediction without failures. (a) Lane-level multi-step traffic volume prediction performance and (b) prediction bias for different ranges of volume. MAPE, mean absolute percentage error; RMSE, root mean square error.

are 4.7 vehs/5 min, 7.7 vehs/5 min, and 15% for lane-level prediction, respectively. Figure 9a also presents the prediction performance for different types of days. As the multi-step prediction progresses, the prediction performance for all day types gradually weakens.

To better examine the prediction bias, the deviation of the traffic volume predictions obtained by the proposed method from the measured traffic volume is analyzed for different traffic volume ranges. As shown in Figure 9b, the x-axis represents the size range of observations, for example, "<0.5(5.1%)" is the case when measured traffic volume is less than 5 vehs /5 min, which occurs in 5.1% of the total test set. The y-axis represents the prediction bias, which is:

$$\frac{prediction - observation}{observation}$$

The bias is huge when the actual observed value is less than 5 vehs/5 min. This is because if the observed value equals 1 veh/5 min, a predicted volume of 2 vehs/5 min results in a 100% prediction bias. When the observed flow is 360–480 vehs/h, the prediction bias is around 12%.

When the amount of data is limited, training a separate model for each detector can have good results because fewer parameters need to be learned with the same amount of data. Several detectors are randomly selected, and a CNN-based model is individually trained for each detector for prediction comparison (denoted by "Per Detector" in Figure 10a). The model is based on four stacked convolutional layers; each layer consists of 32 (3, 1), 32 (3, 1), 32 (2, 1), and 1 (1, 1) convolution kernels. As shown in Figure 10a, the proposed method can obtain prediction results comparable to Per Detector while greatly reducing the number of required models. However, Per Detector does not consider the detector's reliability. It fails prediction when the detector fails because there is no input data, while the proposed method does not. Furthermore, the prediction performance of the proposed method suffers less impairment as the time step progresses. The results indicate that the detectors within each cluster exhibit homogeneity, which helps to reduce noise during learning. Furthermore, each detector can leverage the information and constraints learned from its respective cluster.

