Imperial College
London

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

Short-Term Traffic Prediction: Modelling Temporal-Spatial Features in Local Highway Networks with Deep Neural Networks

Author:
Jingqing Zhang

Supervisor:
Prof. Yike Guo
Dr. Chao Wu

Submitted in partial fulfillment of the requirements for the MRes degree in Advanced Computing of Imperial College London

September 2017

Abstract

Both temporal and spatial features provide significant implications for short-term traffic volume prediction. The problem is challenging due to various non-linear temporal dynamics at different locations, complicated spatial dependencies and difficulty for longer-step ahead forecasting. We propose two deep learning models, CNN-LSTM with attention mechanism (CNN-LSTM-Attn) and Temporal-Spatial-LSTM (TS-LSTM) to incorporate temporal and spatial correlations. Experiments show that both models outperform baselines on the Highways England dataset and the CNN-LSTM-Attn achieves lowest MAPE 9.26% on 2-hour traffic volume prediction. We also evaluate the CNN-LSTM-Attn on the KDDCUP17 dataset and our model defeats the model that got first place in the competition with lower MAPE 10.48%. Our models achieve 2-hour forecasting, which is longer than previous literature, with outstanding accuracy and robustness.

Acknowledgements

I would first like to thank Prof. Yike Guo and Dr. Chao Wu for their supervision and support throughout my master study and this project. Thanks for the opportunity they provide to work in Data Science Institute. Then I would like to thank LexisNexis Corporation for their kindly financial support.

I would also like to thank my parents, my fiancée and my family for their great support and care from home and overseas. I am grateful to my colleagues and friends, Mr. Simiao Yu, Mr. Hao Dong, Mr. Akara Supratak, Dr. David Birch, Dr. Miguel Molina-Solana, Dr. Fangde Liu, Dr. Kai Sun, Mr. Yuanhan Mo, Mr. Xiaoping Fan, Ms. Pan Wang for their kind advice.

Contents

Chapter 1

Introduction

Short-term traffic volume prediction can be essential for Intelligent Transportation Systems (ITSs), traffic management authorities and travellers. During peak hours, accurate prediction and abnormal traffic detection can be especially important for authorities to regulate specific crowded roads and travellers to choose proper routes. Many navigation services have already integrated ETA (Estimated Time of Arrival) calculation services but circumstances in different places can be locally various. Traffic flow prediction has become a challenging task due to various non-linear temporal dynamics at different locations, complicated underlying spatial dependencies and difficulty for longer-step ahead forecasting. Besides, traffic can be potentially influenced by lots of environmental and artificial factors, such as weather, holidays, special events, accidents, and etc..

More sensors and cameras, which have been deployed along roads and the number of which increases rapidly in recent years, make it possible to collect huge amount of data for traffic analysis, though measurement error can be a concern. With the increasing popularity of Internet of Things (IoT), not only larger amount of data and also data from variety of new domains can be expected in the future. In addition to the abundance of data and sources, the exponentially growing computing power allows researchers to implement and run more complex models with efficiency.

Mathematically, the short-term traffic volume prediction can be formalised as follows. Let x_t be one observation of traffic volume at timestamp t at a specific location. Given a sequence of observations $\{x_1, x_2, ..., x_t\}$, a model $\mathcal{M}$ with its parameters θ outputs prediction of observations in the following time period $\{x_{t+1}, x_{t+2}, ...x_{t+H}\}$ with highest probability.

$$\hat{x}_{t+1}, \hat{x}_{t+2}, ..., \hat{x}_{t+H} = \underset{x_{t+1}, x_{t+2}, ..., x_{t+H}}{\arg\max} P(x_{t+1}, x_{t+2}, ..., x_{t+H} | x_1, x_2, ..., x_t; \mathcal{M}; \theta) \quad (1.1)$$

Considering spatial correlation among different locations, we can further define a $K \times T$ matrix $\{X_1, X_2, ..., X_T\}$ which include observations at K locations. The traffic prediction problem with both temporal and spatial information can be rewritten as:

$$\hat{X}_{t+1}, \hat{X}_{t+2}, ..., \hat{X}_{t+H} = \underset{X_{t+1}, X_{t+2}, ..., X_{t+H}}{\arg\max} P(X_{t+1}, X_{t+2}, ..., X_{t+H} | X_1, X_2, ..., X_t; \mathcal{M}; \theta) \tag{1.2}$$

Previous literature analysed short-term traffic prediction commonly based on three approaches, namely classic time series analysis, machine learning and data assimilation. Box et al. (2015) introduces a family of time series analysis methods, especially SARIMA and VAR which are popular choices in traffic prediction. Lippi et al. (2013) quantitatively compares performance of non-parametric (random walk, seasonal mean) and parametric models (SARIMA, ARIMA, SVR, ANN), and SARIMA achieves overall best accuracy. Kamarianakis and Prastacos (2003, 2005) incorporate spatial correlation among neighbouring locations and achieve promising results by VAR and space-time ARIMA (STARIMA). Sun et al. (2006) proposes a new approach based on Baysian Network and treats the joint probability distribution between the cause node and the effect node as a Gaussian mixture model. Besides, Kalman Filter has also been applied for traffic prediction. Considering observations and models are neither assumed to be accurate due to underlying factors that may be unknown, data assimilation methods, e.g. Kalman Filter, can be helpful to balance observations and prior models, and estimate real values in series with posterior distribution. Okutani and Stephanedes (1984); Xie et al. (2007) have shown promising results of Kalman Filter in traffic volume prediction problem. Guo et al. (2014) introduces adaptive Kalman Filter to improve stability and real-time processing for SARIMA+GARCH structure. In addition to the three popular approaches, Xue and Shi (2008) introduces chaos time series theory with two-step optimization for traffic volume forecasting and Yang et al. (2014) demonstrates similarity search in historical traffic database can improve prediction accuracy.

Recent research interest on traffic prediction has been moved to deep learning models. Chen et al. (2016) proposes a hybrid method combining KNN and fully connected neural network and suggests its advantage in highways traffic prediction. Duan et al. (2016) utilises RNN with LSTM on highways traffic time prediction and achieve good accuracy on 1-step ahead prediction. Shahsavari and Abbeel (2015) uses graph neural networks which is able to extract non-linear spatial and temporal relations in transportation network. Similarly, Li et al. (2017) proposes a graph convolutional RNN to capture both spatial and temporal correlations.

The tasks in this project aim to forecast traffic volume in the next 2 hours in highways network based on historical observations. Prior data analysis have demonstrated consistent temporal seasonality, special patterns on special days as well as significant spatial correlation among neighbouring locations. We propose two deep learning models, CNN-LSTM with temporal attention mechanism (CNN-LSTM-Attn)

and Temporal-Spatial-LSTM (TS-LSTM) with spatial relational module, to model temporal and spatial dependencies on traffic flow. The CNN-LSTM-Attn outperforms the baselines with lowest MAPE 9.26% on the Highways England dataset and defeats the model that got first place in the KDDCUP17 competition with lower MAPE 10.48%.

The contributions of this projects can be summarised as below:

1. The CNN-LSTM-Attn is able to extract temporal dependencies without supervision. Though traffic patterns at different locations are various, the CNN-LSTM-Attn is general and only one model is needed for the whole transportation network.
2. The TS-LSTM can incorporate both temporal and spatial dependencies among neighbouring locations and is not necessarily privy to which spatial relation may exist nor the actual meaning of particular relation.
3. Our models can predict 2-hour ahead observations in the future which is longer than that in previous literature (commonly prediction within 30 minutes).
4. The data analysis quantitatively demonstrates and visualises temporal patterns and spatial correlations in the transportation network.

In the Chapter 2, we will introduce background information and related literatures of baselines and deep learning. The Chapter 3 will firstly introduce the datasets we use and then demonstrate discovery on temporal patterns and spatial correlations. Our proposed methodologies based on deep learning and the implementation of our learning models will be covered in the Chapter 4. Experiments and evaluation of our methodologies as well as the visualisation system will be introduced in the Chapter 5. We will finally summarise conclusions and future works in the Chapter 6.

Chapter 2

Background

The traffic prediction problem can be defined as given a sequence $\{x_1, x_2, ..., x_t\}$ of traffic observations (time, speed, volume, etc.) at a specific location, estimation of $x_{t+\Delta}$ with forecasting horizon Δ need to be predicted. With underlying spatial correlations, the $x_{t+\Delta}$ can also be influenced by neighbouring locations in the transportation network. This chapter provides an overview of literatures and detailed introduction of related methods on traffic forecasting tasks. In this chapter, we will firstly introduce classical time series analysis methods and machine learning models which are implemented as baselines, and then fundamental deep neural network architectures that our proposed models are based on.

2.1 ARMA Family

The famous work, originally published in 1970 and republished by Box et al. (2015), proposes a family of classical time series analysis methods, especially the popular autoregressive moving average (ARMA) and its generalizations. Lippi et al. (2013) implemented a quantitative comparison among ARMA and its family with other methods on traffic flow forecasting and SARIMA (Seasonal Auto-Regressive Integrated Moving Average) with Kalman Filter achieves overall best performance but requires high computational resources. In this section, we will introduce ARMA and its family mathematically.

2.1.1 ARMA

Autoregressive moving average, represented by $ARMA(p, q)$, consists of two parts, which are an autoregressive $AR(p)$ model and a moving-average $MV(q)$ model. Typ-

ically, the ARMA is a linear model and can be used to understand and predict values in time series.

