# MemDA: Forecasting Urban Time Series with Memory-based Drift Adaptation

Zekun Cai\* caizekun@csis.u-tokyo.ac.jp The University of Tokyo Tokyo, Japan

Zhaonan Wang znwang@csis.u-tokyo.ac.jp The University of Tokyo Tokyo, Japan Renhe Jiang<sup>†</sup> jiangrh@csis.u-tokyo.ac.jp The University of Tokyo Tokyo, Japan

Diansheng Guo<sup>†</sup> diansheng.guo@gmail.com Tencent Corporation Beijing, China

Xuan Song songxuan@csis.u-tokyo.ac.jp The University of Tokyo Tokyo, Japan

# ABSTRACT

Urban time series data forecasting featuring significant contributions to sustainable development is widely studied as an essential task of the smart city. However, with the dramatic and rapid changes in the world environment, the assumption that data obey Independent Identically Distribution is undermined by the subsequent changes in data distribution, known as concept drift, leading to weak replicability and transferability of the model over unseen data. To address the issue, previous approaches typically retrain the model, forcing it to fit the most recent observed data. However, retraining is problematic in that it leads to model lag, consumption of resources, and model re-invalidation, causing the drift problem to be not well solved in realistic scenarios. In this study, we propose a new urban time series prediction model for the concept drift problem, which encodes the drift by considering the periodicity in the data and makes on-the-fly adjustments to the model based on the drift using a meta-dynamic network. Experiments on real-world datasets show that our design significantly outperforms state-ofthe-art methods and can be well generalized to existing prediction backbones by reducing their sensitivity to distribution changes.

## **CCS CONCEPTS**

 Computing methodologies → Knowledge representation and reasoning; Artificial intelligence; • Applied computing;

\*This work was fulfilled when Zekun Cai interned at Tencent.
<sup>†</sup>Corresponding Author

CIKM '23, October 21-25, 2023, Birmingham, United Kingdom.

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 979-8-4007-0124-5/23/10...\$15.00 https://doi.org/10.1145/3583780.3614962 Xinyu Yang xinyuyang@tencent.com Tencent Corporation Beijing, China

Hill Hiroki Kobayashi kobayashi@ds.itc.u-tokyo.ac.jp The University of Tokyo Tokyo, Japan

Ryosuke Shibasaki shiba@csis.u-tokyo.ac.jp The University of Tokyo Tokyo, Japan

## **KEYWORDS**

time series, concept drift, domain adaptation, urban computing

#### **ACM Reference Format:**

Zekun Cai, Renhe Jiang, Xinyu Yang, Zhaonan Wang, Diansheng Guo, Hill Hiroki Kobayashi, Xuan Song, and Ryosuke Shibasaki. 2023. MemDA: Forecasting Urban Time Series with Memory-based Drift Adaptation. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/ 3583780.3614962

#### **1** INTRODUCTION

Accurate and reliable forecasting of urban multivariate time series data, such as traffic speed, travel demand, and electricity consumption, has been intensively studied as an essential task since it contributes to reducing traffic congestion, energy waste, and greenhouse gas emissions [13, 15, 17, 27, 39]. The advancement of machine learning, particularly deep learning [24, 32], has facilitated both research and practical applications in this field. However, the underlying assumption that these algorithms rest on - independent and identical distributions (i.i.d.) - may not always hold true in the fast-paced, dynamic urban environment teeming with implicit event-induced uncertainty. As an illustrative example, consider the changes in daily traffic speeds in Beijing as shown in Fig. 1. The grayscale ribbon plot in Fig. 1(a) demonstrates that the road speed decreases over Mid-2022, indicating changes in the road speed distribution. The evolution of this distribution, as captured week-by-week in the ridgeline plots of Fig. 1(b), shows the shift from a concentration around 75 km/h to a broader distribution around 10-75 km/h. The drift distribution imposes vulnerability on models trained from earlier data as they are not expected to generalize well to the new coming data, known as the Concept Drift problem [34]. As a result, Fig. 1(c) highlights an increase in the error of a pre-trained model over time, with error distribution spreading from low to medium and high. These changes reflect the realities of a city in flux, where social, economic, and environmental

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.



Figure 1: Illustration of traffic speed concept drift in Beijing from May 8 to July 25, 2022. (a) Grayscale ribbon of road speeds, with highlighted data for three sampled roads. (b) The ridgeline plot exhibits weekly changes in the road speed distribution. (c) The ridgeline plot exhibits weekly changes in the error distribution of the predictive model, and the subplot shows the trend of the daily error.

factors can drastically affect urban patterns, leading to concept drift. It is considered a primary reason for the declining effectiveness of intelligent information systems, as the change in data distribution weakens the induced correlation patterns between new and past data [26]. Addressing this challenge can enhance the equity, safety, and sustainability of cities and communities, assisting urban decision-makers and stakeholders in using data-driven approaches to tackle urban problems amidst continuous social changes.

Learning from changing data has attracted considerable research interest. For non-stationary time series with concept drift, studies are conducted to: (1) detect the occurrence of concept drift by error rate-based [4, 9] or data distribution-based [3, 20] detectors; (2) train a new model, fine-tune the current model, or ensemble the new and existing model [7, 23, 40] to fit the latest data. However, such incremental learning approaches can only adjust the model to narrow the gap among the drifting distributions. Awaiting the accumulation of new data inevitably leads to lagging. The latest adapted model could fail again if the drift continues. And even many scenarios do not qualify for frequent retraining as periodic updates are costly. Considering that urban models are often associated with substantial economic benefits and sustainable social development, this prompts us to explore an evolving approach beyond retraining that can be proactively adjusted on-the-fly based on inputs.

Nevertheless, developing a self-configuring learner is non-trivial. A crucial consideration is the computational efficiency. Addressing concept drift implies the necessity for the model to process extensive historical data, providing a comprehensive understanding of past data distribution to accurately perceive ongoing drifts. However, the assimilation and retention of large volumes of historical data impose a significant computational burden. The model must therefore exhibit the capacity to manage historical data in an intelligent manner, maintaining long-term historical data whilst picking the most valuable information. Second, the navigation of the stability-plasticity dilemma cannot be overlooked. An adaptive model with essential levels of responsiveness to new trends must achieve an equilibrium between maintaining prior learned knowledge (stability) and accommodating new data patterns (plasticity). This poses high requirements in deep learning models as the parameter space of the model can be extremely large. Searching this space for the optimal configuration involves considerable degrees of freedom and can be practically infeasible for large models.