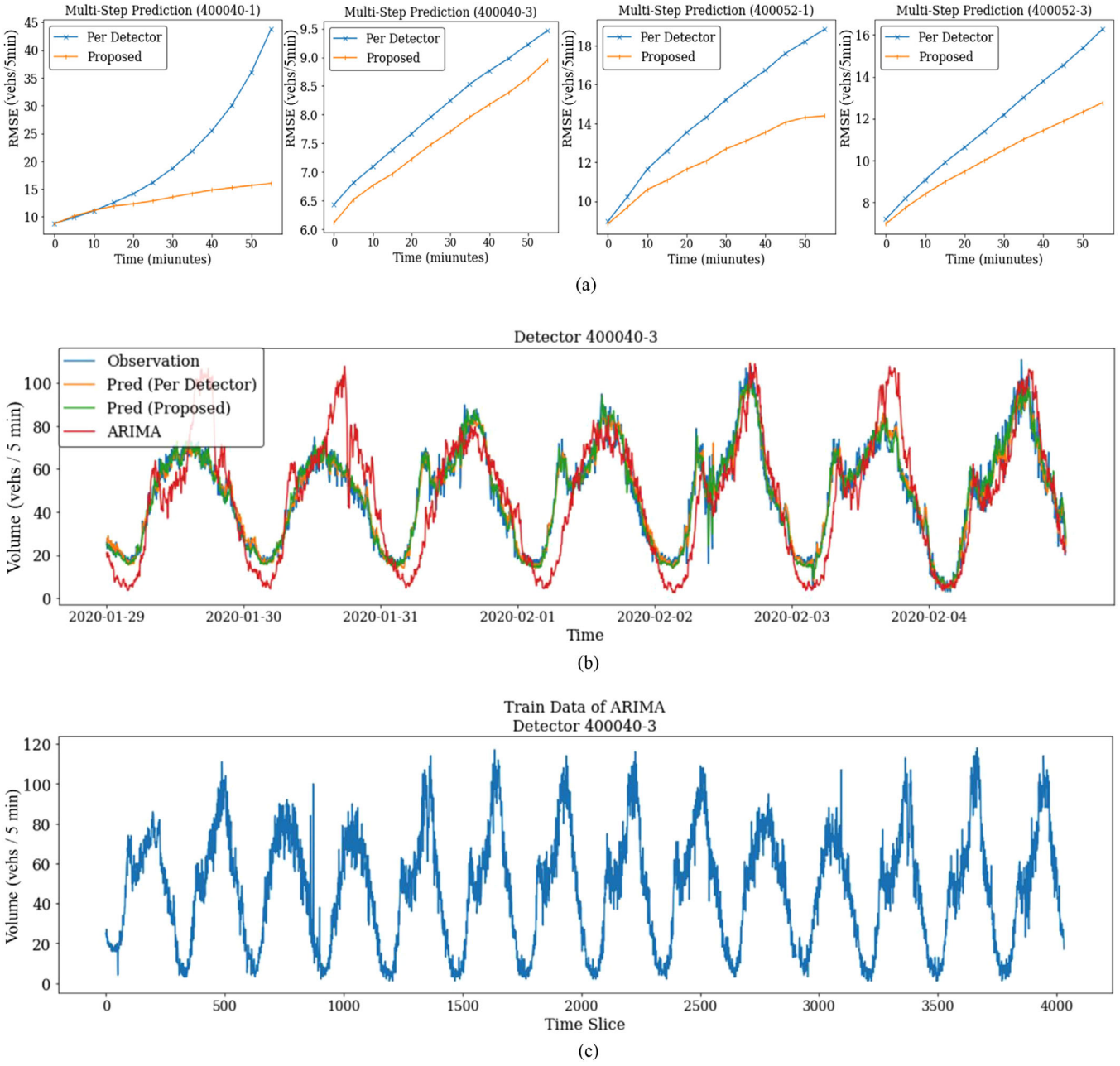

FIGURE 10 Comparison of a few arbitrary detectors. (a) Multi-step prediction results of a few arbitrary detectors, (b) visualization of the prediction of one arbitrary detector, and (c) visualization of training data of the autoregressive integrated moving average model (ARIMA).

In addition, we compare the proposed DL model with the autoregressive integrated moving average model (ARIMA). The (d, p, q) corresponding to each detector are obtained by observing the first-order difference, second-order difference, autocorrelation, and partial correlation diagrams. Specifically, for "400040-1," the RMSE of the DL-based method is about 8 vehs/5 min and of the ARIMA(5,1,0) model is 9.9 vehs/5 min. For "400040-3," the RMSE of the DL-based method is about 6 vehs/5 min, and of the ARIMA(4,1,0) model is 7.4 vehs/5 min. For "400052-1," the RMSE of the DL-based method is about 8.9 vehs/5 min, and of ARIMA(3,1,0) model is 9.1 vehs/5 min. For "400052-3," the RMSE of the DL-based method is about 7.2 vehs/5 min, and of the ARIMA(4,1,0) model is 7.7 vehs/5 min.

Figure 10b visualizes predictions for a randomly selected week. It also shows that DL-based methods can fit real traffic changes well. However, the ARIMA model only performs well on some days. Figure 10c visualizes the corresponding training data of the ARIMA model. Compared with the prediction (Figure 10b), it is found that ARIMA can capture the stability rules that exist in most training