Autoregressive model If a Gaussian random noise is assumed, the conditional probability of x_t can be written as

$$p(x_t|x_1, ..., x_{t-1}, \epsilon_t) \sim \mathcal{N}(f(x_1, ..., x_{t-1}), \sigma^2) \tag{2.1}$$

$$x_t = f(x_1, ..., x_{t-1}) + \epsilon_t \tag{2.2}$$

where ϵ_t is an independent white noise and f is a regression function. In autoregressive model, given a integer p, the $AR(p)$ model is a linear combination of previous values with a random noise term, and is defined as:

$$x_t = c + \sum_{i=1}^{p} \psi_i x_{t-i} + \epsilon_t \tag{2.3}$$

where $c, \psi_1, ..., \psi_p$ are parameters and $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$. A future estimation x_t linearly depends on past observation $\{x_{t-p}, ..., x_{t-1}\}$. With lag operator, the model can be equivalently written as:

$$x_t = c + \sum_{i=1}^{p} \psi_i B^i x_t + \epsilon_t \tag{2.4}$$

As classic AR is a linear model, kernel functions can be applied to incorporate nonlinear dependence relations. For example, RBF kernel is popular in traffic problems (Wu et al. (2004); Sun and Chen (2008); Wang et al. (2009); Lippi et al. (2013)) and the regression function can be formatted as:

$$x_t = c + \sum_{j=1}^{h} \psi_j \phi_j(x_{t-1}, ..., x_{t-p}) \tag{2.5}$$

Moving-Average model The moving-average model captures dependence relations between variables and unobserved noise. The conditional probability is defined as $p(x_t|\epsilon_{t-q}, ..., \epsilon_t)$. As ϵ_i are unobservable, the interpretability of moving-average model is less convincing than the autoregressive model. Given a integer q, the $MV(q)$ is defined as:

$$x_t = \mu + \epsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} \tag{2.6}$$

where μ is an expectation of x_t, $\theta_1, \theta_2, ..., \theta_q$ are parameters and $\epsilon_t, \epsilon_{t-1}$ are white noise term. The $MV(q)$ model is typically a linear regression of current expectation with current and previous white noise. It can be equivalently written with lag operator:

$$x_t = \mu + \epsilon_t(1 + \sum_{i=1}^{q} \theta_i B^i) \tag{2.7}$$

ARMA The $ARMA(p,q)$ as illustrated in Figure 2.1, is a summation of $AR(p)$ and $MV(q)$, modelling conditional probability $p(x_t|x_{t-1},...,x_{t-p},\epsilon_t,...,\epsilon_{t-q})$, defined as:

$$\begin{aligned} x_t &= c + \epsilon_t + \sum_{i=1}^{p} \psi_i x_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} \\ &= c + \sum_{i=1}^{p} \psi_i B^i x_t + \epsilon_t (1 + \sum_{i=1}^{q} \theta_i B^i) \end{aligned} \tag{2.8}$$

Alternatively the ARMA can be written with lag operator:

$$\begin{aligned} \psi(B)x_t &= c + \theta(B)\epsilon_t \\ \text{where } \psi(B) &= 1 - \sum_{i=1}^{p} \psi_i B^i \\ \theta(B) &= 1 + \sum_{i=1}^{q} \theta_i B^i \end{aligned} \tag{2.9}$$

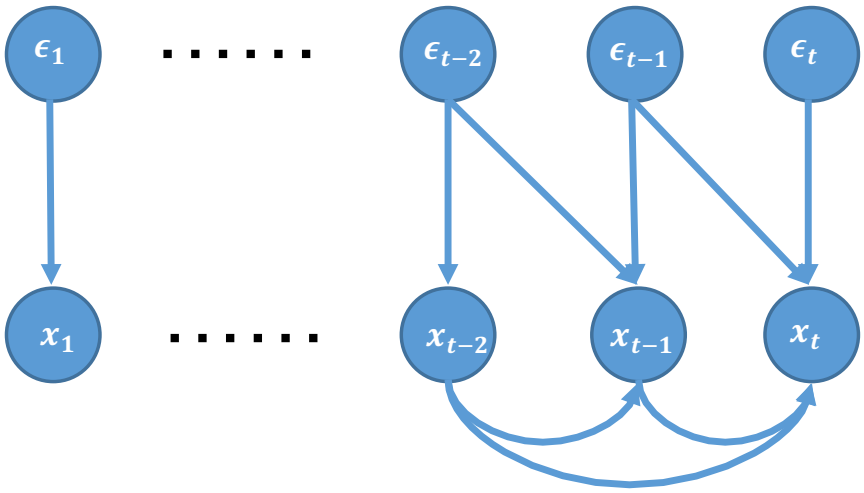


Figure 2.1: An illustration of dependence and independence in $ARMA(2,2)$. Given $\{x_{t-2}, x_{t-1}, \epsilon_{t-1}, \epsilon_{t-1}, \epsilon_t\}$, x_t is independent with other previous values.

ARIMA The autoregressive integrated moving-average model $ARIMA(p,d,q)$, as a generalization of ARMA, introduces another hyper-parameter d to handle non-stationarity. Based on ARIMA Moorthy and Ratcliffe (1988) investigate time series analysis on monthly traffic forecasting.

$$\psi(B)(1-B)^d x_t = c + \theta(B)\epsilon_t \tag{2.10}$$

2.1.2 SARMA

Seasonal ARMA A key nature of traffic data is periodic features. As traffic can be potentially significantly affected by holidays and working hours, the periodicity of traffic peak and off-peak flow patterns can help to improve accuracy of prediction. In ARMA, x_t is independent with $\{x_1, ..., x_{t-p-1}\}$, if given $\{x_{t-p}, ..., x_{t-1}\}$. SARMA, as shown in Figure 2.2, introduces a group of hyper-parameters $(P, Q)_S$, which defines dependence between x_t with $\{x_{t-S}, ..., x_{t-PS}, \epsilon_{t-S}, ..., \epsilon_{t-QS}\}$. The SARMA denoted by $SARMA(p, q) \times (P, Q)_S$, can be defined as:

$$\begin{aligned} \Psi(B^S)\psi(B)x_t &= c + \Theta(B^S)\theta(B)\epsilon_t \\ \text{where } \Psi(B^S) &= 1 - \sum_{i=1}^{P} \psi_i B^{iS} \\ \Theta(B^S) &= 1 + \sum_{i=1}^{Q} \Theta_i B^{iS} \end{aligned} \tag{2.11}$$

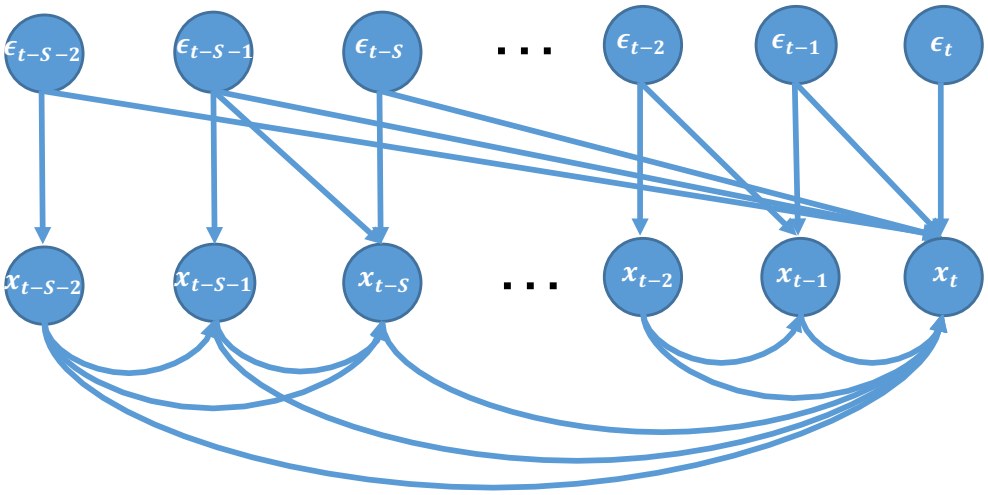


Figure 2.2: An illustration of dependence and independence in $SARMA(2, 2) \times (1, 1)_S$. x_t not only depends on previous neighboring values and also takes seasonality into account.

SARIMA Similarly, SARIMA can be denoted as $SARIMA(p, d, q) \times (P, D, Q)_S$, and defined as:

$$\Psi(B^S)\psi(B)(1 - B)^d(1 - B^S)^D x_t = c + \Theta(B^S)\theta(B)\epsilon_t \tag{2.12}$$

2.1.3 VAR

Vector autoregressive models (VAR) is a special case of vector ARMA and is capable of capturing linear interdependencies among multiple time series (see also Box et al.

(2015); Lütkepohl (2005)). Given $K \times T$ multivariate time series $\{X_1, X_2, ..., X_T\}$, VAR can be defined as:

$$X_t = A_1 X_{t-1} + A_2 X_{t-2} + ... + A_p X_{t-p} + \varepsilon_t \tag{2.13}$$

where A_i is $K \times K$ coefficient matrix and $\varepsilon_t \sim \mathcal{N}(0, \Sigma_K)$. In a transportation network, considering that each node has a traffic series, VAR can be used to find spatial relations among multiple nodes. Kamarianakis and Prastacos (2003, 2005) incorporate spatial information into traffic prediction tasks and demonstrate promising results by VAR and space-time ARIMA (STARIMA).

2.1.4 Autocorrelation

Autocorrelation Function (ACF) Repeating pattern of a given sequence helps to analyse and forecast, especially in traffic prediction which has strong seasonality. Autocorrelation, also called serial correlation, measures similarity between observations and observations with previous time steps, also called lags. In statistics, the autocorrelation is a function of the time lag and autocorrelation at lag k is defined as

$$\rho_k = \frac{\mathbb{E}[(x_t - \mu)(x_{t+k} - \mu)]}{\sqrt{\mathbb{E}[(x_t - \mu)^2]\,\mathbb{E}[(x_{t+k} - \mu)^2]}} = \frac{\mathbb{E}[(x_t - \mu)(x_{t+k} - \mu)]}{\sqrt{\sigma^2_{x_t}\sigma^2_{x_{t+k}}}} \tag{2.14}$$

In a stationary process, the assumption holds that the process is a specific state of a statistical equilibrium. The stationary assumption implies constant variance, so the variance $\sigma^2_{x_t}$ is the same at time $t + k$. Thus, if the sequence is stationary, the autocorrelation at lag k can be rewritten as

$$\rho_k = \frac{\mathbb{E}[(x_t - \mu)(x_{t+k} - \mu)]}{\mathbb{E}[(x_t - \mu)^2]} = \frac{\gamma_k}{\gamma_0} \tag{2.15}$$

where $\gamma_k = cov[x_t, x_{t+k}] = \mathbb{E}[(x_t - \mu)(x_{t+k} - \mu)]$ is the autocovariance at lag k. Thus, the autocorrelation function satisfies the same form of difference equation as autoregressive process defined in Equation 2.3.

$$\rho_k = \psi_1 \rho_{k-1} + \psi_2 \rho_{k-2} + ... + \psi_p \rho_{k-p} = \sum_{i=1}^{p} \psi_i \rho_{k-i} \tag{2.16}$$

Intuitively, observations are supposed to have strong autocorrelation at (up to) lag p (q) in $AR(p)$ ($MV(q)$), and the autocorrelation should be sharply weakened afterwards. However, in SARMA, autocorrelation should also have strong and continuous seasonal pattern.

