

Hybrid hidden Markov LSTM for short-term traffic flow prediction

Agnimitra Sengupta¹, Adway Das^{2*}, S. Ilgin Guler^{3*}

^{1,2,3}Department of Civil & Environmental Engineering, The Pennsylvania State University
University Park, PA 16802, USA

¹avs6897@psu.edu, ²amd7293@psu.edu, ³sig123@psu.edu

Abstract

Deep learning (DL) methods have outperformed parametric models such as historical average, ARIMA and variants in predicting traffic variables into short and near-short future, that are critical for traffic management. Specifically, recurrent neural network (RNN) and its variants (e.g. long short-term memory) are designed to retain long-term temporal correlations and therefore are suitable for modeling sequences. However, multi-regime models assume the traffic system to evolve through multiple states (say, free-flow, congestion in traffic) with distinct characteristics, and hence, separate models are trained to characterize the traffic dynamics within each regime. For instance, Markov-switching models with a hidden Markov model (HMM) for regime identification is capable of capturing complex dynamic patterns and non-stationarity. Interestingly, both HMM and LSTM can be used for modeling an observation sequence from a set of latent or hidden state variables. In LSTM, the latent variable is computed in a deterministic manner from the current observation and the previous latent variable, while, in HMM, the set of latent variables is a Markov chain. Inspired by research in natural language processing, a hybrid hidden Markov-LSTM model that is capable of learning complementary features in traffic data is proposed for traffic flow prediction. Results indicate significant performance gains in using hybrid architecture compared to conventional methods such as Markov switching ARIMA and LSTM.

Introduction

Accurate traffic predictions in the short or near-short term future, spanning from 5 minutes to 1 hour, play a vital role in efficient traffic management, encompassing traffic control and congestion mitigation. The effectiveness of various traffic control strategies, such as ramp metering or detour suggestions, heavily depends on precise traffic forecasting in the near future. However, achieving precise forecasting across both free-flow and congested traffic states is often challenging due to the inherent uncertainty and chaotic characteristics of transportation systems. Numerous statistical models, both parametric and non-parametric, have been developed to accurately model the temporal aspects of traffic data. Parametric models including historical average (HA) algorithms, autoregressive integrated moving average (ARIMA)

(Ahmed and Cook 1979; Levin and Tsao 1980) fails to uncover complex traffic dynamics as shown in (Davis and Nihan 1991; Hamed, Al-Masaeid, and Said 1995). To partially adapt to the complexities of traffic dynamics, multi-variable prediction models (Innamaa 2000; Dougherty and Cobbett 1997; Florio and Mussone 1996; Lyons et al. 1996) and state-space models (Okutani and Stephanedes 1984; Chen and Chien 2001; Chien and Kuchipudi 2003) were developed. Additionally, trend retrieval using simple-average, principle component analysis (PCA), and wavelet methods have been discussed in the literature (Chen et al. 2012; Xie, Zhang, and Ye 2007) to account for the apparent similarity of daily traffic flow time series.

Alternately, multi-regime prediction models assume that a traffic system evolves through multiple regimes or states (say, free-flow, congestion) with distinct characteristics, and separate regression models are developed to predict traffic flow within each regime (Cetin and Comert 2006; Kamarianakis, Oliver Gao, and Prastacos 2010; Kamarianakis, Shen, and Wynter 2012). These model often use a Hidden Markov model (HMM) (Rabiner 1989) for the identification of traffic regimes. For example, see (Li, Li, and Li 2014; Qi and Ishak 2014; Zhu et al. 2016). Although the multi-regime models are observed to identify the local trend within the time series more efficiently, the overall performance of these models was not significantly improved due to errors incurred while switching between regimes (Li, Li, and Li 2014).

Despite their reasonable performances, the specific functional form and methodological assumptions of the parametric models limit their capabilities to adapt to non-linearities associated with short-term trends – which is a major shortcoming. Moreover, traffic data is observed to exhibit chaotic behaviour during congestion which makes it highly unstable (Disbro and Frame 1989). To the contrary, non-parametric techniques do not specify any functional form, rather they rely on pattern recognition to handle large data quantities. As a result, these approaches can better model traffic patterns with greater transferability and robustness across datasets (Smith and Demetsky 1997; Clark 2003). Nearest neighbors algorithms (Smith, Williams, and Oswald 2002), support vector machine (Mingheng et al. 2013), and Bayesian network (Sun, Zhang, and Yu 2006) are among the machine learning (ML) models that have demonstrated successful performances in small-scale traffic prediction problems.

*These authors contributed equally.

However, the success of such models often depend on suitable feature definition that require engineering judgement (Lin et al. 2019). Deep learning (DL) models use a multi-layer neural framework to capture complex relations in non-linear data (LeCun, Bengio, and Hinton 2015), that require few to negligible feature engineering. Following its success, DL models have been extensively used in traffic time series modeling (Chang and Su 1995; Innamaa 2000; Dia 2001; Park and Rilett 1999). Specifically, recurrent neural networks (RNN) (Rumelhart, Hinton, and Williams 1986) and its variants like long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997) are designed to preserve temporal correlations between observations in a time series, and hence better suited for traffic forecasting as well (Ma et al. 2015; Yu et al. 2017; Cui et al. 2020; Yao et al. 2019, 2018).

Recent researches in natural language processing highlights the structural similarity between RNN and HMM, and their capability to learn complementary features from language data (Krakovna and Doshi-Velez 2016; Salaün, Petetin, and Desbouvries 2019). TConsequently, the use of hybrid models combining both RNN (specifically, Long Short-Term Memory or LSTM) and HMM can provide improved modeling capabilities for complex sequential data, such as traffic time series. This study focuses on investigating hybrid models that leverage the joint usage of HMM and LSTM for the task of traffic flow prediction. Additionally, we explore the feasibility of incorporating duration-based state transitions within the HMM framework in these models.

The remainder of the paper is organized as follows. First, we present an overview on hidden (semi-) Markov models and LSTM, followed by the proposed hybrid models in this study. Next, the performance of hybrid models are compared with the baseline models. Finally, some concluding remarks are presented.

Background

In this section, we provide a background on hidden Markov models (HMMs) and discuss the sojourn time distributions that are considered in this study. Subsequently, we present an overview of the long short-term memory (LSTM) model, which serves as the basis for the modifications that will be described in the following section.

