Prediction of traffic state variability with an integrated model-based and data-driven Bayesian framework

Xinyue Wu[a], Andy H.F. Chow[a*]
[a]Systems Engineering, City University of Hong Kong
Kowloon Tong, Hong Kong SAR, China

Wei Ma[b], William H.K. Lam [b]
[b]Civil and Environmental Engineering, The Hong Kong Polytechnic University
Hung Hom, Hong Kong SAR, China

S.C. Wong[c]
[c]Civil Engineering, The University of Hong Kong
Pokfulam, Hong Kong SAR, China

November 29, 2024

Abstract

Deriving statistical description of uncertainties associated with prediction of traffic states is essential to development of reliability-based intelligent transportation systems. This paper presents a Bayesian learning approach framework for predicting evolution of both traffic states and the associated variability. The proposed framework ensures the interpretability and stability of the predictions with an underlying state space model, and captures sophisticated dynamics of traffic variability via a data-driven recurrent neural network component. By maintaining the filtering structure in the specialized neural network component, the proposed integrated model overcomes the key limitations of deep learning systems by improving the data efficiency and providing interpretability. The framework is trained with a multivariate Gaussian negative log-likelihood loss function for quantifying both model and stochastic uncertainties. It is implemented and tested with actual traffic data collected from a Hong Kong Strategic Route. The case study shows that the proposed prediction framework can simultaneously retain the interpretability of the results while capture the complex dynamics of the evolution of traffic variability with the recurrent neural network component. This study contributes to the development of reliability-based intelligent transportation systems through the use of advanced statistical modeling and deep learning methods.

Keywords: stochastic prediction, Bayesian learning, recurrent neural network, temporal differences, automatic vehicle identification

[*] corresponding author: andychow@cityu.edu.hk

1 Introduction

With the complexity and uncertainties encountered in real life, deriving the statistical description of uncertainties associated with the prediction of traffic states is essential to the development of reliability-based intelligent transportation systems (R-ITS) (Zhou et al., 2019). Capturing simultaneously the dynamic and stochastic nature of traffic flow makes the prediction a difficult task and it turns the prediction tasks into a sophisticated modeling and computational process (Zhong et al., 2017b). It also induces the challenge of real-time deployment under which the prediction framework would have to process feeding data and derive plausible results in reasonable computational time.

The majority of previous work on stochastic traffic prediction can be categorized into two groups: model-based algorithms and data-driven approaches (Karlaftis and Vlahogianni, 2011). Model-based approaches rely on assumption and knowledge of the underlying dynamics of traffic flow. Statistical volatility models, such as generalized autoregressive conditional heteroskedasticity (GARCH) models, are the most widely used approaches to deal with variability in traffic systems (Zhang et al., 2014b; Tsekeris and Stathopoulos, 2006; Yang et al., 2009). Kamarianakis et al. (2005) apply the GARCH time series model to represent the dynamic of traffic flow variability using loop detector data. Zhang et al. (2014a) adopt GARCH models for predicting travel times and associated reliability with automatic vehicle identification data. Cheng et al. (2014) presents a dynamic and localized space-time autoregressive approach for modeling network journey times. Nevertheless, it is shown that these model-based time series modeling approaches are computationally expensive to run and inefficient in capturing abrupt changes in data pattern. To address the computational issue, an approach is to adopt a Bayesian filtering framework with an underlying and simplified state space model (Wang and Papageorgiou, 2005; Marinică et al., 2013; Ottaviano et al., 2017). In particular, non-linear variations of Kalman filter approaches are among the most widely used approaches (Wang et al., 2007; Ngoduy and Sumalee, 2010). Guo et al. (2014) adopt an adaptive Kalman filter with a GARCH structure for traffic flow prediction and uncertainty quantification. Chen and Rakha (2014) develop a particle filter approach for travel time prediction using probe vehicle data. Nantes et al. (2016) and Saeedmanesh et al. (2021) present an incremental extended Kalman filter algorithm for arterial traffic prediction using multiple sources. Ngoduy and Sumalee (2010) and Trinh et al. (2022) use a unscented Kalman filter approach to estimate traffic state and the model noise distribution with multiple data sources. Li et al. (2023) apply the filter for real-time path travel time estimation without ground truth known. Bai et al. (2024) propose a state estimation with use of speed-density relationships with multi-resolution data. Li et al. (2024) present an estimation method for multi-class path travel times using multi-source traffic data. Wu et al. (2024) present an alternative estimation method based on a Bayesian data fusion with use of mixture model.

The common limitation of the model-based approaches above is that their performances are sensitive to the underlying model assumptions and the domain knowledge involved during the model formulation process. Nevertheless, the complexity of system dynamics and non-stationary noise make it a challenging task to accurately characterize the temporal prediction noise distributions with pure model-based approaches (Rajamani, 2007).

As an alternative, the data-driven approaches have demonstrated in recent years their ability to extract features from complex traffic dynamic systems given sufficient data sources. These approaches learn the features directly from the data without a full understanding of the underlying physical characteristics (Goodfellow et al., 2016). Common data-driven approaches are developed based on machine learning approaches (Zheng and Su, 2014; Cai et al., 2016; Feng et al., 2018) and various deep learning approaches (Cui et al., 2019; Gu et al., 2019a; Cui et al., 2020a,b; Wang et al., 2020). In the literature, we see Zhong et al. (2017a) which present a journey time prediction framework with abnormal conditions using functional principal component analysis in a machine learning framework. Zhang et al. (2019) propose an attention graph convolutional sequence-to-sequence model to predict traffic network speed. Bogaerts et al. (2020) apply a convolutional neural network (CNN) and long short-term memory (LSTM) for traffic forecast using trajectory data. Zhang et al. (2022) propose an adaptive graph learning algorithm to predict traffic flow and could effectively exploit the hidden correlations of the nodes. To incorporate associated uncertainties in the prediction process, Li et al. (2020) predict both the deterministic and uncertainty of travel time with the hybrid deep belief network with quantile regression. Zhou et al. (2020) propose a variational graph recurrent attention neural networks to capture the ambiguity of the predicted results. Markos et al. (2021) and Zhu et al. (2022) apply the Bayesian deep learning to derive the mean and variance of the prediction distribution for unsupervised GPS trajectory segmentation. Zhuang et al. (2024) present a interval journey time prediction with two-stream deep learning data fusion framework. Without incorporation of any domain knowledge and underlying model assumption, the data-driven approaches operate as a *black-box*. The drawback is their lack of interpretability and generalizability, and the possibility of being overfit with overly sophisticated structure and large number of parameters involved (Zhang and Haghani, 2015).

To overcome these drawbacks, recent studies have started exploring Explainable Artificial Intelligence (XAI) techniques to enhance the interpretability of prediction outputs. Extreme gradient boosting (XGBoost) and SHapley Additive exPlanationsis (SHAP) are the most commonly used techniques to understand the prediction outputs (Parsa et al., 2020). There have been a number of studies investigating the feature importance with the use of XGBoost due to its scalability and efficiency (Sun et al., 2021). Chikaraishi et al. (2020) compare various machine learning models for predicting traffic states during disasters and find random forest and XGBoost could achieve better prediction performance. Kang et al. (2020) combine the empirical dynamic modeling and complex networks with an XGBoost model to predict short-term travel time. Lartey et al. (2021) use XGBoost to predict hourly traffic volume under different traffic conditions. More recently, SHAP has been utilized to provide a deeper understanding of the complex interactions between features. Parsa et al. (2020) apply SHAP to interpret the traffic accident detection results and analyze the importance of individual features. Lee (2023) uses SHAP to select critical links and introduces the XGBoost to predict urban traffic speed. Despite their explanatory power, these explainable data-driven approaches rely on post-hoc explanations to provide insights into predictions while the underlying mechanics are still difficult to interpret (Kenny et al., 2021). Furthermore, explanations generated from XAI-based models have limited generalizability and inconsistent representation across models and datasets (Ali et al., 2023).

The consideration of advantages and disadvantages of the model-based and data-driven approaches calls for their integration with an aim to reduce the dependence on domain knowledge while improve the interpretability and generalizability of the resulting framework simultaneously. We see in the literature Abdi et al. (2012), Ma et al. (2020) and Yang et al. (2021) integrate a model-based Autoregressive Integrated Moving Average (ARIMA) time-series model with an artificial neural network for traffic flow prediction. Hou et al. (2019) apply the fuzzy logic to fuse the outputs obtained from ARIMA and wavelet neural network. Gu et al. (2019b) combine the gated recurrent unit neural network, radial basis function neural network and ARIMA through a Bayesian learning framework for short-term traffic prediction. Sattarzadeh et al. (2023) combines ARIMA, CNN and LSTM for traffic flow prediction. Pan et al. (2024) adopt a Markovian-based model to predict regular patterns and a LSTM to capture residual time series. However, these hybrid methods integrate multiple parallel models and produce the final predictions following an ensemble-based fashion (Guo et al., 2018; Alsolami et al., 2020). Despite the current progress, most work on integrated framework only focus on point prediction and fail to provide uncertainty quantification. Moreover, existing ensemble-based integrated models often result in sophisticated structures and hence become computationally expensive in the training process.

This paper presents an integrated model-based and data-driven traffic speed prediction framework through a Bayesian learning approach. Given the previous estimates and real-time observations of the road network, we focus on predicting the mean traffic speed and quantifying the associated uncertainty (*i.e.* covariance) at the next time step in the short future. The proposed algorithm applies an underlying state space model characterizing the system dynamics, and a data-driven recurrent neural network component for tracking the evolution of the associated state variability. The prediction framework is trained with the use of a multivariate Gaussian negative log-likelihood loss function. We introduce a Monte Carlo (MC) dropout during the training and testing processes to update the structure of the recurrent neural network component with consideration of both model and stochastic uncertainties. The performance of the proposed model is tested with real-world traffic data collected from the Hong Kong Strategic Route. The case study shows that the proposed prediction framework can simultaneously retain the interpretability of the results while capture the complex dynamics of the evolution of traffic variability with the recurrent neural network component. Different from the most explainable traffic prediction models, the proposed model offers interpretability from two aspects: 1) The point predictions generated by the statistical dynamic linear model provide clear interpretations of the parameters associated with traffic input features. 2) Preserving the operation process of filtering algorithms in the data-driven module provides explanations of internal input-output mappings and less abstraction. This study contributes to the development of reliability-based intelligent transportation systems through the use of advanced statistical modeling (*i.e.* dynamic linear model-based Kalman filter) and deep learning methods (*i.e.* recurrent neural network). The modular nature of the state space model enables the adaptation and extension of the framework to accommodate different modeling techniques and improve the model generalization in future studies. In general, it contributes to the state-of-the-art in three aspects: First, it contributes to the reliability-based traffic state prediction with an interpretable integrated model-based and data-driven framework. Second, we consider different sources of uncertainties in the traffic prediction processes constructed with MC dropout through an end-to-end multivariate Gaussian likelihood training loss. Finally, the incorporation of state space model could reduce the dependence on the data, and the recursive nature makes it well-suited for real-time traffic predictions.

The rest of the paper is organized as follows: Section 2 presents the proposed methodology. Section 3 shows the case study with data collected from a selected Hong Kong highway corridor. Finally, Section 4 provides some concluding remarks.

2 Methodology

The proposed framework aims to predict traffic states (including flow, speed, concentration) of interest in a given road network system in vector $\hat{\mathbf{x}}^d_{t|t-1}$ and their associated statistical characteristics $\mathbf{\Sigma}^d_{t|t-1}$ at time step t on day d based on previous estimate $\hat{\mathbf{x}}^d_{t-1|t-1}$ at time step $t-1$ and feeding observations $\mathbf{o}^d_t$. The prediction framework consists of a model-based core and a data-driven state variability prediction module which are to be introduced in the following.

2.1 Model-based core framework

In the model-based core framework, each link of the given road network is first indexed by $n = \{1, 2, \cdots, \mathbf{N}\}$, where $\mathbf{N}$ represents the number of links in the network. Each link is associated with traffic state variable $x_t^d(l_n)$ (e.g., flows, speeds) at a given time t on day d. The state vector $\mathbf{x}_t^d = \left[x_t^d(l_1), x_t^d(l_2), \ldots, x_t^d(l_N)\right]^\top \in \mathbb{R}^{\mathbf{N}\times 1}$ represents the underlying state of the system to be estimated. The discrete-time traffic state dynamics are described by the dynamic linear equation based on the physical laws of the system as:

$$\text{(State equation)} \qquad \mathbf{x}_{t+1}^d = \mathbf{F}_t \mathbf{x}_t^d + \mathbf{w}_{t+1}^d \tag{1}$$

where $\mathbf{w}_t^d$ is the noise or model error vector in which the errors are assumed to follow a Gaussian distribution with a zero mean and a known covariance matrix $\mathbf{Q}_t^d$; $\mathbf{F}_t \in \mathbb{R}^{\mathbf{N}\times\mathbf{N}}$ is the transition space-time covariance matrix in the state equation which is assumed to be estimated and known from historical space-time data (Kwak and Geroliminis, 2020).

We further denote $\mathbf{X}_t^{\mathcal{D}} = [\mathbf{x}_t^{d_1}, \mathbf{x}_t^{d_2}, \ldots, \mathbf{x}_t^{d_{\mathbf{D}}}] \in \mathbb{R}^{\mathbf{N}\times\mathbf{D}}$; $\mathbf{W}_t^{\mathcal{D}} = [\mathbf{w}_t^{d_1}, \mathbf{w}_t^{d_2}, \ldots, \mathbf{w}_t^{d_{\mathbf{D}}}] \in \mathbb{R}^{\mathbf{N}\times\mathbf{D}}$, be the concatenations of all traffic states and associated modeling errors in the system over days $d = 1, 2, ..., \mathbf{D}$. The state equation (1) can now be rewritten as:

$$\mathbf{X}_{t+1}^{\mathcal{D}} = \mathbf{F}_t \mathbf{X}_t^{\mathcal{D}} + \mathbf{W}_{t+1}^{\mathcal{D}}. \tag{2}$$

The dynamic transition matrix $\mathbf{F}_t$ is calibrated by historical training dataset via the least-square method (Kwak and Geroliminis, 2020), in which the objective function of the least-square method is set as:

$$\underset{\mathbf{F}_t}{\text{minimize}}\ \eta\omega^{\mathbf{D}}\|\mathbf{F}_t\|^2 + \left\|\left(\mathbf{X}_{t+1}^{\mathcal{D}} - \mathbf{F}_t\mathbf{X}_t^{\mathcal{D}}\right)\mathbf{\Omega}_{\mathbf{D}}^{\frac{1}{2}}\right\|^2, \tag{3}$$

in which $\mathbf{\Omega}_{\mathbf{D}}$ is the adaptive matrix which is defined as

$$\mathbf{\Omega}_{\mathbf{D}} = \begin{bmatrix} \omega^{\mathbf{D}-1} & 0 & \cdots & 0 \\ 0 & \omega^{\mathbf{D}-2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \omega^0 \end{bmatrix} \in \mathbb{R}^{\mathbf{D}\times\mathbf{D}}.$$

with forgetting factor ω to penalize the historical data with different weights. The parameter η is a regularization parameter, the operator $\|Y\| = \sqrt{\text{tr}(YY^\top)}$ and $\text{tr}(YY^\top)$ is the sum of all diagonal elements of the matrix $YY^\top$.

To infer the states $\mathbf{x}_t^d$ in (1), we also have $\mathbf{M}$ observations, $\mathbf{o}_t^d \in \mathbb{R}^{\mathbf{M}\times 1}$, collected from the site by sensors. These observations are directly related to the system state itself or to other associated state variables. We establish the following observation equation to map the true state variables to the observations:

$$\text{(Observation equation)} \quad \mathbf{o}_{t+1}^{d} = \mathbf{H}_{t+1}\mathbf{x}_{t+1}^{d} + \mathbf{e}_{t+1}^{d} \tag{4}$$

over all days $d = 1, 2, ..., \mathbf{D}$, where $\mathbf{e}_t^d$ is a zero-mean Gaussian observation noise with covariance matrix $\mathbf{R}_t^d$; $\mathbf{H}_t \in \mathbb{R}^{\mathbf{M}\times\mathbf{N}}$ is the observation transition matrix which describes the underlying relationships between the state $\mathbf{x}_t^d$ and observations $\mathbf{o}_t^d$. This allows for flexible integration of data from multiple sources into a cohesive model and helps to determine the weight assigned to each state variable when correcting the observed data.