Partial Autocorrelation Function (PACF) In the autoregressive process $AR(p)$, it is not clear for us to find an appropriate order p to fit observations. Partial autocorrelation function is an efficient tool to exploit autocorrelation function in terms of p. We define ϕ_{pj} as the jth coefficient in an $AR(p)$ process.

$$\rho_j = \phi_{p1}\rho_{j-1} + \phi_{p2}\rho_{j-2} + ... + \phi_{pp}\rho_{j-p} = \sum_{i=1}^{p} \phi_{pi}\rho_{j-i} \tag{2.17}$$

where the value of ϕ_{pp} is called the partial autocorrelation function. In intuition, in $AR(p)$, the value of partial autocorrelation should be nonzero for lag less than or equal to p and zero for lag greater than p.

2.2 Support Vector Regression

SVM Given data points $x_i \in \mathbb{R}^n$ with binary labels $y_i \in \{1, -1\}, i = 1, ..., n$, the support vector machine (SVM) tries to find a hyperplane to classify positive and negative data points. The SVM maps data points from high dimensional space into feature space by $\phi(x)$ and finds the optimal hyperplane by maximizing the margin between two classes. Classic SVM constructs a linear hyperplane but can be extended to nonlinear space by kernel functions.

SVR While SVM is designed to solve binary classification problem, SVR tries to estimate a function that can predict future values. Considering a set of labeled data $\{(x_1, y_1), (x_2, y_2), ..., (x_t, y_t)\}$, where $x_i \in \mathbb{R}^n$ and $y_i \in \mathbb{R}$, SVR defines a generic regression function to fit them:

$$f(x) = w * \phi(x) + b \tag{2.18}$$

where $\phi(x)$ is a nonlinear transformation from $\mathbb{R}^n$ to feature space. Kernel functions like radial basis function (RBF) can be incorporated into SVR by writing weights w as:

$$w = \sum_{i=1}^{t} (\alpha_i - \alpha_i^*)\phi(x_i) \tag{2.19}$$

and the generic function can be updated as:

$$\begin{aligned} f(x) &= \sum_{i=1}^{t} (\alpha_i - \alpha_i^*)\phi(x_i)\phi(x) + b \\ &= \sum_{i=1}^{t} (\alpha_i - \alpha_i^*)k(x_i, x) + b \end{aligned} \tag{2.20}$$

Therefore, the overall optimization problem in SVR can be summarized as:

$$\begin{aligned} &\min C \sum_{i=1}^{t} L(f(x_i) - y_i) + \frac{1}{2}||w||^2 \\ &\text{s.t.} \sum_{i=1}^{t} \alpha_i - \alpha_i^* = 0, \alpha_i, \alpha_i^* \in [0, C] \end{aligned} \tag{2.21}$$

where $L(f(x_i)-y_i)$ defines loss function. Further improvement with ϵ-insensitive loss function and slack variables ζ_i can be applied. In traffic applications, the prediction function can be defined as:

$$x_{t+l} = f(t, l, x_t, x_{t-1}, ..., x_{t-n}) \tag{2.22}$$

Wu et al. (2004); Lippi et al. (2013) implement SVR with RBF/linear kernel and demonstrate its feasibility and promising results. However, only data points in a short path at rush hours are used for prediction and evaluation, and in order to simplify the problem, other potential factors like weather, holidays and transportation network topology are not taken into consideration.

2.3 Gaussian Process

Gaussian Process is a statistical model for continuous domain that casts prior distribution on functions and treats a function as an infinite long vector of values $\{f_1, f_2, ...\}$ (see also Bishop (2006); Williams and Rasmussen (2006)). However, in practice, Gaussian Process needs to work in finite space as training set and testing set are both finite and discrete. Gaussian Process has been widely researched in regression and classification problem and its derivations include ARMA and Kalman Filter. Xie et al. (2010) and Zhao and Sun (2016) have explored Gaussian Process in traffic forecasting and shown promising improvement. However, Gaussian Process is not suitable for relatively longer term prediction like more than 1 hour, as the variance can be accumulated to extremely large. A Gaussian Process model can be notated by $GP(m, k)$, where m is a mean function and k is a covariance function. Based on the Bayes theorem, the Gaussian Process can be written as:

$$\begin{aligned} p(f|X, y) &= \frac{p(y|f, X)p(f)}{p(y|X)} \\ \text{where} \quad p(f) &= GP(m, k) && \text{as prior} \\ p(y|f, X) &= \mathcal{N}(f(X), \sigma_n^2 I) && \text{as likelihood} \\ p(y|X) &= \int p(y|f, X)p(f|X)df && \text{as evidence} \\ p(f|X, y) &= GP(m_p, k_p) && \text{as posterior} \end{aligned} \tag{2.23}$$

In prediction stage, posterior distribution $p(f_t|X, y, X_t)$ at test point X_t can be obtained:

$$\begin{aligned} p(f_t|X, y, X_t) &= \mathcal{N}(\mathbb{E}(f_t|X, y, X_t), \mathbb{V}(f_t|X, y, X_t)) \\ \mathbb{E}(f_t|X, y, X_t) &= m_p(X_t) = m(X_t) + k(X_t, X)(K + \sigma_n^2 I)^{-1}(y - m(X)) \\ \mathbb{V}(f_t|X, y, X_t) &= k(X_t, X_t) - k(X_t, X)(K + \sigma_n^2 I)^{-1}k(X, X_t) \end{aligned} \tag{2.24}$$

With a mean function and covariance function, the $GP(m, k)$ is fully specified. Covariance function leads to different structural patterns such as smoothness and periodicity. Therefore, for traffic prediction problem, proper covariance function would be essential. There are several popular choices of covariance functions:

$$\begin{aligned} &\text{Gaussian covariance function} && k(x_i, x_j) = \sigma_f^2 \exp(-(x_i - x_j)^T(x_i - x_j)/l^2) \\ &\text{Matérn covariance function} && k_{\frac{3}{2}}(x_i, x_j) = \sigma_f^2(1 + \frac{\sqrt{3}||x_i - x_j||}{l}) \exp(-\frac{\sqrt{3}||x_i - x_j||}{l}) \\ &\text{Periodic covariance function} && k(x_i, x_j) = \sigma_f^2 \exp(-\frac{2\sin^2(\frac{\omega(x_i - x_j)}{2\pi})}{l^2}) \end{aligned} \tag{2.25}$$

2.4 Deep Learning Models

Deep learning, or Deep Neural Networks (DNNs), was previously known as artificial neural networks (ANNs) and was initially inspired by information communication in biological nervous system (Goodfellow et al. (2016)). Deep learning models are assembled by multiple layers, which include visible layers and hidden layers. Visible layers contain input and output, and hidden layers are designed to extract features by feedforward operations. Rumelhart et al. (1988) introduces back propagation based on gradient to train neural networks. With stronger representation learning capability, increasing dataset sizes and real world challenges, deep learning has rapidly growth in applications. Deep learning models, e.g. multilayer perceptron (MLP), convolutional neural network (CNN), recurrent neural network (RNN), generative adversarial networks (GAN, Goodfellow et al. (2014)), have been widely applied in computer vision (Krizhevsky and Hinton (2009); Radford et al. (2015)), natural language processing(Yu et al. (2017)), audio generation (van den Oord et al. (2016)) and etc. Recent research have also explored LSTM neural network in travel time forecasting (Duan et al. (2016); Ma et al. (2017)) and residual networks in crowd flow prediction (Zhang et al. (2017)).

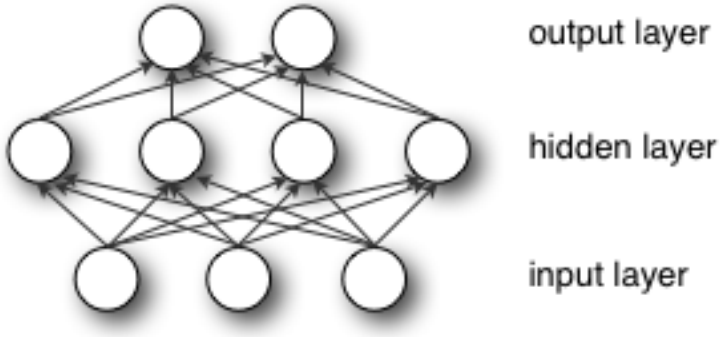
2.4.1 Multilayer Perceptron

Multilayer perceptron, also known as MLP, fully connected network and dense network, is a matrix multiplication operations mapping inputs to outputs and now has been widely applied in decoder networks. A typical MLP is a three-layer neural network, as demonstrated in Figure 2.3. Mathematically, the MLP can be written as:

$$y = f^{(2)}(W^{(2)}(f^{(1)}(W^{(1)}x + b^{(1)})) + b^{(2)}) \tag{2.26}$$

with input x, output y, parameters W, b and activation function f.

Figure 2.3: An example [a] of three-layer MLP.