To confront the identified challenges, we propose a novel drift adaptation network designed around two key components: a dual memory module and a strategically adjustable meta-dynamic network. Firstly, our model incorporates a dual memory module to enhance computational efficiency while handling extensive historical information. This module consists of a Replay Memory for time-efficient drift embedding and a Pattern Memory for preserving significant patterns over extended periods. Then a novel driftadaptive meta-dynamic network is utilized to generate adjustable parameters for bridging the encoder and decoder, allowing the model to respond to changing data patterns with minimal degrees of freedom. Our approach is both simple and general, as it only requires tracking a few parameters to model the drift and can be highly compatible with any modern urban deep learning models as a plug-and-play module for handling concept drift. Compared to existing models, the new network improves prediction under non-stationary time series in a robust and self-generating manner. Our contribution is summarized in the following sections.

- We first propose to model the urban concept drift by finding learnable components in non-stationarity, which facilitates a more nuanced understanding of urban dynamics and introduces new perspectives for prediction in such environments.
- We propose MemDA, an on-the-fly adaptation architecture that implicitly perceives drift without retraining and dynamically adjusts the corresponding model parameters.
- The proposed model is pluggable and backbone-agnostic. Extensive experiments on four urban datasets demonstrate the superior performance of our design<sup>1</sup>.

## 2 RELATED WORK

#### 2.1 Concept Drift and Model Adaptation

Time-varying patterns in the data render obsolete the knowledge previously gained from old and static data, thereby reducing the predictive capability of the model for new data [19, 26]. As the gap between the latest and upcoming data distribution is more likely narrow, the fresh training model can be utilized as a tool to address the issue. There are three types of approaches based on trade-offs for preserving historical information: the straightforward way is to retrain a new model with the most recent data. An explicit sliding

<sup>&</sup>lt;sup>1</sup>https://github.com/deepkashiwa20/Urban\_Concept\_Drift

CIKM '23, October 21-25, 2023, Birmingham, United Kingdom.

window drift detector is commonly employed to determine when to retrain [4, 23]. Alternatively, in some cases, retaining and finetuning the old model can save efforts in training the new model. New base predictors can be added to existing collections of models, or data-adaptive learning models be created. [40] designs multiple predictors using a deep ensemble approach and trains a classifier to predict which predictor to use. [44] proposes a weighted method for reusing previous models, where each model is associated with a metric that represents its reusability for the current observation. [7] uses a segmentation algorithm to divide the time series and uses a transfer learning approach to integrate knowledge under various concepts. However, training-based approaches are just attempting to catch up with the evolution of the concept. They can only build models after drift and inevitably suffer from model aging.

Normalizing the input into a normalized distribution is also used to deal with non-stationarity. RevIN[21] to remove and rebuild the discrepancy in the mean and standard deviation of the input sequences. The Non-stationary Transformer[25] mitigates the overstationarization problem by calculating the normalized attention. Dish-TS[8] learns sample-level normalization parameters. However, The normalization-based approaches simply attempt to align the magnitudes of the samples, with a weak capacity for modeling changes in data patterns and conditional probability distributions.

#### 2.2 Urban Time Series Forecasting

Building time series models to forecast future traffic speed, taxi demand, electricity consumption, crowd in/outflow, etc., are widely studied as key technologies for smart city applications. [39] and [6] employ an LSTM structure to connect hidden embeddings for urban data forecasting. [22], [24], [1], and [43] use GRU for more efficient modeling of traffic volume and speed. Meanwhile, convolutionbased modules, such as 1D CNN and TCN, are shown to be competent with temporal data as well. [41], [28], [36], and [35] for predicting urban data demonstrate the efficiency and effectiveness of such designs. Attention mechanisms in natural language processing have also been adapted. [12, 38] design models with multiple inputs based on the periodicity of the data (hours, days, or weeks) and subsequently use attention to capture the dynamic temporal correlation and assign greater weight to important moments. [45] and [37] also propose similar attention modules for better global/longterm temporal modeling. However, mainstream efforts in urban data forecasting assume constant data patterns. The pre-trained models are highly vulnerable to the drift in data.

Another line focuses on the prediction of urban data under events. [30] predicts human movements during earthquakes using a Hidden Markov Model. [15, 16] models the crowd dynamics during four big events by mobility momentum; EAST-Net [33] enhances the robustness of the deep model using heterogeneous information networks and memory bank. However, prediction in the event case is closer to emergency response, which is concerned with the effect of the model on localized outliers, whereas the concept drift is focused on persistent changes in the patterns underlying the data.

## **3 PRELIMINARIES**

**Concept Drift.** Concept drift refers to the underlying distribution of streaming data varies over time caused by changes in unobserved

hidden variables [26]. It will lead to model performance degradation as the statistical properties of the target domain do not match the history. Formally, given a set of instances obtained at different timesteps { $(x_0, y_0), ..., (x_T, y_T)$ }, where  $x_t$  is the feature and  $y_t$  is the target, concept drift at timestamp t can be described as  $p_{0:t-1}(x, y) \neq p_{t:T}(x, y)$ , where  $p(\cdot)$  denotes the joint probability distribution. The source of concept drift can be decomposed into two parts: changes in the marginal probability distribution p(x)and changes in the conditional probability distribution p(y|x).

Adaptive Forecasting. Given a specified granularity of urban time series data, the tensor  $x_t \in \mathbb{R}^{N \times C}$  can be represented as the observation at timestamp t, where N denotes the number of urban sensors (or regions) and C denotes the feature channel (e.g., traffic speed, crowd). Then given the historical observations  $\{x_i | i \in [0, t]\}$ , building adaptive model for the future  $\alpha$  steps of urban data  $\hat{Y}_t =$  $\{\hat{x}_i | i \in [t + 1, t + \alpha]\}$  is to obtain such parameters  $\theta_t$  that can minimize the objective function  $\mathcal{L}(\cdot)$  as follows:

$$\theta_t = \underset{\theta_t}{\operatorname{argmin}} \mathcal{L}(\hat{Y}_t, \{x_i | i \in [t+1, t+\alpha]\}). \tag{1}$$

where  $x_i$  are drawn from a time-varying distribution.  $\theta_t$  is learned from the training data and will be continuously adapted to test data.