The value of $\mathbf{H}_t$ can also be estimated and derived from historical data based on the physical laws of the system. Define $\mathbf{O}_t^{\mathcal{D}} = [\mathbf{o}_t^{d_1}, \mathbf{o}_t^{d_2}, \ldots, \mathbf{o}_t^{d_\mathbf{D}}] \in \mathbb{R}^{\mathbf{M}\times\mathbf{D}}$; $\mathbf{E}_t^{\mathcal{D}} = [\mathbf{e}_t^{d_1}, \mathbf{e}_t^{d_2}, \ldots, \mathbf{e}_t^{d_\mathbf{D}}] \in \mathbb{R}^{\mathbf{M}\times\mathbf{D}}$, the observation equation (4) could further be consolidated like (2) as

$$\mathbf{O}_{t+1}^{\mathcal{D}} = \mathbf{H}_{t+1}\mathbf{X}_{t+1}^{\mathcal{D}} + \mathbf{E}_{t+1}^{\mathcal{D}}. \tag{5}$$

The low-complexity Kalman filter could provide the optimal solution to the above Gaussian state space model. Three assumptions are made for the application of Kalman filtering: 1) the dynamic systems are linear, and this assumption can be relaxed by the model linearization techniques for non-linear systems (*e.g.*, extended Kalman filter); 2) the modeling noise $\mathbf{w}_t^d$ and observation noise $\mathbf{e}_t^d$ are white; 3) both noises are Gaussian distributed, *i.e., $\mathbf{w}_t^d \sim \mathcal{N}(0, \mathbf{Q}_t^d)$ and* $\mathbf{e}_t^d \sim \mathcal{N}(0, \mathbf{R}_t^d)$. Following the state space framework established above, we now define $\hat{\mathbf{x}}_{t|t-1}^d$ be the prior estimate and $\hat{\mathbf{x}}_{t|t}^d$ be the posterior estimate of the traffic state vector $\mathbf{x}_t^d$ respectively. Based on the conventional Kalman filter framework, we have (Wang and Papageorgiou, 2005):

$$\text{Predicted a priori state estimate:} \qquad \hat{\mathbf{x}}_{t|t-1}^d = \mathbf{F}_{t-1} \cdot \hat{\mathbf{x}}_{t-1|t-1}^d \tag{6a}$$

$$\text{Innovation:} \qquad \Delta\hat{\mathbf{o}}_t^d = \mathbf{o}_t^d - \mathbf{H}_t \cdot \hat{\mathbf{x}}_{t|t-1}^d \tag{6b}$$

$$\text{Updated a posterior state estimate:} \qquad \hat{\mathbf{x}}_{t|t}^d = \hat{\mathbf{x}}_{t|t-1}^d + \mathbf{K}_t^d \cdot \Delta\hat{\mathbf{o}}_t^d \tag{6c}$$

in which the prior estimate $\hat{\mathbf{x}}_{t|t-1}^d$ is an estimate produced by the underlying state equation model (1), and $\mathbf{K}_t^d \in \mathbb{R}^{\mathbf{N}\times\mathbf{M}}$ is known as the Kalman gain matrix for correcting this prior estimate from $\hat{\mathbf{x}}_{t|t-1}^d$ to posterior estimate $\hat{\mathbf{x}}_{t|t}^d$ with feeding observations. The measure of the estimated accuracy of the state estimate is also tracked by the Kalman filtering with the following equations:

$$\text{Predicted a priori estimate covariance:} \quad \mathbf{\Sigma}_{t|t-1}^d = \mathbf{F}_{t-1} \cdot \mathbf{\Sigma}_{t-1|t-1}^d \cdot \mathbf{F}_{t-1}^\top + \mathbf{Q}_t^d \tag{7a}$$

$$\text{Innovation covariance:} \quad \mathbf{S}_t^d = \mathbf{H}_t \cdot \mathbf{\Sigma}_{t|t-1}^d \cdot \mathbf{H}_t^\top + \mathbf{R}_t^d \tag{7b}$$

$$\text{Updated a posteriori estimate covariance:} \quad \mathbf{\Sigma}_{t|t}^d = \mathbf{\Sigma}_{t|t-1}^d - \mathbf{K}_t^d \cdot \mathbf{S}_t^d \cdot (\mathbf{K}_t^d)^\top \tag{7c}$$

where $\mathbf{\Sigma}_{t|t-1}^d = \mathbb{E}[(\mathbf{x}_t^d - \hat{\mathbf{x}}_{t|t-1}^d)(\mathbf{x}_t^d - \hat{\mathbf{x}}_{t|t-1}^d)^\top]$ and $\mathbf{\Sigma}_{t|t}^d = \mathbb{E}[(\mathbf{x}_t^d - \hat{\mathbf{x}}_{t|t}^d)(\mathbf{x}_t^d - \hat{\mathbf{x}}_{t|t}^d)^\top]$. Here, the optimal value of $\mathbf{K}_t^d$ is found by minimizing the mean-square error vector, which gives:

$$\text{Optimal Kalman Gain:} \qquad \mathbf{K}_t^d = \mathbf{\Sigma}_{t|t-1}^d \cdot \mathbf{H}_t^\top \cdot (\mathbf{S}_t^d)^{-1}. \tag{8}$$

The Kalman gain is constructed based on the uncertainties in the system established by the noise covariance matrices $\mathbf{Q}_t^d$ and $\mathbf{R}_t^d$ (Ngoduy and Sumalee, 2010). The model-based Kalman filtering framework could be carried out in an online fashion by allowing recursive updates with continuous feeding of new observations. Nevertheless, the prediction accuracy of the eventual state and associated variabilities depend on the correctness of the underlying state equation model (Tampère and Immers, 2007) which could be difficult to construct and calibrate due to availability and deployment of on-site sensors as well as the complex dynamics of the underlying traffic state variability.

2.2 Data driven state variability prediction module

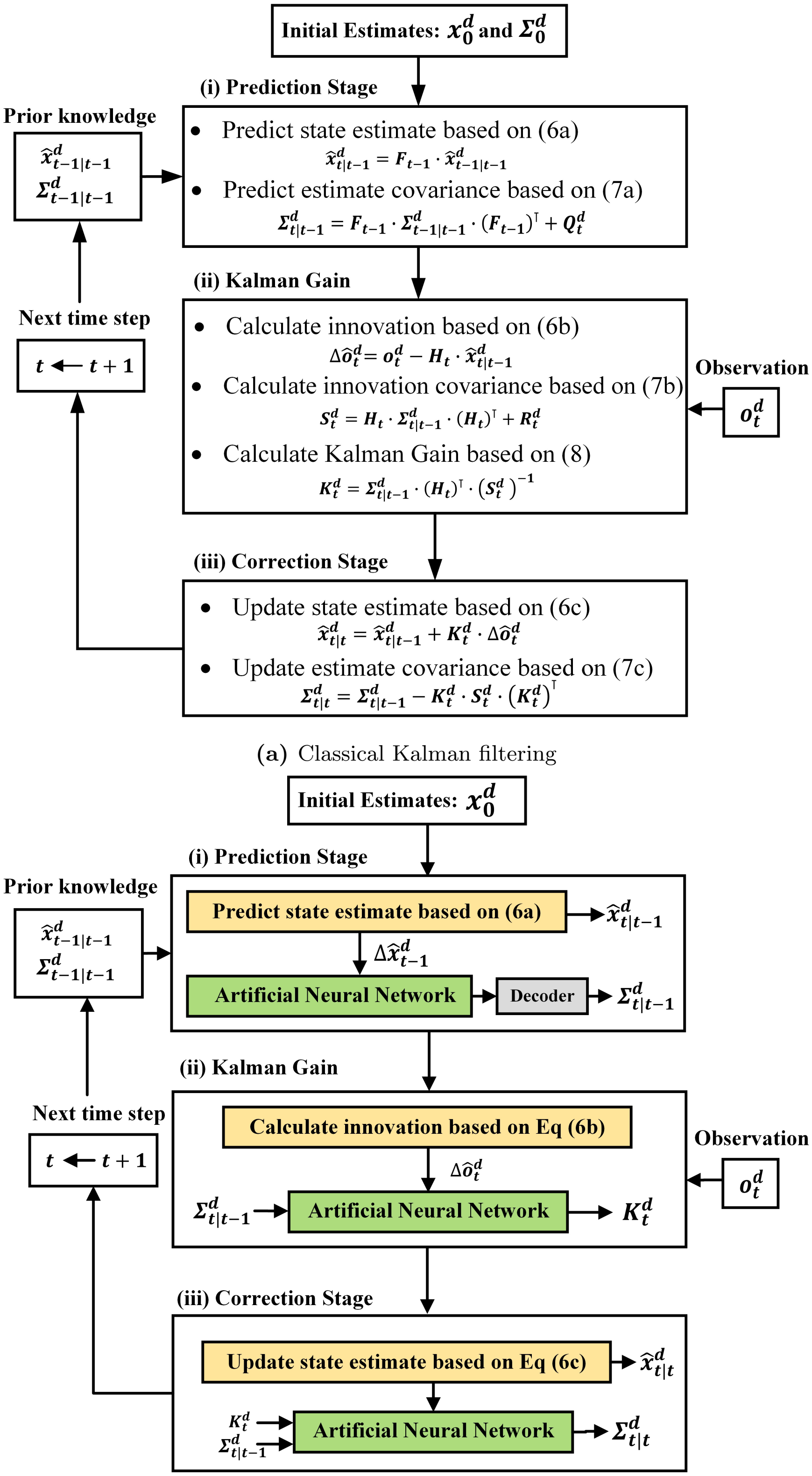
This study aims to incorporate artificial neural networks into the model-based core framework above to address the challenge of estimating the Kalman gain and traffic state variability. Fig. 1 shows and compares the algorithmic structures of the classical model-based Kalman filter prediction framework and the proposed one with incorporation of the data-driven module. As illustrated in Fig. 1a, the computation of state estimates and associated uncertainty measures in the classical Kalman filtering are simultaneous and are integrated via the Kalman gain $\mathbf{K}_t^d$. It motivates us to learn the measurement of uncertainties and $\mathbf{K}_t^d$ directly from the data instead of domain knowledge.

The problem of simply replacing the measurement of uncertainties with artificial neural networks shown in Fig. 1b is that such design requires a large number of neurons to achieve satisfying prediction performance. Therefore, we propose an architecture that strictly follows the operation process of filtering algorithms to reduce the hyperparameters need to be tuned in the data-driven module. In this section, we describe the details of the data-driven module shown in Fig. 2, which consists of three stages following the operation flow of the model-based core.

2.2.1 Feature extraction

In this paper, we assume the transition matrix $\mathbf{F}_t$ and observation matrix $\mathbf{H}_t$ are known but the noise $\mathbf{Q}_t^d$ and $\mathbf{R}_t^d$ are unknown. The unknown $\mathbf{Q}_t^d$ and $\mathbf{R}_t^d$ are to be inferred in the data-driven component in the proposed model via various features incorporated as shown in Fig. 2a. Several temporal difference features are proposed and used for training the neural network data-driven component:

- **Feature 1:** the evolution difference between two consecutive posterior state estimates: $\Delta\tilde{\mathbf{x}}_t^d = \hat{\mathbf{x}}_{t|t}^d - \hat{\mathbf{x}}_{t-1|t-1}^d$. The available feature $\Delta\tilde{\mathbf{x}}_{t-1}^d$ is used at time step t on day d. This feature captures the variability in the state estimates across consecutive time steps in the evolution process, which is directly related to the state covariance $\mathbf{\Sigma}_t^d$.


Initial Estimates: x_0^d and Σ_0^d
(i) Prediction Stage
Prior knowledge
$\hat{x}_{t-1|t-1}^d$
$\Sigma_{t-1|t-1}^d$
Predict state estimate based on (6a)
$\hat{x}_{t|t-1}^d = F_{t-1} \cdot \hat{x}_{t-1|t-1}^d$
Predict estimate covariance based on (7a)
$\Sigma_{t|t-1}^d = F_{t-1} \cdot \Sigma_{t-1|t-1}^d \cdot (F_{t-1})^\top + Q_t^d$
(ii) Kalman Gain
Next time step
$t \leftarrow t+1$
Calculate innovation based on (6b)
$\Delta\hat{o}_t^d = o_t^d - H_t \cdot \hat{x}_{t|t-1}^d$
Calculate innovation covariance based on (7b)
$S_t^d = H_t \cdot \Sigma_{t|t-1}^d \cdot (H_t)^\top + R_t^d$
Calculate Kalman Gain based on (8)
$K_t^d = \Sigma_{t|t-1}^d \cdot (H_t)^\top \cdot (S_t^d)^{-1}$
Observation
o_t^d
(iii) Correction Stage
Update state estimate based on (6c)
$\hat{x}_{t|t}^d = \hat{x}_{t|t-1}^d + K_t^d \cdot \Delta\hat{o}_t^d$
Update estimate covariance based on (7c)
$\Sigma_{t|t}^d = \Sigma_{t|t-1}^d - K_t^d \cdot S_t^d \cdot (K_t^d)^\top$
Initial Estimates: x_0^d
(i) Prediction Stage
Prior knowledge
$\hat{x}_{t-1|t-1}^d$
$\Sigma_{t-1|t-1}^d$
Predict state estimate based on (6a)
$\hat{x}_{t|t-1}^d$
$\Delta\hat{x}_{t-1}^d$
Artificial Neural Network
Decoder
$\Sigma_{t|t-1}^d$
(ii) Kalman Gain
Next time step
$t \leftarrow t+1$
Calculate innovation based on Eq (6b)
$\Delta\hat{o}_t^d$
$\Sigma_{t|t-1}^d$
Artificial Neural Network
K_t^d
Observation
o_t^d
(iii) Correction Stage
Update state estimate based on Eq (6c)
$\hat{x}_{t|t}^d$
K_t^d
$\Sigma_{t|t-1}^d$
Artificial Neural Network
$\Sigma_{t|t}^d$

(a) Classical Kalman filtering

(b) Filter with artificial neural networks

Fig. 1. Algorithmic structure of state prediction filter

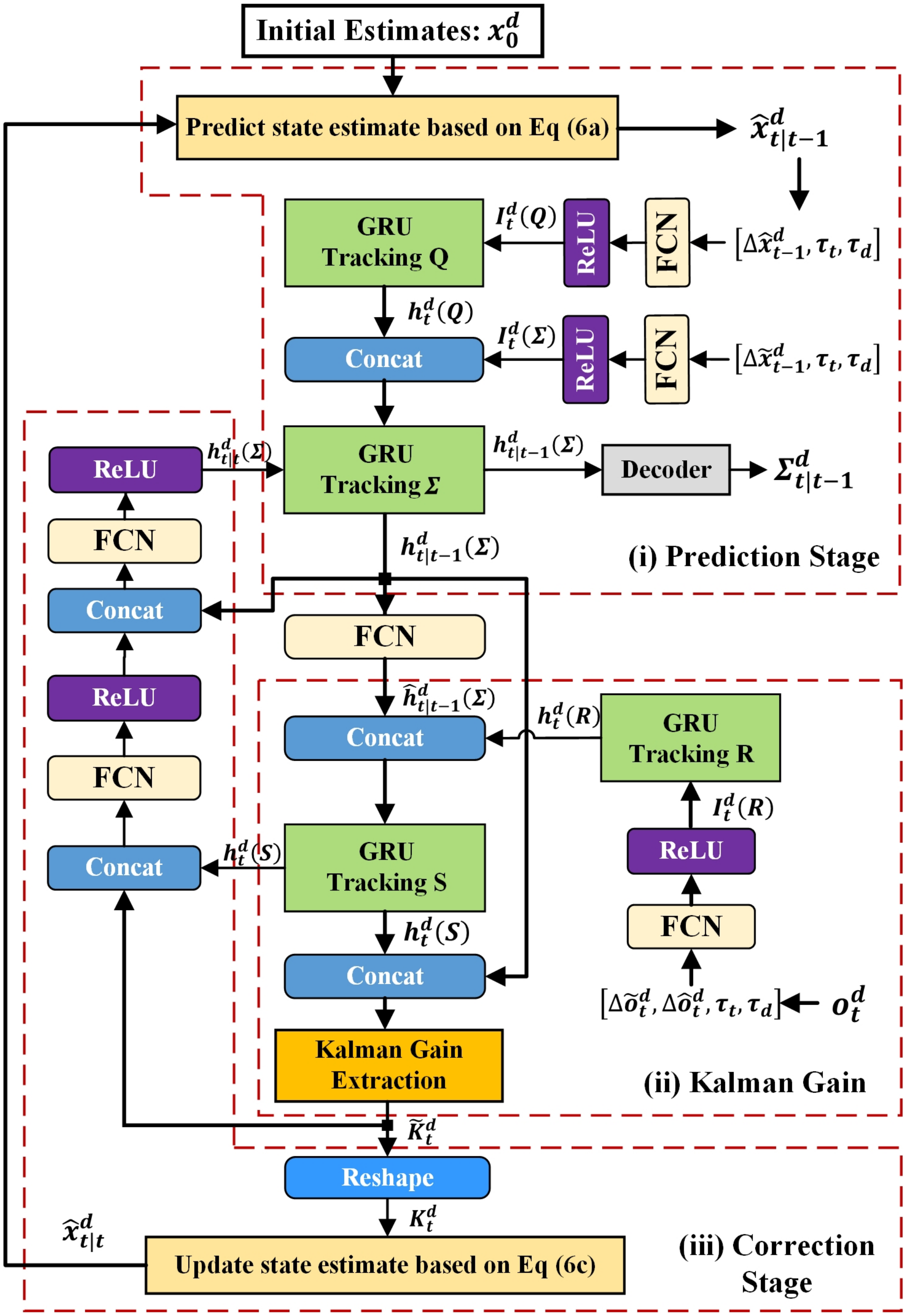


Fig. 2. Overall structure of the proposed framework

- **Feature 2**: the update difference between the posterior state estimate and the prior state estimate: $\Delta\hat{\mathbf{x}}_t^d = \hat{\mathbf{x}}_{t|t}^d - \hat{\mathbf{x}}_{t|t-1}^d$. We also use the available feature $\Delta\hat{\mathbf{x}}_{t-1}^d$ at time step t to capture the statistical information. A larger update difference suggests greater uncertainty in predicting future states of the dynamic system, which is crucial for estimating the model error covariance $\mathbf{Q}_t$.

- **Feature 3:** the evolution difference between two consecutive observations: $\Delta\tilde{\mathbf{o}}_t^d = \mathbf{o}_t^d - \mathbf{o}_{t-1}^d$. This variability is directly related to the measurement noise covariance $\mathbf{R}_t$ as it reflects the inherent noise in the measurement evolution process.

- **Feature 4**: the innovation difference between the predicted observation and the current observation: $\Delta\hat{\mathbf{o}}_t^d = \mathbf{o}_t^d - \hat{\mathbf{o}}_t^d$. This innovation also reflects the alignment between the predictions and the actual observations, and thus should be accounted for in the estimation of measurement noise $\mathbf{R}_t$.