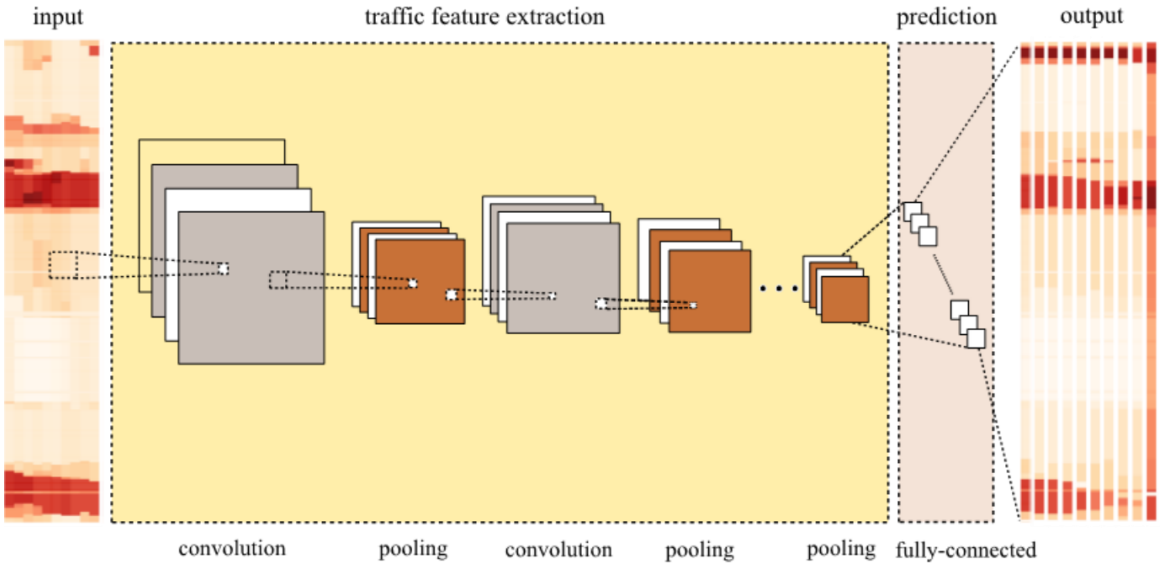
[a] Source from `http://deeplearning.net/tutorial/mlp.html`, accessed: 2017-08-24

2.4.2 Convolutional Neural Networks

Convolutional neural networks, also knows as CNNs, has been widely applied in many areas and has shown outstanding capability in processing grid-like data (LeCun et al. (1989)). Examples include 1-D time series samples with regular time intervals, and images which can be thought as 2-D grid pixels (Goodfellow et al. (2016)). In the mid 20th century, neurophysiologists tried to understand how the brain responds to images and discovered that some neurons in the brains of cats and monkeys are highly sensitive to edges in small regions in visual fields (Hubel and Wiesel (1959, 1962, 1968)). Thus, as a artificial intelligence model, CNNs is fundamentally supported by neuron science. Recent paper Ma et al. (2017) treats traffic data as 2-D images, one dimension of time, one dimension of location, and demonstrates better accuracy with CNNs on the speed prediction (Figure 2.4).

Convolution Operation The convolution is designed to scan a sequence x with a weighting function w. A continuous convolution operation can be written and de-

Figure 2.4: An example of CNNs. The CNNs based framework implemented by Ma et al. (2017) for transportation network speed prediction.



fined as:

$$s(t) = (x \star w)(t) = \int x(a)w(t-a)da \tag{2.27}$$

In machine learning applications, the input data is discrete and usually multidimentional. Using 2-D images as a example, we can define a 2-D discrete convolution:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n) \tag{2.28}$$

where the $I(m,n)$ refers to the input image, and $K(m,n)$ is called kernel or feature map. Through visualisation, Yosinski et al. (2015) helps us understand how CNN works. Convolution operations (Figure 2.5) use feature maps to scan images, measure their similarity, and outputs a heatmap $S(i,j)$ which highlights regions of interest. For example, if a feature map, which is extracted from images without supervision, is a outline of human face, the heatmap $S(i,j)$ can indicate location of human faces unless there is no faces. The feature maps for raw images can be straightforward, but after several hidden layers, the feature maps can be highly abstract.

Pooling Pooling can be seen as a summarization of responses over a whole neighbourhood. Pooling drops useless information by reducing the output size, which helps to make the network invariant to small changes of the input. Max pooling is one of the most popular pooling operations, which reports the maximum value

within a rectangular neighbourhood, as shown in Figure 2.6.

Figure 2.5: Convolution operation. The input (blue) is element-wisely multiplied by feature maps (dark blue) and the output (green) is computed by applying the feature maps to the corresponding region of the input. Source from: Dumoulin and Visin (2016).

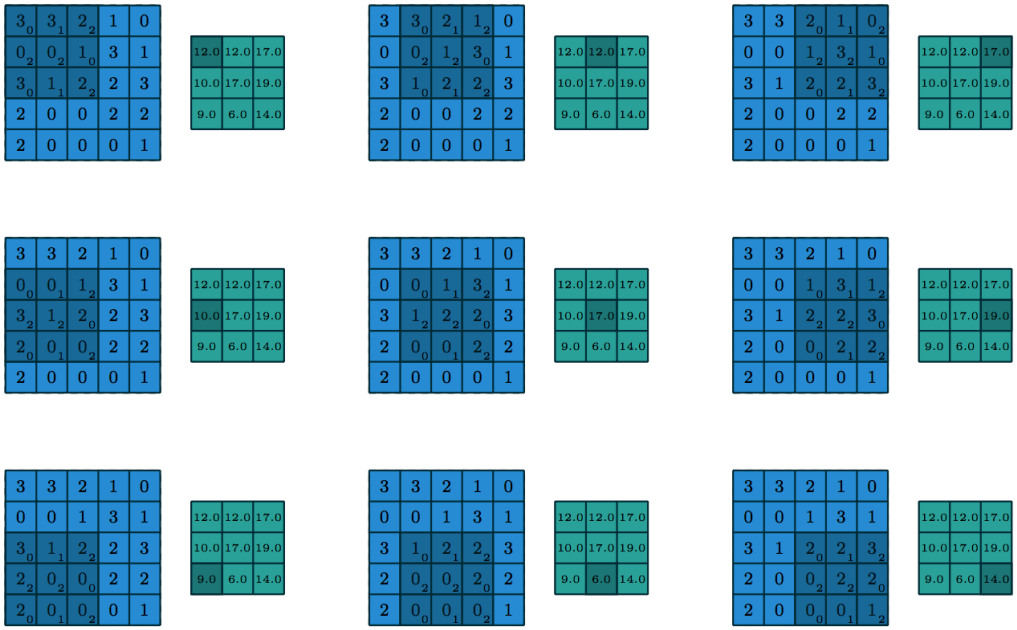
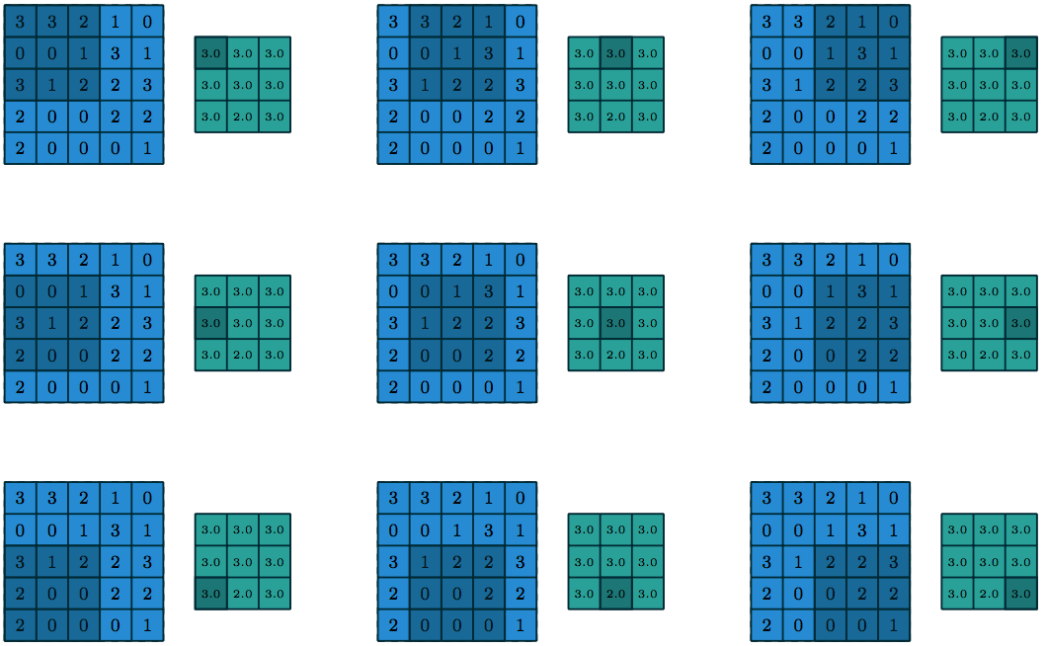


Figure 2.6: Max pooling. The max pooling operation returns the maximum value (dark green) in the rectangular region (dark blue). Source from: Dumoulin and Visin (2016).

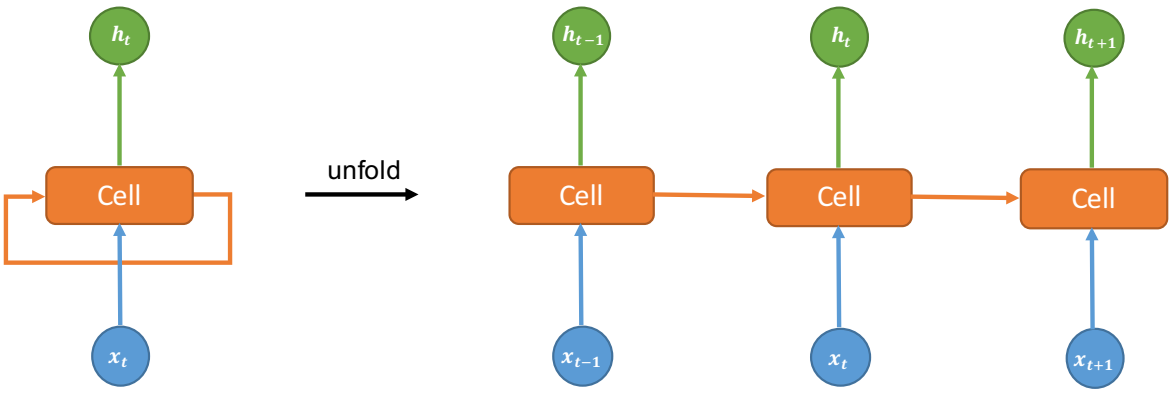


2.4.3 Recurrent Neural Network

As illustrated in Figure 2.7, the basic idea of recurrent neural network (RNN) is that an output is produced at each time step and the hidden units are recurrently connected, which means the output h_t depends on both the input x_t and previous information h_{t-1} (see also Goodfellow et al. (2016)). The cell decides how h_t is computed given the sequential data. A simple choice for the cell can be a linear operation which accumulates sequential information and outputs the hidden states.