#### **Table 1: Notations**

Symbol	Description
$\overline{x_t}$	Urban data observation tensor at timestamp $t$
N, C	Dimension of $x_t$
Т	Total number of training samples
p(x, y)	Joint probability distribution of variables $x$ and $y$
α	Prediction leading time
$X_t$	Observations of urban data over time $[t - \alpha, t]$
$X_t$	A collection of observations constructed by periodicity
p	Data sampling volume in one day
k	the k-th look-back day for drift embedding
$P_k$	Look-back timestamp for drift embedding, $P_k = p \times k$
Κ	Total look-back days
$\mathcal{F}$	Neural network functions
$Z_t$	Embedding result of $X_t$ by Encoder
Ce	Dimension of embedding generated by encoder
$E_t$	Embedding result of $X_t$ by Encoder and Replay Memory
M	Pattern Memory
L, D	Dimension of Pattern Memory
$V_t$	The query result of $E_t$ returned by Pattern Memory
$H_t$	Drift Embedding result of $X_t$ by Dual-Memory
$W_t$	Drift adaptation parameters
Ns	Dimension of similarity generated by NTN
I(X;Y)	Mutual information between variables $X$ and $Y$

## 4 METHODOLOGY

## 4.1 Rationale

Although the distribution of urban data may change due to a variety of factors, the inherent periodicity of urban dynamics follows a consistent temporal regularity[11]. This persistent feature enables the model to maintain its predictive capabilities, as the accumulation and integration of fresh patterns facilitate adaptive learning even when distribution undergoes significant transformations. It serves as an anchor for our model, guiding the model toward the interpretation and management of concept drift. Moreover, we acknowledge the sustained features to be encoded in the pre- and



Figure 2: The framework of the memory-based drift adaptation model.

post-drift data retain significant commonalities. These sustained features can be efficiently captured by the stable component of the model, which means modifications to some key model parameters should be prioritized instead of updating all parameters. This strategy strikes a balance between stability and plasticity, ensuring the resilience of the model to varying data patterns and distributions.

Motivated by these, we design a **Mem**ory-based **D**rift Adaptation (**MemDA**) network illustrated in Fig. 2, where we first propose to use a Dual-Memory to encode periodicity implied in the long sequential inputs to counteract the drift, and later use meta dynamic networks to generate adaptation parameters governing the integration of input information. The notations are listed in Table 1.

## 4.2 Dual-Memory for Drift Embedding

Periodic inputs are essential for coping with urban data dynamics, while long sequential inputs are the informational foundation for perception to drift. Therefore, as shown in Fig. 3(a), we accordingly construct **Drift-Aware Inputs**  $X_t$  along the time axis. Assuming that the data are sampled at a frequency of p per day, the inputs are constructed by fetching up to K days of data as follows:

$$X_t = \{X_t\} \cup \{X_{t-P_k}, X_{t+\alpha-P_k} | k \in [1, K]\}$$

where  $P_k = p \times k$  and  $X_t, X_{t-P_k}, X_{t+\alpha-P_k}$  are:

$$X_{t} = \{x_{t-\alpha}, x_{t-\alpha+1}, ..., x_{t}\},$$
  

$$X_{t-P_{k}} = \{x_{t-\alpha-P_{k}}, x_{t-\alpha+1-P_{k}}, ..., x_{t-P_{k}}\},$$
  

$$X_{t+\alpha-P_{k}} = \{x_{t+1-P_{k}}, x_{t+2-P_{k}}, ..., x_{t+\alpha-P_{k}}\}.$$
(2)

The segment  $X_t$  consists of the closest  $\alpha$  observations up to time t. The segments  $X_{t-P_k}$  and  $X_{t+\alpha-P_k}$  consist of the same number of observations as  $X_t$  but from before and after the same clock t prior to k days. Here,  $P_k$  represents the time gap between the backtrack and the present. Thus,  $X_t$  contains 2K + 1 segments and each segment has  $\alpha$  observations.

However, for such plain long time series (i.e.,  $(2K + 1) \times \alpha$  steps), traditional approaches [12, 14, 42] that utilize multi-branch to encode the sequence will be no longer applicable as the computational complexity exponentially increases as the input grows. Therefore,



Figure 3: Design of model inputs switches from Plain Blocks (a) to Replay Memory enhanced (b) inputs.

we propose to use a **Replay Memory (RM)** for efficient drift embedding. RM is essentially a queue that stores all the hidden embedding of  $X_t$  computed by the encoder, forming  $Z \in \mathbb{R}^{T \times N \times C_e}$ , where Tis the number of training samples and  $C_e$  is the output dimension of the encoder. During training, the RM fetches the embeddings  $Z_{t-P_k}$  and  $Z_{t+\alpha-P_k}$  of the past segments  $X_{t-P_k}$  and  $X_{t+\alpha-P_k}$  to avoid duplicate computing. Without loss of generality, we use  $\mathcal{F}_e$ to denote the encoder network with parameter  $\theta_e$ . As shown in Fig. 3(b), the embedding result  $E_t$  for Drift-Aware Inputs  $X_t$  is converted to the  $X_t$  encoding and RM replaying as:

$$E_{t} = \underbrace{\{\mathcal{F}_{e}(X_{t}, \theta_{e})\}}_{\text{Encoding}} \cup \underbrace{\{Z_{t-P_{k}}, Z_{t+\alpha-P_{k}} | k \in [1, K]\}}_{\text{Replaying}}.$$
 (3)

The embedding of  $X_t$  is subsequently updated to RM by  $Z_t = \mathcal{F}_e(X_t, \theta_e)$ . Note that replayed memories are collected from the previous epoch while  $Z_t$  is obtained at the current epoch (i.e., one epoch lag) when training. However, as the epoch proceeds, they will gradually converge to the same embedding space.

While periodic long inputs ensure the possibility of tracking data drift, the model cannot extend segments indefinitely, and distribution information may not be limited to periodicity. Therefore, we further augment the drift embedding  $E_t$  with a **Pattern Memory** (**PM**). The PM explicitly records the prototype of normal patterns during training and serves to perceive out-of-distribution during inference. Specifically, the Pattern Memory  $M \in \mathbb{R}^{L \times D}$  is a parameter matrix with *L* real-valued *D* dimensional prototypes. For the *i*-th embedding in  $E_t$ , *M* is queried to find similar prototypes by attention. Formally, we have:

$$\begin{cases} Q_i = E_t^i W_Q + b_Q \\ \varphi_j = \frac{e^{Q_i * M_j}}{\sum_{j=1}^L e^{Q_i * M_j}} \\ V_t^i = (\sum_{j=1}^L \varphi_j \cdot M_j) W_V + b_V, \end{cases}$$
(4)

where  $\varphi_j$  is the attention score corresponding to *j*-th prototype;  $W_Q, b_Q, W_V, b_V$  are linear transformation parameters;  $V_t^i$  denotes the query result of  $E_t^i$ .