Hidden Markov models

Markov chains model dynamical systems with the assumption that the state of the system at a time t *only* depends on the state in the immediately prior time step, $t - 1$. However, such an assumption often does not hold true for complex dynamic systems. An alternative to Markov chains, the hidden Markov model (HMM) (Rabiner 1989, 1993) assumes the existence of a latent (hidden) process that follows a Markov chain from which observations \mathbf{X} are generated. Therefore, for an observation sequence $\mathbf{X} = \{x_1, x_2, \dots, x_T\}$ in $[1, T]$, there exists an unobserved state sequence $\mathbf{Z} = \{z_1, z_2, \dots, z_T\}$, where the hidden states, z_t belonging to state-space $Q \triangleq \{q_1, q_2, \dots, q_M\}$ follow a Markov chain governed by:

- a state-transition probability matrix $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{M \times M}$ where $a_{ij} = p(z_{t+1} = q_j | z_t = q_i)$
- initial state matrix $\pi = [\pi_i] \in \mathbb{R}^{1 \times M}$ with $\pi_i = p(z_1 = q_i)$ (i.e., the prior)

Further, for each hidden state z_t , corresponding observation x_t is a realization of an emission process $\mathbf{B} = [b_j(x)]$ where $b_j(x) = p(x | z = q_j)$. We assume $b_j(x)$ follows a Gaussian mixture model (GMM) as defined in Equation 1.

$$p(x_t | z = q_j) = \sum_{l=1}^k c_{jl} \mathcal{N}(x_t | \mu_{jl}, \Sigma_{jl}) \quad (1)$$

where $\sum_{l=1}^k c_{jl} = 1, \forall j = \{1, \dots, M\}$, k is the number of Gaussian mixture components and $\mathcal{N}(x_t | \mu_{jl}, \Sigma_{jl})$ denotes a Gaussian probability density with mean μ_{jl} and covariance Σ_{jl} for state j and mixture component l . The number of hidden states (M) and mixture components (k) are the two hyperparameters of the model which have to be provided apriori.

Therefore, the joint probability density function of the observation \mathbf{X} can be expressed as:

$$p(\mathbf{X}) = p(z_1) \prod_{t=1}^{T-1} p(z_{t+1} | z_t) \prod_{t=1}^T p(x_t | z_t) \quad (2)$$

The optimum parameters $[\mathbf{A}, \mathbf{B}, \pi]$ that locally maximize the total observation likelihood (Equation 2) of observation \mathbf{X} , are estimated using an expectation-maximization algorithm, known as the Baum-Welch algorithm (Rabiner 1993). Furthermore, the probability of the system being in a given latent state, z_t corresponding to \mathbf{x}_t is computed using the Viterbi algorithm.

Sojourn time distribution An inherent assumption of the HMM is that the number of time steps (u) spent in a given state q_j (a.k.a sojourn time) is geometrically distributed (denoted by d_j) as shown below.

$$d_j(u) = a_{jj}^u (1 - a_{jj}) \quad (3)$$

However for some dynamical systems, the probability of a state change depends on the time spent in the current state. Therefore, geometrically distributed sojourn time fails to model such systems. An alternate solution is to explicitly estimate the duration density $d(u)$, which results in a hidden *semi*-Markov model (HSMM). In this study, we compare Gamma, Weibull and logarithmic distributions for sojourn density in addition to the default choice of geometric distribution. For each of these assumptions, the parameters of the HSMM model is estimated by maximizing the likelihood of the joint probability density function of \mathbf{X} , as shown in Equation 4 (Guédon 2005).

$$p(\mathbf{X}) = p(z_1) d_{z_1}(u_1) \left\{ \prod_{t=1}^{T-1} p(z_{t+1} | z_t) d_{z_t}(u_t) \right\} p(z_T | z_{T-1}) D_{z_T}(u_T) \prod_{t=1}^T p(x_t | z_t) \quad (4)$$

In the above equation, the survival function, $D_j(u)$, is used to represent the time spent in the last state since the system is not observed beyond time T . Using this survival function improves the parameter estimation and, provides a more accurate prediction of the last state visited.

Long short-term memory

Feed-forward neural network architectures are not explicitly designed to handle sequential data. A class of DL approaches, recurrent neural network (RNN), uses a feedback mechanism where the output from a previous time step is fed as an input to the current time step such that information from the past can propagate into future states. This feedback mechanism preserves the temporal correlation and makes it suitable to capture the temporal evolution of traffic parameters. However, RNNs are incapable of handling the long-term dependencies in temporal data due to the vanishing gradient problem (Hochreiter 1998). Long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997), a type of RNN, consists of memory cells in its hidden layers and several gating mechanisms, which control information flow within a cell state (or, memory) to selectively preserve long-term information.

The objective is to update the cell, C_t , over time using the input x_t and the previous time step's hidden state, h_{t-1} . This process involves several key operations. First, a forget gate, f_t , selectively filters information from the past. Then, an input gate, i_t , regulates the amount of information from the candidate memory cell, \tilde{C}_t , that should be incorporated into the current cell state, C_t . Finally, an output gate, o_t , governs the update of the hidden state, h_t . See Figure 1. The computations are represented as follows:

$$\begin{aligned}\tilde{C}_t &= \tanh(W_c[h_{t-1}, x_t] + b_c) \\ C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \\ h_t &= o_t \odot \tanh(C_t)\end{aligned}\quad (5)$$

The outputs from the forget gate, f_t , input gate, i_t , and output gate, o_t are computed as shown below:

$$\begin{aligned}f_t &= \sigma(W_f[h_{t-1}, x_t] + b_f) \\ i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i) \\ o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o)\end{aligned}\quad (6)$$

Here, σ and \tanh represent non-linear activation functions, while W_f , W_i , W_o , and W_c denote weight matrices corresponding to the forget gate, input gate, output gate, and candidate memory cell, respectively. Similarly, b_f , b_i , b_o , and b_c represent the corresponding bias vectors.

Modeling traffic data

Vehicle arrival rates in traffic are commonly modeled as a Poisson process, assuming that vehicles arrive independently at a specific location or time according to Poisson distribution with a fixed average rate, λ . However, in real-world traffic scenarios, the average arrival rate $\lambda(t)$ fluctuates throughout the day, resulting in a non-homogeneous Poisson process. This means that the rate of vehicle arrivals

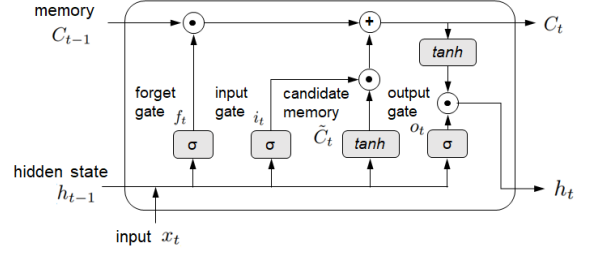


Figure 1: Schematic diagram of LSTM module

vary over time, reflecting changes in traffic flow and congestion levels. This variation in the Poisson average rate captures the dynamic nature of traffic patterns, aligning with real-world observations of different traffic conditions at different times of the day, such as high flow periods during rush hours or lower flow periods during off-peak times.

In our study, we analyze the data representing the fluctuations in vehicle counts, i.e., the changes in vehicle arrival at each time step. While the arrival rates in traffic are commonly modeled as a Poisson process, the observed fluctuation data follows a Skellam distribution, which arises from taking the difference between two Poisson random variables with parameters $\lambda(t)$ and $\lambda(t + \delta t)$, where δt represents the time gap between consecutive observations. Throughout the day, the average rate of vehicle arrivals fluctuates, leading to variations in the parameters of the Skellam distribution. Consequently, the distribution of traffic fluctuation can be described as a mixture of Skellam distributions, with each component representing a specific average arrival rate.