The chain structure decides that RNN is intimately suitable for sequence data analysis. Therefore, RNN has shown powerful learning capability in machine translation (Sutskever et al. (2014)), text-to-image synthesis (Reed et al. (2016); Dong et al. (2017b)), speech generation (van den Oord et al. (2016)), traffic prediction (Duan et al. (2016); Li et al. (2017)) and many other applications.

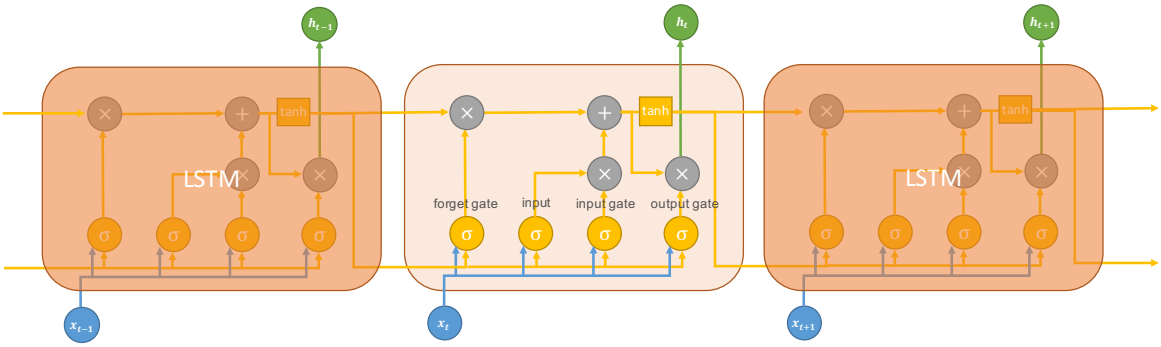
Figure 2.7: Recurrent neural network[a]. A chunk of neural network is integrated in the cell with x_t as input, h_t as output. The cell allows information to be passed from current step to its successor. The left figure shows the basic structure of RNN and an equivalent unfolded version in the right.



[a]The figure is reproduced from `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`, accessed: 2017-08-24

LSTM The long short-term memory (LSTM), proposed by Hochreiter and Schmidhuber (1997), based on gated recurrent units, has become a very popular and effective choice for the RNN cell in practical applications. Intuitively, there are cases where we need more context to decide h_t. Recent information closely before time step t may be not enough and information further back is necessary. The LSTM introduces a highway to achieve accumulation of information for a long duration and also a forget gate layer to decide whether to drop an old state or not. The key idea is that LSTM can keep or add information to the cell state where is regulated by gates. An illustration of LSTM is shown in Figure 2.8.

Figure 2.8: LSTM[a]. The diagram illustrates the recurrent connection between LSTM cells. After sigmoid function σ, the values will be put into the gates. Previous information will be accumulated into the next step.



[a]The figure is reproduced from `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`, accessed: 2017-08-24

2.5 Global Data Observatory

Global Data Observatory (GDO) [1], largest of its kind in the Europe, built by Data Science Institute at Imperial College London, assembled by 64 monitors and 32 computers, features an enveloping circular screen wall with ultra high resolution (Figure 2.9). GDO provides 313-degree surround vision and friendly interactive system, which leverage powerful visualisation of big data to help analysts extract useful knowledge and make best quantitative decisions.

GDO Software Development Kit (SDK) provides two development modes, Cluster Mode and Section Mode. In the Cluster Mode, all 64 monitors work individually and information for each monitor will be distributed by a central server. The Section Mode splits the screen wall into five sections, as shown in Figure 2.10. Visualisation on each section is implemented based on a web browser and information can be transferred among the sections by connection tools such as Socket.IO [2], PeerJS [3] and SignalR [4]. In this project, we finished visualisation of traffic data and evaluation of volume prediction on the Section Mode of GDO, which will be further discussed in Section 5.5.

[1]GDO, see also: `https://www.imperial.ac.uk/data-science/data-observatory-/`, accessed: 2017-09-04

[2]Socket.IO is a real-time tool based on Node.js: `https://socket.io/`, accessed 2017-08-27.

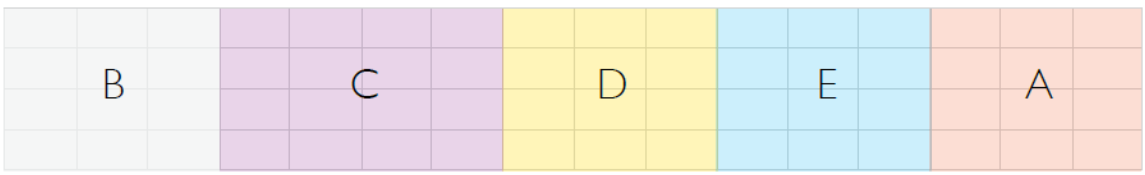
[3]PeerJS supports peer-to-peer data transfer: `http://peerjs.com/`, accessed 2017-08-27.

[4]SignalR is a real-time library for ASP.NET: `http://signalr.net/`, accessed 2017-08-27.

Figure 2.9: Global Data Observatory in Data Science Institute at Imperial College London [a].

[a]Source from `https://pbs.twimg.com/media/C4oYInAXUAAiMEN.jpg`, accessed: 2017-08-25

Figure 2.10: Section Mode of GDO [a]. The screen wall is split into five sections and each section is based on web browser.



[a]Source from `https://wiki.imperial.ac.uk/display/DG/1+-+Setup`, accessed: 2017-08-27, privilege may be needed to view this link.

Chapter 3

Data Analysis and Refinery

This chapter provides introduction and analysis on the traffic datasets we use. We will introduce discovery on temporal patterns of traffic volume including peak hours, seasonality and special patterns on special days. Additionally, spatial correlations in highway networks will be demonstrated. Moreover, data refinery as prerequisite for proposed models is also essential and will be covered at the end of this chapter.

3.1 Datasets

In this project, we investigate and implement experiments on two traffic datasets, the Highways England dataset and the KDDCUP 2017 dataset. The analysis on temporal and spatial relations in this chapter (Section 3.2 and 3.3) is mainly based on the Highways England dataset, while the KDDCUP 2017 dataset will be used as benchmark in Chapter 5 to demonstrate outstanding performance of our proposed methodologies.

3.1.1 Highways England Dataset

Since 2015, the UK government and Highways England have published traffic data on all motorways and 'A' roads regulated by Highways England [1]. The dataset provides average speed, average journey time, traffic flow volume of different types of vehicles for 15-minute periods from January 2006 to June 2017, and it is still updating. There are more than 14,000 monitoring nodes collecting real time traffic

[1]Highways England network journey time and traffic flow data, published under Open Government Licence on `https://data.gov.uk/dataset/highways-england-network-journey-time-and-traffic-flow-data`, accessed: 2017-08-17

data in the entire England region as shown in Figure 3.1. The distribution of nodes is closely compact in suburban area around huge cities like London, Birmingham and Manchester. According to the official introduction of the dataset, the data are collected and estimated from combination of different sources including automatic number plate recognition cameras, in-vehicle GPS and inductive loops in the road surface. Quality index based on number of valid record is provided and can be used to drop unreliable data points.

In this project, as we focus on short-term traffic flow prediction, we select data of traffic volume from 1st January 2016 to 30th June 2017, including 52 nodes around Uxbridge and Heathrow Airport, as shown in Figure 3.1. More information about the nodes can be found in Table 7.1 in Appendices. For simplicity, the nodes are named by simplified numerical ID, e.g. 1017, 2209 and 345. Among the 52 nodes, 46 of them are located on the motorways M4 and M25, and the other 6 nodes are selected from 'A' roads which are closed to entrances or exits of the motorways. There is one and only one data point each 15-minute slice, so 96 data points per day and 672 per week. Each data point provides several fields of information as shown in Table 3.1. Both total carriageway flow and type of days will be applied as inputs (features) to forecast traffic flow in the short-term future. You can find an example of data points in the Table 3.2. We will demonstrate interesting discoveries and analysis based on the Highways England dataset in the following sections of this Chapter.

3.1.2 KDDCUP17 Dataset

KDDCUP 2017 [2] released highway tollgate traffic flow prediction challenges with two tasks, which are journey time prediction and traffic volume prediction. As our project focuses on traffic volume prediction, we select the second task and corresponding data as benchmark. The training set provides traffic data (volume, weather, holiday, etc.) from 19th Sep to 17th Oct 2016 with 20-minute intervals. More information can be found in Table 3.3.

Given traffic data during 6-8am and 15-17pm (green slots in Figure 3.2) in the following week, from 18th Oct to 24th Oct 2016, the task is to predict traffic volume during 8-10am and 17-19pm each day in the week (red slots in Figure 3.2) at each tollgate. There are only three tollages in the dataset and they are labelled as 1, 2, 3. In Chapter 5, we will demonstrate our proposed model outperforms state-of-the-art methods in KDDCUP 2017 competition.

[2]More information about KDDCUP 2017 can be found on: `http://www.kdd.org/kdd2017/News/view/announcing-kdd-cup-2017-highway-tollgates-traffic-flow-prediction`,accessed: 2017-08-23.

Table 3.1: Fields of information provided in the Highways England dataset.

Fields	**Descriptions**	
Local Time	Local time (British Standard Time) with 15-minute time intervals. Normally, one and only data point for each 15-minute time slice.	
Day Type ID	Days are classified into several categories, and valid ID includes:	
	0,1,2,3,4	Monday to Friday (Workdays).
	5,6	Saturday and Sunday (Weekends).
	7,9,11	School holidays but excluding days falling within type 12,13,14.
	12	Bank Holiday.
	13	Christmas period holiday between Christmas Day and New Year's Day but excluding days falling within type 14.
	14	Christmas Day and New Year's Day.
Total Carriageway Flow	The number of vehicles counted on any lane within the 15-minute time window.	
Quality Index	The indication of quality of data collected. The number of valid 1-minute record within the 15-minute time window. 0 indicates no valid record in the time window and 15 indicates the best quality.	

Table 3.2: An example of data points at Node 1017 in the Highways England dataset. In this example, the data points are collected on Friday morning with good quality. The local time has 15-minute time intervals and total carriageway flow is the number of vehicles that go through Node 1017 in corresponding time slice.