Then,  $E_t$  and  $V_t$  are concatenated to form the final drift embedding  $H_t = [E_t, V_t]$ , which encodes the changes in data distribution and implies the essential information for prediction. In general, the weighted fusion of embedding can integrate all the information and can be fed to the decoder for future predictions. One mainstream approach [1, 38, 42] is to use the trainable parameter to estimate the influence weights of each segment as:

$$\hat{Y}_t = \mathcal{F}_{proj.}(\sum_{i=1}^{4K+2} W^i \odot H^i_t)$$
(5)

where  $\odot$  is Hadamard product;  $W \in \mathbb{R}^{4K+2}$  are learnable parameters, and  $\mathcal{F}_{proj.}$  projects the result to the prediction. Next, we introduce a dynamic network to make the key parameters W self-configurable.

## 4.3 Meta-Dynamic for Drift Adaptation

We characterize model adaptation in terms of Information Bottlenecks (IB). The IB problem was introduced in [31] as an informationtheoretic framework for learning. It considers quantifying the amount of information of hidden representation H extracted from the observation X contains about another relevant target signal Y by using Mutual Information (MI). The goal is to find a H that minimizes the mutual information I(X; H) while keeping I(H; Y) above the threshold. IB theory suggests that deep neural networks construct Markov chains of the hidden layer representations about the input X, and DNNs trained using SGD are essentially solving the compression-prediction trade-off problem [10], which means deeper layers correspond to smaller I(X; H) but larger I(H; Y).

When it comes to the concept drift problem, in the encoderdecoder network, the encoder essentially acts as an information bottleneck that filters the noise from the data and forces the network to extract the typical patterns of high-dimensional data, of which outputs are then transmitted to the decoder. This makes I(X; H)and I(H; Y) a function of the en/decoder parameters. When drift occurs, the encoder is more resistant to drift as it is more focused on information compression. This leads us to fix the encoder parameters  $\theta_e$  to reduce the degrees of freedom while preserving the ability to extract features over different target domains. For the adaptor, since we want to boost the MI between the hidden embedding  $H_t$  and  $Y_t$ , the fusion parameter W should be actively and dynamically adjusted to different magnitudes under different concepts, i.e., varying with time t. For example, the weight of historical information is evenly distributed when the data is stable, while the influence of history is reduced when the data is out of distribution. To generate the appropriate  $W_t$ , it is necessary to decide how much each segment embedding contributes to the future, which means it should be proportional to the MI of  $H_t$  and  $Y_t$ , i.e.,

$$\begin{split} W_t &= softmax(I(H_t^1; Y_t), ..., I(H_t^{4K+2}; Y_t)) \\ &= softmax(\mathbb{E}_{Y_t} \{ D_{KL}(p(H_t^1) \| p(H_t^1 | Y_t)), ..., 0 \} \\ D_{KL}(p(H_t^{4K+2}) \| p(H_t^{4K+2} | Y_t)) \}). \end{split}$$

However, since the computation of MI requires knowledge of the joint and marginal distributions of two random variables, the distributions that can be depicted for high-dimensional urban datasets are stretched compared to the true continuous distribution, especially when the joint distribution changes in drift scenarios. This makes it tough to compute MI directly. Nevertheless, it is possible to approximate the MI between high-dimensional variables using a neural network with gradient descent optimization[2]. Therefore, we propose to use a meta-dynamic network to approximate the contribution of each segment by dynamically generating the corresponding  $W_t^i$  conditional on the input. Specifically, to perceive the drift, we construct three lists of temporal alignment tuples,

$$Pair_{x} = \{(Z_{t}, Z_{t-P_{1}})\} \cup \{(Z_{t-P_{k}}, Z_{t-P_{k+1}})\},\$$

$$Pair_{y} = \{(Z_{t+\alpha-P_{k}}, Z_{t+\alpha-P_{k+1}})\},\$$

$$Pair_{xy} = \{([Z_{t-P_{k}}, Z_{t+\alpha-P_{k}}], [Z_{t-P_{k+1}}, Z_{t+\alpha-P_{k+1}}])\}.$$
(7)

The constructed embedding pairs are used to measure the evolution of the distribution over the marginal probability p(x) and the conditional probability p(y|x). Each pair serves as an anchor to facilitate determining the contribution of each segment embedding to the prediction. Each  $(Z_i, Z_j) \in Pair$  is fed into a Neural Tensor Network (NTN) [29] to model the relation:

$$sim_k = \sigma(Z_i^T W_S Z_j + V[Z_i, Z_j]^T + b_S),$$
  

$$W_t = \mathcal{F}_{meta}([sim_1, sim_2, ...]),$$
(8)

where  $W_S \in \mathbb{R}^{C_e \times N_e}$ ,  $V \in \mathbb{R}^{N_s \times 2C_e}$ , and  $b_S \in \mathbb{R}^{N_s}$  are weight tensor.  $N_s$  is a hyperparameter that controls the drift scores generated by the model for each pair from different perspectives.  $\sigma$ is an activation function.  $\mathcal{F}_{meta}$  denotes a meta layer that can be implemented in various ways, which generates fusion parameters based on the similarity matrix to adjust to the drift on-the-fly. The generated  $W_t$  are loaded into the fusion layer replacing the static Win Eq. 5, and a multilayer CNN is employed as a decoder to project the result to the prediction. The groundtruth values are taken to calculate the loss with prediction to optimize the model.

#### 4.4 Training Details

ŀ

During the training phase, we train the model by random sampling batch, and the RM is updated between epochs. During the testing phase, we feed the samples into the model in chronological order, and the RM is updated sequentially. Note that in the training phase, we maintain a full number of embeddings in RM as many as *T*,

while in the testing phase and online environment, we only retain the most recent  $p \times K$  embeddings to prevent unlimited growth. The PM is updated as the gradient descends and is fixed in the test.