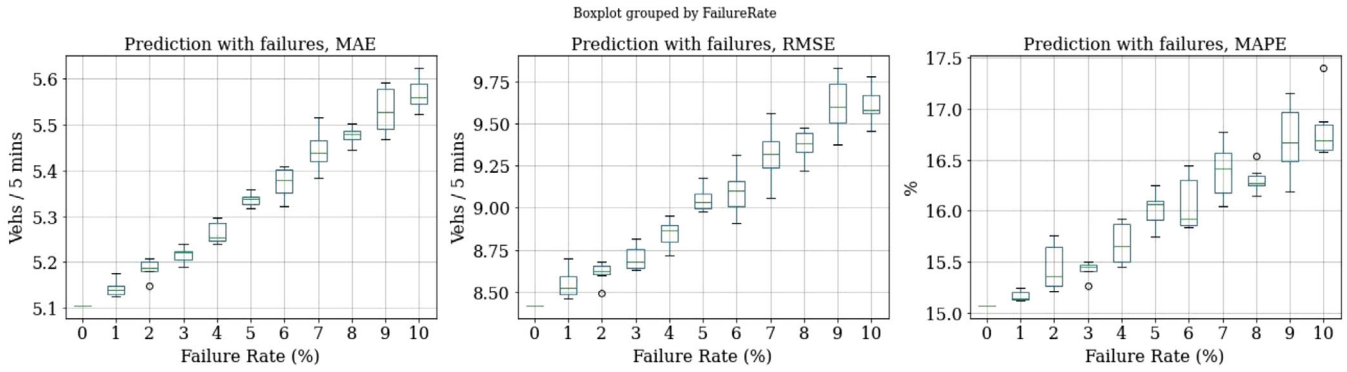

FIGURE 11 Lane-level traffic volume prediction with failures. MAPE, mean absolute percentage error; RMSE, root mean square error.

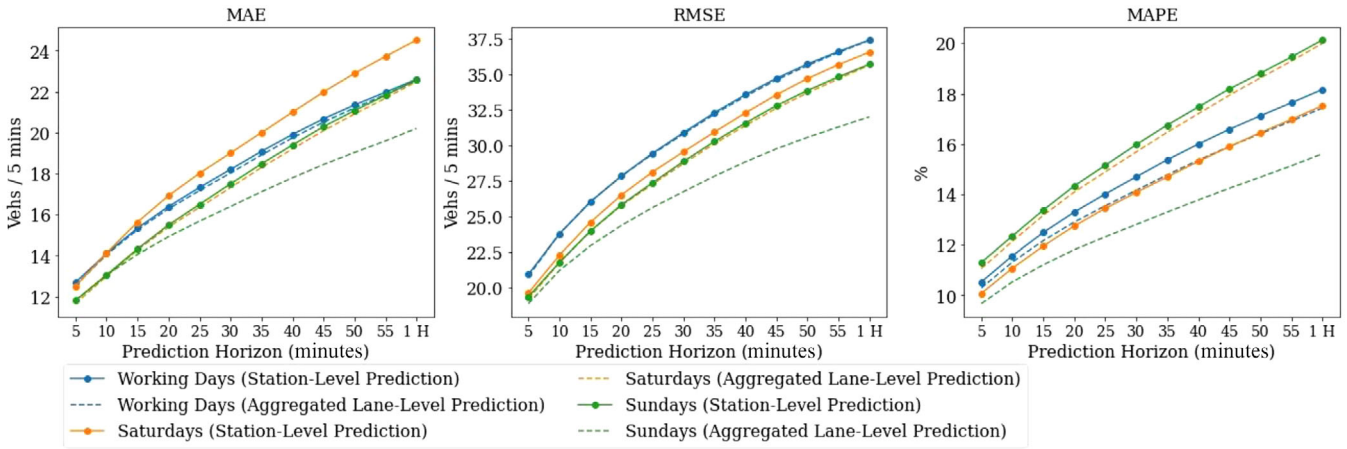

FIGURE 12 Comparison of station-level prediction and aggregated lane-level prediction. MAPE, mean absolute percentage error; RMSE, root mean square error.

data, which is suitable for stationarity data but not suitable for fluctuation.