Local Time	Day Type ID	Total Carriageway Flow	Quality Index
01/07/2016 08:14:00	4	1107	15
01/07/2016 08:29:00	4	1104	15
01/07/2016 08:44:00	4	1015	15
01/07/2016 08:59:00	4	966	15
01/07/2016 09:14:00	4	913	15
01/07/2016 09:29:00	4	932	15
01/07/2016 09:44:00	4	1030	15
01/07/2016 09:59:00	4	901	15

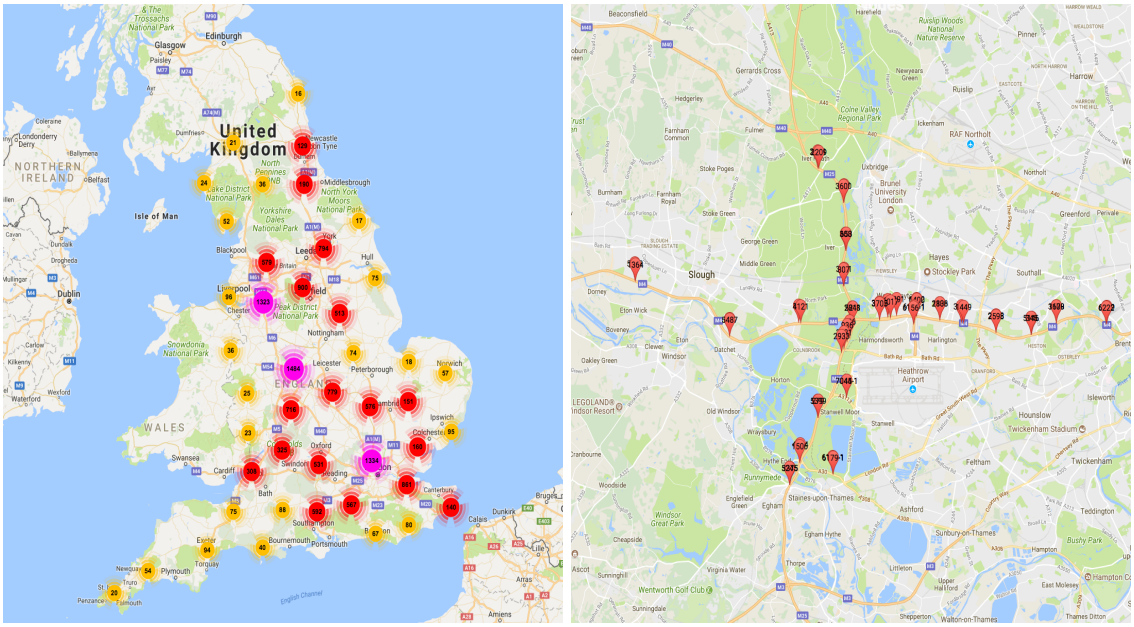


Figure 3.1: Highways England dataset. (a) The left figure [a] shows distribution of monitoring nodes in England region. The nodes are closely compacted around cities like London, Birmingham and Manchester. (b) The right figure shows the selected 52 nodes around Uxbridge and Heathrow Airport. The nodes are mainly distributed on motorways M4 and M25, and 'A' roads closed to their entrances or exits. More information about the selected nodes can be found in Table 7.1.

[a] Source from `http://webtris.highwaysengland.co.uk/`, accessed: 2017-08-23

Figure 3.2: Time windows for traffic volume prediction in KDDCUP 2017 dataset [a].

[a]Source from https://tianchi.aliyun.com/competition/information.htm?raceId=231597, accessed: 2017-08-23

Table 3.3: Fields of information provided in the KDDCUP 17 dataset.

Fields	Descriptions
Time	The time when a vehicle goes through the tollgate.
Tollgate ID	ID of the tollgate, which can be 1, 2 or 3. There are only three tollgates.
Holiday	0: normal, or 1: public holidays.
Air Pressure	Air pressure (hPa: Hundred Pa).
Wind Speed	Wind speed (m/s).
Temperature	Temperature (°C).
Precipitation	Precipitation (mm).

3.2 Temporal Pattern

Compared with minor roads in urban area, the highways have fewer entrances and exits per unit distance, fewer traffic lights that can block vehicles, higher average speed and higher traffic volume. Visualisation of traffic volume indicates strong daily pattern like peak hours, and strong continuous seasonality by autocorrelation. Furthermore, special days e.g. weekends and Christmas holidays have significantly different traffic pattern with workdays.

3.2.1 Peak Hours

Statistics based on traffic data of 52 nodes from 1st January 2016 to 30th June 2017 show there are on average 744.12 vehicles per 15 minutes on motorways (M4 and M25), with maximum 2512 during peak hours, and average 255.51 vehicles per 15 minutes on 'A' roads, with maximum 773 (Table 3.4). The traffic volume during weekends is lower than workdays and usually has only one peak in most nodes but there are some exceptions.

During workdays, the peak hours start 07:00-08:00 in the morning and ends 19:00-20:00 in the evening. As illustrated in the Figure 3.3, the morning peak is usually higher and lasts shorter than the evening peak during workdays, which may be explained by intuition that most people drive for work crowded in a short period of time early in the morning, but the time for going home is diverse and lasts longer.

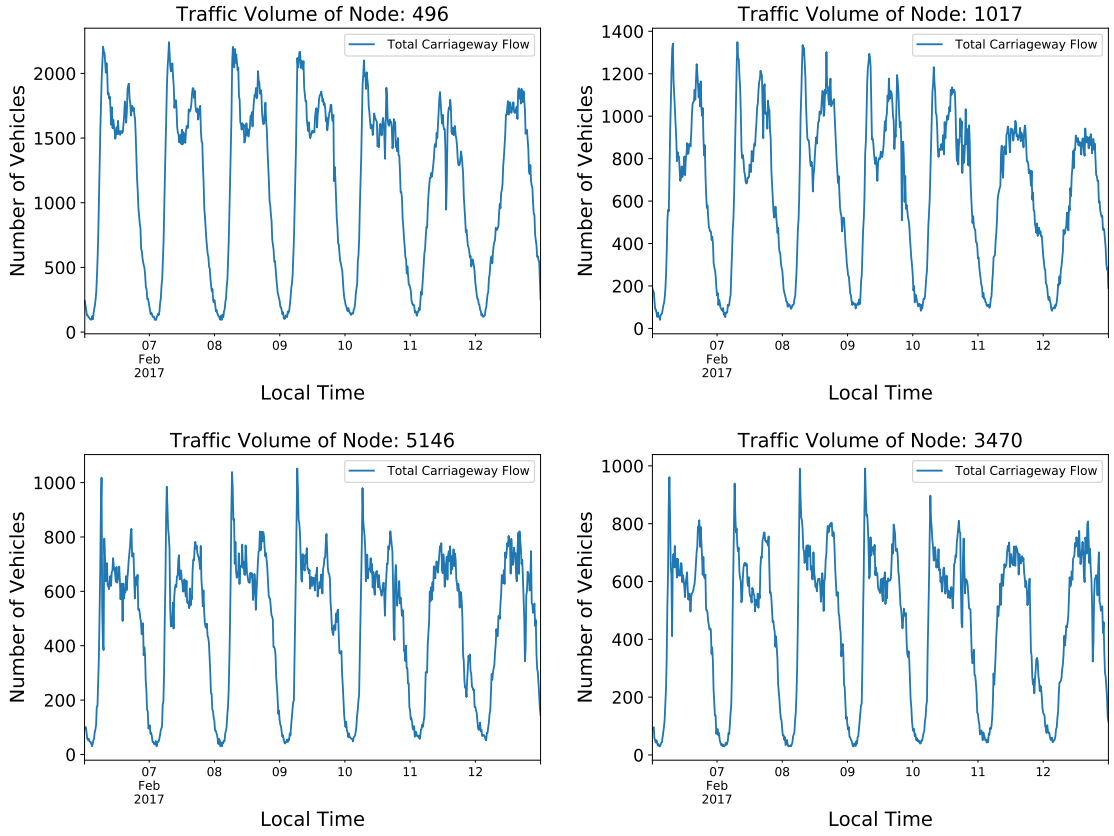
In addition, we can find symmetry in pairs of nodes. For example, Figure 3.4 demonstrates that Node 7044-1 has higher morning peak but, on the contrary, Node 7045-1 has higher evening peak. On the map, we can easily find that Node 7044-1 heads towards Heathrow Airport while Node 7045-1 heads for the opposite direction to M25. Such symmetry may help us unearth potential moving pattern of people in a small local area.

Moreover, dramatic decreasing of traffic volume in a short time window as demonstrated in Figure 3.5 attract our attention, which may be caused by abnormal conditions including unexpected accidents and planned events, but such record is unavailable in the dataset. Though the traffic volume has overall strong recurrent pattern, interference of variance and those abnormal outliers can be challenging for our models.

Table 3.4: Statistics on number of vehicles per 15 minutes. The traffic volume on motorways (M4 and M25) is much higher than that on 'A' roads.

	Max	Min	Mean	Median
Motorways	2512	0	774.12	732.00
'A' Roads	773	0	225.51	214.00

Figure 3.3: Traffic volume from 6th Feb 2017 (Monday) to 12th Feb 2017 (Sunday) at Node 496, 1017 and 5146. During workdays, the morning peak usually has higher traffic volume and lasts shorter than the evening counterpart, which may be explained by intuition that people drive towards work crowded in a short time window in early morning but the time for going home is very different. However, such pattern does not hold on weekends.



3.2.2 Seasonality and ARMA Order

Intuitively, traffic volume should have continuous daily or weekly pattern. For example, during workdays, given a specific location, traffic volume should be similar with that one day before. Weekends are different from workdays but similarity between this and previous weekend is not easy to be ignored. The seasonality is a fundamental assumption in short-term traffic prediction problems and the models we use, not

Figure 3.4: Symmetry between Node 7044-1 and 7045-1. On workdays, the traffic volume during morning peak of Node 7044-1 is significantly higher than evening peak. However, the Node 7045-1 has an opposite traffic pattern. On the map, both Node 7044-1 and 7045-1 are located on road A3113, while 7044-1 heads towards Heathrow Airport but 7045-1 towards the opposite direction to motorway M25. Such symmetry may unearth moving behaviour of people in the small area.