## **5 EXPERIMENTS**

## **Table 2: Summary of Experimental Datasets**

Dataset	Period	#N	Temporal	Drift
PeMS	2020/01/01~2020/07/31	325	5 minutes	Sudden
Beijing	2022/05/12~2022/07/25	3126	5 minutes	Sudden
Electricity	2012/01/01~2012/06/30	370	1 hour	Incremental
COVID-CHI	2019/07/01~2020/12/31	112	2 hour	Incremental

#### 5.1 Datasets

We collected data from four cities covering different data sources and concept drift types, which are presented in Table 2. The PeMS and Beijing datasets are collected from the traffic speeds of major roads in California and Beijing. The Electricity dataset contains the electricity consumption. And the COVID-CHI [33] dataset is the demand for shared bicycles collected from Chicago. To better understand how the data distribution has changed, we show sampled data in Fig. 4, where each dataset exhibits different types of concept drift. These drifts are caused by known events (i.e., COVID, Seasonality) and result in significant variations in the data distribution.



Jul 2019 Sep 2019 Nov 2019 Jan 2020 Mar 2020 May 2020 Jul 2020 Sep 2020 Nov 2020

Figure 4: Two nodes from each of the four datasets are drawn for time series plots. The black vertical lines indicate the cut-off positions of the training and testing datasets.

#### 5.2 Settings

We annotate the time of events causing drift on each dataset. Based on this we divide the data before the drift into the training set and the rest as the test set, and further selected 20% of the training set as the validation set. The cut-off positions for training and testing are shown in Fig. 4. As for the model, the predicted leading time  $\alpha$  is set to 12. The hyperparameters are chosen based on the performance of the validation set. The look-back days K for drift embedding is set to 2, with the encoding dimension  $C_e$  being 256, the number of prototypes L of Pattern Memory being 20, and the dimension D being 32. The GW-Net [36] is chosen as the encoder backbone for the model as its proven superior performance [18]. We set the  $N_s$  in the drift adaptation module as 5, and the meta-layer is implemented by a linear layer. As for the training, MAE is chosen to be optimized using Adam, with a batch size set to 64 and a learning rate of 0.001. The best model is selected by early stop with a maximum of 200 epochs. The experiments were performed on a GPU server with four GeForce RTX 3090 graphic cards. Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) are selected to evaluate the results.

## 5.3 Baselines

*5.3.1 Non-adaptive Models.* We implemented state-of-the-art methods on urban data prediction for comparison, including:

- CopyLastDay (CLD): Observations from the same moment of the previous day are used as predicted values.
- ARIMA: AutoRegressive Integrated Moving Average is a widely used statistical model for time series prediction by identifying patterns in data.
- AGCRN [1]: A node parameter learning and graph generation model for traffic forecasting tasks. It automatically captures the spatial and temporal correlation of covariates in time series data by adaptive modules.
- **DCRNN** [24]: A model for modeling multivariate time-series prediction tasks using bidimensional graph diffusion and recurrent neural networks.
- MTGNN [35]: An advanced multivariate time-series prediction model using active graph learning and multi-kernel TCN modules.
- **STGCN** [41]: A spatio-temporal network that combines graph convolution with 1D convolution.
- ASTGCN [12]: A novel attention-based spatial-temporal graph convolution model. It features three temporally separate components to model periodic dependencies.
- GMAN [45]: A model using multi-attentive networks with gated fusion to fit nonlinear temporal correlations and adaptively fuse information.
- StemGNN [5]: A predictive model for modeling the spatial and temporal dependence of multivariate time series in the spectral domain.
- GW-Net [36]: A state-of-the-art multivariate modeling model using parametric graph inputs and a WaveNet-like temporal dilated structure.
- STTN [37]: A model in which various relationships related to different factors in covariates are modeled jointly by spatial and temporal transformers.
- EAST-Net [33]: A novel model for modeling event-induced non-stationary prediction problems in multimodal data via heterogeneous mobility information networks.

MemDA: Forecasting Urban Time Series with Memory-based Drift Adaptation

Table 3: Performance Comparisons with state-of-the-art Urban Time Series Prediction Models

	PeMS			Beijing			Electricity			COVID-CHI		
Model	RMSE	MAF	MAPF	RMSF	MAF	MAPE	RMSF	MAF	MAPF	RMSF	MAF	MAPF
CLD	4.022	1 784	3 140%	11.028	5 225	15.020%	81 325	41.088	15 422%	15.879	6 780	86.680%
CLD	4.022	1.704	5.140%	11.028	5.225	13.02976	01.525	41.000	13.42270	13.079	0.789	80.080%
ARIMA	3.135	1.215	2.301%	9.184	4.640	16.400%	94.334	64.808	18.222%	16.721	6.903	80.334%
AGCRN	2.572	1.191	2.094%	8.715	4.238	14.376%	95.242	50.422	15.451%	16.279	7.087	85.224%
DCRNN	2.500	1.158	2.030%	8.280	3.811	13.838%	125.335	125.335	25.754%	17.453	7.175	87.617%
MTGNN	2.426	1.148	2.007%	10.976	5.120	19.928%	96.475	51.562	15.410%	18.721	7.434	89.097%
STGCN	2.556	1.212	2.114%	10.284	5.000	15.284%	132.209	72.802	22.580%	22.462	8.687	85.064%
ASTGCN	2.402	1.157	2.078%	9.156	4.250	17.130%	74.074	39.388	13.584%	15.242	6.830	77.223%
GMAN	2.849	1.298	2.308%	11.555	6.448	21.941%	138.436	81.285	26.263%	22.462	8.687	85.064%
StemGNN	2.414	1.144	2.030%	8.575	4.294	15.605%	103.375	56.803	19.366%	17.231	7.132	85.32%
GW-Net	2.385	1.125	1.976%	7.754	3.564	11.916%	97.087	51.684	16.120%	16.210	6.857	80.038%
STTN	2.441	1.168	2.040%	9.886	4.748	17.150%	93.125	50.671	15.776%	14.853	6.934	79.155%
EAST-Net	2.399	1.118	1.988%	11.671	5.970	21.119%	100.789	55.158	18.714%	17.543	7.832	82.321%
MemDA	2.297	1.053	1.861%	6.720	3.192	9.913%	67.413	34.814	12.186%	14.003	6.115	72.088%
$\Delta\%$	3.70%	5.82%	5.78%	13.34%	10.43%	16.81%	8.99%	11.61%	10.29%	5.73%	9.93%	9.93%