3.4.2 | Resilience of lane-level traffic volume prediction against failures

Figure 11 showcases the lane-level traffic volume prediction results when 1%–10% of the detectors are faulty. The corresponding MAE, RMSE, and MAPE values range from 5.2 to 5.6 vehs/5 min, 8.5 to 9.8 vehs/5 min, and approximately 15% to 17.5%, indicating that the model exhibits a favorable tolerance toward detector failures.

3.4.3 | Comparison of station-level and aggregated lane-level traffic volume prediction

Furthermore, the lane-level predictions can be aggregated into station-level predictions, denoted by dashed lines in Figure 12. It can be observed that the performance of the aggregated predictions is comparable to, and in some cases even superior to, the results obtained by training and predicting directly with station-level data.

4 | CONCLUSION

A framework for short-term traffic volume forecasting with limited available data is proposed in this paper. It becomes challenging for a single model to effectively capture all the patterns in complex traffic networks with diverse traffic patterns and limited learning data. The problem of overfitting in training is serious because the data are insufficient to match the task's difficulty. This paper solves the challenge by dividing detectors into a few clusters and developing one predictive model for one cluster. Detectors are grouped into multiple clusters according to the similarity of their profiles. Each cluster corresponds to a series of similar traffic trends. Hence, a complex multi-feature learning task is decomposed into multiple homogeneous feature learning tasks. DL techniques are

applied to extract similar trends in each cluster from the limited data for traffic volume prediction. For the problem that extracting representative profiles of detectors is challenging when learning data are limited, a profile model based on few-shot learning is proposed. Specifically, the proposed framework consists of the profile extraction model, detector clustering, and predictive model. The profile model is designed to extract profiles with the intrinsic characteristics of each detector from limited training data. The extracted profiles are sent to a clustering algorithm to group all detectors into clusters, each with one predictive model. A CNN-LSTM predictive model is designed to capture the spatiotemporal features of all detectors in a cluster to predict the traffic volume for each detector at the next time interval.

The effectiveness of the proposed short-term traffic volume prediction method has been demonstrated through empirical experiments based on the PeMS dataset. The proposed method can forecast lane- and station-level traffic volume with limited training data. Moreover, it reduces the required predictive models for the traffic network and considers the detectors' reliability. In the case of 10% faulty detectors, the average MAE, RMSE, and MAPE of station-level traffic volume prediction increase from 12.7 vehs/5 min, 20.9 vehs/5 min, and 10.5% to 13.9 vehs/5 min, 24.2 vehs/5 min, and 11.7%, respectively. For lane-level traffic volume prediction, the average MAE, RMSE, and MAPE increase from 4.7 vehs/5 min, 7.7 vehs/5 min, and 15% to 5.6 vehs/5 min, 9.6 vehs/5 min, and 16.8%, respectively.

The proposed framework is also applicable for predicting traffic speed and occupancy using traffic speed/occupancy data. The traffic flow theory shows a correlation between flow, speed, and occupancy. Hence, we propose combining the proposed prediction model with TL to obtain knowledge transfer from traffic volume to occupancy/speed prediction tasks, achieving better performance. With training data less than a month, the MAE, RMSE, and MAPE for traffic speed prediction are 0.7 km/h, 1.3 km/h, and 1.3%, respectively. For traffic occupancy prediction, the corresponding values are 0.5%, 1.1%, and 11%.

The proposed method is based on learning patterns from historical data to make predictions. It is unsuitable for unforeseen situations, such as non-recurring congestion. When traffic patterns change, retraining the models using new data is necessary. Since the proposed method is designed for limited data, it allows for retraining the models suitable for new traffic conditions without collecting a large amount of data over a long period. Additionally, considering the different strengths of multiple models and the availability of new data generated daily, future work will incorporate dynamic ensemble learning (Alam et al., 2020) and finite element machines for fast learning (Pereira et al., 2020) to update previously learned models in real time when new data become available. Furthermore, in the future, efforts will be made to explore alternative approaches, such as neural dynamic classification algorithms, to find better methods for determining the optimal number of clusters. Another future direction will focus on incorporating additional information such as weather conditions, visibility, and accident data to address predictions for non-recurring congestion and enhance the predictive capabilities of the models.