The four features above are selected because they contain statistical information on traffic state uncertainty in the "*predict-correct*" filtering processes. Other intrinsic temporal features, TimeoOfDay $\tau_{(t)}$ ($\tau_{(t)} \in \{0, 1, \cdots, \mathbf{T}-1\}$ and $\mathbf{T}$ is the length of time sequence) and DayOfWeek $\tau_{(d)}$ ($\tau_{(d)} \in \{0, 1, 2, \cdots, 6\}$), are used in conjunction with temporal difference features as inputs since they are direct attributes to the traffic state. We use the fully-connected network (FCN) to extract the features hidden in the data and the Rectified Linear Unit (ReLU) as an activation function to introduce non-linearity. Thus, we have

$$\begin{aligned}
\mathbf{I}_t^d(\mathbf{Q}) &= \mathrm{ReLU}\Big(\boldsymbol{\theta}^{(\mathbf{Q})} \cdot \mathrm{concat}[\Delta\hat{\mathbf{x}}_t^d, \tau_{(t)}, \tau_{(d)}] + \boldsymbol{c}^{(\mathbf{Q})}\Big) \\
\mathbf{I}_t^d(\boldsymbol{\Sigma}) &= \mathrm{ReLU}\Big(\boldsymbol{\theta}^{(\boldsymbol{\Sigma})} \cdot \mathrm{concat}[\Delta\tilde{\mathbf{x}}_t^d, \tau_{(t)}, \tau_{(d)}] + \boldsymbol{c}^{(\boldsymbol{\Sigma})}\Big) \\
\mathbf{I}_t^d(\mathbf{R}) &= \mathrm{ReLU}\Big(\boldsymbol{\theta}^{(\mathbf{R})} \cdot \mathrm{concat}[\Delta\tilde{\mathbf{o}}_t^d, \Delta\hat{\mathbf{o}}_t^d, \tau_{(t)}, \tau_{(d)}] + \boldsymbol{c}^{(\mathbf{R})}\Big)
\end{aligned} \tag{9}$$

where $\mathbf{I}_t^d(*)$ is the output of the temporal difference features and intrinsic features used for tracking corresponding second-order statistics (*i.e.,* $\mathbf{Q}_t^d$, $\mathbf{R}_t^d$, $\boldsymbol{\Sigma}_t^d$ and $\mathbf{S}_t^d$) at time t on day d, $\boldsymbol{\theta}^{(*)}$ and $\boldsymbol{c}^{(*)}$ are the learnable weight and bias respectively. The influence of the intrinsic features is discussed in Section 3.2 through the numerical study. Our empirical results suggest that good combinations is temporal difference features with TimeoOfDay intrinsic feature.

As illustrated in Fig. 2, the unknown second-order statistics in Kalman filtering algorithms are tracked by independent Gated Recurrent Unit (GRU) cells. The GRU cell is selected as the hidden state could retain the learned information in the previous time step, which aligns with the recursive nature of Kalman filtering. Moreover, GRU has fewer parameters and less complexity compared to the LSTM, making it more computationally efficient in capturing temporal features (Yamak et al., 2019). Fig. 3 shows the basic structure of the GRU cell. Each GRU cell has two gating units, including an update gate $\boldsymbol{z}_t^d$ and a reset gate $\boldsymbol{r}_t^d$. The reset gate $\boldsymbol{r}_t^d$ is integrated with the current input $\boldsymbol{i}_t^d$ and hidden state from the previous time step $\boldsymbol{h}_{t-1}^d$ to compute the *candidate* hidden state $\tilde{\boldsymbol{h}}_t^d$. The formulation of GRU is presented in Eq (10).

$$\begin{aligned} \text{(Update gate)} \qquad & \boldsymbol{z}_t^d = \sigma\big(\boldsymbol{V}^{(z)}\boldsymbol{i}_t^d + \boldsymbol{U}^{(z)}\boldsymbol{h}_{t-1}^d + \boldsymbol{c}^{(z)}\big) \\ \text{(Reset gate)} \qquad & \boldsymbol{r}_t^d = \sigma\big(\boldsymbol{V}^{(r)}\boldsymbol{i}_t^d + \boldsymbol{U}^{(r)}\boldsymbol{h}_{t-1}^d + \boldsymbol{c}^{(r)}\big) \\ \text{(Candidate hidden state)} \qquad & \tilde{\boldsymbol{h}}_t^d = \tanh\Big(\boldsymbol{V}^{(h)}\boldsymbol{i}_t^d + \boldsymbol{U}^{(h)}\big(\boldsymbol{r}_t^d \odot \boldsymbol{h}_{t-1}^d\big) + \boldsymbol{c}^{(h)}\Big) \end{aligned} \tag{10}$$

where $\boldsymbol{V}^{(*)}$, $\boldsymbol{U}^{(*)}$ are weight matrices and $\boldsymbol{c}^{(*)}$ are bias parameters, the symbol $\odot$ is the element-wise product, and $\sigma(\cdot)$ and $\tanh(\cdot)$ are the sigmoid function and hyperbolic tangent function.

Given the candidate hidden state $\tilde{\boldsymbol{h}}_t^d$, the layer output vector $\boldsymbol{h}_{t-1}^d$ at of GRU cell at the current time step is defined as:

$$\text{(Hidden state)} \qquad \boldsymbol{h}_t^d = \big(1 - \boldsymbol{z}_t^d\big) \odot \tilde{\boldsymbol{h}}_t^d + \boldsymbol{z}_t^d \odot \boldsymbol{h}_{t-1}^d. \tag{11}$$

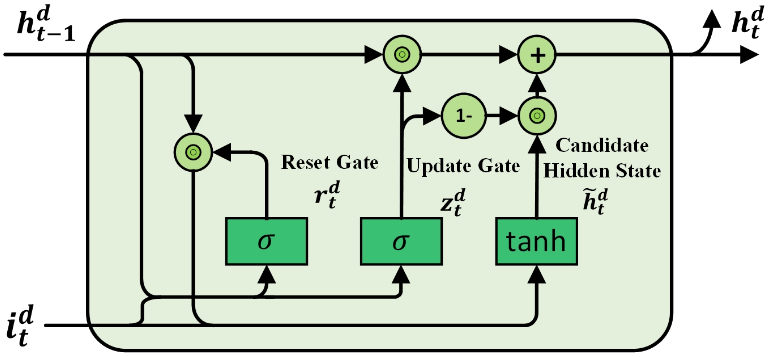


Fig. 3. Structure of a basic GRU cell

In our method, the unknown $\mathbf{Q}_t^d \in \mathbb{R}^{\mathbf{N}\times\mathbf{N}}$ and $\mathbf{R}_t^d \in \mathbb{R}^{\mathbf{M}\times\mathbf{M}}$ in the state space model are learned from the extracted features $\mathbf{I}_t^d(\mathbf{Q})$ and $\mathbf{I}_t^d(\mathbf{R})$ with the use of GRU cell, and the output of the GRU cell can be written as:

$$\begin{aligned} \boldsymbol{h}_t^d(\mathbf{Q}) &= \mathrm{GRU}\Big(\mathbf{I}_t^d(\mathbf{Q}), \boldsymbol{h}_{t-1}^d(\mathbf{Q})\Big), \\ \boldsymbol{h}_t^d(\mathbf{R}) &= \mathrm{GRU}\Big(\mathbf{I}_t^d(\mathbf{R}), \boldsymbol{h}_{t-1}^d(\mathbf{R})\Big) \end{aligned} \tag{12}$$

where $\boldsymbol{h}_t^d(\mathbf{Q})$ and $\boldsymbol{h}_t^d(\mathbf{R})$ denote the learned output of the GRU cells tracking $\mathbf{Q}_t^d$ and $\mathbf{R}_t^d$ respectively, and GRU$(\cdot)$ denotes the GRU operation described in Eq.(10) and (11). Therefore, the number of features in the hidden state of GRU cells tracking $\mathbf{Q}_t^d \in \mathbb{R}^{\mathbf{N}\times\mathbf{N}}$ and $\mathbf{R}_t^d \in \mathbb{R}^{\mathbf{M}\times\mathbf{M}}$ is set to be $\mathbf{N}^2$ and $\mathbf{M}^2$.

2.2.2 Architecture of prediction framework

We now present the structure of the proposed framework by combining the model-based core framework with the feature extraction GRU cell as illustrated in Fig. 2a. In each time step t on day d, the proposed model predicts $\hat{\mathbf{x}}^d_{t|t-1}$ and its statistical estimate covariance $\mathbf{\Sigma}^d_{t|t-1}$ following three stages.

(i) Prediction Stage

In the prediction stage, the prior predictor for the current state $\hat{\mathbf{x}}^d_{t|t-1}$ is obtained from the a priori state equation (Eq.(6a)) as in the basic filtering framework discussed in Section 2.1. Different from the classical Kalman filtering setting, the estimate covariance $\mathbf{\Sigma}^d_t$ is to be learned through the GRU cell instead of being computed explicitly. Following Eq.(7a), $\mathbf{Q}^d_t$ is used to produce $\mathbf{\Sigma}^d_{t|t-1}$, thus we concatenate the learned $\boldsymbol{h}^d_t(\mathbf{Q})$ and $\mathbf{I}^d_t(\mathbf{\Sigma})$ as the final input feature of the GRU cell tracking $\mathbf{\Sigma}^d_t$. To make the output feature $\boldsymbol{h}^d_{t|t-1}(\mathbf{\Sigma})$ keep the same size with $\mathbf{\Sigma}^d_t$, the number of features in the hidden state of GRU cells tracking $\mathbf{\Sigma}^d_t$ is set to be $\mathbf{N}^2$.

The extracted features $\boldsymbol{h}^d_{t|t-1}(\mathbf{\Sigma})$ are then fed into a FCN decoder which yields the predicted a prior estimate covariance $\mathbf{\Sigma}^d_{t|t-1}$ at current time step. Fig 4 shows the structure of the FCN decoder. The output of the decoder can be written as

$$\begin{aligned}\boldsymbol{\alpha}^d_t &= \exp\left(\boldsymbol{\theta}^{(\boldsymbol{\alpha})}\boldsymbol{h}^{\mathbf{Q}}_{t|t-1} + \boldsymbol{c}^{(\boldsymbol{\alpha})}\right)\\ \boldsymbol{\beta}^d_t &= \tanh\left(\boldsymbol{\theta}^{(\boldsymbol{\beta})}\boldsymbol{h}^{\mathbf{Q}}_{t|t-1} + \boldsymbol{c}^{(\boldsymbol{\beta})}\right)\end{aligned} \tag{13}$$

where $\exp(\cdot)$ denotes the exponential activation function. The $\mathbf{\Sigma}^d_{t|t-1}$ is estimated from the vector $\boldsymbol{\alpha}^d_t \in \mathbb{R}^{\mathbf{N}}_+$ and $\boldsymbol{\beta}^d_t \in \mathbb{R}^{\frac{\mathbf{N}\times(\mathbf{N}-1)}{2}}$ using Cholesky decomposition $\mathbf{\Sigma}^d_{t|t-1} = \mathbf{A}^d_t(\mathbf{A}^d_t)^\top$ where $\mathbf{A}^d_t$ is a lower triangular matrix. Vector $\boldsymbol{\alpha}^d_t$ and $\boldsymbol{\beta}^d_t$ are used as the diagonal and off-diagonal elements of the matrix $\mathbf{A}^d_t$.

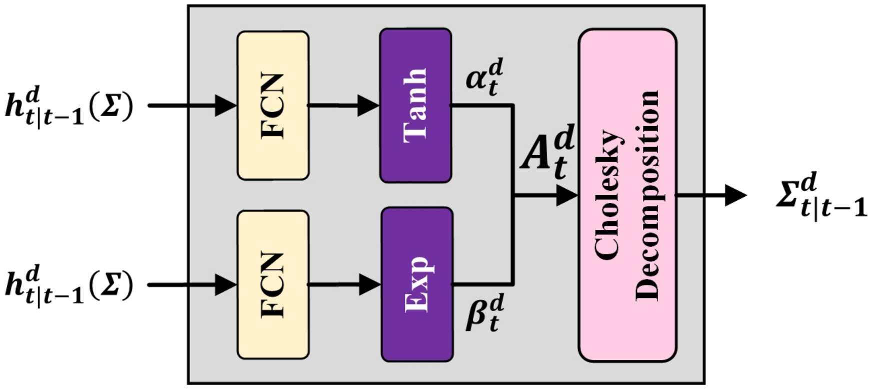


Fig. 4. Covairance decoder block

(ii) Calculation of gain

When the current new observation $\mathbf{o}^d_t$ is available, the innovation $\Delta\hat{\mathbf{o}}^d_t$ can be obtained using Eq.(6b) as the basic filtering framework. To calculate $\mathbf{S}^d_t$ based on Eq.(7b), the dimension of learned features $\boldsymbol{h}^d_{t|t-1}(\boldsymbol{\Sigma})$ is transformed into $\mathbf{M}^2$ to ensure compatibility with the dimensionality of $\mathbf{R}^d_t$ using a FCN and ReLU activation. Then, the learned $\boldsymbol{h}^d_t(\mathbf{R})$ and $\hat{\boldsymbol{h}}^d_{t|t-1}(\boldsymbol{\Sigma})$ are concatenated and further passed to the last GRU cell tracking the innovation covariance $\mathbf{S}^d_t$ whose number of features in the hidden state is $\mathbf{M}^2$. Therefore, the output of GRU cell tracking $\mathbf{S}^d_t$ can be written as:

$$\begin{aligned}\hat{\boldsymbol{h}}^d_{t|t-1}(\boldsymbol{\Sigma}) =&\text{ReLU}\Big(\boldsymbol{\theta}^{(\boldsymbol{\Sigma})}_1 \boldsymbol{h}^d_{t|t-1}(\boldsymbol{\Sigma}) + \boldsymbol{c}^{(\boldsymbol{\Sigma})}_1\Big)\\ \boldsymbol{h}^d_t(\mathbf{S}) =& \text{ GRU}\Big(\text{concat}\big[\boldsymbol{h}^d_t(\mathbf{R}), \hat{\boldsymbol{h}}^d_{t|t-1}(\boldsymbol{\Sigma})\big], \boldsymbol{h}^d_{t-1}(\mathbf{S})\Big).\end{aligned} \tag{14}$$

As $\boldsymbol{\Sigma}^d_{t|t-1}$ and $\mathbf{S}^d_t$ are involved in computing $\mathbf{K}^d_t$ shown in Eq. (8), these two learned features $\hat{\boldsymbol{h}}^d_{t|t-1}(\boldsymbol{\Sigma})$ and $\boldsymbol{h}^d_t(\mathbf{S})$ are concatenated and processed through a multi-layer Kalman Gain extraction block shown in Fig. 5. During the feature extraction, dropout (Srivastava et al., 2014) is introduced to avoid overfitting. The number of neurons in the ouput FCN layer of the Kalman Gain extraction block is set to be $\mathbf{N} \cdot \mathbf{M}$. Therefore, the output of the Kalman Gain extraction block can be written as

$$\tilde{\mathbf{K}}^d_t = \boldsymbol{\theta}^{(\mathbf{K})}_2 \Bigg[\boldsymbol{\gamma}^{(\mathbf{K})} \odot \text{ReLU}\Big(\boldsymbol{\theta}^{(\mathbf{K})}_1 \cdot \text{concat}\big[\boldsymbol{h}^d_{t|t-1}(\boldsymbol{\Sigma}), \boldsymbol{h}^d_t(\mathbf{S})\big] + \boldsymbol{c}^{(\mathbf{K})}_1\Big)\Bigg] + \boldsymbol{c}^{(\mathbf{K})}_2 \tag{15}$$

where $\boldsymbol{\gamma}^{(\mathbf{K})}$ is a binary mask matrix and each element is independently sampled from a Bernoulli distribution with probability p.

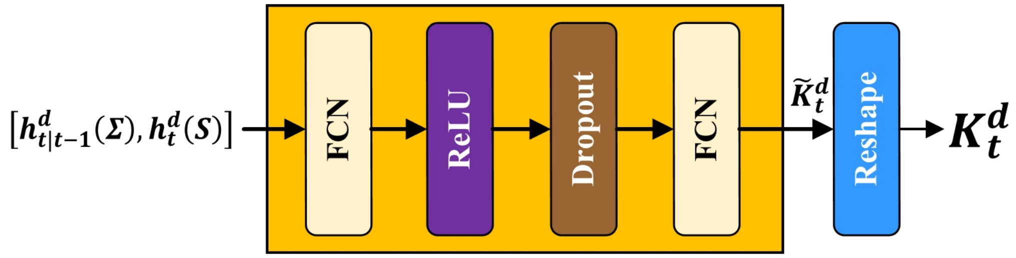


Fig. 5. Kalman Gain extraction

(iii) Correction Stage

In the correction stage, the output of Kalman Gain extraction block $\tilde{\mathbf{K}}^d_t$ is reshaped into the matrix $\mathbf{K}^d_t \in \mathbb{R}^{\mathbf{N}\times\mathbf{M}}$ such that the learned $\mathbf{K}^d_t$ can be used to update the state estimate $\hat{\mathbf{x}}^d_{t|t}$ following the basic Kalman filtering shown in Eq (6c).

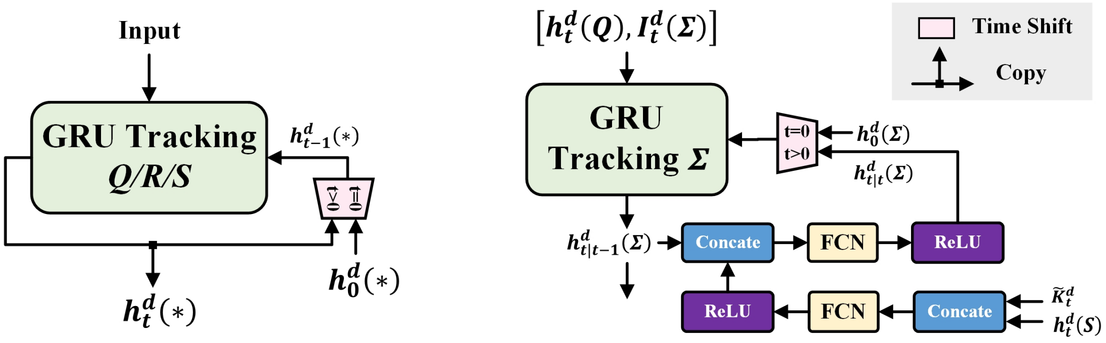
Based on Eq. (7c), $\mathbf{K}^d_t$, $\mathbf{S}^d_t$ and $\boldsymbol{\Sigma}^d_{t|t-1}$ are in turn used to update the posterior covariance $\boldsymbol{\Sigma}^d_{t|t}$. To follow this formulation, the concatenation of $\tilde{\mathbf{K}}^d_t$ and $\boldsymbol{h}^d_t(\mathbf{S})$ are mapped into the estimated $\mathbf{K}^d_t \cdot \mathbf{S}^d_t \cdot (\mathbf{K}^d_t)^\top \in \mathbb{R}^{\mathbf{N}\times\mathbf{N}}$ through an output FCN with $\mathbf{N}^2$ neurons. Then, the outputs, together with learned $\boldsymbol{h}^d_{t|t-1}(\boldsymbol{\Sigma})$, are used to estimate the features in $\boldsymbol{\Sigma}^d_{t|t}$ (*i.e.* $\boldsymbol{h}^d_{t|t}(\boldsymbol{\Sigma})$) using a FCN.