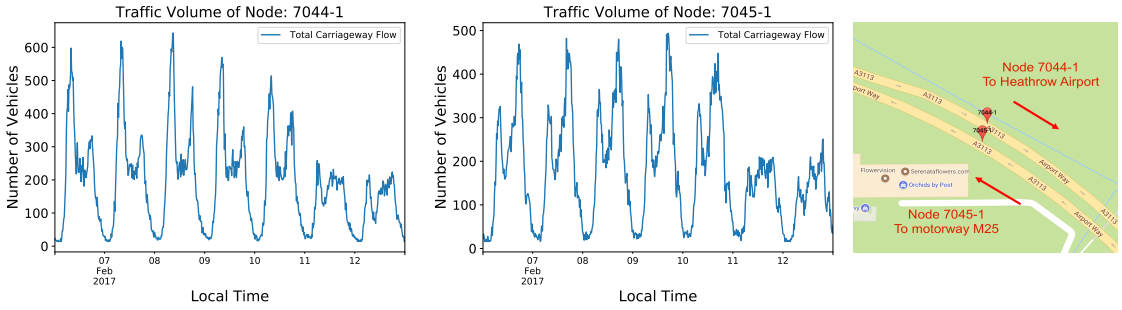
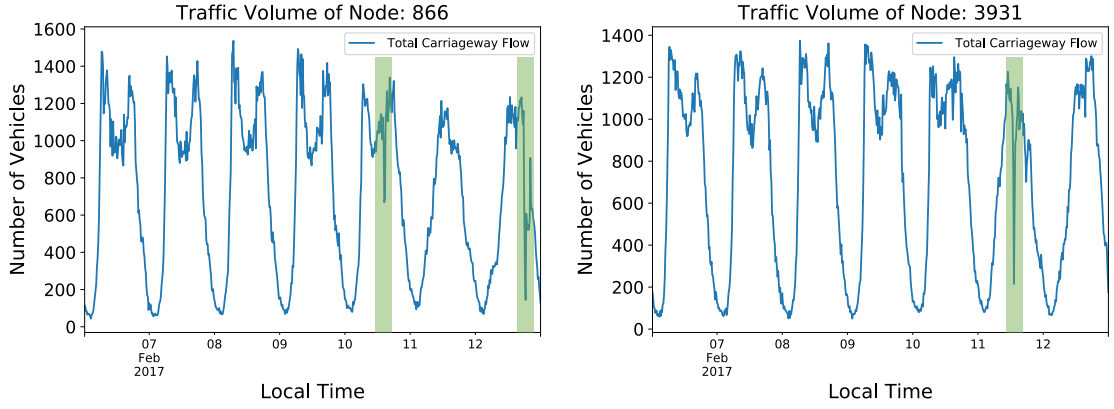


Figure 3.5: Abnormal conditions at Node 866 and 3931. The green background highlight outliers. The dramatic decreasing of traffic volume may be caused by accidents or maintenance events but such record is not available in our dataset.



only SARIMA and also neural networks. Autocorrelation function (ACF), which are introduced in Section 2.1.4, can help us find out the order of seasonality. Besides, we also explore partial autocorrelation function (PACF), which is created to decide order of ARMA, at each node.

Figure 3.6 illustrates that daily seasonality is strong and consistent at Node 2291, 5146 and 6157-1, and the results of PACF indicates that an appropriate order for ARMA is 1. In fact, such conclusions hold for all 52 nodes. As mentioned in Section 3.1, the number of lags for each day is 96. Therefore, the value at a specific time is positively correlated with neighouring values and those days before, while negatively correlated with those half days in advance. Moreover, Figure 3.7 demonstrates special cases at Node 7044-1 and 7045-1, and such pattern is caused by the special symmetry between Node 7044-1 and 7045-1 and the asymmetry between morning and evening peak as shown in Figure 3.4.

Figure 3.6: Autocorrelation function (ACF) and partial autocorrelation function (PACF) at Node 2291, 5146 and 6157-1. The three figures on the top computed by ACF show strong and consistent daily seasonality. Results of PACF at the bottom find that an appropriate order for ARMA is 1. Such conclusions hold for all 52 nodes.

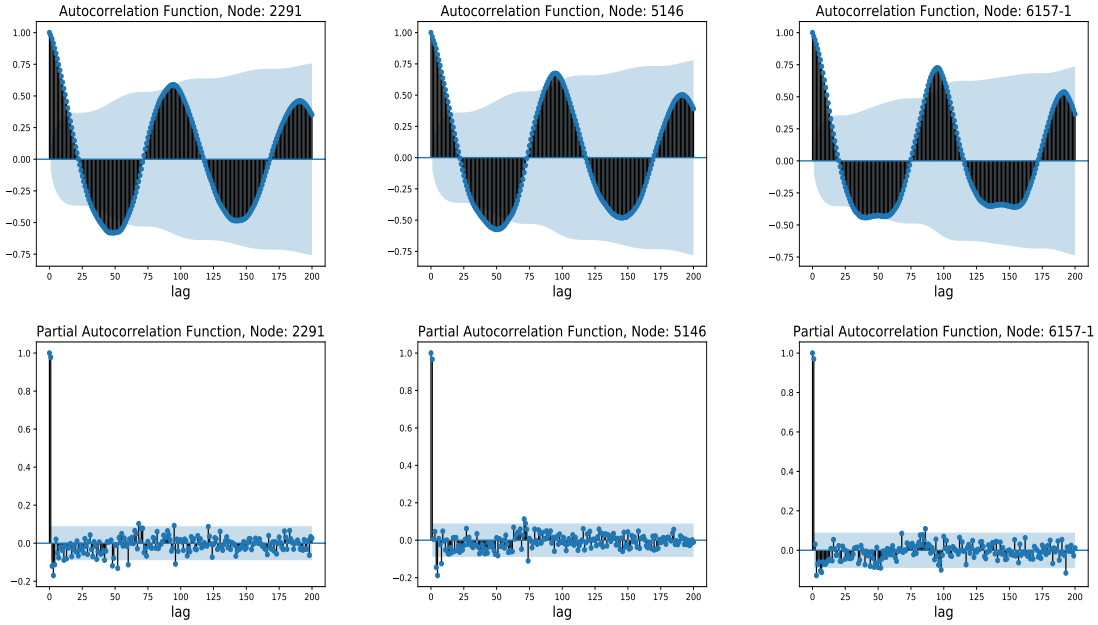
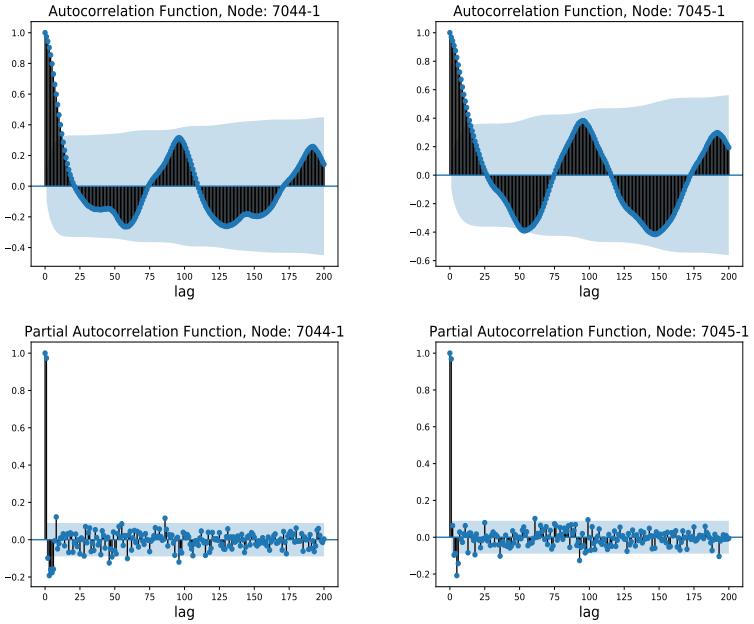


Figure 3.7: Special cases of autocorrelation function (ACF) and partial autocorrelation function (PACF) at Node 7044-1 and 7045-1. Such pattern is caused by the special symmetry between Node 7044-1 and 7045-1 and the asymmetry between morning and evening peak as shown in Figure 3.4.



3.2.3 Special Days

In the previous sections, we have analysed typical temporal patterns in traffic, and such patterns hold strong and consistent on most days. However, there are some special days that the series may behave differently. The Highways England dataset provides indication of type of days as showed in Table 3.1. There are six types of special days including weekends, school holidays, bank holidays, Christmas Day, Christmas period and New Year's Day. We firstly introduce some measurements to help us find difference between workdays and special days.

1. Peak hours: The moments have highest 25% traffic volume.
2. Start of peak hours: The earliest moment each day of peak hours.
3. End of peak hours: The latest moment each day of peak hours.
4. Percentage difference of all hours:

$$\frac{mean(x_t \mid t \in T^{all}_{workdays}) - mean(x_t \mid t \in T^{all}_{specialdays})}{mean(x_t \mid t \in T^{all}_{workdays})} \tag{3.1}$$

   where $T^{all}_{workdays}$ is a set of all hours on workdays, $T^{all}_{specialdays}$ is a set of all hours on special days.

5. Percentage difference of peak hours:

$$\frac{mean(x_t \mid t \in T^{peak}_{workdays}) - mean(x_t \mid t \in T^{peak}_{specialdays})}{mean(x_t \mid t \in T^{peak}_{workdays})} \tag{3.2}$$

   where $T^{peak}_{workdays}$ is a set of peak hours on workdays, $T^{peak}_{specialdays}$ is a set of peak hours on special days.

Weekends: On the 52 nodes, compared with workdays, traffic on weekends has 10.9% lower volume on average. And during peak hours, the traffic volume on weekends is 9.3% lower than workdays. Besides, the average start of peak hours is postponed from 07:38am on workdays to 11:03am on weekends, and the average end of peak hours is also delayed from 19:28 on workdays to 19:56 on weekends. Usually, there are two peaks during workdays (morning peak and evening peak), but such pattern does not hold on weekends. Figure 3.8 and Table 3.5 compare traffic volume between workdays and weekends.

Bank Holidays: Bank holidays have similar traffic pattern as weekends, which is easily to understand as Bank holidays are Mondays following weekends. On average, the bank holidays have 14.8% less traffic (all hours) and 11.4% less in peak hours. Additionally, the start of peak hours is further postponed to 11:24am while the end of peak hours is 19:47 about 20 minutes later than workdays. Figure 3.9 and Table 3.6 illustrate differences between bank holidays and workdays.