REFERENCES

Abduljabbar, R. L., Dia, H., & Tsai, P.-W. (2021). Unidirectional and bidirectional LSTM models for short-term traffic prediction. *Journal of Advanced Transportation*, *2021*, 5589075.

Adeli, H. (2001). Neural networks in civil engineering: 1989–2000. *Computer-Aided Civil and Infrastructure Engineering*, *16*(2), 126–142.

Alam, K. M. R., Siddique, N., & Adeli, H. (2020). A dynamic ensemble learning algorithm for neural networks. *Neural Computing and Applications*, *32*, 8675–8690.

Cao, M., Li, V. O., & Chan, V. W. (2020). A CNN-LSTM model for traffic speed prediction. *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, Antwerp, Belgium (pp. 1–5).

Chen, C. (2002). *Freeway performance measurement system (PeMS)* [Doctoral dissertation, University of California].

Cui, Z., Ke, R., Pu, Z., & Wang, Y. (2020). Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values. *Transportation Research Part C: Emerging Technologies*, *118*, 102674.

Dharia, A., & Adeli, H. (2003). Neural network model for rapid forecasting of freeway link travel time. *Engineering Applications of Artificial Intelligence*, *16*(7-8), 607–613.

Do, L. N., Vu, H. L., Vo, B. Q., Liu, Z., & Phung, D. (2019). An effective spatial-temporal attention based neural network for traffic flow prediction. *Transportation Research Part C: Emerging Technologies*, *108*, 12–28.

Doğan, E. (2021). LSTM training set analysis and clustering model development for short-term traffic flow prediction. *Neural Computing and Applications*, *33*(17), 11175–11188.

Duan, Y., Yisheng, L., & Wang, F.-Y. (2016). Travel time prediction with LSTM neural network. *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Rio de Janeiro, Brazil (pp. 1053–1058).

Fang, Z., Long, Q., Song, G., & Xie, K. (2021). Spatialtemporal graph ode networks for traffic flow forecasting. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, Singapore (pp. 364–373).

Ghosh-Dastidar, S., & Adeli, H. (2003). Wavelet-clustering-neural network model for freeway incident detection. *Computer-Aided Civil and Infrastructure Engineering*, *18*(5), 325–338.

Gu, Y., Lu, W., Qin, L., Li, M., & Shao, Z. (2019). Short-term prediction of lane-level traffic speeds: A fusion deep learning model. *Transportation Research Part C: Emerging Technologies*, *106*, 1–16.

Haghighat, A. K., Ravichandra-Mouli, V., Chakraborty, P., Esfandiari, Y., Arabi, S., & Sharma, A. (2020). Applications of deep learning in

intelligent transportation systems. *Journal of Big Data Analytics in Transportation*, *2*(2), 115–145.

Han, L., Zheng, K., Zhao, L., Wang, X., & Shen, X. (2019). Short-term traffic prediction based on DeepCluster in large-scale road networks. *IEEE Transactions on Vehicular Technology*, *68*(12), 12301–12313.

He, T., Bao, J., Li, R., Ruan, S., Li, Y., Song, L., He, H., & Zheng, Y. (2020). What is the human mobility in a new city: Transfer mobility knowledge across cities. *Proceedings of the Web Conference 2020*, Taipei, Taiwan (pp. 1355–1365).

Innamaa, S. (2000). Short-term prediction of traffic situation using MLP-neural networks. *Proceedings of the 7th World Congress on Intelligent Transport Systems*, Turin, Italy (pp. 6–9).

Jiang, X., & Adeli, H. (2004). Clustering-neural network models for freeway work zone capacity estimation. *International Journal of Neural Systems*, *14*(3), 147–163.

Ke, R., Li, Z., Kim, S., Ash, J., Cui, Z., & Wang, Y. (2016). Real-time bidirectional traffic flow parameter estimation from aerial videos. *IEEE Transactions on Intelligent Transportation Systems*, *18*(4), 890–901.