In our approach, we model the temporal sequence of traffic fluctuations using a Hidden Markov Model (HMM). In other words, the HMM tries to categorize the outcome of the traffic fluctuations, which are assumed to be a random variable that is the mixture of Skellam distributions. More specifically, the HMM approximates the output space by employing a mixture of Gaussian distributions, allowing for effective modeling and inference of the underlying traffic patterns. However, since the HMM assumes a continuous distribution while the Skellam is a discrete distribution, the Skellam distribution is normalized using the mean and standard deviation of the Skellam distribution.

Furthermore, it is worth noting that the independence of observations justifies the use of an HMM, as each observation in the traffic data is considered to be independent of previous observations. Moreover, despite the varying average trend in real traffic data throughout the day, the fluctuations tend to exhibit sporadic behavior, indicating little to no duration dependency. To account for this characteristic, multiple duration densities are employed to model the sojourn durations and find the distribution(s) that best fit the data.

Methodology

The hidden Markov model (HMM) and long short-term Memory (LSTM) are two distinct models that generate latent space representations for modeling the distribution of an

observed sequence. While they have structural similarities and divergences, they can be considered as special instances of a more comprehensive framework called the Generative Unified Model (GUM).

In the GUM framework, there exists a hidden or latent state which provides information about the observation. In the HMM, the hidden state z follows Markovian dynamics, meaning that the current state z_t depends only on the previous state z_{t-1} and is conditionally independent of the observation x (as represented by Equation 2). On the other hand, in the LSTM model, the latent variable h_t is deterministic and is a function of the previous latent variable h_{t-1} and the current observation x_t (see Equation 5). The LSTM captures temporal dependencies in the sequence by updating its hidden state representation based on the previous state and current observation.

These structural dissimilarities between the HMM and LSTM result in the two models learning complementary feature representations of an observed sequence. This phenomenon has been highlighted in research conducted in the field of natural language processing (NLP) (Salaün, Petetin, and Desbouvries 2019; Liu, Lin, and Reid 2019; Krakovna and Doshi-Velez 2016). For instance, hybrid RNN-HMM architectures were explored in text sequence modeling (Krakovna and Doshi-Velez 2016), where the HMM modeled the underlying statistical patterns in the input sequence, while the LSTM captured the temporal dependencies using its hidden state representation.

In the context of traffic prediction, we aim to leverage this complementary feature learning phenomenon by combining HMM and LSTM in two proposed architectures. By combining these two models, we can effectively capture both the statistical patterns and the temporal dynamics of the traffic data, leading to enhanced prediction accuracy. This stands in contrast to the baseline model, which is a simple LSTM model (See Figure 2(a)) that solely relies on the temporal history of x to predict its future value. Additionally, unlike multi-regime models that employ separate prediction models for different states, the hybrid models train a single prediction model that captures the system’s evolution within the latent space.

Model architectures

In this study, two architectures of a hybrid HMM-LSTM model are considered: the sequential hybrid (S-Hybrid) and the concatenated hybrid (C-Hybrid).

Sequential hybrid In the sequential hybrid model (S-Hybrid), the first step is to train an HMM on the input sequence \mathbf{X} to learn the probability of the system being in each hidden state $q \in Q$ at time t . The HMM captures the time-evolution of state probabilities based on the observed sequence. These HMM features, representing the probabilities of being in different hidden states, are then used as input to train the LSTM to learn the temporal dependencies in the sequence.

In the sequential Hybrid Model (S-Hybrid), the initial step involves training an HMM on the input sequence \mathbf{X} to estimate the likelihood of the system being in each hidden state

$q \in Q$ at time t . The HMM effectively captures the dynamic changes in state probabilities based on the observed sequence. These HMM features, which represent the probabilities associated with different hidden states, are subsequently utilized as input for training the LSTM network to learn the temporal dependencies within the sequence. The latent outputs (h) from the LSTM are processed through a series of dense layers to generate the final prediction. This S-Hybrid approach effectively combines the probability information obtained from the HMM with the LSTM’s capabilities, enhancing the model’s ability to anticipate state transitions. As a result, it is expected that this modeling approach can potentially lead to improved prediction performance by leveraging the complementary strengths of the HMM and LSTM. See Figure 2(b) for the architecture.

Concatenated hybrid In the concatenated hybrid model (C-Hybrid), the latent outputs from two distinct LSTM networks are combined. One LSTM network is trained on the input sequence \mathbf{X} , while the other LSTM network is trained on the sequence of hidden states obtained from the HMM. These two sets of latent outputs, representing the learned temporal dependencies from both the input sequence and the HMM state sequence, are concatenated together. The concatenated features are then fed into a series of densely connected layers to generate the final prediction. By integrating the latent outputs from the LSTM networks trained on different sequences, the model potentially captures the complementary information and leverages it to enhance prediction accuracy. This approach allows the model to benefit from both the statistical patterns captured by the HMM and the temporal dynamics captured by the LSTM. See Figure 2(c) for the architecture.

Model training

The baseline deep learning (DL) model employed in this study consists of a stacked LSTM architecture. Our model consists of three LSTM layers with 20, 20 and 10 units, followed by four dense layers with 10, 10, 6 and 2 units respectively, with LeakyReLU activation function (Maas et al. 2013) for dense layers respectively. Additionally, a statistical benchmark model called the HMM-based regime-switching autoregressive model (AR-HMM) (Kim, Piger, and Startz 2008) was chosen for comparison. The AR-HMM is a widely recognized model frequently used in traffic data analysis literature.

The hyperparameters of the proposed architectures were selected to ensure that the number of trainable parameters is comparable for each DL model. Specifically, the same architecture is utilized for the S-Hybrid model as the LSTM model, with the only difference being the number of feature channels in S-Hybrid, which is set equal to the number of hidden HMM states considered. In the case of the C-hybrid model, it consists of two branches. Each branch comprises two LSTM layers with 20 and 10 units, respectively. The feature outputs from these branches are merged, and then passed through four dense layers with 10, 6, 6, and 1 units, respectively with LeakyReLU activation.