Finally, the hidden state of the GRU cell tracking $\mathbf{\Sigma}_t^d$ is updated with the learned $\boldsymbol{h}_{t|t}^d(\mathbf{\Sigma})$. This forward update process is presented in Eq (16).

$$\boldsymbol{h}_{t|t}^d(\mathbf{\Sigma}) = \mathrm{ReLU}\Big[\boldsymbol{\theta}_3^{(\mathbf{\Sigma})} \cdot \mathrm{concat}\Big[\mathrm{ReLU}\big(\boldsymbol{\theta}_2^{(\mathbf{\Sigma})} \cdot \mathrm{concat}[\tilde{\mathbf{K}}_t^d, \boldsymbol{h}_t^d(\mathbf{S})] + \boldsymbol{c}_2^{(\mathbf{\Sigma})}\big), \boldsymbol{h}_{t|t-1}^d(\mathbf{\Sigma})\Big] + \boldsymbol{c}_3^{(\mathbf{\Sigma})}\Big]$$

$$\boldsymbol{h}_{t|t-1}^d(\mathbf{\Sigma}) = \mathrm{GRU}\Big(\mathrm{concat}\big[\mathbf{I}_t^d(\mathbf{\Sigma}), \mathbf{h}_t^d(\mathbf{Q})\big], \boldsymbol{h}_{t-1|t-1}^d(\mathbf{\Sigma})\Big) \tag{16}$$

Fig. 6 compares the update process of the hidden state in GRU cells in the proposed framework. Different from other GUR cells, GRU cell tracking $\mathbf{\Sigma}_t^d$ utilizes the updated posterior covariance features $\boldsymbol{h}_{t|t}^d(\mathbf{\Sigma})$ instead of directly using the output of GRU cells from the previous time step. Such design allows to track $\mathbf{\Sigma}_{t|t-1}^d$ and $\mathbf{\Sigma}_{t|t}^d$ simultaneously with only one GRU cell and stick to the "*predict-correct*" principle of Kalman filtering. The proposed deep GRU-aided filtering model is designed strictly following the operation flow of Kalman filtering instead of being a pure incomprehensible black box. The interconnections between different components are tailored following the formulation of basic Kalman filtering, which makes it benefit from both model-based and data-driven approaches.



(a) GRU cell tracking *Q*, *R* and *S*

(b) GRU cell tracking *Σ*

Fig. 6. The update process of the hidden state in GRU cells

2.3 Training, validation, and testing

We now present the training, validation, and testing procedures for the prediction framework.

2.3.1 Loss function for training

The training is to be conducted with a set of labeled data, in which it is assumed that the probability of making an actual observation being $\mathbf{x}_t^d \in \mathbb{R}^{\mathbf{N}\times 1}$ given the associated prediction is $\hat{\mathbf{x}}_{t|t-1}^d$ at time t on day d follows a multivariate Gaussian distribution as:

$$p(\mathbf{x}_t^d|\hat{\mathbf{x}}_{t|t-1}^d) = \frac{1}{\sqrt{(2\pi)^n|\mathbf{\Sigma}_{t|t-1}^d|}} \times \exp\Big[-\frac{1}{2}\big(\mathbf{x}_t^d - \hat{\mathbf{x}}_{t|t-1}^d\big)^\top (\mathbf{\Sigma}_{t|t-1}^d)^{-1}\big(\mathbf{x}_t^d - \hat{\mathbf{x}}_{t|t-1}^d\big)\Big]. \tag{17}$$

The state and estimate covariance are trained simultaneously by maximizing the likelihood of Eq.(17), following which the parameters in the deep neural network Θ are determined by minimizing the negative log-likelihood of the multivariate Gaussian distribution model:

$$-\log p(\mathbf{x}_t^d|\hat{\mathbf{x}}_{t|t-1}^d) \propto \frac{1}{2}\big(\mathbf{x}_t^d - \hat{\mathbf{x}}_{t|t-1}^d\big)^\top (\mathbf{\Sigma}_{t|t-1}^d)^{-1}\big(\mathbf{x}_t^d - \hat{\mathbf{x}}_{t|t-1}^d\big) + \frac{1}{2}\ln|\mathbf{\Sigma}_{t|t-1}^d|. \quad (18)$$

It is observed that Eq.(18) would be sensitive with respect to $\mathbf{\Sigma}_{t|t-1}^d$. Therefore, a regularization term of $\mathbf{\Sigma}_{t|t-1}^d$ is added to Eq.(18) with which we formulate the following loss function for training and validation:

$$\begin{aligned}\ell_\Theta(\mathbf{x}_t^d, \hat{\mathbf{x}}_{t|t-1}^d, \mathbf{\Sigma}_{t|t-1}^d) &= \lambda\Big[\frac{1}{2}\big(\mathbf{x}_t^d - \hat{\mathbf{x}}_{t|t-1}^d\big)^\top (\mathbf{\Sigma}_{t|t-1}^d)^{-1}\big(\mathbf{x}_t^d - \hat{\mathbf{x}}_{t|t-1}^d\big) + \frac{1}{2}\ln|\mathbf{\Sigma}_{t|t-1}^d|\Big] \\ &+ (1-\lambda)\Big(\ln|\mathbf{\Sigma}_{t|t-1}^d|\Big),\end{aligned} \quad (19)$$

where $\lambda \in [0,1]$ is the hyperparameter that can be tuned during the validation process. In the training phase, we assume each day in the training dataset has the same length of time steps. By defining $\mathbf{T}$ be the length of the time sequence of interest, the loss value on the sequence of $\mathbf{T}$ frames in a day is calculated as:

$$\mathcal{L}_\Theta = \frac{1}{\mathbf{T}}\sum_{t=1}^{\mathbf{T}} \ell_\Theta(\mathbf{x}_t^d, \hat{\mathbf{x}}_{t|t-1}^d, \mathbf{\Sigma}_{t|t-1}^d). \quad (20)$$

Given the loss function (20), the parameters Θ are to be solved by Adam stochastic gradient search method (Kingma and Ba, 2014).

2.3.2 Measures of uncertainty

It is a challenging problem to provide an accurate uncertainty prediction in the applications of dynamic systems relying on probabilistic filters. The fixed noise used in the conventional approach might lead to unreliable predictions for systems with dynamic and correlated errors (Russell and Reale, 2021). Therefore, *heteroscedastic* noise uncertainty is considered in the proposed model. The source of the heteroscedastic uncertainty includes two types: one is *model* uncertainty, and the other is *stochastic* uncertainty (Punzo and Montanino, 2020). We assume that the model and stochastic uncertainties are independent, thus the overall uncertainty can be calculated as the sum of the model and stochastic uncertainty, as detailed below.

A. Model uncertainty: Model uncertainty refers to the uncertainty resulting from the model parameters in the prediction framework. The most popular and practical approach to estimate the model uncertainty is the MC dropout considering the trade-off between training time and accuracy (Russell and Reale, 2021). As a regularization technique, MC dropout not only prevents overfitting in neural networks but also enables multiple forward passes through the network with different subsets of active neurons (Gal and Ghahramani, 2016). These characteristics make it easy to incorporate into the existing neural networks for the extraction of uncertainty estimates. In this paper, we follow the procedure of MC dropout method proposed by Gal and Ghahramani (2016) and train the proposed prediction framework using the random dropout with a specific dropout probability. We generate $\boldsymbol{B}$ (dropout iterations) different prior state $\hat{\mathbf{x}}_{t|t-1}^{d,b}$ and estimate covariance $\mathbf{\Sigma}_{t|t-1}^{d,b}$ predictions, where $b = 1, 2, \cdots, \boldsymbol{B}$, given the fixed input from a ensemble model $E^b(\cdot)$ at each time step. The model uncertainty is then estimated as:

$$(\mathbf{\Sigma}_{t|t-1}^{d})^{\mathrm{m}} = \frac{1}{\boldsymbol{B}} \sum_{b=1}^{\boldsymbol{B}} \left(\hat{\mathbf{x}}_{t|t-1}^{d,b} - \bar{\mathbf{x}}_{t|t-1}^{d}\right)\left(\hat{\mathbf{x}}_{t|t-1}^{d,b} - \bar{\mathbf{x}}_{t|t-1}^{d}\right)^{\top}, \tag{21}$$

where $\bar{\mathbf{x}}_{t|t-1}^{d} = \sum_b \hat{\mathbf{x}}_{t|t-1}^{d,b}/\boldsymbol{B}$ is the mean of all state predictions.

B. Stochastic uncertainty: Stochastic uncertainty arises due to the randomness inherent in the dynamic system. It is associated with the intrinsic process and measurement noises in the filtering model. The prior estimate covariance $\mathbf{\Sigma}_{t|t-1}^{d}$ obtained from the FCN decoder by the proposed algorithm describes the stochastic uncertainty. During the inference phase, the stochastic uncertainty is computed by averaging all prior estimate covariance obtained from MC dropout method:

$$(\mathbf{\Sigma}_{t|t-1}^{d})^{\mathrm{s}} = \bar{\mathbf{\Sigma}}_{t|t-1}^{d} = \frac{1}{\boldsymbol{B}} \sum_{b=1}^{\boldsymbol{B}} \mathbf{\Sigma}_{t|t-1}^{d,b}. \tag{22}$$

The overall uncertainty $(\mathbf{\Sigma}_{t|t-1}^{d})^{\mathrm{all}}$ evaluated in the training process as the sum of model uncertainty and stochastic uncertainty with respect the predicted mean $\bar{\mathbf{x}}_{t|t-1}^{d}$ as:

$$\begin{aligned}(\mathbf{\Sigma}_{t|t-1}^{d})^{\mathrm{all}} &= (\mathbf{\Sigma}_{t|t-1}^{d})^{\mathrm{m}} + (\mathbf{\Sigma}_{t|t-1}^{d})^{\mathrm{s}} \\ &= \underbrace{\frac{1}{\boldsymbol{B}} \sum_{b=1}^{\boldsymbol{B}} \left(\hat{\mathbf{x}}_{t|t-1}^{d,b} - \bar{\mathbf{x}}_{t|t-1}^{d}\right)\left(\hat{\mathbf{x}}_{t|t-1}^{d,b} - \bar{\mathbf{x}}_{t|t-1}^{d}\right)^{\top}}_{\text{model uncertainty}} + \underbrace{\frac{1}{\boldsymbol{B}} \sum_{b=1}^{\boldsymbol{B}} \mathbf{\Sigma}_{t|t-1}^{d,b}}_{\text{stochastic uncertainty}}\end{aligned} \tag{23}$$

2.4 Performance metrics for validation and testing

The performance of the trained prediction framework is to be evaluated from both point estimation and interval estimation metrics in the validation and testing phases.

1) Point estimation metrics: four commonly used point estimation metrics are used, including mean absolute percentage error (MAPE), mean absolute error (MAE), root mean square error (RMSE) and coefficient of determination (R^2):

$$\text{MAE} = \frac{1}{\mathbf{D} \times \mathbf{T} \times \mathbf{N}} \sum_{d=1}^{\mathbf{D}} \sum_{t=1}^{\mathbf{T}} \sum_{n=1}^{\mathbf{N}} \left| \hat{x}_{t|t-1}^{d}(l_n) - x_t^d(l_n) \right| \tag{24a}$$

$$\text{MAPE} = \frac{1}{\mathbf{D} \times \mathbf{T} \times \mathbf{N}} \sum_{d=1}^{\mathbf{D}} \sum_{t=1}^{\mathbf{T}} \sum_{n=1}^{\mathbf{N}} \frac{\left| \hat{x}_{t|t-1}^{d}(l_n) - x_t^d(l_n) \right|^2}{x_t^d(l_n)} \tag{24b}$$

$$\text{RMSE} = \sqrt{\frac{\sum_d \sum_t \sum_n \left[\hat{x}_{t|t-1}^{d}(l_n) - x_t^d(l_n) \right]^2}{\mathbf{D} \times \mathbf{T} \times \mathbf{N}}} \tag{24c}$$

$$R^2 = 1 - \frac{\sum_d \sum_t \sum_n \left[x_t^d(l_n) - \hat{x}_{t|t-1}^{d}(l_n) \right]^2}{\sum_d \sum_t \sum_n \left[x_t^d(l_n) - \bar{x} \right]^2} \tag{24d}$$

where $\hat{x}_{t|t-1}^{d}(l_n)$ is the predicted value obtained from $\bar{\mathbf{x}}_{t|t-1}^{d}$, $x_t^d(l_n)$ is the ground truth value and $\bar{x} = \frac{1}{\mathbf{D}\times\mathbf{T}\times\mathbf{N}} \sum_d \sum_t \sum_n x_t^d(l_n)$ is the mean of the ground truth value, $\mathbf{T}$ is the length of time sequence in a day, $\mathbf{D}$ is the number of days in validation or testing dataset and $\mathbf{N}$ is the number of links in the network.

2) Interval estimation metrics: We further have the prediction interval coverage probability (PICP) and mean prediction interval width (MPIW) as the uncertainty prediction metrics for evaluating the variance estimates. The diagonal elements in the covariance matrix $(\mathbf{\Sigma}_{t|t-1}^{d})^{\text{all}}$ describe the variances of the corresponding prediction states, thus the 95% confidence interval of the prediction state $\hat{x}_{t|t-1}^{d}(l_n)$ can be estimated by:

$$\hat{x}_{t|t-1}^{d}(l_n) \pm 1.96 \times \sqrt{(\mathbf{\Sigma}_{t|t-1}^{d})^{\text{all}}(n,n)}. \tag{25}$$

We use $\hat{U}_t^d(l_n)$ and $\hat{L}_t^d(l_n)$ to denote the upper and lower bound of the confidence interval derived from the speed prediction $\hat{x}_{t|t-1}^{d}(l_n)$ and corresponding predictive uncertainty $(\mathbf{\Sigma}_{t|t-1}^{d})^{\text{all}}(n,n)$ respectively. We define a binary variable:

$$k_t^d(l_n) = \begin{cases} 1, & \text{if } x_t^d(l_n) \in [\hat{L}_t^d(l_n), \hat{U}_t^d(l_n)] \\ 0, & \text{otherwise} \end{cases} \tag{26}$$

which denotes whether the ground truth value is captured by the confidence interval.

Then, the PICP can be calculated as

$$\text{PICP} = \frac{\sum_{d=1}^{D} \sum_{t=1}^{T} \sum_{n=1}^{N} k_t^d(l_n)}{\mathbf{D} \times \mathbf{T} \times \mathbf{N}} \tag{27}$$

where PICP $\in [0,1]$ and a satisfying PICP should closer to the corresponding confidence interval (*i.e.*, 95%).

In addition to the PICP, we use MPIW to quantify the mean width of the confidence interval which could capture the ground truth value in order to measure the reliability of the estimates, which is defined as

$$\mathrm{MPIW} = \frac{1}{\mathbf{D} \times \mathbf{T} \times \mathbf{N}} \sum_{d=1}^{\mathbf{D}} \sum_{t=1}^{\mathbf{T}} \sum_{n=1}^{\mathbf{N}} [\hat{U}_t^d(l_n) - \hat{L}_t^d(l_n)] \cdot k_t^d(l_n). \tag{28}$$

Moreover, we use the calibration figure (Kuleshov et al., 2018) to determine whether the uncertainty is predicted reliably and accurately. Intuitively, the empirical frequency should match the corresponding confidence interval for a well-qualified uncertainty. Mathematically, we select $\mathcal{I}$ confidence levels $0 \leq c_1 \leq \cdots \leq c_i \leq \cdots c_{\mathcal{I}} \leq 1$ and compute each corresponding empirical frequency

$$\hat{\boldsymbol{p}}(c_i) = \frac{\left| \left\{ x_t^d(l_n) \middle| \mathcal{F}_\Phi \left[x_t^d(l_n) \right] \leq c_i \right\} \right|}{\mathbf{D} \times \mathbf{T} \times \mathbf{N}} \tag{29}$$

where $\mathcal{F}_\Phi \left[x_t^d(l_n) \right]$ denotes the cumulative distribution function (CDF) of the predicted state $\hat{x}_{t|t-1}^d(l_n)$ and predictive uncertainty $(\boldsymbol{\Sigma}_{t|t-1}^d)^{\mathrm{all}}$, which can be expressed as follows:

$$\mathcal{F}_\Phi \left[x_t^d(l_n) \right] = \boldsymbol{\Phi} \left(\frac{\left[x_t^d(l_n) - \hat{x}_{t|t-1}^d(l_n) \right]}{\sqrt{(\boldsymbol{\Sigma}_{t|t-1}^d)^{\mathrm{all}}(n,n)}} \right). \tag{30}$$

The calibration figure visualizes the comparison by plotting $\left\{ \left(c_i, \hat{\boldsymbol{p}}(c_i) \right) \right\}_{i=1}^{\mathcal{I}}$. Therefore, a better uncertainty prediction would be closer to the diagonal line with a slope of 1. To qualitatively illustrate the forecast calibration quality, we use the difference between the empirical frequency and corresponding confidence interval, *i.e.* the *expected calibration error* (ECE), as the numerical score (Guo et al., 2017):

$$\mathrm{ECE} = \sum_{i=1}^{\mathcal{I}} \Big(c_i - \hat{\boldsymbol{p}}(c_i) \Big)^2. \tag{31}$$

Finally, to further evaluate the quality of covariance matrix prediction, we compare the empirical frequency of the squared Mahalanobis distance ξ_t^d with the theoretical probability distribution, *i.e.,* χ^2 distribution with n degree of freedom, due to the Gaussian assumption in loss function. The Mahalanobis distance ξ_t^d is calculated as $\xi_t^d = \left(\mathbf{x}_t^d - \bar{\mathbf{x}}_{t|t-1}^d \right)^\top \left[\left(\boldsymbol{\Sigma}_{t|t-1}^d \right)^{\mathrm{all}} \right]^{-1} \left(\mathbf{x}_t^d - \bar{\mathbf{x}}_{t|t-1}^d \right)$. This comparison is crucial for validating the assumption of multivariate normality in the loss function, and significant deviations from the expected χ^2 distribution might suggest non-normality or model misspecification.

3 Case study

We now present the case study for testing the performance of the proposed state prediction method.