Khajeh Hosseini, M., & Talebpour, A. (2019). Traffic prediction using time-space diagram: A convolutional neural network approach. *Transportation Research Record*, *2673*(7), 425–435.

Li, G., Knoop, V. L., & Van Lint, H. (2021). Multistep traffic forecasting by dynamic graph convolution: Interpretations of real-time spatial correlations. *Transportation Research Part C: Emerging Technologies*, *128*, 103185.

Li, J., Guo, F., Wang, Y., Zhang, L., Na, X., & Hu, S. (2020). Short-term traffic prediction with deep neural networks and adaptive transfer learning. *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, Rhodes, Greece (pp. 1–6).

Li, M., & Zhu, Z. (2021). Spatial-temporal fusion graph neural networks for traffic flow forecasting. *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence*, Virtual (pp. 4189–4196).

Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2018). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *6th International Conference on Learning Representations, ICLR 2018*, Vancouver, BC, Canada, April 30–May 3, 2018, Conference Track Proceedings.

Liu, Q., Wang, B., & Zhu, Y. (2018). Short-term traffic speed forecasting based on attention convolutional neural network for arterials. *Computer-Aided Civil and Infrastructure Engineering*, *33*(11), 999–1016.

Liu, Y., Lyu, C., Zhang, Y., Liu, Z., Yu, W., & Qu, X. (2021). DeepTSP: Deep traffic state prediction model based on large-scale empirical data. *Communications in Transportation Research*, *1*, 100012.

Lu, W., Rui, Y., & Ran, B. (2020). Lane-level traffic speed forecasting: A novel mixed deep learning model. *IEEE Transactions on Intelligent Transportation Systems*, *23*(4), 3601–3612.

Lv, M., Hong, Z., Chen, L., Chen, T., Zhu, T., & Ji, S. (2021). Temporal multi-graph convolutional network for traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, *22*(6), 3337–3348.

Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., & Wang, Y. (2017). Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*, *17*(4), 818.

Mallick, T., Balaprakash, P., Rask, E., & Macfarlane, J. (2021). Transfer learning with graph neural networks for short-term highway traffic forecasting. *2020 25th International Conference on Pattern Recognition (ICPR)*, Milan, Italy (pp. 10367–10374).

Manibardo, E. L., Laña, I., & Del Ser, J. (2020). Transfer learning and online learning for traffic forecasting under different data availability conditions: Alternatives and pitfalls. *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, Rhodes, Greece (pp. 1–6).

Nguyen, H., Kieu, L.-M., Wen, T., & Cai, C. (2018). Deep learning methods in transportation domain: A review. *IET Intelligent Transport Systems*, *12*(9), 998–1004.

Pereira, D. R., Piteri, M. A., Souza, A. N., Papa, J. P., & Adeli, H. (2020). FEMa: A finite element machine for fast learning. *Neural Computing and Applications*, *32*, 6393–6404.

Rajalakshmi, V., & Ganesh Vaidyanathan, S. (2022). Hybrid CNN-LSTM for traffic flow forecasting. In G. Mathur, M. Bundele, M. Lalwani, & M. Paprzycki (Eds.), *Proceedings of 2nd international conference on artificial intelligence: Advances and applications* (pp. 407–414). Springer Nature Singapore.

Ren, Y., Chen, X., Wan, S., Xie, K., & Bian, K. (2019). Passenger flow prediction in traffic system based on deep neural networks and transfer learning method. *2019 4th International Conference on Intelligent Transportation Engineering (ICITE)*, Singapore (pp. 115–120).

Samant, A., & Adeli, H. (2000). Feature extraction for traffic incident detection using wavelet transform and linear discriminant analysis. *Computer-Aided Civil and Infrastructure Engineering*, *15*(4), 241–250.