To ensure the generalizability of the models, the models

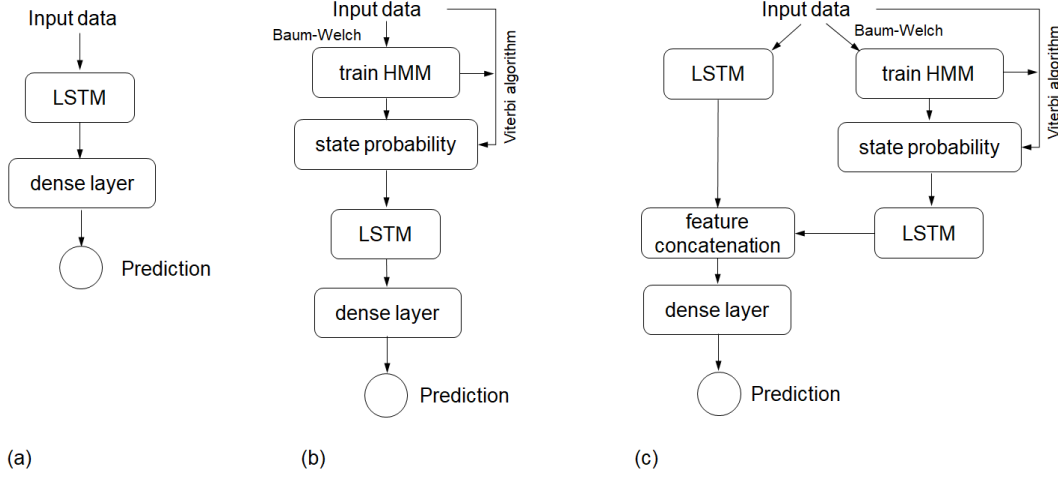


Figure 2: DL architectures considered (a) Stacked LSTM; (b) Sequential Hybrid; (c) Concatenated Hybrid

are trained, validated, and tested on three separate sets. The dataset is divided into three parts: 60% for model training, 15% for validation, and 25% for testing. This division allows for assessing the model's performance on unseen data and helps prevent overfitting. To address overfitting, the model parameters are tuned throughout the training process based on their performance on the validation set. The models are trained to minimize the mean squared error (MSE) loss function for sufficiently large number of epochs until the validation loss starts to increase. The model with the lowest validation error is selected as the final model. We use Adadelta (Zeiler 2012) as the optimizer with the learning rate of 0.20, ρ value of 0.95, and epsilon of $1e-7$ to train the models.

Data

The performance evaluation of the proposed models is conducted on a dataset obtained from the California Department of Transportation's Performance Measurement System (PeMS). This dataset is widely used for traffic data modeling. The traffic data, including flow, occupancy, and speed, is collected from vehicle detector stations (VDS) located along freeways and ramps. The raw data is sampled at 30-second intervals and then aggregated at 5-minute intervals. Flow data for one year (i.e., 104942 samples) from VDS 1114805 on California Interstate 05 NB in District 11 were used for training and testing of the models. The performance of the models are evaluated on 25% of the dataset that was never used in training.

In this study, the models are employed to predict fluctuations in traffic flow, specifically the change in flow between successive time steps (Δ Flow), instead of directly predicting the absolute flow values at a detector location. It is important to note that modeling the first-order difference, i.e., the flow fluctuations, can be advantageous for capturing short-term dynamics in the time series, whereas, modeling the actual time series can be more suitable for capturing long-term trends and patterns (Li, Li, and Li 2014). Figure 3 shows the detrended time series (Δ Flow) corresponding to

the flow observed at the detector during a 48-hour cycle.

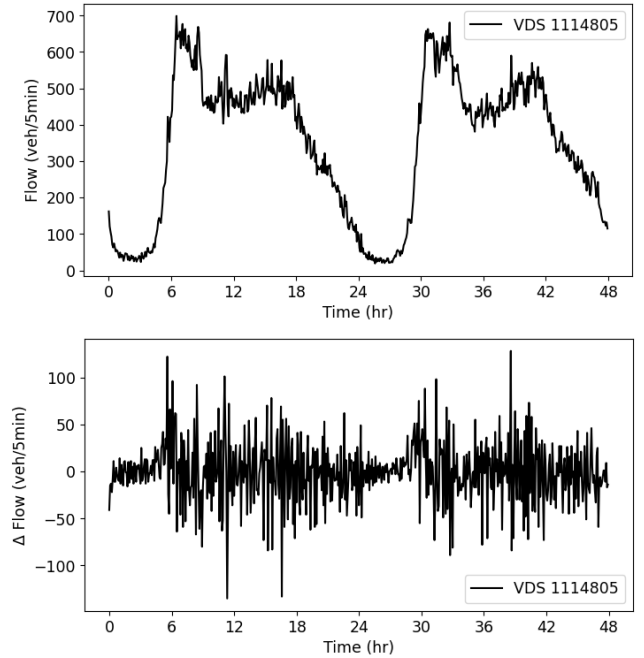


Figure 3: (a) Traffic flow; (b) detrended flow or fluctuation for 24-hr period

Results

In this study, three DL models – 1) a stacked LSTM model trained on flow fluctuations, 2) a stacked LSTM model trained on probabilities of hidden state transition (or, S-Hybrid) and 3) merged LSTM model that takes both flow fluctuations and probabilities of hidden state transition as inputs, are considered for prediction tasks.

We evaluate the model performances for single-step pre-

diction horizons, by comparing the prediction mean with the corresponding true values using three metrics: root mean squared error (RMSE), mean absolute percentage error (MAPE) and R^2 as defined below.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^N [y_i - \hat{y}_i]^2} \quad (7)$$

$$\text{MAPE} = \frac{1}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (8)$$

$$R^2 = 1 - \frac{\sum_{i=1}^N [y_i - \hat{y}_i]^2}{\sum_{i=1}^N [y_i - \bar{y}]^2} \quad (9)$$

where y_i represents the 'ground truth' or true value of the observation i , \hat{y}_i is the predicted value of y_i for $i = 1, 2, \dots, T$.

In the 'Methodology' section, the hybrid models are described as utilizing HMM features derived from the input data, specifically the flow fluctuations (Δ Flow), to perform the prediction task. To enable this, HMMs are trained on the input data with different configurations. In this study, HMMs are trained assuming 3 and 5 latent states, and a Gaussian mixture model is used with either 1 or 2 mixture components.

To characterize the duration densities of the states, various distributions such as geometric, logarithmic, gamma, and Weibull distributions are assessed. For each case studied, the AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion) are computed to select the models based on a trade-off between model fit and complexity. The results are presented in Table 1. It is worth noting that the AIC and BIC values for models with 1 and 2 Gaussian mixture components are consistently similar. Hence, for the sake of brevity and reduced model complexity, results obtained with 1 Gaussian component are reported. Across all the evaluated distributions, the AIC and BIC values indicate that a model with 3 latent states is preferable. This configuration provides a good balance between model fit and complexity. On the other hand, using 5 latent states introduces more complexity, without substantial improvement in model performance, resulting in higher AIC and BIC values.¹

Using the trained hidden (semi) Markov models, the most likely states for each time instant is identified as shown in Figures 4 and 5. As can be seen, the system dynamically transitions between the hidden states within the day. However, due to different sojourn densities, the identified states and their duration are quite different. In case of geometric and logarithmic sojourn density, the system is observed to exhibit abrupt state changes as flow increases and congestion builds on the roadway. Therefore, traffic is observed to be unstable during congestion, as highlighted by (Disbro and Frame 1989). Conversely, the system is observed to remain in each state for an increased time duration when Gamma

and Weibull sojourn distributions are assumed. Therefore, abrupt fluctuations in flow during congestion are not efficiently captured in these two cases.