Table 4: Performance Comparisons with state-of-the-art Adaptive Models

Model	PeMS			Beijing			Electricity			COVID-CHI		
Model	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
Backbone	2.385	1.125	1.976%	7.754	3.564	11.916%	97.087	51.684	16.120%	16.210	6.857	80.038%
RevIN	2.541	1.129	1.956%	8.116	3.707	11.200%	85.352	45.434	15.847%	15.472	6.789	80.049%
$\Delta\%$	No	Improver	nent	No	No Improvement		12.1%	12.1%	1.7%	4.6%	1.0%	-
Dish-TS	2.519	1.144	1.973%	7.738	3.586	10.499%	73.509	41.465	13.726%	14.807	6.435	74.182%
$\Delta\%$	No	Improver	nent	No	No Improvement		24.3%	19.8%	14.8%	14.8%	12.7%	14.0%
MemDA	2.297	1.053	1.861%	6.720	3.192	9.913%	67.413	34.814	12.186%	14.003	6.115	72.088%
$\Delta\%$	3.7%	6.4%	5.8%	13.3%	10.4%	16.8%	30.6%	32.6%	24.4%	19.4%	17.1%	16.4%

## **Table 5: Variants Performance Evaluation**

Variant	PeMS			Beijing			Electricity			COVID-CHI		
varialit	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
Backbone	2.385	1.125	1.976%	7.754	3.564	11.916%	97.087	51.684	16.120%	16.210	6.857	80.038%
RM	2.359	1.114	1.975%	7.747	3.607	12.422%	69.945	36.093	12.411%	17.578	7.362	84.249%
RM+PM	2.319	1.060	1.856%	8.748	4.048	14.627%	71.184	36.652	12.875%	14.470	6.296	74.447%
MemDA	2.297	1.053	1.861%	6.720	3.192	9.913%	67.413	34.814	12.186%	14.003	6.115	<b>72.088</b> %

*5.3.2* Adaptive Models. The latest advanced adaptive models for non-stationary time series are also chosen to compare the effect on the concept drift problem, including:

- **RevIN** [21]: An instance normalization method for against distribution shift that suppresses non-stationary information in one input layer and restores it in the output layer.
- **Dish-TS** [8]: An advanced version of RevIN, which designs a coefficient net that maps input sequences into learnable distribution to relieve distribution change.

For a fair comparison, we select adaptive models that can also function pluggable for comparison. The Backbone model is chosen as GW-Net based on its optimal performance on the four datasets.

#### 5.4 Performance Evaluation

*5.4.1* Overall Performance Comparison. Table 3 shows the results of MemDA and baselines under three evaluation metrics. The proposed MemDA consistently outperforms the best baseline model on all four datasets, achieving a reduction in *MAE* of approximately

6% to 12%. These results demonstrate the effectiveness of the adaptive model under drift scenarios. Furthermore, we observed that the model performance on test data varied. Among the two speed datasets, the Beijing dataset presented a greater challenge due to the impact of lifting lockdowns, which caused traffic speeds to drift from uniformly smooth to fluctuating. The proposed model exhibited a greater advantage over the baselines when dealing with such new patterns, resulting in an average improvement of 13% for the three evaluation metrics. The PeMS dataset, on the other hand, drifted from congested to uncongested, which can be interpreted as an oversampling from the original distribution. Therefore, the concept drift problem may not cause significant issues, but our approach still achieved a 6% improvement in the MAE metric. For the Electricity datasets, ASTGCN, which also utilized periodically constructed inputs, achieved the best results among the baseline models. However, our proposed framework still achieves an additional 11% improvement, demonstrating the potential of self-configuration capabilities in fully exploiting long-term historical information. Finally, all baseline models showed poor results on the COVID-CHI

dataset, failing to match the simplest rule-based model CopyLast-Day in terms of *MAE*. Nevertheless, MemDA still stands out and shows strong robustness in the drift.

Table 4 shows the results of MemDA and SOTA adaptive models. The experimental results presented clearly demonstrate that our proposed MemDA model outperforms the other existing adaptive models. Moving onto the analysis of these results, RevIN and Dish-TS dynamically normalize the input data during neural network computation, which can only alleviate issues caused by changes in the marginal probability distribution p(x). However, they overlook changes in the conditional probability distribution, which makes them disadvantaged in comparison. In addition, the Beijing and PeMS datasets exhibit significant variations in data scale (refer to Fig. 4). Both RevIN and Dish-TS fail to show improvement in these scenarios, indicating a potential inability to handle drastic distribution drift effectively. In contrast, MemDA shows improved performance, demonstrating its robust prediction logic that can be particularly effective for prediction in challenging environments characterized by substantial concept drift.



Figure 5: Visualization of the variation of the adaptation parameter  $W_t$  on PeMS dataset. Where (a) is from the RM+PM, (b) is from the Meta, and (c) is from the proposed model.

5.4.2 Ablation Test. A series of variant experiments are conducted to verify the effectiveness of each module. The variant with only the original encoder network (GW-Net) is referred to as the Backbone, to which we add RM and PM modules. The Meta variant removes the meta-dynamic layer and directly generates  $W_t$  through a linear network connected to  $E_t$ . Results are shown in Table 5, where we can observe that (1) Simply augmenting the long-term input via memory modules does not always improve performance, with negative effects for Beijing and COVID-CHI datasets. (2) Simply generating  $W_t$  through the embedding of input, as is done in the Meta variant, does not lead to satisfactory results. In contrast, the MemDA evaluates drift by embedding pairs and generates  $W_t$  accordingly, demonstrating a proven adaptation capability and the effectiveness of the meta-dynamic layer. To interpret how MemDA copes with drift, we visualize the fusion parameter  $W_t$  in Fig. 5. Specifically, we show the solidified W(the variant of RM+PM) in (a),  $W_t$  generated by the Simple Meta module (the variant of RM+PM+Meta) in (b), and  $W_t$  of MemDA in (c). The visualization results demonstrate that the MemDA module enables the adaptation model to be aware of changes in data distribution and to make proactive adjustments to the model parameters in response to different concepts at different times. In contrast, the solidified model can only maintain the prediction logic that fits the training set, and the Simple Meta variant, although making some adjustments in response to different concepts, demonstrates a weak adaptation capability that cannot effectively track drift.



**Figure 6: Sensitivity Analysis** 

5.4.3 Sensitivity Analysis. We experimented the effects of Pattern Memory size L and dimension D of MemDA on the PeMS dataset. The results are presented in Fig. 6(a,b). The model achieves optimal performance with a memory size of 32. This optimal point represents a balance between storing enough prototype patterns to accurately detect out-of-distribution instances and the ability to generalize by discarding noise. Similarly, optimal performance can also only be achieved with a suitable memory dimension.