Shen, G., Yu, K., Zhang, M., & Kong, X. (2021). ST-AFN: A spatial-temporal attention based fusion network for lane-level traffic flow prediction. *PeerJ Computer Science*, *7*, e470.

Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems* 30, Long Beach, CA.

Song, X., Li, W., Ma, D., Wang, D., Qu, L., & Wang, Y. (2018). A match-then-predict method for daily traffic flow forecasting based on group method of data handling. *Computer-Aided Civil and Infrastructure Engineering*, *33*(11), 982–998.

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018). A survey on deep transfer learning. In V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, & I. Maglogiannis (Eds.), *Artificial neural networks and machine learning—ICANN 2018* (pp. 270–279). Springer International Publishing.

Tarunesh, I., & Chung, E. (2020). Predicting traffic volume and occupancy at failed detectors. *Transportation Research Procedia*, *48*, 1072–1083.

Wang, B., Yan, Z., Lu, J., Zhang, G., & Li, T. (2018). Road traffic flow prediction using deep transfer learning. *Conference on Data Science and Knowledge Engineering for Sensing Decision Support*, Belfast, UK (pp. 331–338).

Wang, J., Gu, Q., Wu, J., Liu, G., & Xiong, Z. (2016). Traffic speed prediction and congestion source exploration: A deep learning method. *2016 IEEE 16th International conference on Data Mining (ICDM)*, Barcelona, Spain (pp. 499–508).

Wang, L., Geng, X., Ma, X., Liu, F., & Yang, Q. (2019, 7). Cross-city transfer learning for deep spatio-temporal prediction. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, Macao, China (pp. 1893–1899).

Yao, B., Chen, C., Cao, Q., Jin, L., Zhang, M., Zhu, H., & Yu, B. (2017). Short-term traffic speed prediction for an urban corridor. *Computer-Aided Civil and Infrastructure Engineering*, *32*(2), 154–169.

Yao, H., Liu, Y., Wei, Y., Tang, X., & Li, Z. (2019). Learning from multiple cities: A meta-learning approach for spatial-temporal prediction. *The World Wide Web Conference*, San Francisco, CA (pp. 2181–2191).

Yu, B., Yin, H., & Zhu, Z. (2018). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *International Joint Conferences on Artificial Intelligence Organization*, Stockholm, Sweden (pp. 3634–3640).

Yu, X., Ma, S., Zhu, N., Lam, W. H., & Fu, H. (2023). Ensuring the robustness of link flow observation systems in sensor failure events. *Transportation Research Part B: Methodological*, *178*, 102849.

Zhang, S., Yao, Y., Hu, J., Zhao, Y., Li, S., & Hu, J. (2019). Deep autoencoder neural networks for short-term traffic congestion prediction of transportation networks. *Sensors*, *19*(10), 2229.

Zhang, S., Zhou, L., Chen, X., Zhang, L., Li, L., & Li, M. (2020). Network-wide traffic speed forecasting: 3D convolutional neural network with ensemble empirical mode decomposition. *Computer-Aided Civil and Infrastructure Engineering*, *35*(10), 1132–1147.

Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., & Li, H. (2019). T-GCN: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, *21*(9), 3848–3858.

Zhou, J., Shuai, S., Wang, L., Yu, K., Kong, X., Xu, Z., & Shao, Z. (2022). Lane-level traffic flow prediction with heterogeneous data and dynamic graphs. *Applied Sciences*, *12*(11), 5340.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., & He, Q. (2021). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, *109*(1), 43–76.

Zou, X., Chung, E., Zhou, Y., Long, M., & Lam, W. H. (2024). A feature extraction and deep learning approach for network traffic volume prediction considering detector reliability. *Computer-Aided Civil and Infrastructure Engineering*, *39*(1), 102–119.

How to cite this article: Zou, X., & Chung, E. (2024). Traffic prediction via clustering and deep transfer learning with limited data. *Computer-Aided Civil and Infrastructure Engineering*, *39*, 2683–2700. https://doi.org/10.1111/mice.13207