3.1 Scenario settings

We adopt real traffic data collected from a 9-km Hong Kong strategic route section. The strategic route starts from the Island Eastern Corridor and ends at the exit of Western Harbour Crossing (shown in Fig. 7). Along the route section, five Autoscope Video Detection Stations are deployed at different locations to collect local spot speed data through video detection and analysis every two minutes. Besides, a pair of Automatic Vehicle Identification (AVI) detectors are set up at the entry and exit of the route section respectively. It could automatically recognize the plate numbers of the vehicles from the cameras by image processing technique and derive corresponding travel time by matching the identifications between two consecutive camera sites.

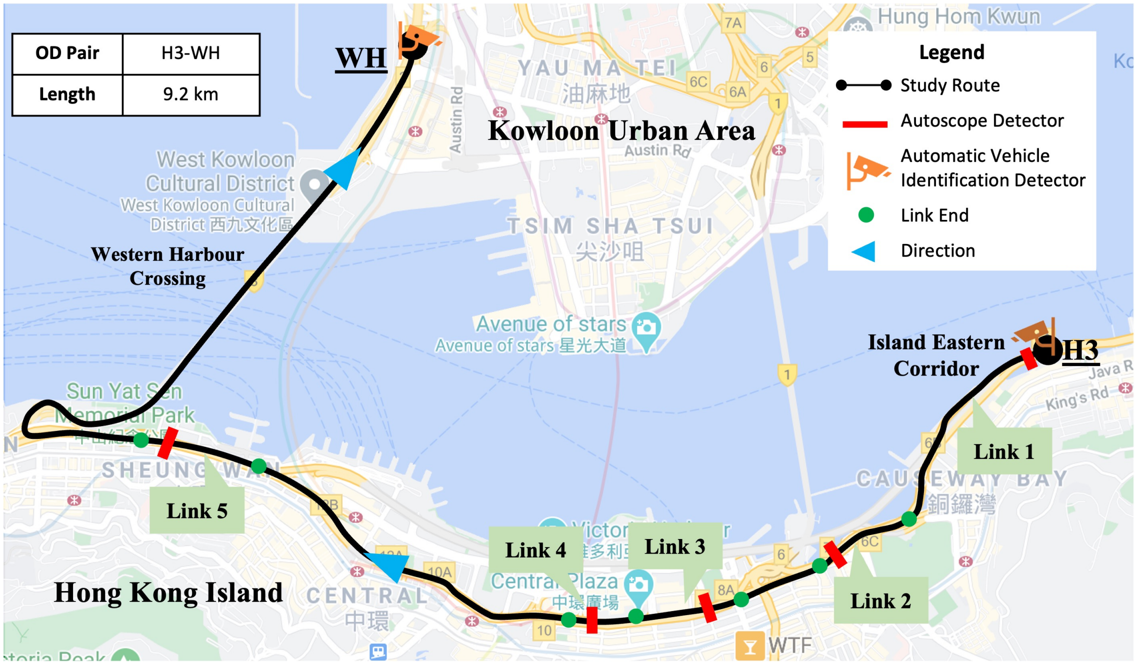


Fig. 7. Case study corridor - Hong Kong strategic route

The data were collected and processed over an 8-month period of weekdays from 1 August 2017 to 25 March 2018, 07:00 to 21:00. However, the travel time data derived from AVI detectors contains various disturbances and outliers due to the detouring, mismatching and parking problems. The Spatio-Temporal Density-Based Spatial Clustering of Applications with Noise method (ST-DBSCAN) (Birant and Kut, 2007) is adopted to remove invalid data points. All the filtered AVI travel time dataset and Autoscope speed dataset are aggregated or interpolated

Table 1: Statistics summary of the Autoscope and AVI datasets

Statistics (Unit: km/h)	Link 1	Link 2	Link 3	Link 4	Link 5	Entire route
Mean	51.5	48.8	43.7	46.2	79.9	46.5
Median	57.1	55.5	53.3	57.6	80.6	47.9
Mode	56.6	58.0	55.7	73.9	81.6	56.0
Standard Deviation	15.6	16.8	18.3	25.5	4.4	13.7
Range	74.5	78.7	75.3	94.9	84.7	85.6

into 5-min intervals, and we have 26,026 records used in the case study. To construct the state equation, the travel time records in AVI dataset are converted to the average speed. Therefore, the speed vector $\mathbf{x}_t^d$ can be rewritten as $\mathbf{x}_t^d = \left[x_t^d(l_1), x_t^d(l_2), \cdots, x_t^d(l_5), \bar{x}_t^d(l)\right]^\top$, where $x_t^d(l_n)$ represents the local link speed and $\bar{x}_t^d(l)$ represents the average speed of the whole route section transferred from travel time data collected from AVI system. Table 1 summarizes the descriptive statistics of traffic speed data collected from Autoscope and AVI datasets. For evaluation, validated ground truths of the traffic speed are also provided by the Journey Time Indication System by the Hong Kong Transport Department. The processed dataset is divided into three subsets: data collected from 1 August 2017 to 31 January 2018 are used as the training dataset; data collected from 1 February 2018 to 28 February 2018 are used as the validation dataset and data collected from 1 March 2018 to 25 March 2018 are used as the testing dataset.

3.2 Model setup and training

We implemented our model based on PyTorch and Python 3.9 on a machine with 10-core CPU, 16-core GPU and 16 GB unified memory. In this case study, the dimension of the true state vector $\mathbf{x}_t^d$ and observation state vector $\mathbf{o}_t^d$ are both 6 (*i.e.,* $N = M = 6$). Therefore, the hidden size of GRU cells tracking $\mathbf{Q}_t^d$, $\mathbf{R}_t^d$, $\mathbf{S}_t^d$ and $\mathbf{\Sigma}_t^d$ are all 36 and the number of stacked GRU layers are set to be 1. Following Lakshminarayanan et al. (2017), the MC dropout samples $\boldsymbol{B}$ is set to be 5 considering the balance between the computational efficiency and prediction accuracy. The training dataset is used to train the hyperparameters in the model-based and data-driven components simultaneously. Grid searches of the hyperparameter space of the network are conducted to find the most appropriate network settings. Following a series of validation tests with data collected on 1 February 2018 - 28 February 2018 as aforementioned, the parameter controlling the size of the input and output fully connected layers is set to be 10. The proposed framework is learned through the Adam stochastic gradient algorithm with a learning rate of 0.0001 and a weight decay of 0.00001.

The key design hyperparameters should be discussed in detail are listed as follows: (A) the regularization term η and forgetting term ω in the state equation; (B) the weight λ in the loss function; (C) the dropout rate p; (D) the time sequence length $\mathbf{T}$ in the loss function; (E) the selection of intrinsic temporal features.

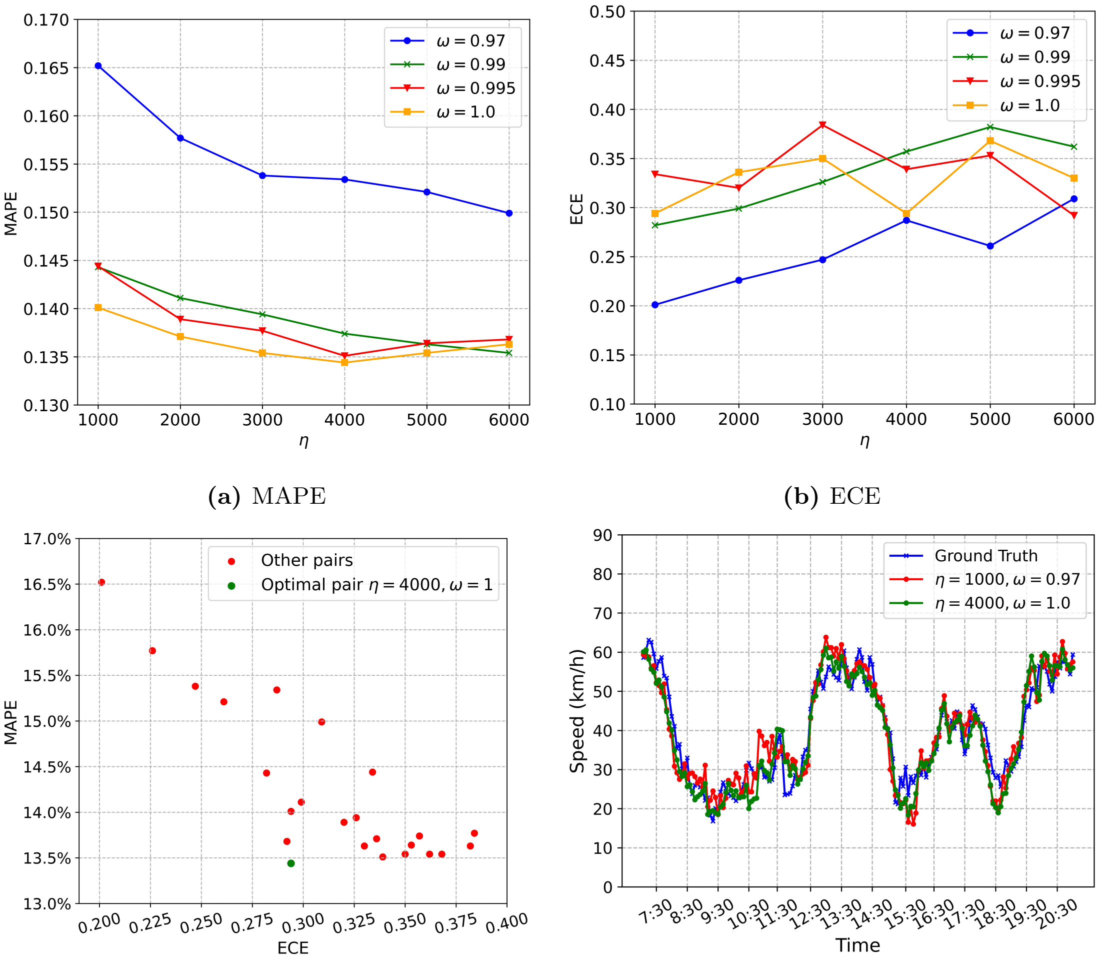
A. Choice of hyperparameters

To investigate the effects of the model-based module on the prediction performance, we explore different pairs of hyperparameters of regularization term η and forgetting term ω. Empirically, we set the λ in the loss function as 0.5, the dropout rate p as 0.5 and the time sequence length $\mathbf{T}$ as 168. The optimal pair is selected based on the prediction results on the validation dataset. To select the optimal pair of hyperparameters, we train the model with different combinations of (η, ω) from the sets $\eta \in \{1000, 2000, 3000, 4000, 5000, 6000\}$ and $\omega \in \{0.97, 0.99, 0.995, 1.0\}$, and evaluate the prediction results on the validation dataset. Fig. 8 compares the performance of different combination pairs of η and ω, where smaller values of MAPE and ECE indicate better point and interval prediction accuracy. It can be observed from Fig. 8c that the pair $(4000, 1.0)$ achieves the best performance considering the balance between the point and interval prediction.

Fig. 8a and Fig. 8b compare the changes of MAPE and ECE with the increase of η under different ω settings. The validation results demonstrate that the prediction results are sensitive to the model-based module settings. However, both MAPE and ECE are more sensitive to the forgetting factor ω than to the regularization term η. Specifically, we observe that increasing ω from 0.97 to 1.0 under the same value of η significantly reduces both MAPE and ECE. We speculate that this may be due to the fact that a larger value of ω makes the training dataset insufficient. Therefore, the value of ω should be carefully selected based on the size of the training dataset. All these results suggest that a reliable state space model has a positive effect on improving the prediction accuracy. To further evaluate this finding, we compare the point prediction of the average speed of the entire route using the best combination $(4000, 1.0)$ and the worst combination $(1000, 0.97)$ shown in Fig. 8d. It is clearly shown in the figure that the predicted average speed is more accurate with the best combination, especially during the time period 10:30-12:30. These figures demonstrate that the proposed integrated model is capable of learning from the underlying dynamics in the state space model. This finding also supports the application of the proposed model with different state space dynamics for other traffic prediction tasks.

B. Choice of weight λ

Fig. 9 provides an illustration of how the hyperparameter λ can be determined through the validation process. The figure compares the point estimation metric MAPE with the interval estimation metric ECE as λ increases. The results show that increasing λ reduces ECE



(a) MAPE

(b) ECE

(c) Scatter plot

(d) Average speed prediction

Fig. 8. Sensitivity analysis of hyperparameters in the state equation. (a) Performance comparison on MAPE. (b) Performance comparison on ECE. (c) Performance comparison between MAPE and ECE with different combinations of η and ω. (d) Average speed prediction of the entire route on 01 Feb. 2018 with the best combination $\eta = 4000, \omega = 1$ and the worst combination $\eta = 1000, \omega = 0.97$.

but increases MAPE. This behaviour is consistent with the goal of preventing the explosion of uncertainty by the introduction of λ in the loss function. It also indicates that λ is a crucial hyperparameter in the loss function that should be carefully selected based on the specific traffic prediction accuracy requirements for different tasks. As observed from the figure, there is no substantial reduction in MAPE for values of λ greater than 0.8. Consequently, we conclude from this validation process that a value of 0.8 would be an appropriate choice for λ in this case, as it balances the performance for both point and interval predictions.

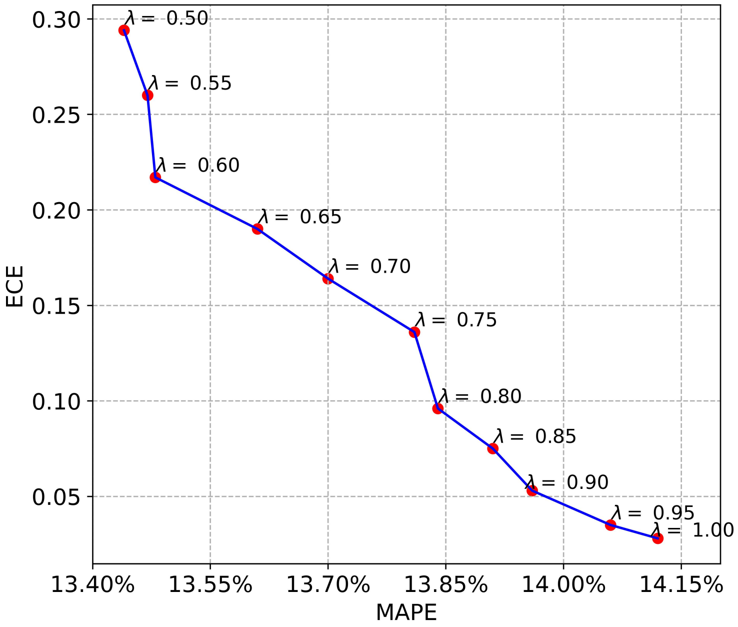
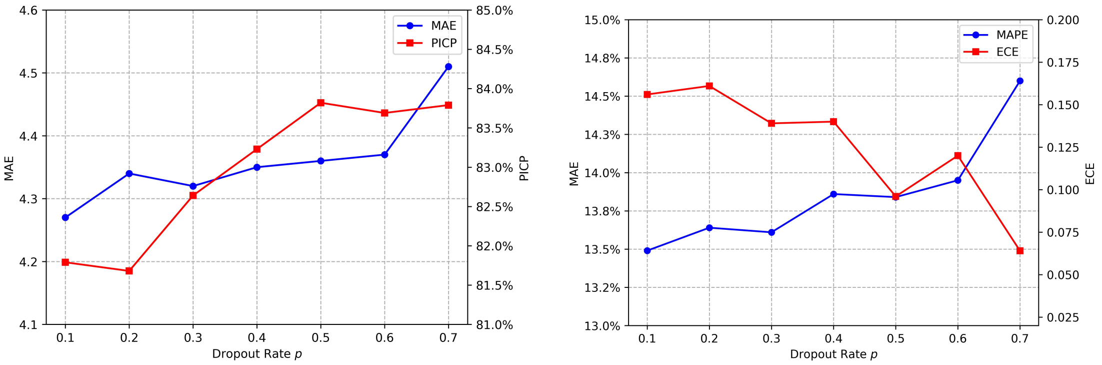


Fig. 9. Parameter tuning for λ in loss function during validation

C. Choice of dropout rate p

In practical applications, the trade-off between the point and interval predictions could be affected by the dropout rate. The increase in dropout rate introduces more noise to the model by randomly deactivating some units in the neural network during the training process, which improves the robustness and generalization of the model but reduces the representational capacity. Therefore, the optimal dropout rate should be carefully selected to avoid being overconfident or underconfident in corresponding predictions. zA well-selected dropout rate could enhance model generalization under different experiment scenarios and improve prediction reliability. Fig. 10 compares the point estimation metrics (*i.e.,* MAE and MAPE) with the interval estimation metrics (*i.e.,* PICP and ECE) under different dropout rates. It can be observed that a higher dropout rate leads to a less accurate point prediction (higher MAE and MAPE) but a better interval prediction (higher PICP and lower ECE). We set the dropout rate $p = 0.5$ as it could achieve relatively accurate interval estimations while maintaining an acceptable level of point prediction performance. This finding on the choice of dropout rate echoes some previous findings (Srivastava et al., 2014).



(a) MAE vs. PICP

(b) MAPE vs. ECE

Fig. 10. Prediction results with respect to different dropout rates in validation process

D. Choice of the time sequence length T

One potential issue of the Kalman filtering is the performance degradation over time. The length of time sequence $\mathbf{T}$ which the filter is applied to estimate the point and interval of the traffic state might impact the stability of the training process. To investigate this, we compare the validation results achieved with different choices of $\mathbf{T}$. In the case study, we select $\mathbf{T}$ from the set $\{11, 23, 83, 168\}$, representing time horizons of 1 hour, 2 hours, 7 hours, and 14 hours (one day) respectively shown in Fig. 11, and the validation performance is presented in Table 2. Our results indicate the influence of $\mathbf{T}$ on the point prediction is relatively limited (MAPE reduction is around 1% when $\mathbf{T}$ increases from 11 to 168), which indicates that the proposed integrated model is able to learn the system dynamics from the data and thus overcome the performance degradation problem caused by inaccurate or incomplete knowledge of the underlying system. This finding also indicates that the value of $\mathbf{T}$ has limited impacts on the generalizability of our model and can be selected based on the propagation duration of the filter. Meanwhile, increasing the time sequence length could result in greater uncertainty as both PICP and MPIW tend to show increasing trends. We reckon this could be because the filter propagates the state estimates over a longer period of time and thus must consider a greater amount of uncertainty in the system. Given the real-world application of traffic prediction, we could set the value of $\mathbf{T}$ as 168 to ensure prediction consistency in the duration of traffic patterns over a day.