We further tested the effect of look-back days *K*. As a comparison, we constructed a Plain model, in which naive uses multiple branches with a linear fusion layer to process the input segments from different days (same as Fig. 3 (a)). The results are shown in Fig. 6(c). MemDA demonstrates a continuous improvement of the effect with increasing look-back days, while the Plain model fluctuates and does not show significant trends. This demonstrates that dealing with concept drift is not a simple matter of adding inputs. And even sometimes the increase in look-back days isn't beneficial (Plain model 1 to 2). As the distribution of data changes over time, a model that heavily relies on past data could be misled by outdated information that no longer applies to the current. This is a risk when increasing input without an adaptive mechanism.

A comparison of the running speed of the Plain and MemDA is shown in Fig. 6(d). It demonstrates that the computational pressure of the Plain model shows an exponential increase as the number of backtracking days increases, and it takes more than 10 times more time to look back at a week. While MemDA enhanced by Replay Memory presents an efficient ability of slow linear increase MemDA: Forecasting Urban Time Series with Memory-based Drift Adaptation

CIKM '23, October 21-25, 2023, Birmingham, United Kingdom.



Figure 7: Case Study. (a) Average speed change of Beijing test set. (b) The daily error of Backbone and MemDA model. (c,d) Density error gray ribbon of Backbone/MemDA. (e,f) The model prediction results of two roads at different time periods in the test set.

in computation time, which is crucial for enhancing model effects in many real-time scenarios.

#### 5.5 Generalizability Evaluation

The properties of MemDA enable it to be easily integrated with existing urban data forecasting models. We utilize the backbone of three widely used models, MTGNN, GMAN, and GW-Net as the encoder to test the encoder-agnostic performance of the proposed model. The results are presented in Table 6, which demonstrates the generalizability and applicability of MemDA across various models.

**Table 6: Encoder-agnostic Performance Evaluation** 

Paalshana		Beijing			Electricity	7
Dackbolle	RMSE	MAE	MAPE	RMSE	MAE	MAPE
MTGNN	10.976	5.120	19.928%	96.475	51.562	15.410%
+MemDA	9.411	4.379	16.923%	69.381	35.416	11.517%
$\Delta\%$	14.26%	14.48%	15.08%	28.08%	31.31%	25.26%
GMAN	11.555	6.448	21.941%	138.436	81.285	26.263%
+MemDA	10.005	6.011	19.041%	91.030	49.400	15.072%
$\Delta\%$	13.41%	6.78%	13.22%	34.24%	39.23%	42.61%
GW-Net	7.754	3.564	11.916%	97.087	51.684	16.120%
+MemDA	6.720	3.192	9.913%	67.413	34.814	12.186%
$\Delta\%$	13.37%	10.43%	16.81%	30.56%	32.64%	24.40%

## 6 CASE STUDY

We do case studies from both macro and micro perspectives. Fig. 7 (a,b,c,d) shows the city-wide average speed of the Beijing test dataset, the daily error time series of the Backbone (GW-Net) and MemDA model, the daily error density gray ribbon distribution of the Backbone and MemDA model, respectively. As the average road speed in Beijing dwindles, traffic conditions progressively worsen. This downturn significantly disrupts the predictability of traffic patterns, triggering a substantial decline in the performance of the

Backbone model. This is reflected in the model's increasing mean prediction error, with a considerable spread ranging from low to high errors. In stark contrast, the incorporation of MemDA into the Backbone manages to keep the error distribution tightly controlled and centered on the lower values.

A microscopic analysis of the performance was focused on two representative sample roads selected from the initial, intermediate, and final phases of the test set. The zoomed-in views are shown in Fig. 7(e,f). As can be seen, both roads with different speed patterns undergo significant pattern shifts over time (marked with yellow boxes), and the Backbone model just barely works near the training periods. However, as the concept drifts, Backbone's predictions stray further from the actual values. In essence, the Backbone model appears to be memorizing speed patterns rather than predicting them. The introduction of MemDA to it, however, imparts a remarkable transformation to the Backbone, enabling it to track the changes in data patterns and always maintain robustness after drift.

# 7 CONCLUSION

In this study, we target the problem of concept drift in urban time series data prediction. A general adaptive urban model leverages a dual-memory module, and a meta-dynamic network is designed to enable self-configuration. Extensive experiments demonstrate the effectiveness of the proposed framework. In the future, we aim to explore online approaches for constructing robust models as well as additional strategies for addressing spatial drift.

# ACKNOWLEDGMENTS

This work was supported by Japan Science and Technology Agency (JST) SPRING, Grant Number JPMJSP2108, and the 2022 Tencent Rhino-Bird Research Elite Program. CIKM '23, October 21-25, 2023, Birmingham, United Kingdom.