The performance of the S-Hybrid and C-Hybrid DL models for different configurations of hidden (semi) Markov models are compared in Tables 2 and 3 respectively. Upon analyzing the results, it is evident that models utilizing geometric and logarithmic sojourn densities demonstrate superior performance compared to models using Gamma and Weibull sojourn densities. This observation is consistent with the characteristics of the observed state transitions in the time-series data as depicted in Figures 4 and 5.

The geometric distribution assumes a constant probability of transitioning from one state to another, regardless of the duration already spent in the current state. On the other hand, the logarithmic distribution considers the time already spent in a state, allowing for a wider range of durations and better adaptation to the observed patterns in the traffic data. This flexibility in capturing varying durations enables the logarithmic sojourn density to more effectively model the complex dynamics of traffic behavior, leading to slightly better performance compared to the geometric sojourn density. This aligns well with the assumption of vehicle arrival being modeled as a Poisson process and the (near) independence assumption.

In this study, a logarithmic sojourn density with 5 states and 1 Gaussian emission is found to marginally outperform the geometric sojourn density in terms of prediction accuracy for both the S-Hybrid and C-Hybrid DL models. The optimized parameters for the fitted HMM model with logarithmic sojourn density, 5 states, and 1 Gaussian emission distribution are denoted by Equation 10.

$$\begin{aligned} A &= \begin{pmatrix} 0.000 & 0.0161 & 0.258 & 0.714 & .012 \\ 0.109 & 0.000 & 0.009 & 0.046 & 0.836 \\ 0.441 & 0.002 & 0.000 & 0.548 & 0.009 \\ 0.768 & 0.008 & 0.144 & 0.000 & 0.081 \\ 0.038 & 0.572 & 0.029 & 0.361 & 0.000 \end{pmatrix} \\ \pi &= [1 \quad 0 \quad 0 \quad 0 \quad 0] \\ p &= [0.307 \quad 0.455 \quad 0.934 \quad 0.420 \quad 0.245] \\ s &= [1 \quad 1 \quad 1 \quad 1 \quad 1] \\ \mu &= [-0.8594 \quad -0.7903 \quad -0.0098 \quad 0.8144 \quad -0.8524] \\ \sigma &= [0.3430 \quad 6.1145 \quad 0.1494 \quad 0.4009 \quad 2.3412] \end{aligned} \quad (10)$$

where A represents the hidden state-transition matrix, π corresponds to the initial state probabilities, p and s denote the scale and shift parameters of the sojourn density, and μ and σ represent the parameters of the Gaussian emission function.

The proposed DL models with logarithmic sojourn density with 5 states and 1 Gaussian emission are compared with two baseline models – a stacked LSTM and HMM-based regime-switching autoregressive model (AR-HMM) with lags 1, 10 and 20. As observed from Table 4, the hybrid models perform significantly better than the baseline models, with C-Hybrid outperforming S-Hybrid. Figure 6

¹The computations for these evaluations are conducted using the Hidden Markov Model package (Amini, Bayat, and Salehian 2022).

Table 1: Comparison of AIC and BIC values for different HMM configurations

Metric	State	Sojourn distribution			
		Geometric	Logarithmic	Gamma	Weibull
AIC	3	196652.6	192302.1	216200.9	213386.6
	5	216445.4	201293.3	242818.6	233851.3
BIC	3	196819.5	192496.8	216395.6	213581.4
	5	216816.3	201692	243217.3	234250

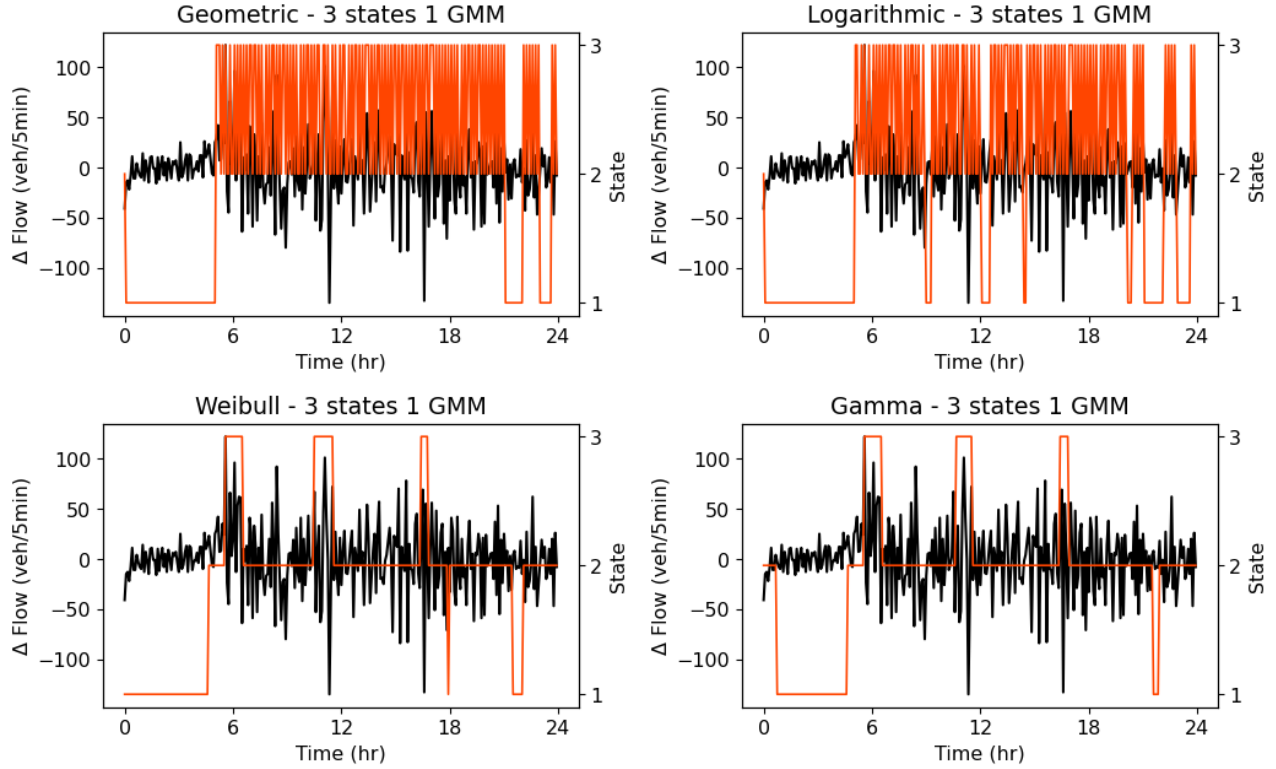


Figure 4: Identified hidden states by different sojourn densities with 3 latent states and 1 Gaussian mixture component