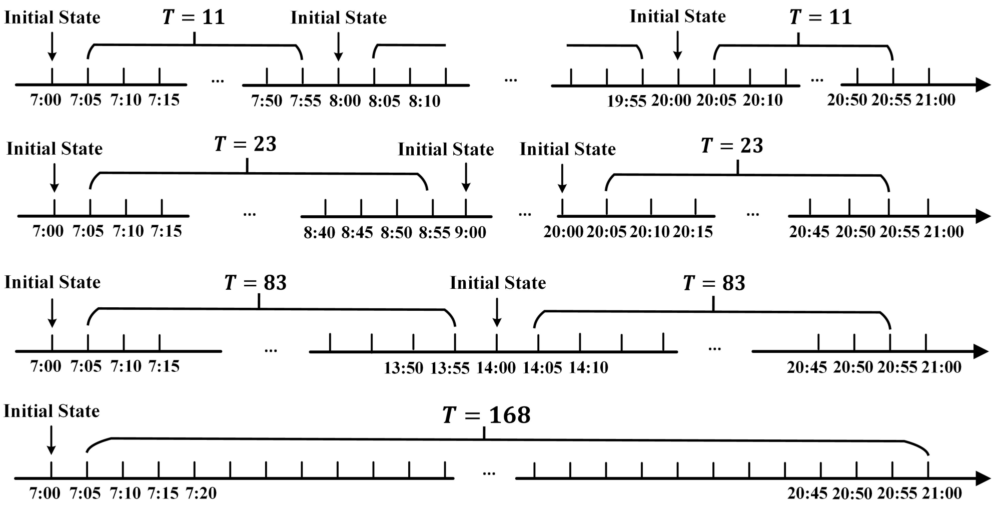


Fig. 11. Time sequence settings

Table 2: Performances of different lengths of time sequence in the validation process

Length of time	Point estimates				Interval estimates		
sequence **T**	MAE	MAPE	RMSE	R^2	PICP	MPIW	ECE
11 (1 hour)	4.07	12.71%	5.55	0.907	81.96%	13.81	0.098
23 (2 hours)	4.17	12.58%	5.59	0.905	80.17%	13.78	0.448
83 (7 hours)	4.21	13.13%	5.68	0.902	82.60%	14.52	0.087
168 (14 hours)	4.36	13.84%	5.84	0.897	83.82%	15.21	0.096

E. Choice of intrinsic temporal features

Fig. 12 displays the validation loss versus the training epoch with different intrinsic temporal features ($\tau_{(t)}$ and $\tau_{(d)}$). None of the validation loss shows an increasing sign, indicating that overfitting does not occur during the training process with or without different intrinsic temporal features. Comparatively, it is observed that traffic speed data with the TimeOfDay $\tau_{(t)}$ converge faster than all the other combinations. Moreover, the incorporation of the DayOfWeek $\tau_{(d)}$ does not contribute to the improvement of prediction accuracy. It suggests that the temporal traffic patterns fluctuate more significantly over the period of a day than over different days of the week, and thus these patterns could be captured more accurately by $\tau_{(t)}$ rather than $\tau_{(d)}$. Therefore, we only incorporate $\tau_{(t)}$, along with other evolution and update features (as introduced in Section 2.2.1), into the proposed integrated prediction framework. However, it is important to note that the choice of intrinsic temporal features should be carefully considered based on the specific traffic patterns in different scenarios and datasets.

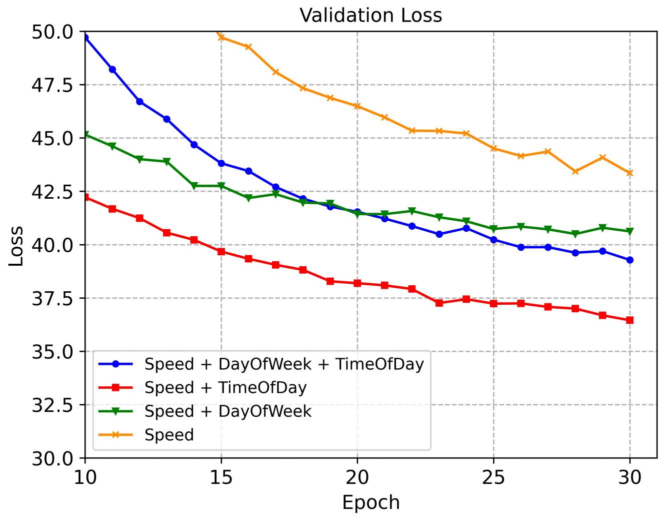


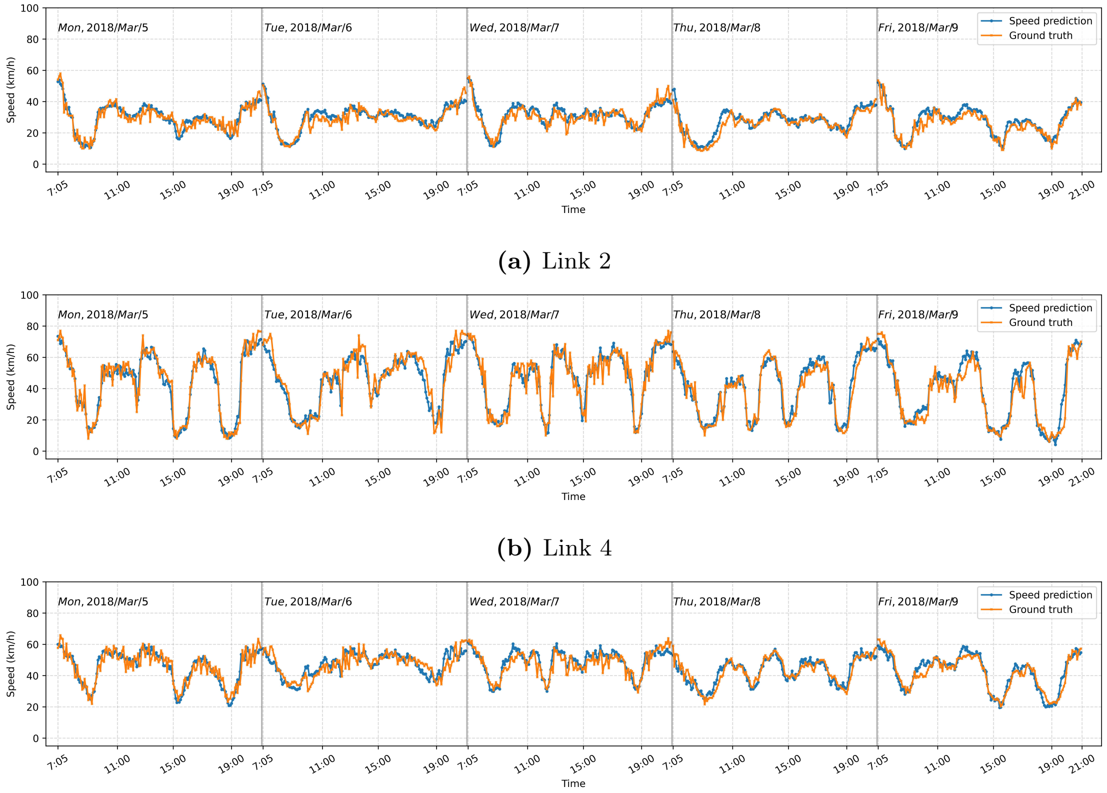
Fig. 12. Validation loss versus training epoch with different combination of intrinsic temporal features

3.3 Testing Results and Discussion

3.3.1 Point estimates

In this section, we compare the prediction results against the ground truth value as shown in Fig. 13. Fig. 13a and Fig. 13b show the predicted speeds over two different links inferred from the Autoscope detectors with distinct traffic patterns. It can be observed that all predicted speed values align well with the associated ground truths. Fig. 13c presents the predicted average speed extracted from AVI data over the entire study route. The time spans of all the sub-figures in Fig. 13 last for one week. The results generally show that the proposed prediction can derive accurate predictions despite different traffic patterns encountered at different locations and times.

We now investigate the spatial-temporal features of the road network incorporated into the prediction framework via the model-based module. Fig. 14 depicts the heatmap of transition matrix parameters under congested and free-flow conditions. The higher diagonal values in the heatmaps suggest that the interactions between different road segments might be less significant due to the constraints imposed by high traffic density during congested conditions. In contrast, the fact that traffic flow tends to propagate more smoothly under free-flow conditions typically results in higher off-diagonal values. In addition, the diagonal value of link 4 (*i.e.,* 0.75) is the highest and it also has the greatest influence (*i.e.,* 0.20) on the journey time under congested conditions. This implies that link 4 can be considered as the bottleneck of the road corridor. The analysis of spatial-temporal features using the statistical dynamic linear model demonstrates



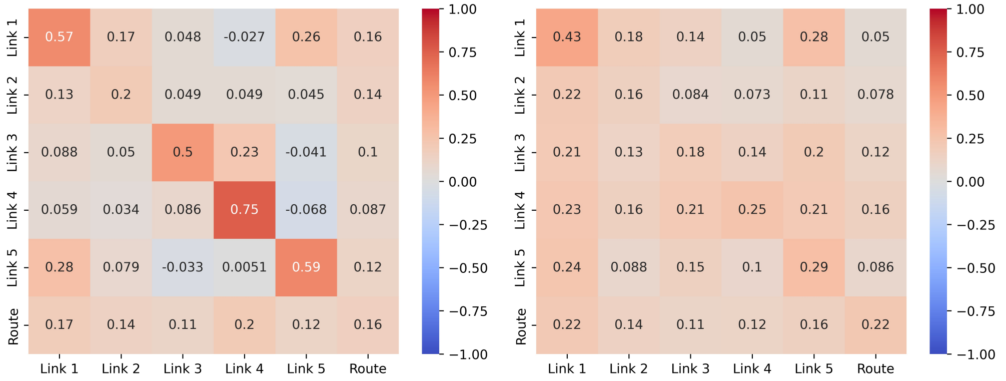
(a) Link 2

(b) Link 4

(c) Average speed of the entire route extracted from AVI

Fig. 13. Results of point estimates of traffic speed from 5 Mar. 2018 to 9 Mar. 2018

that the proposed framework has the ability to deliver interpretable traffic predictions rather than functioning as a black box.



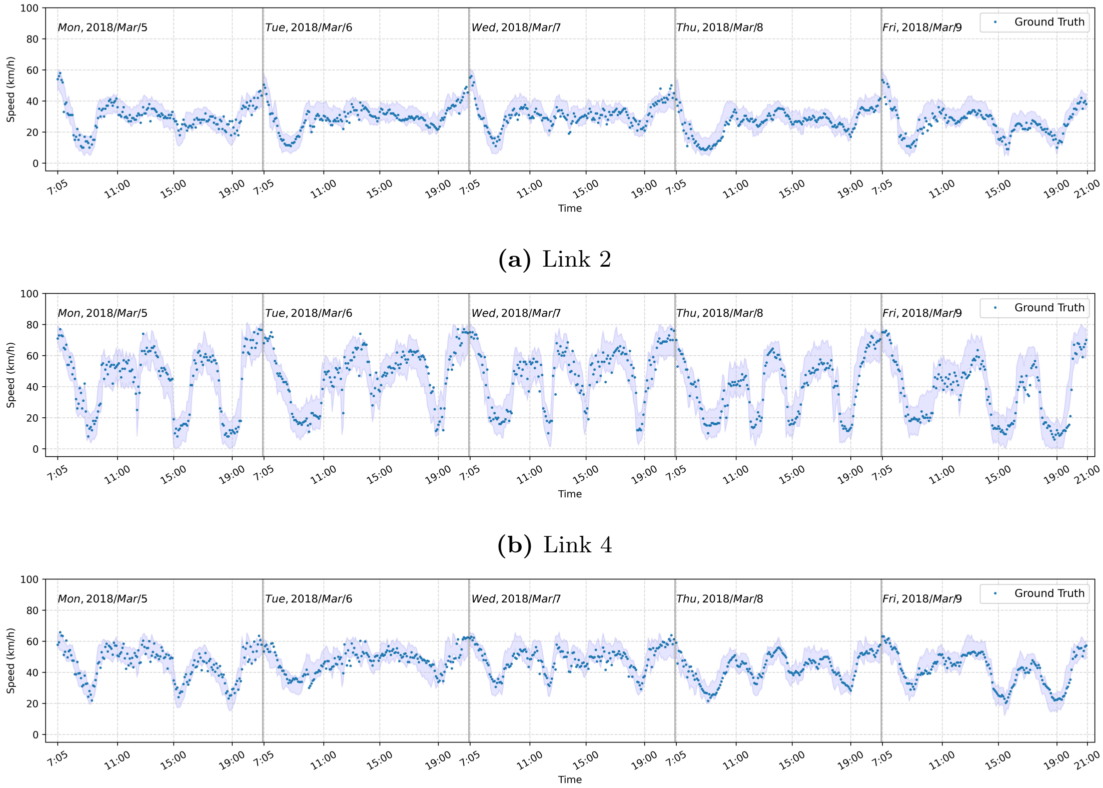
(a) 18:00 (congested condition)

(b) 20:30 (free-flow condition)

Fig. 14. Heatmap of transition matrix parameters under different traffic conditions (a) Under congested conditions; (b) Under free-flow conditions

3.3.2 Interval estimates

To evaluate the interval estimates, we first compare the confidence interval constructed based on the Eq.(25) as shown in Fig. 15. It is observed that the prediction algorithms are able to estimate the surges in traffic speed as well as the associated variability during peaks. As discussed in Section 3.3.1, the more dynamic speed variations at link 4 (see Fig. 15b) will have to be captured by a wider interval compared to the speed profile at link 2 (see Fig. 15a). Table 3 compares the prediction performance of link 2 and link 4 under free-flow (16:00-17:00) and congested (08:00-09:00) conditions to further explore the prediction accuracy at different road links under different traffic periods. It can be found that the best prediction performance is achieved during free-flow periods in link 2 compared to link 4. This comparison result indicates a potential improvement for the proposed model in accurately estimating uncertainty in less predictable traffic scenarios with significant fluctuations in speed.



(a) Link 2

(b) Link 4

(c) Average speed of the entire route extracted from AVI

Fig. 15. Results of interval estimates of traffic speed from 5 Mar. 2018 to 9 Mar. 2018

With regard to the correlation coefficients among the multivariate traffic features, Fig. 16 shows the heatmap of the correlation matrix under different traffic conditions on 13 March 2018. The heatmap of the correlated matrix could provide insights into the spatio-temporal features of

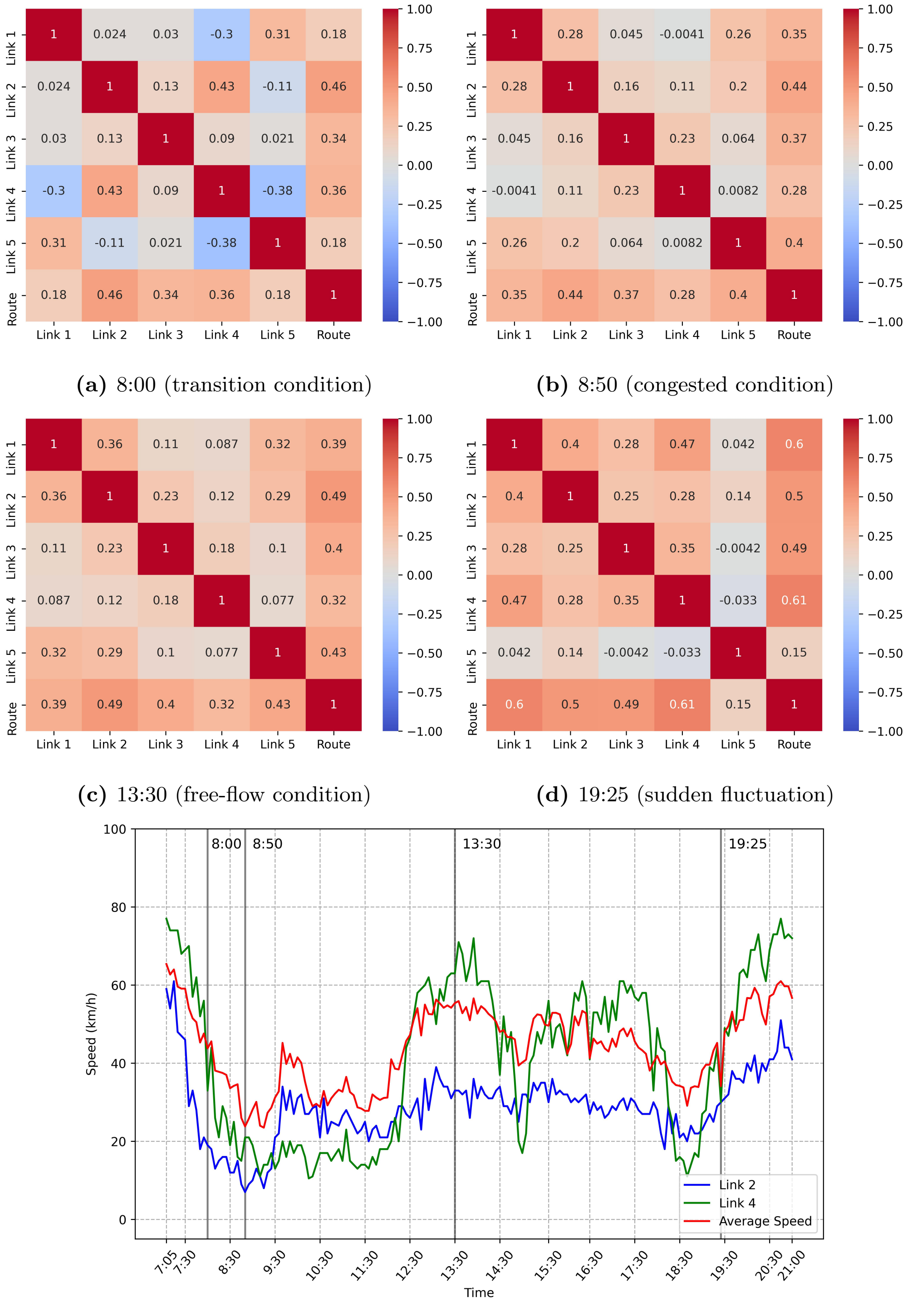
Table 3: Testing results of selected links under free-flow and congested conditions

Traffic condition	Link	Point estimates			Interval estimates	
		MAE	MAPE	RMSE	PICP	MPIW
Free-flow (16:00-17:00)	Link 2	2.57	11.43%	3.29	91.86%	10.39
	Link 4	4.88	15.76%	6.59	88.24%	20.03
Congestion (08:00-09:00)	Link 2	2.22	13.31%	3.11	94.12%	11.67
	Link 4	6.20	15.93%	7.73	80.54%	18.71

the traffic speed patterns. It can be observed that the correlation coefficients between localized speed and the generalized average speed are comparatively larger than the relations between different pairs of localized speed. This implies the average speed of the entire route incorporates the information on traffic patterns from all the link segments. Fig. 16e shows the actual traffic patterns of link 2, link 4 and the entire route. It can be found that the traffic pattern of the entire route is more similar to the traffic pattern at link 2, but less relevant to the traffic pattern at link 4. Consequently, the correlation coefficients between the traffic speed of link 2 and the entire route are the highest under different conditions. Moreover, it can be seen from Fig. 16d that a sudden deduction in traffic speed for the entire route occurs at 19:25, which is only observed in the spot speed at link 4. Therefore, the correlation coefficient between the spot speed at link 4 and the average speed of the entire route increases significantly. This again demonstrates that the proposed integrated model has the capability to predict the heteroscedastic uncertainty.