## REFERENCES

- L Bai, L Yao, C Li, X Wang, and C Wang. 2020. Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting. In 34th Conference on Neural Information Processing Systems.
- [2] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. 2018. Mine: mutual information neural estimation. arXiv preprint arXiv:1801.04062 (2018).
- [3] Siddharth Bhatia, Arjit Jain, Shivin Srivastava, Kenji Kawaguchi, and Bryan Hooi. 2022. MemStream: Memory-Based Streaming Anomaly Detection. In Proceedings of the ACM Web Conference 2022. 610–621.
- [4] Albert Bifet and Ricard Gavalda. 2007. Learning from time-changing data with adaptive windowing. In Proceedings of the 2007 SIAM international conference on data mining. SIAM, 443–448.
- [5] Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Conguri Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, et al. 2020. Spectral temporal graph neural network for multivariate time-series forecasting. In Proceedings of the 34th International Conference on Neural Information Processing Systems. 17766–17778.
- [6] Zhiyong Cui, Kristian Henrickson, Ruimin Ke, and Yinhai Wang. 2019. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. *IEEE Transactions on Intelligent Transportation Systems* 21, 11 (2019), 4883–4894.
- [7] Yuntao Du, Jindong Wang, Wenjie Feng, Sinno Pan, Tao Qin, Renjun Xu, and Chongjun Wang. 2021. Adarnn: Adaptive learning and forecasting of time series. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 402–411.
- [8] Wei Fan, Pengyang Wang, Dongkun Wang, Dongjie Wang, Yuanchun Zhou, and Yanjie Fu. 2023. Dish-TS: A General Paradigm for Alleviating Distribution Shift in Time Series Forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37. 7522–7529.
- Joao Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues. 2004. Learning with drift detection. In *Brazilian symposium on artificial intelligence*. Springer, 286–295.
- [10] Ziv Goldfeld and Yury Polyanskiy. 2020. The information bottleneck problem and its applications in machine learning. *IEEE Journal on Selected Areas in Information Theory* 1, 1 (2020), 19–38.
- [11] Marta C Gonzalez, Cesar A Hidalgo, and Albert-Laszlo Barabasi. 2008. Understanding individual human mobility patterns. *nature* 453, 7196 (2008), 779–782.
- [12] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 922–929.
- [13] Chuanmin Jia, Shiqi Wang, Xinfeng Zhang, Shanshe Wang, and Siwei Ma. 2017. Spatial-temporal residue network based in-loop filter for video coding. In 2017 IEEE Visual Communications and Image Processing (VCIP). IEEE, 1–4.
- [14] Renhe Jiang, Zekun Cai, Zhaonan Wang, Chuang Yang, Zipei Fan, Quanjun Chen, Kota Tsubouchi, Xuan Song, and Ryosuke Shibasaki. 2021. DeepCrowd: A deep model for large-scale citywide crowd density and flow prediction. *IEEE Transactions on Knowledge and Data Engineering* (2021).
- [15] Renhe Jiang, Xuan Song, Dou Huang, Xiaoya Song, Tianqi Xia, Zekun Cai, Zhaonan Wang, Kyoung-Sook Kim, and Ryosuke Shibasaki. 2019. Deepurbanevent: A system for predicting citywide crowd dynamics at big events. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2114–2122.
- [16] Renhe Jiang, Zhaonan Wang, Yudong Tao, Chuang Yang, Xuan Song, Ryosuke Shibasaki, Shu-Ching Chen, and Mei-Ling Shyu. 2023. Learning Social Metaknowledge for Nowcasting Human Mobility in Disaster. In *Proceedings of the ACM Web Conference 2023*. 2655–2665.
- [17] Renhe Jiang, Zhaonan Wang, Jiawei Yong, Puneet Jeph, Quanjun Chen, Yasumasa Kobayashi, Xuan Song, Shintaro Fukushima, and Toyotaro Suzumura. 2023. Spatio-temporal meta-graph learning for traffic forecasting. In *Proceedings* of the AAAI Conference on Artificial Intelligence, Vol. 37. 8078–8086.
- [18] Renhe Jiang, Du Yin, Zhaonan Wang, Yizhuo Wang, Jiewen Deng, Hangchen Liu, Zekun Cai, Jinliang Deng, Xuan Song, and Ryosuke Shibasaki. 2021. Dl-traff: Survey and benchmark of deep learning models for urban traffic prediction. In Proceedings of the 30th ACM international conference on information & knowledge management. 4515–4525.
- [19] Imen Khamassi, Moamar Sayed-Mouchaweh, Moez Hammami, and Khaled Ghédira. 2018. Discussion and review on evolving data streams and concept drift adapting. *Evolving systems* 9, 1 (2018), 1–23.
- [20] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. 2004. Detecting change in data streams. In Proceedings of the Thirtieth international conference on Very large data bases-Volume 30. 180–191.
- [21] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. 2021. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*.

- [22] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. 2018. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st* international ACM SIGIR conference on research & development in information retrieval. 95–104.
- [23] Wendi Li, Xiao Yang, Weiqing Liu, Yingce Xia, Jiang Bian, et al. 2022. DDG-DA: Data Distribution Generation for Predictable Concept Drift Adaptation. In Proceedings of the AAAI Conference on Artificial Intelligence.
- [24] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In International Conference on Learning Representations.
- [25] Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. 2022. Non-stationary Transformers: Exploring the Stationarity in Time Series Forecasting. In Advances in Neural Information Processing Systems.
- [26] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. 2018. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering* 31, 12 (2018), 2346–2363.
- [27] Sijie Ruan, Jie Bao, Yuxuan Liang, Ruiyuan Li, Tianfu He, Chuishi Meng, Yanhua Li, Yingcai Wu, and Yu Zheng. 2020. Dynamic public resource allocation based on human mobility prediction. Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies 4, 1 (2020), 1–22.
- [28] Shun-Yao Shih, Fan-Keng Sun, and Hung-yi Lee. 2019. Temporal pattern attention for multivariate time series forecasting. *Machine Learning* 108, 8 (2019), 1421– 1441.
- [29] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. Advances in neural information processing systems 26 (2013).
- [30] Xuan Song, Quanshi Zhang, Yoshihide Sekimoto, and Ryosuke Shibasaki. 2014. Prediction of human emergency behavior and their mobility following largescale disaster. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. 5–14.
- [31] Naftali Tishby, Fernando C Pereira, and William Bialek. 2000. The information bottleneck method. arXiv preprint physics/0004057 (2000).
  [32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones,
- [32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. Advances in neural information processing systems 30 (2017).
- [33] Zhaonan Wang, Renhe Jiang, Hao Xue, Flora D Salim, Xuan Song, and Ryosuke Shibasaki. 2022. Event-Aware Multimodal Mobility Nowcasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36. 4228–4236.
- [34] Gerhard Widmer and Miroslav Kubat. 1996. Learning in the presence of concept drift and hidden contexts. *Machine learning* 23, 1 (1996), 69–101.
- [35] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 2020. Connecting the dots: Multivariate time series forecasting with graph neural networks. In Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. 753–763.
- [36] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Graph wavenet for deep spatial-temporal graph modeling. In Proceedings of the 28th International Joint Conference on Artificial Intelligence. 1907–1913.
- [37] Mingxing Xu, Wenrui Dai, Chunmiao Liu, Xing Gao, Weiyao Lin, Guo-Jun Qi, and Hongkai Xiong. 2020. Spatial-temporal transformer networks for traffic flow forecasting. arXiv preprint arXiv:2001.02908 (2020).
- [38] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, and Zhenhui Li. 2019. Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In Proceedings of the AAAI conference on artificial intelligence, Vol. 33. 5668–5675.
- [39] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.
- [40] Xiaoyu You, Mi Zhang, Daizong Ding, Fuli Feng, and Yuanmin Huang. 2021. Learning to learn the future: Modeling concept drifts in time series prediction. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 2434–2443.
- [41] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In Proceedings of the 27th International Joint Conference on Artificial Intelligence. 3634–3640.
- [42] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Thirty-first AAAI conference on* artificial intelligence.
- [43] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. 2019. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems* 21, 9 (2019), 3848–3858.
- [44] Peng Zhao, Le-Wen Cai, and Zhi-Hua Zhou. 2020. Handling concept drift via model reuse. Machine Learning 109, 3 (2020), 533–568.
- [45] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. 2020. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI* conference on artificial intelligence, Vol. 34. 1234–1241.