Table 2: Performance comparison of S-Hybrid DL models with different sojourn densities

Metric	State	Sojourn distribution			
		Geometric	Logarithmic	Gamma	Weibull
RMSE	3	0.6163	0.6229	0.9486	0.9493
	5	0.5635	0.5326	0.9457	0.9447
R^2	3	0.5882	0.5794	0.0246	0.0230
	5	0.6557	0.6925	0.0305	0.0323
MAPE	3	168.8370	155.0041	52.4545	48.9196
	5	167.3981	142.0781	47.6088	48.2944

demonstrates the prediction performance for a 24-hour period. It is evident from the figure that AR-HMM and LSTM follow similar trends, while the hybrid models perform significantly better to capture the abrupt flow changes. However, in the free-flow regime (2 to 6 hrs), S-Hybrid fails to capture the trend. This is due to the fact that the system pre-

dominantly remains in state 3 during free-flow and hence, the input to the LSTM i.e., the hidden state probabilities over time does not change appreciably. Therefore, the model generates a near constant output. To the contrary, the predictions of C-Hybrid performs comparatively better than S-Hybrid to capture the low-fluctuations.

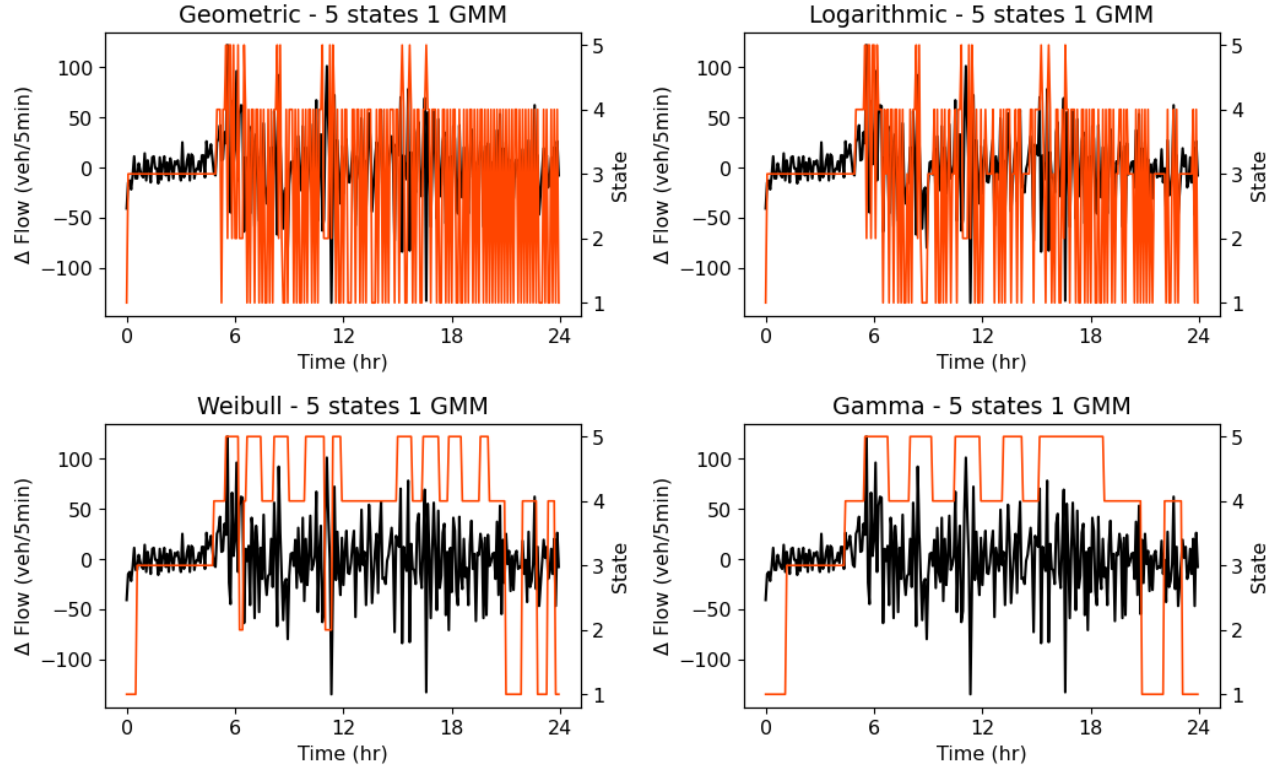


Figure 5: Identified hidden states by different sojourn densities with 5 latent states and 1 Gaussian mixture component

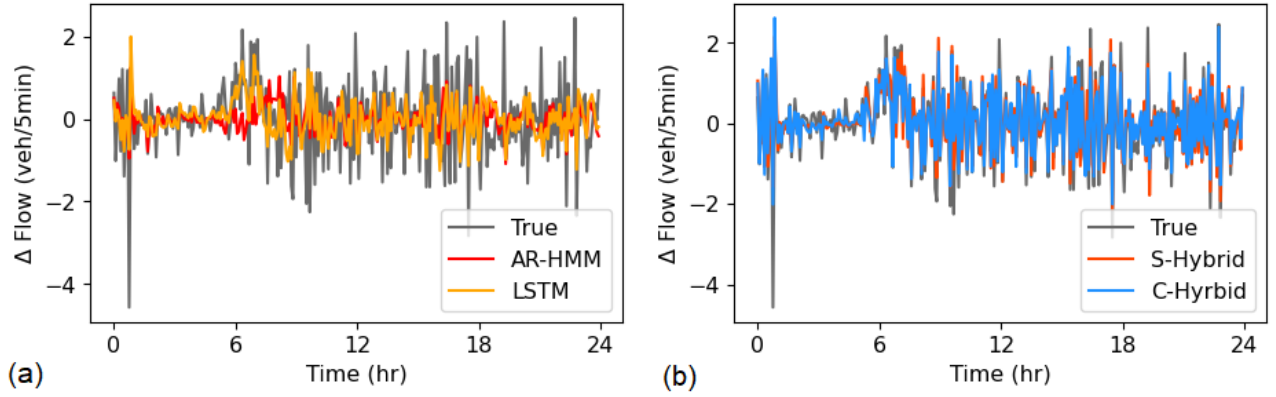


Figure 6: Performance comparison of (a) LSTM and AR-HMM models, (b) Hybrid models in 24-hr period

Further, we compare the feature-space representations of the penultimate layers of the models to identify specific patterns that enhance prediction capabilities of hidden Markov-LSTM models. We use *t*-Stochastic Neighbor Embedding, a non-linear technique for dimension reduction, to reduce the high dimensional feature output to two dimensions (Van der Maaten and Hinton 2008). Flow fluctuation data were labeled into four traffic regimes: 1) low flow (0 to 6 hr), 2) increasing flow (6 to 8 hr), 3) high flow (8 - 18 hr) or congestion and 4) decreasing flow (18 to 24 hr) which are suitably color-coded as shown below.