On the effect of congestion, the figures reveal that the correlation coefficients between different localized links under free flow conditions are stronger than under transition and congested conditions. We attribute the stronger correlations to the more consistent traffic states under free flow conditions due to relatively smaller traffic volume and weaker speed fluctuations. It is also observed that correlation coefficients can be negative under transition conditions, while they tend to be positive under both congested and free-flow conditions. This can be interpreted as the fact that the congestion or disruption propagates backward through the traffic stream during transition conditions. Such hysteresis phenomenon in traffic flow causes negative correlation coefficients.

Fig. 17 further shows the calibration plot incorporated with different sources of uncertainty, together with the comparison between the empirical and the theoretical distribution of the squared Mahalanobis distance to quantify the accuracy of uncertainty. The calibration plot demonstrates the effectiveness of the interval prediction. As shown in Fig. 17a, only considering the single source of uncertainty could result in the inaccurate estimation of overall uncertainty. Moreover, it can be found that stochastic uncertainty is the dominant source of uncertainty



(a) 8:00 (transition condition)

(b) 8:50 (congested condition)

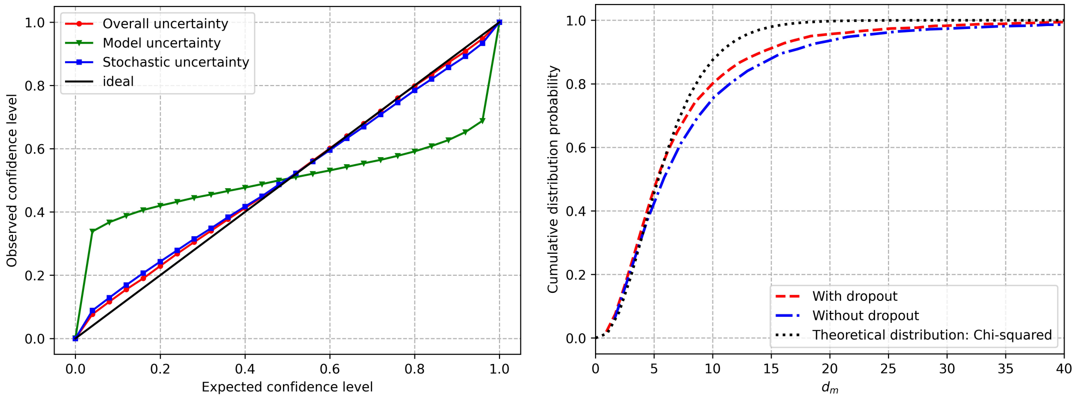
(c) 13:30 (free-flow condition)

(d) 19:25 (sudden fluctuation)

(e) Ground truth values

Fig. 16. Correlation heatmap under different traffic conditions - 13 March 2018. (a) Under transition conditions; (b) Under congested conditions; (c) Under free-flow conditions; (d) Under dramatic fluctuation conditions; (e) Ground truth values of speed for link 2, link 4 and average speed of the entire route.

compared to model uncertainty in this scenario, which indicates that the variability of traffic prediction is primarily caused by inherent randomness rather than the lack of knowledge. Therefore, the physical model in the proposed integrated method could incorporate the domain knowledge, and thus could contribute to reducing the uncertainty that arises from the limited understanding of the underlying mechanism of the dynamic system when compared to pure data-driven methods. Furthermore, as discussed in Section 2.4, the empirical distribution of the squared Mahalanobis distance for an ideal uncertainty calibration should yield a perfect match with the theoretical χ^2 distribution. It also can be seen from Fig. 17b that an improvement could be achieved with the incorporation of MC dropout during uncertainty inference. This also suggests the benefit of considering different sources of uncertainty for prediction accuracy and reliability.



(a) Calibration figure

(b) Cumulative distribution of ξ_t^d

Fig. 17. Uncertainty quantification with different sources. (a) Calibration plot after incorporating model and stochastic uncertainty; (b) Comparison between the empirical distribution of the squared Mahalanobis distance with theoretical distribution with (dropout rate $p = 0.5$ and MC samples $\boldsymbol{B} = 5$) and without uncertainty measurement.

3.4 Comparison with baseline algorithms

The proposed method is compared with the following baseline methods with the inclusion of both point and interval metrics to prove its competitiveness:

1. KF (Kwak and Geroliminis, 2020): Kalman Filtering with linear state space model is selected as our main model-based algorithm benchmark.

2. AKF (Guo et al., 2014): Adaptive Kalman Filtering with linear state space model could update the process and measurement noises to track the heteroscedastic traffic conditions.

3. LSTM (Ma et al., 2015): Conventional-based Long short-term memory network with fully connected layers is used to extract the temporal features of traffic speed.

4. GRU (Zhang and Kabuka, 2018): Conventional-based Gated recurrent unit is another type of RNN architecture used for processing time-series data.

5. TCN (Ren et al., 2020): Temporal convolutional network utilizes a series of 1-D convolutional layers to extract temporal features in sequential data.

6. GCN (Yu et al., 2020): Graph convolutional network is selected to extract the structural spatial features in the road network nodes.

7. ConvLSTM (Petersen et al., 2019): Convolutional long short-term memory neural network is used to capture sptail-temporal features. ConvLSTM is a unified network which combines convolutional operations with LSTM units.

8. TCN-LSTM (Bi et al., 2021): Temporal convolutional network with long short-term memory network model. TCN module is used to extract short-term features and the LSTM module is used to capture the long-term dependence.

9. CNN-LSTM (Zhuang et al., 2024): Convolutional neural network with long short-term memory network model, which combines a CNN module to extract spatial-temporal features of Autoscope data and an LSTM module to extract temporal features of AVI data.

10. SE-CNN-LSTM-SA (Hu et al., 2018; Zheng et al., 2020): squeeze-and-excitation convolutional neural network with long short-term memory and self-attention. SE block is plugged into the CNN as a channel-wise attention mechanism, and the self-attention layer is used to capture the relationships between different spatial-temporal features in the sequence.

11. GCN-LSTM (Li et al., 2019): Graph convolutional neural network with long short-term memory network model. This framework shares a similar structure with the CNN-LSTM to effectively capture both spatial dependencies in graph-structured Autoscope data and temporal dynamics of AVI data.

12. ChebNet-LSTM (Hou et al., 2021): Chebyshev graph convolutions with long short-term memory network model. Based on the structure of GCN-LSTM, ChebNet uses Chebyshev polynomials to approximate graph convolutions for capturing higher-order neighborhood information.

13. ChebNet-LSTM-SA (Hou et al., 2021; Zhao et al., 2022): Based on ChebNet-LSTM structure, self-attention is incorporated into the network to dynamically weigh the importance of different spatial-temporal features.

14. MDBF-GRU: Integrated model-based and data-driven Bayesian framework with direct artificial neural network (shown in Fig. 1b). This structure also utilizes the GRU and the same hyperparameters as the proposed model.

Table 4 shows the testing results of all selected baseline models. We compare the proposed integrated model with the common model-based benchmarks, namely the KF and AKF. Next, we consider the performance of pure data-driven algorithms. Data-driven models are utilized as a black box, directly mapping the observed data to the end-to-end point and interval estimations using the loss function introduced in Section 2.3.1 without explicit knowledge. The structure and the hyperparameters of all the models are initially set based on the experiences and tuned based on the validation results respectively. The early stopping technique is applied to avoid overfitting. Among the selected data-driven algorithms, LSTM and GRU models are the typical and most widely applied recurrent neural networks to handle sequence data. TCN is another type of time-series prediction model that could capture longer-term dependencies across multiple time steps thanks to its convolutional layers. To better capture the short-term and long-term dependencies simultaneously, the combination of TCN with LSTM is also introduced. To further consider the spatio-temporal feature extraction, ConvLSTM is introduced to simultaneously process the spatial and temporal information. In addition, a two-stream novel combination of CNN and LSTM is selected as another powerful approach to extract the spatial-temporal features separately. SE-CNN-LSTM-SA model incorporates the squeeze-and-excitation block into the CNN to improve channel interdependencies and uses the self-attention mechanism to effectively integrate spatial and temporal information from these two parallel modules. Graph-based models are also selected as baselines due to their ability to incorporate the topology of transportation networks. Based on the foundation of GCN models, GCN-LSTM combines GCN with LSTM to capture spatial and temporal dynamics in sequential data. ChebNet-LSTM enhances this framework by utilizing Chebyshev polynomials to approximate graph convolutions, while the ChebNet-LSTM-SA model further incorporates self-attention mechanisms to weight the importance of different features. Lastly, we validate the effectiveness of the architectural design by comparing the proposed model to the integrated model which directly replaces the second-order statistics calculation process with the GRU network.

Table 4: Testing results of baseline algorithms

Algorithm	Training Time	Point estimates				Interval estimates		
	(seconds)	MAE	MAPE	RMSE	R^2	PICP	MPIW	ECE
(I) Model-based algorithms								
KF	-	7.26	24.24%	8.60	0.717	78.14%	20.88	3.260
AKF	-	7.30	24.19%	8.62	0.711	82.08%	22.91	3.149
(II) Data-driven algorithms								
LSTM	1564.50	5.01	16.85%	6.81	0.820	92.87%	23.93	0.008
GRU	1268.99	4.58	15.76%	6.26	0.848	93.26%	22.51	0.010
TCN	1448.77	4.51	14.44%	5.94	0.863	97.29%	27.64	0.038
GCN	1806.73	4.74	14.67%	6.19	0.851	93.41%	28.07	0.013
ConvLSTM	1365.59	4.49	14.71%	6.09	0.872	98.98%	34.43	0.134
TCN-LSTM	1523.35	4.49	13.74%	5.91	0.864	96.17%	25.67	0.035
CNN-LSTM	1561.74	4.22	12.07%	5.59	0.879	97.75%	28.96	0.086
SE-CNN-LSTM-SA	1946.74	3.84	11.71%	5.19	0.896	98.85%	26.91	0.115
GCN-LSTM	1925.56	3.96	14.02%	5.66	0.876	91.76%	17.90	0.026
ChebNet-LSTM	2054.74	4.34	13.35%	5.69	0.874	97.53%	31.14	0.131
ChebNet-LSTM-SA	2156.35	4.00	12.21%	5.32	0.890	96.53%	26.42	0.071
(III) Integrated algorithms								
MDBF-GRU	1695.19	3.96	11.88%	5.16	0.898	89.53%	17.45	0.281
Proposed	1311.41	3.49	**10.56%**	4.65	0.917	90.41%	15.27	**0.008**

As shown in Table 4, the proposed integrated model can deliver the best point prediction compared with the baseline algorithms. The model-based algorithms have the worst performance among all the selected baselines due to the difficulties of estimating noise measurement in real-world applications. It can also be observed that all the pure data-driven algorithms perform better than the model-based algorithms, indicating that the data-driven model can better deal with complicated dynamic systems. Furthermore, among all the pure data-driven models, the SE-CNN-LSTM-SA shows the best performance by utilizing spatial-temporal traffic information. Unlike other models, the CNN module allows it to extract spatial features across different link segments. However, graph-based models do not improve the prediction accuracy but increase the model training time due to the simplistic topology of the road corridor compared to CNN-based models. Although these models could deliver relatively acceptable performance in point predictions, they fail to provide reliable interval prediction resulting in wider intervals. These data-driven models require large datasets to train the numerical parameters even for relatively simple tasks. This characteristic increases model uncertainty and leads to an overestimation of overall uncertainty. Therefore, the limited data in our experimental scenario makes the data-driven less efficient in point prediction. Moreover, it can be observed that data-driven models tend to improve the point prediction accuracy at the expense of uncertainty estimation when comparing SE-CNN-LSTM-SA to LSTM.

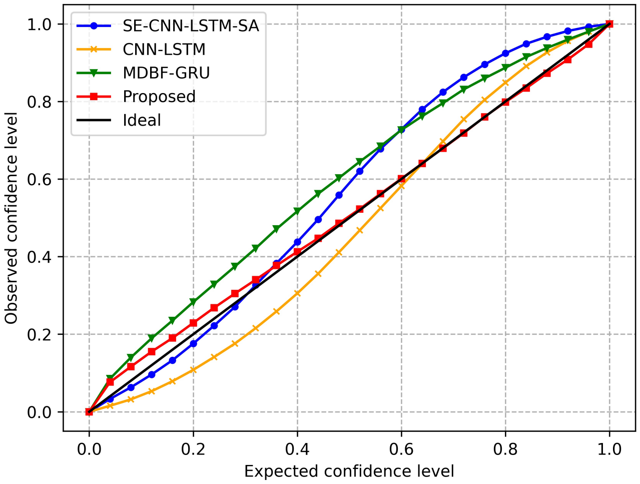


Fig. 18. Comparisons of uncertainty calibration plots among different baseline algorithms

Compared to purely model-based and data-driven algorithms, integrated algorithms could improve the prediction performance from the following three aspects: first, the incorporation of the state space model into the data-driven approaches could improve model interpretability, reduce the model complexity and enhance the data efficiency. Then, the constrained input-output mappings in the data-driven module have specific theoretical meanings instead of being a black box, resulting in less trainable parameters and explainable data-driven architecture in tracking second-order statistics. Finally, by following the "*predict-correct*" structure of Kalman filtering, the proposed model could calibrate predictions with new observations and thus improve the prediction accuracy. We can observe that the proposed algorithm has improved the value of R^2 by 2.34% compared to the SE-CNN-LSTM-SA. In addition, MBDF-GRU can offer comparative results on point prediction, but it still fails to achieve better uncertainty prediction as shown in Fig. 18. This also reveals and supports the benefits of utilizing separate GRU cells for tracking the second-order statistics in uncertainty quantifications by preserving the operational manner of the Bayesian filter. In terms of computational complexity, the training time of the proposed model is comparable to that of a conventional GRU-based baseline while significantly improving prediction accuracy. CNN-LSTM-based models could achieve similar point prediction performance but are relatively computationally intensive due to their large number of trainable parameters when compared with the proposed model. These facts support the proposed integrated method as a promising and computationally efficient approach for traffic prediction compared with purely data-driven or model-based algorithms.

4 Conclusions

This paper proposes a Bayesian framework which integrates the model-based Kalman filtering with the data-driven artificial neural network to provide stochastic traffic predictions. The correlated and heteroscedastic uncertainty, along with the point predictions, are quantified simultaneously in the form of the state vector and estimate covariance matrix. The architecture of the proposed integrated model is designed to maintain the operation flow of the Kalman filtering to keep its interpretability. We incorporate the dynamic linear state space model as the predictor of traffic states and use four separate GRU cells to track the second-order statistics. These data-driven modules interact and exchange their features following the calculation operation of corresponding noise-dependent second-order statistics. By using the learned features from GRU cells as inputs to produce the $\mathbf{K}_t^d$, the integrated model escapes from the dependence on the knowledge of noise measurements and benefits from both the strengths of model-based and data-driven approaches. This constrained network architecture provides less abstraction by preserving the general state space model of Kalman filtering as its model-based core, thus providing generalizability to related problems without modifications to the network architecture. The model is trained end-to-end using the multivariate Gaussian negative log-likelihood loss, and we present how to incorporate both the model and stochastic uncertainty during the inference by MC dropout.

The proposed prediction framework is implemented and tested with traffic data collected from a selected Hong Kong corridor and compared with eight other established prediction methods for benchmarking. The results reveal that the proposed model could yield accurate point and interval predictions. It outperforms both the pure model-based and data-driven approaches, demonstrating its capability to leverage data efficiency and domain knowledge. Such performance can be achieved with a relatively small dataset, which is more applicable to real-world problems. Furthermore, the integration of model uncertainty by MC dropout during inference could also improve prediction accuracy and reliability, and it is found that stochastic uncertainty is the dominant source of uncertainty in our proposed model. We reckon this could be due to the fact that the incorporation of domain knowledge into the artificial neural network contributes to reducing the model uncertainties arising from the model parameters. Moreover, on the effect of including different intrinsic temporal features, our experiment shows that the inclusion of the time-of-day indicator contributes more to the effectiveness of traffic prediction. This suggests the temporal patterns fluctuate more significantly over a day than over different days of the week.

This work contributes to the development of reliability-based intelligent transportation systems. The proposed integrated model could be applied in real-world travel guidance systems in an online manner, making it possible to continuously update the estimates of the dynamic traffic systems when new observations are available. It also offers the potential to take advantage of both model-based and data-driven approaches to provide more reliable and robust predictions. However, it is noted that the physical model presented herein is based on a statistical approach applied to a small-scale road corridor. In the future, we will integrate the proposed model into traffic networks through the use of more advanced traffic flow modeling techniques (*e.g.* cell transmission model and kinematic wave traffic model (Su et al., 2021)) and high-dimensional statistical techniques (*e.g.* Lasso regression (Kamarianakis et al., 2012), high dimensional regression (Bouchouia and Portier, 2021) and principal component analysis (Zhong et al., 2023)) via the model-based core to produce interpretable results. Moreover, we would like to point out that the state space model incorporated in the framework is a linear system, and a future direction we have been working on is to improve the current framework based on the extended Kalman filtering to handle the traffic non-linearity. Moreover, it is important to note that the state space model incorporated in our framework is a linear system. A future direction which we are working on is to improve the current framework based on the extended Kalman filtering. This approach is suitable for non-linear traffic systems, as it linearizes the state and observation functions using a first-order Taylor expansion to address the traffic non-linearity problem.