Figure 8 illustrates the learned feature space for the four different regimes obtained using the stacked LSTM, S-Hybrid, and C-Hybrid models. When considering the LSTM model, it is evident that different traffic states overlap, making it challenging to distinguish between them. However, with the incorporation of HMM features into the model, we observe clear separations in the feature space for the hybrid models. Notably, the C-Hybrid model exhibits superior separability for low flow traffic states (as shown in Figure 8), which likely contributes to its superior performance. Table 5 presents a comparison of the variances of outputs belonging

Table 3: Performance comparison of C-Hybrid DL models with different sojourn densities

Metric	State	Sojourn distribution			
		Geometric	Logarithmic	Gamma	Weibull
RMSE	3	0.4631	0.4898	0.8106	0.8167
	5	0.4245	0.4203	0.8055	0.8019
R^2	3	0.7675	0.7400	0.2876	0.2769
	5	0.8046	0.8085	0.2967	0.3029
MAPE	3	150.3153	148.6007	146.0438	147.1233
	5	138.7888	116.3367	143.3267	145.9820

Table 4: Performance comparison of models

Model name	RMSE	R^2	MAPE
LSTM	0.8235	0.2648	164.9522
1-lag AR-HMM	0.9083	0.1046	98.0325
10-lag AR-HMM	0.8843	0.1522	109.7145
20-lag AR-HMM	0.8766	0.1669	125.6203
S-Hybrid	0.5326	0.6925	142.0781
C-Hybrid	0.4203	0.8085	116.3367

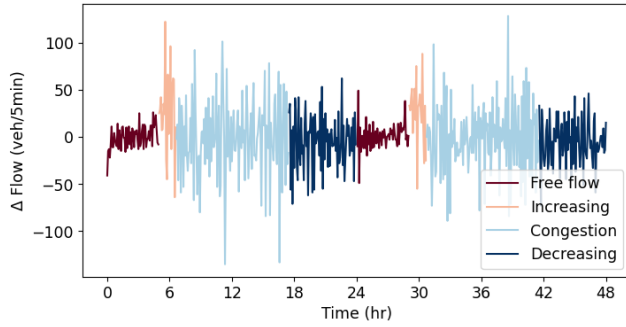


Figure 7: Flow and fluctuation segmented based on regimes

to specific traffic regimes in the 6-dimensional feature space for each model. It is worth noting that the LSTM model demonstrates high dispersion of outputs within the feature space for data from the same traffic states, along with significant overlap between data from multiple regimes. In contrast, the hybrid models incorporating HMM features effectively localize features in the space, resulting in enhanced performance.

Table 5: Variance of 6-dimensional outputs of models for different traffic flow regimes

Regime	LSTM	S-Hybrid	C-Hybrid
Low	0.1709	0.0470	0.0724
Increasing	0.1458	0.0294	0.0783
High	0.2890	0.1284	0.0862
Decreasing	0.2399	0.1046	0.0769

Conclusions

Hidden Markov model (HMM) and recurrent neural networks like Long short-term memory (LSTM) are capable of modeling an observation sequence from a set of latent (hidden) state variables. The latent variables in LSTM are determined in a deterministic manner from the current observation and the previous latent variable, while, in HMM, the set of latent variables is a Markov chain. Recent research highlights the structural similarity between LSTM and HMM, and their capability to learn complementary features from input data. Therefore, appropriate hybridization of these models could lead to a better modeling of the data compared to the individual models.

Our study adopts a hybrid approach combining a HMM and LSTM to model the temporal sequence of traffic fluctuations. Specifically, the HMM allows to capture the underlying patterns and state changes in traffic dynamics that typically are assumed to have a Poisson distribution. The HMM's capability to model the short-term fluctuations in traffic, often resembling a Skellam distribution, enables us to accurately characterize the variability in traffic behavior. Additionally, we incorporate the LSTM, which retains long-term temporal correlations, allowing us to capture complex dynamic patterns and non-stationarity within the traffic data. As shown in this study, hybrid models that jointly use HMM and LSTM to perform the task of traffic flow prediction outperform in terms of prediction accuracy the LSTM and auto-regressive HMM regime switching models in capturing chaotic behavior in traffic data by learning complementary features. The testing of the models on loop detector data shows that the the LSTM and AR-HMM models result in an RMSE of 0.8235 and 0.8766, respectively, while the S-Hybrid and C-Hybrid models result in an RMSE of 0.5326 and 0.4203, respectively, which corresponds to an approximately 31-52% improvement in performance. Transferability of hybrid hidden Markov-LSTM models to predict traffic on out-of-distribution datasets will be explored in future studies.

References

- Ahmed, M. S.; and Cook, A. R. 1979. *Analysis of freeway traffic time-series data by using Box-Jenkins techniques*. 722.
- Amini, M.; Bayat, A.; and Salehian, R. 2022. hhsmm: An

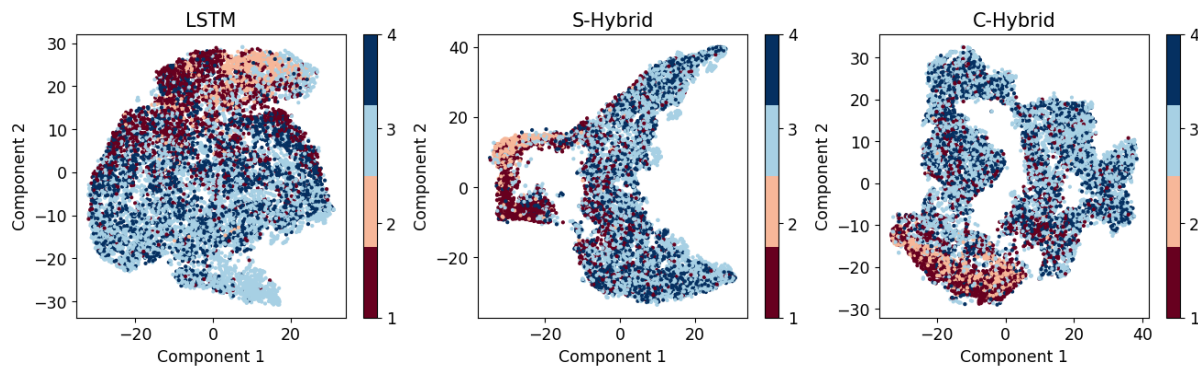


Figure 8: Feature space visualizations of layer outputs of LSTM, S-Hybrid and C-Hybrid models

R package for hidden hybrid Markov/semi-Markov models. *Computational Statistics*, 1–53.

Cetin, M.; and Comert, G. 2006. Short-term traffic flow prediction with regime switching models. *Transportation Research Record*, 1965(1): 23–31.

Chang, G.-L.; and Su, C.-C. 1995. Predicting intersection queue with neural network models. *Transportation Research Part C: Emerging Technologies*, 3(3): 175–191.

Chen, C.; Wang, Y.; Li, L.; Hu, J.; and Zhang, Z. 2012. The retrieval of intra-day trend and its influence on traffic prediction. *Transportation Research Part C: Emerging Technologies*, 22: 103–118.

Chen, M.; and Chien, S. I. 2001. Dynamic freeway travel-time prediction with probe vehicle data: Link based versus path based. *Transportation Research Record*, 1768(1): 157–161.

Chien, S. I.-J.; and Kuchipudi, C. M. 2003. Dynamic travel time prediction with real-time and historic data. *Journal of transportation engineering*, 129(6): 608–616.

Clark, S. 2003. Traffic prediction using multivariate non-parametric regression. *Journal of transportation engineering*, 129(2): 161–168.

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

Davis, G. A.; and Nihan, N. L. 1991. Nonparametric regression and short-term freeway traffic forecasting. *Journal of Transportation Engineering*, 117(2): 178–188.

Dia, H. 2001. An object-oriented neural network approach to short-term traffic forecasting. *European Journal of Operational Research*, 131(2): 253–261.

Disbro, J. E.; and Frame, M. 1989. Traffic flow theory and chaotic behavior. Technical report, New York (State). Dept. of Transportation.

Dougherty, M. S.; and Cobbett, M. R. 1997. Short-term inter-urban traffic forecasts using neural networks. *International journal of forecasting*, 13(1): 21–31.

Florio, L.; and Mussone, L. 1996. Neural-network models for classification and forecasting of freeway traffic flow stability. *Control Engineering Practice*, 4(2): 153–164.

Guédon, Y. 2005. Hidden hybrid Markov/semi-Markov chains. *Computational Statistics & Data Analysis*, 49(3): 663–688.

Hamed, M. M.; Al-Masaeid, H. R.; and Said, Z. M. B. 1995. Short-term prediction of traffic volume in urban arterials. *Journal of Transportation Engineering*, 121(3): 249–254.

Hochreiter, S. 1998. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02): 107–116.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Innamaa, S. 2000. Short-term prediction of traffic situation using MLP-neural networks. In *Proceedings of the 7th world congress on intelligent transport systems, Turin, Italy*, 6–9.

Kamarianakis, Y.; Oliver Gao, H.; and Prastacos, P. 2010. Characterizing regimes in daily cycles of urban traffic using smooth-transition regressions. *Transportation Research Part C: Emerging Technologies*, 18(5): 821–840. Applications of Advanced Technologies in Transportation: Selected papers from the 10th AATT Conference.

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.

Kim, C.-J.; Piger, J.; and Startz, R. 2008. Estimation of Markov regime-switching regression models with endogenous switching. *Journal of Econometrics*, 143(2): 263–273.

Krakovna, V.; and Doshi-Velez, F. 2016. Increasing the interpretability of recurrent neural networks using hidden markov models. *arXiv preprint arXiv:1606.05320*.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *nature*, 521(7553): 436–444.

Levin, M.; and Tsao, Y.-D. 1980. On forecasting freeway occupancies and volumes (abridgment). *Transportation Research Record*, (773).

- Li, Z.; Li, Y.; and Li, L. 2014. A comparison of detrending models and multi-regime models for traffic flow prediction. *IEEE Intelligent Transportation Systems Magazine*, 6(4): 34–44.
- Lin, F.; Xu, Y.; Yang, Y.; and Ma, H. 2019. A spatial-temporal hybrid model for short-term traffic prediction. *Mathematical Problems in Engineering*, 2019.
- Liu, L.; Lin, Y.-C.; and Reid, J. 2019. Improving the Performance of the LSTM and HMM Models via Hybridization. *ArXiv*, abs/1907.04670.
- Lyons, G.; McDonald, M.; Hounsell, N.; Williams, B.; Cheese, J.; and Radia, B. 1996. Urban traffic management; the viability of short term congestion forecasting using artificial neural networks. PTRC.
- 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.
- Maas, A. L.; Hannun, A. Y.; Ng, A. Y.; et al. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, 3. Atlanta, Georgia, USA.
- Mingheng, Z.; Yaobao, Z.; Ganglong, H.; and Gang, C. 2013. Accurate multisteps traffic flow prediction based on SVM. *Mathematical Problems in Engineering*, 2013.
- Okutani, I.; and Stephanedes, Y. J. 1984. Dynamic prediction of traffic volume through Kalman filtering theory. *Transportation Research Part B: Methodological*, 18(1): 1–11.
- Park, D.; and Rilett, L. R. 1999. Forecasting freeway link travel times with a multilayer feedforward neural network. *Computer-Aided Civil and Infrastructure Engineering*, 14(5): 357–367.
- Qi, Y.; and Ishak, S. 2014. A Hidden Markov Model for short term prediction of traffic conditions on freeways. *Transportation Research Part C: Emerging Technologies*, 43: 95–111.
- Rabiner, L. 1993. Fundamentals of speech recognition. *Fundamentals of speech recognition*.
- Rabiner, L. R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2): 257–286.
- Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1986. Learning representations by back-propagating errors. *nature*, 323(6088): 533–536.
- Salaün, A.; Petetin, Y.; and Desbouvries, F. 2019. Comparing the Modeling Powers of RNN and HMM. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 1496–1499.
- Smith, B. L.; and Demetsky, M. J. 1997. Traffic flow forecasting: comparison of modeling approaches. *Journal of transportation engineering*, 123(4): 261–266.
- Smith, B. L.; Williams, B. M.; and Oswald, R. K. 2002. Comparison of parametric and nonparametric models for traffic flow forecasting. *Transportation Research Part C: Emerging Technologies*, 10(4): 303–321.
- Sun, S.; Zhang, C.; and Yu, G. 2006. A Bayesian network approach to traffic flow forecasting. *IEEE Transactions on intelligent transportation systems*, 7(1): 124–132.
- Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).
- Xie, Y.; Zhang, Y.; and Ye, Z. 2007. Short-term traffic volume forecasting using Kalman filter with discrete wavelet decomposition. *Computer-Aided Civil and Infrastructure Engineering*, 22(5): 326–334.
- Yao, H.; Tang, X.; Wei, H.; Zheng, G.; and Li, Z. 2019. Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 5668–5675.
- Yao, H.; Wu, F.; Ke, J.; Tang, X.; Jia, Y.; Lu, S.; Gong, P.; Ye, J.; and Li, Z. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
- Yu, R.; Li, Y.; Shahabi, C.; Demiryurek, U.; and Liu, Y. 2017. Deep learning: A generic approach for extreme condition traffic forecasting. In *Proceedings of the 2017 SIAM international Conference on Data Mining*, 777–785. SIAM.
- Zeiler, M. D. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.
- Zhu, G.; Song, K.; Zhang, P.; and Wang, L. 2016. A traffic flow state transition model for urban road network based on Hidden Markov Model. *Neurocomputing*, 214: 567–574.