Acknowledgments

Constructive comments from the anonymous referees are gratefully acknowledged. The authors would like to thank Hong Kong Transport Department and Autotoll Limited for providing the traffic data. This study is supported by two Research Impact Funds (RIF) sponsored by the Hong Kong Research Grants Council (Refs: R5029-18 and R7027-18).

References

Abdi, J., Moshiri, B., Abdulhai, B., and Sedigh, A. K., 2012. Forecasting of short-term traffic-flow based on improved neurofuzzy models via emotional temporal difference learning algorithm. Engineering Applications of Artificial Intelligence, 25(5):1022–1042.

Ali, S., Abuhmed, T., El-Sappagh, S., Muhammad, K., Alonso-Moral, J. M., Confalonieri, R., Guidotti, R., Del Ser, J., Díaz-Rodríguez, N., and Herrera, F., 2023. Explainable artificial intelligence (xai): What we know and what is left to attain trustworthy artificial intelligence. Information fusion, 99:101805.

Alsolami, B., Mehmood, R., and Albeshri, A., 2020. Hybrid statistical and machine learning methods for road traffic prediction: A review and tutorial. Smart Infrastructure and Applications: Foundations for Smarter Cities and Societies, pages 115–133.

Bai, L., Wong, W., Xu, P., Liu, P., Chow, A. H. F., Lam, W. H. K., Ma, W., Han, Y., and Wong, S. C., 2024. Fusion of multi-resolution data for estimating speed-density relationships. Transportation Research Part C, 165:104742.

Bi, J., Zhang, X., Yuan, H., Zhang, J., and Zhou, M., 2021. A hybrid prediction method for realistic network traffic with temporal convolutional network and lstm. IEEE Transactions on Automation Science and Engineering, 19(3):1869–1879.

Birant, D. and Kut, A., 2007. ST-DBSCAN: An algorithm for clustering spatial–temporal data. Data & Knowledge Engineering, 60(1):208–221.

Bogaerts, T., Masegosa, A. D., Angarita-Zapata, J. S., Onieva, E., and Hellinckx, P., 2020. A graph cnn-lstm neural network for short and long-term traffic forecasting based on trajectory data. Transportation Research Part C: Emerging Technologies, 112:62–77.

Bouchouia, M. and Portier, F., 2021. High dimensional regression for regenerative time-series: An application to road traffic modeling. Computational Statistics & Data Analysis, 158: 107191.

Cai, P., Wang, Y., Lu, G., Chen, P., Ding, C., and Sun, J., 2016. A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting. Transportation Research Part C: Emerging Technologies, 62:21–34.

Chen, H. and Rakha, H. A., 2014. Real-time travel time prediction using particle filtering with a non-explicit state-transition model. Transportation Research Part C: Emerging Technologies, 43:112–126.

Cheng, T., Wang, J., Haworth, J., Heydecker, B., and Chow, A., 2014. A dynamic spatial weight matrix and localized space-time autoregressive integrated moving average for network modeling. Geographical Analysis, 46:75–97.

Chikaraishi, M., Garg, P., Varghese, V., Yoshizoe, K., Urata, J., Shiomi, Y., and Watanabe, R., 2020. On the possibility of short-term traffic prediction during disaster with machine learning approaches: An exploratory analysis. Transport Policy, 98:91–104.

Cui, Z., Henrickson, K., Ke, R., and Wang, Y., 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):4883–4894.

Cui, Z., Ke, R., Pu, Z., Ma, X., and Wang, Y., 2020a. Learning traffic as a graph: A gated graph wavelet recurrent neural network for network-scale traffic prediction. Transportation Research Part C: Emerging Technologies, 115:102620.

Cui, Z., Ke, R., Pu, Z., and Wang, Y., 2020b. 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.

Feng, X., Ling, X., Zheng, H., Chen, Z., and Xu, Y., 2018. Adaptive multi-kernel svm with spatial–temporal correlation for short-term traffic flow prediction. IEEE Transactions on Intelligent Transportation Systems, 20(6):2001–2013.

Gal, Y. and Ghahramani, Z., 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. Proceedings of The 33rd International Conference on Machine Learning, volume 48, pages 1050–1059.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning*. MIT press, 2016.

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

Gu, Y., Lu, W., Xu, X., Qin, L., Shao, Z., and Zhang, H., 2019b. An improved bayesian combination model for short-term traffic prediction with deep learning. IEEE Transactions on Intelligent Transportation Systems, 21(3):1332–1342.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K., 2017. On calibration of modern neural networks. Proceedings of the 34th International Conference on Machine Learning, 70:1321–1330.

Guo, F., Polak, J. W., and Krishnan, R., 2018. Predictor fusion for short-term traffic forecasting. Transportation Research Part C: Emerging Technologies, 92:90–100.

Guo, J., Huang, W., and Williams, B. M., 2014. Adaptive kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification. Transportation Research Part C: Emerging Technologies, 43:50–64.

Hou, F., Zhang, Y., Fu, X., Jiao, L., and Zheng, W., 2021. The prediction of multistep traffic flow based on ast-gcn-lstm. Journal of Advanced Transportation, 2021(1):9513170.

Hou, Q., Leng, J., Ma, G., Liu, W., and Cheng, Y., 2019. An adaptive hybrid model for short-term urban traffic flow prediction. Physica A: Statistical Mechanics and its Applications, 527: 121065.

Hu, J., Shen, L., and Sun, G., 2018. Squeeze-and-excitation networks. Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7132–7141.

Kamarianakis, Y., Kanas, A., and Prastacos, P., 2005. Modeling traffic volatility dynamics in an urban network. Transportation Research Record, 1923(1):18–27.

Kamarianakis, Y., Shen, W., and Wynter, L., 2012. Real-time road traffic forecasting using regime-switching space-time models and adaptive lasso. Applied stochastic models in business and industry, 28(4):297–315.

Kang, L., Hu, G., Huang, H., Lu, W., and Liu, L., 2020. Urban traffic travel time short-term prediction model based on spatio-temporal feature extraction. journal of advanced transportation, 2020(1):3247847.

Karlaftis, M. G. and Vlahogianni, E. I., 2011. Statistical methods versus neural networks in transportation research: Differences, similarities and some insights. Transportation Research Part C: Emerging Technologies, 19(3):387–399.

Kenny, E. M., Ford, C., Quinn, M., and Keane, M. T., 2021. Explaining black-box classifiers using post-hoc explanations-by-example: The effect of explanations and error-rates in xai user studies. Artificial Intelligence, 294:103459.

Kingma, D. and Ba, J., 2014. Adam: A Method for Stochastic Optimization. arXiv:1412.6980.

Kuleshov, V., Fenner, N., and Ermon, S., 2018. Accurate uncertainties for deep learning using calibrated regression. Proceedings of the 35th International Conference on Machine Learning, pages 2796–2804.

Kwak, S. and Geroliminis, N., 2020. Travel time prediction for congested freeways with a dynamic linear model. IEEE Transactions on Intelligent Transportation Systems, 22(12):7667–7677.

Lakshminarayanan, B., Pritzel, A., and Blundell, C., 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. Advances in Neural Information Processing Systems, 30.

Lartey, B., Homaifar, A., Girma, A., Karimoddini, A., and Opoku, D., 2021. Xgboost: a tree-based approach for traffic volume prediction. 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pages 1280–1286. doi: 10.1109/SMC52423.2021.9658959.

Lee, E. H., 2023. Traffic speed prediction of urban road network based on high importance links using xgboost and shapley additive explanation. IEEE Access.

Li, A., Lam, W. H. K., Ma, W., Chow, A. H. F., Wong, S. C., and Tam, M. L., 2023. Filtering limited automatic vehicle identification data for real-time path travel time estimation without ground truth. IEEE Transactions on Intelligent Transportation Systems.

Li, A., Lam, W. H. K., Ma, W., Wong, S. C., Chow, A. H. F., and Tam, M. L., 2024. Real-time estimation of multi-class path travel times using multi-source traffic data. Expert Systems with Applications, 237:121613.

Li, L., Ran, B., Zhu, J., and Du, B., 2020. Coupled application of deep learning model and quantile regression for travel time and its interval estimation using data in different dimensions. Applied Soft Computing, 93:106387.

Li, Z., Xiong, G., Chen, Y., Lv, Y., Hu, B., Zhu, F., and Wang, F.-Y., 2019. A hybrid deep learning approach with gcn and lstm for traffic flow prediction. 2019 IEEE intelligent transportation systems conference (ITSC), pages 1929–1933.

Ma, T., Antoniou, C., and Toledo, T., 2020. Hybrid machine learning algorithm and statistical time series model for network-wide traffic forecast. Transportation Research Part C: Emerging Technologies, 111:352–372.

Ma, X., Tao, Z., Wang, Y., Yu, H., and Wang, Y., 2015. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transportation Research Part C: Emerging Technologies, 54:187–197.

Marinică, N. E., Sarlette, A., and Boel, R. K., 2013. Distributed particle filter for urban traffic networks using a platoon-based model. IEEE Transactions on Intelligent Transportation Systems, 14(4):1918–1929.

Markos, C., James, J., and Da Xu, R. Y., 2021. Capturing uncertainty in unsupervised gps trajectory segmentation using bayesian deep learning. Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 390–398.

Nantes, A., Ngoduy, D., Bhaskar, A., Miska, M., and Chung, E., 2016. Real-time traffic state estimation in urban corridors from heterogeneous data. Transportation Research Part C: Emerging Technologies, 66:99–118.

Ngoduy, D. and Sumalee, A., 2010. Adaptive estimation of noise covariance matrices in unscented kalman filter for multiclass traffic flow model. Transportation Research Record, 2188(1):119–130.

Ottaviano, F., Cui, F., and Chow, A. H. F., 2017. Modelling and fusion of dynamic highway traffic data. Transportation Research Record, 2644:92–99.

Pan, Y. A., Guo, J., Chen, Y., Cheng, Q., Li, W., and Liu, Y., 2024. A fundamental diagram based hybrid framework for traffic flow estimation and prediction by combining a markovian model with deep learning. Expert Systems with Applications, 238:122219.

Parsa, A. B., Movahedi, A., Taghipour, H., Derrible, S., and Mohammadian, A. K., 2020. Toward safer highways, application of xgboost and shap for real-time accident detection and feature analysis. Accident Analysis & Prevention, 136:105405.

Petersen, N. C., Rodrigues, F., and Pereira, F. C., 2019. Multi-output bus travel time prediction with convolutional lstm neural network. Expert Systems with Applications, 120:426–435.

Punzo, V. and Montanino, M., 2020. A two-level probabilistic approach for validation of stochastic traffic simulations: Impact of drivers' heterogeneity models. Transportation Research Part C: Emerging Technologies, 121:102843.

Rajamani, M. R. *Data-based techniques to improve state estimation in model predictive control.* PhD thesis, University of Wisconsin–Madison, 2007.

Ren, Y., Zhao, D., Luo, D., Ma, H., and Duan, P., 2020. Global-local temporal convolutional network for traffic flow prediction. IEEE Transactions on Intelligent Transportation Systems, 23(2):1578–1584.

Russell, R. L. and Reale, C., 2021. Multivariate uncertainty in deep learning. IEEE Transactions on Neural Networks and Learning Systems, 33(12):7937–7943.

Saeedmanesh, M., Kouvelas, A., and Geroliminis, N., 2021. An extended kalman filter approach for real-time state estimation in multi-region mfd urban networks. Transportation Research Part C: Emerging Technologies, 132:103384.

Sattarzadeh, A. R., Kutadinata, R. J., Pathirana, P. N., and Huynh, V. T., 2023. A novel hybrid deep learning model with arima conv-lstm networks and shuffle attention layer for short-term traffic flow prediction. Transportmetrica A: Transport Science, pages 1–23.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1):1929–1958.

Su, Z., Chow, A. H. F., and Zhong, R. X., 2021. Adaptive network traffic control with an integrated model-based and data-driven approach and a decentralised solution method. Transportation Research Part C: Emerging Technologies, 128:103154.

Sun, B., Sun, T., and Jiao, P., 2021. Spatio-temporal segmented traffic flow prediction with anprs data based on improved xgboost. Journal of Advanced Transportation, 2021(1):5559562.

Tampère, C. M. and Immers, L., 2007. An extended Kalman filter application for traffic state estimation using CTM with implicit mode switching and dynamic parameters. 2007 IEEE Intelligent Transportation Systems Conference, pages 209–216.

Trinh, X.-S., Ngoduy, D., Keyvan-Ekbatani, M., and Robertson, B., 2022. Incremental unscented kalman filter for real-time traffic estimation on motorways using multi-source data. Transportmetrica A: Transport Science, 18(3):1127–1153.

Tsekeris, T. and Stathopoulos, A., 2006. Real-time traffic volatility forecasting in urban arterial networks. Transportation Research Record, 1964(1):146–156.

Wang, Y. and Papageorgiou, M., 2005. Real-time freeway traffic state estimation based on extended kalman filter: a general approach. Transportation Research Part B: Methodological, 39(2):141–167.

Wang, Y., Papageorgiou, M., and Messmer, A., 2007. Real-time freeway traffic state estimation based on extended kalman filter: A case study. Transportation Science, 41(2):167–181.

Wang, Z., Su, X., and Ding, Z., 2020. Long-term traffic prediction based on lstm encoder-decoder architecture. IEEE Transactions on Intelligent Transportation Systems, 22(10):6561–6571.

Wu, X., Chow, A. H. F., Zhuang, L., Ma, W., Lam, W. H. K., and Wong, S. C., 2024. Estimation of vehicular journey time variability by bayesian data fusion with general mixture model. IEEE Transactions on Intelligent Transportation Systems, 25(10):13640–13652.

Yamak, P. T., Yujian, L., and Gadosey, P. K., 2019. A comparison between arima, lstm, and gru for time series forecasting. Proceedings of the 2019 2nd international conference on algorithms, computing and artificial intelligence, pages 49–55.

Yang, H., Li, X., Qiang, W., Zhao, Y., Zhang, W., and Tang, C., 2021. A network traffic forecasting method based on sa optimized arima–bp neural network. Computer Networks, 193:108102.

Yang, M., Liu, Y., and You, Z., 2009. The reliability of travel time forecasting. IEEE Transactions on Intelligent Transportation Systems, 11(1):162–171.

Yu, B., Lee, Y., and Sohn, K., 2020. Forecasting road traffic speeds by considering area-wide spatio-temporal dependencies based on a graph convolutional neural network (gcn). Transportation research part C: emerging technologies, 114:189–204.

Zhang, D. and Kabuka, M. R., 2018. Combining weather condition data to predict traffic flow: a gru-based deep learning approach. IET Intelligent Transport Systems, 12(7):578–585.

Zhang, W., Zhu, F., Lv, Y., Tan, C., Liu, W., Zhang, X., and Wang, F.-Y., 2022. Adapgl: An adaptive graph learning algorithm for traffic prediction based on spatiotemporal neural networks. Transportation Research Part C: Emerging Technologies, 139:103659.

Zhang, Y. and Haghani, A., 2015. A gradient boosting method to improve travel time prediction. Transportation Research Part C: Emerging Technologies, 58:308–324.

Zhang, Y., Haghani, A., and Zeng, X., 2014a. Component garch models to account for seasonal patterns and uncertainties in travel-time prediction. IEEE Transactions on Intelligent Transportation Systems, 16(2):719–729.

Zhang, Y., Zhang, Y., and Haghani, A., 2014b. A hybrid short-term traffic flow forecasting method based on spectral analysis and statistical volatility model. Transportation Research Part C: Emerging Technologies, 43:65–78.

Zhang, Z., Li, M., Lin, X., Wang, Y., and He, F., 2019. Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies. Transportation Research Part C: Emerging Technologies, 105:297–322.

Zhao, J., Liu, Z., Sun, Q., Li, Q., Jia, X., and Zhang, R., 2022. Attention-based dynamic spatial-temporal graph convolutional networks for traffic speed forecasting. Expert Systems with Applications, 204:117511.

Zheng, H., Lin, F., Feng, X., and Chen, Y., 2020. A hybrid deep learning model with attention-based conv-lstm networks for short-term traffic flow prediction. IEEE Transactions on Intelligent Transportation Systems, 22(11):6910–6920.

Zheng, Z. and Su, D., 2014. Short-term traffic volume forecasting: A k-nearest neighbor approach enhanced by constrained linearly sewing principle component algorithm. Transportation Research Part C: Emerging Technologies, 43:143–157.

Zhong, C., Wu, P., Zhang, Q., and Ma, Z., 2023. Online prediction of network-level public transport demand based on principle component analysis. Communications in Transportation Research, 3:100093.

Zhong, R., Luo, J., Cai, H., Sumalee, A., Yuan, F., and Chow, A. H. F., 2017a. Forecasting journey time distribution with consideration to abnormal traffic conditions. Transportation Research Part C: Emerging Technologies, 85:292–311.

Zhong, R., Luo, J., Cai, H., Sumalee, A., Yuan, F., and Chow, A. H. F., 2017b. Forecasting journey time distribution with consideration to abnormal traffic conditions. Transportation Research Part C: Emerging Technologies, 85:292–311.

Zhou, F., Yang, Q., Zhong, T., Chen, D., and Zhang, N., 2020. Variational graph neural networks for road traffic prediction in intelligent transportation systems. IEEE Transactions on Industrial Informatics, 17(4):2802–2812.

Zhou, Y., Wang, J., and Yang, H., 2019. Resilience of transportation systems: concepts and comprehensive review. IEEE Transactions on Intelligent Transportation Systems, 20(12): 4262–4276.

Zhu, Y., Ye, Y., Liu, Y., and Yu, J. J. Q., 2022. Cross-area travel time uncertainty estimation from trajectory data: A federated learning approach. IEEE Transactions on Intelligent Transportation Systems, 23(12):24966–24978.

Zhuang, L., Wu, X., Chow, A. H. F., Ma, W., Lam, W. H. K., and Wong, S. C., 2024. Reliability-based journey time prediction via two-stream deep learning with multi-source data. Journal of Intelligent Transportation Systems, pages 1–19.