School Holidays: The traffic volume during school holidays does not have obvious difference with workdays. Figure 3.10 and Table 3.7 demonstrate that the average percentage difference between workdays and school holidays is about $\pm$ 1%, which can be ignored. Therefore, we can make an conclusion that school holidays do not have significant impact on highway traffic.

Christmas Period: Christmas period refers the days after Christmas Day and before New Year's Day (26th Dec to 30th Dec). In our dataset, we have only one Christmas period which is in 2016. Figure 3.11 and Table 3.8 show that Christmas period has overall 20.0% less traffic and 14.6% less in peak hours than workdays. The start of peak hours is delayed to 11:26am on average, but 24 minutes earlier ending of peak hours in the evening, which is 19:05.

Christmas Day: In our dataset, there is only one Christmas Day which is on 25th Dec 2016. Considering all 52 nodes, Christmas Day has significantly lower traffic volume than workdays.Figure 3.12 and Table 3.9 indicate that 45.1% fewer vehicles would drive on the highways than workdays. The highest volume on the Christmas Day is around the borderline of peak hours on workdays. In addition, the start of peak hours is 10:59am which is similar with that on weekends. However, the ending of peak hours is rather late in the evening about 20:48.

New Year's Day: The dataset includes New Year's Days of 2016 and 2017 which have overall 35.1% less traffic than workdays. Interestingly, there are unique midnight peaks on New Year's Days as highlighted in the left column of Figure 3.13. The midnight peak has relatively higher volume on the roads outwards central London which indicates crowded of people going out of the city after midnight celebration events. Moreover, the start of peak hours is also significantly delayed to 12:32, which is nearly 5 hours later than workdays. Further comparison can be found in Figure 3.13 and Table 3.10.

Table 3.5: Comparison between workdays and weekends on all 52 nodes. The percentage differences are defined by Equation 3.1 and 3.2. As shown, workdays have about 10% higher traffic volume and earlier start/end of peak hours than weekends.

	Max	Min	Mean
Percentage difference (all hours)	0.4105	0.0226	0.1087
Percentage difference (peak hours)	0.4854	-0.0278	0.0927
Start Peak Hours (Workdays)	09:14	06:42	07:38
Start Peak Hours (Weekends)	12:39	07:05	11:03
End Peak Hours (Workdays)	21:36	16:02	19:28
End Peak Hours (Weekends)	21:03	18:11	19:56

Figure 3.8: (Top) Weekends v.s. workdays at Node 279 and 5246. The green background highlights traffic volume on weekends. The red and orange lines indicate borderline of peak hours (highest 25% volume) on workdays and weekends respectively. As shown, weekends have lower traffic volume than workdays. (Middle) Histogram of percentage difference on all 52 nodes. The two figures in the middle indicate the traffic volume is on average 10% higher during workdays than weekends. (Bottom) Histogram of start/end of peak hours on all 52 nodes. The two figures at the bottom illustrate the change of average start/end of peak hours on weekends. The delay of average start of peak hours on weekends is about three and half hours which is clearly demonstrated on the bottom left histogram. The average end of peak hours is postponed about half an hour.

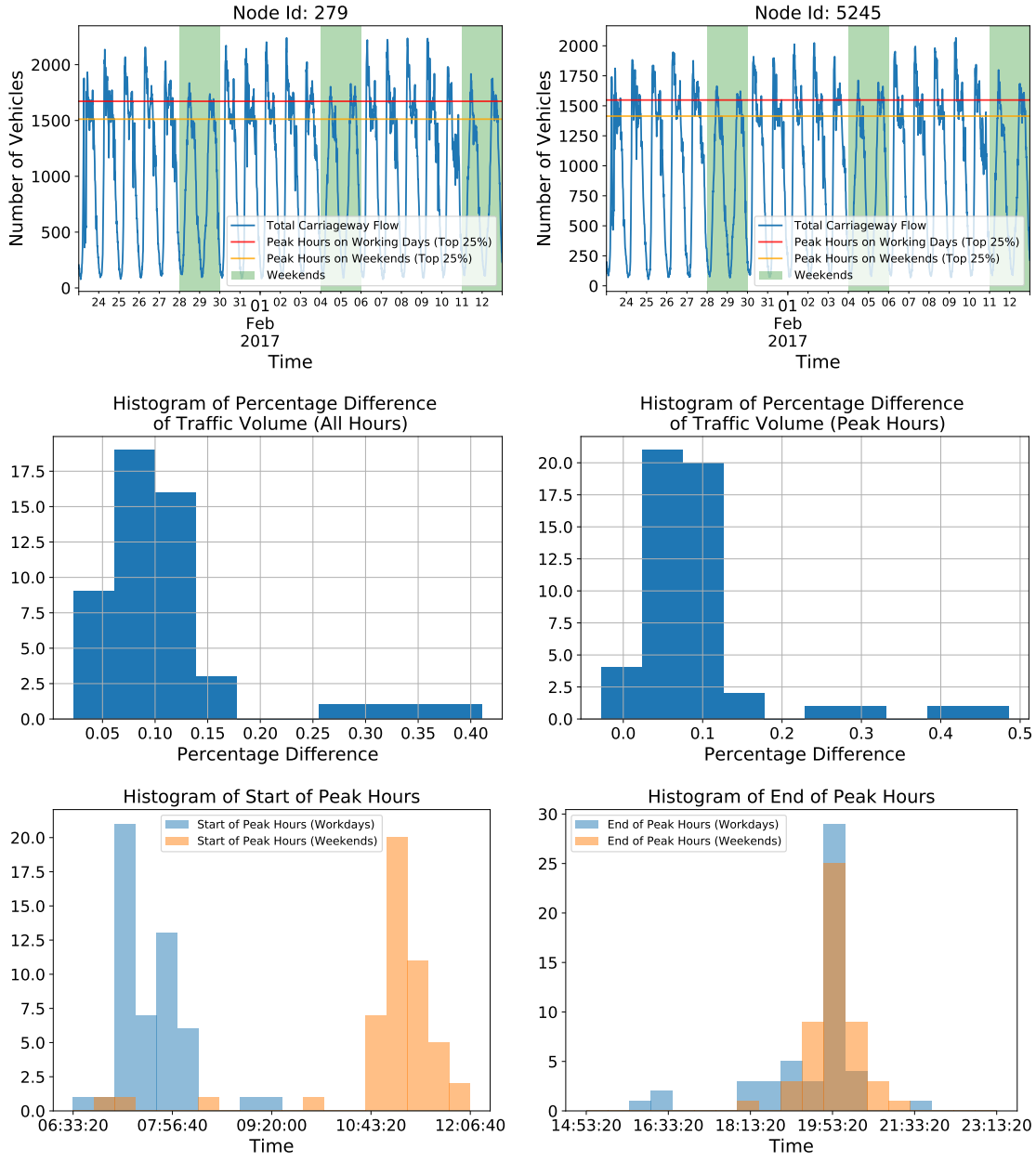


Figure 3.9: (Left) Workdays v.s. bank holidays at Node 279 and 5245. (Middle) Histogram of percentage difference on all 52 nodes. (Right) Histogram of start/end of peak hours on all 52 nodes. There is significant delay of start of peak hours and lower traffic volume compared with workdays.

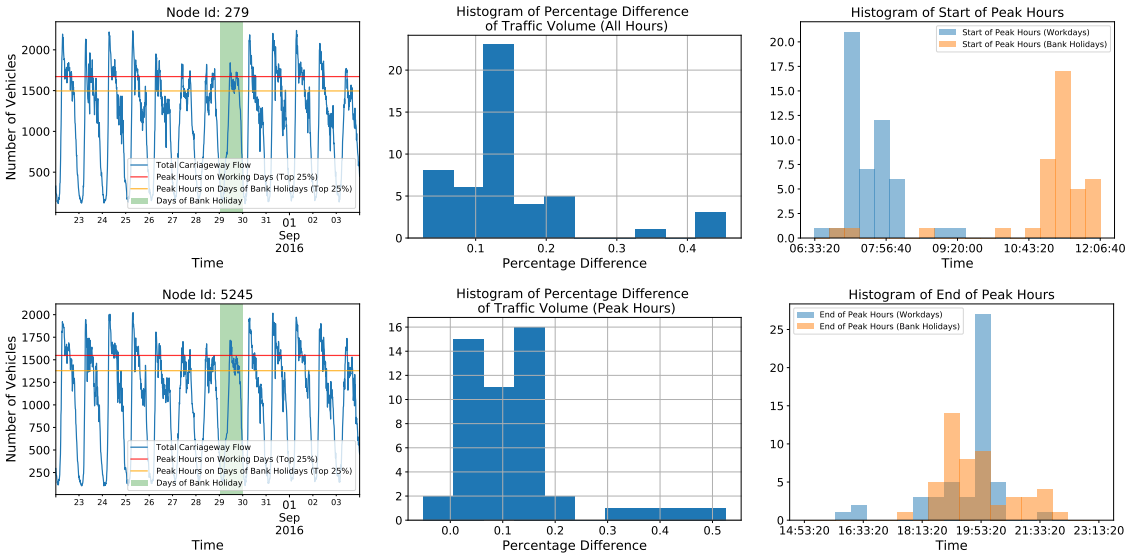


Table 3.6: Comparison between workdays and bank holidays on all 52 nodes. The traffic patterns on bank holidays are similar with weekends as shown in Table 3.5. Workdays have on average 14.76% higher traffic volume than bank holidays.

	Max	Min	Mean
Percentage difference (all hours)	0.4547	0.0258	0.1476
Percentage difference (peak hours)	0.5258	-0.0519	0.1149
Start Peak Hours (Workdays)	09:14	06:42	07:38
Start Peak Hours (Bank Holidays)	12:47	06:55	11:24
End Peak Hours (Workdays)	21:36	16:02	19:28
End Peak Hours (Bank Holidays)	22:05	17:51	19:47

Figure 3.10: (Left) Workdays v.s. school holidays at Node 279 and 5245. (Middle) Histogram of percentage difference on all 52 nodes. (Right) Histogram of start/end of peak hours on all 52 nodes. As shown, there is no much difference between workdays and school holidays.

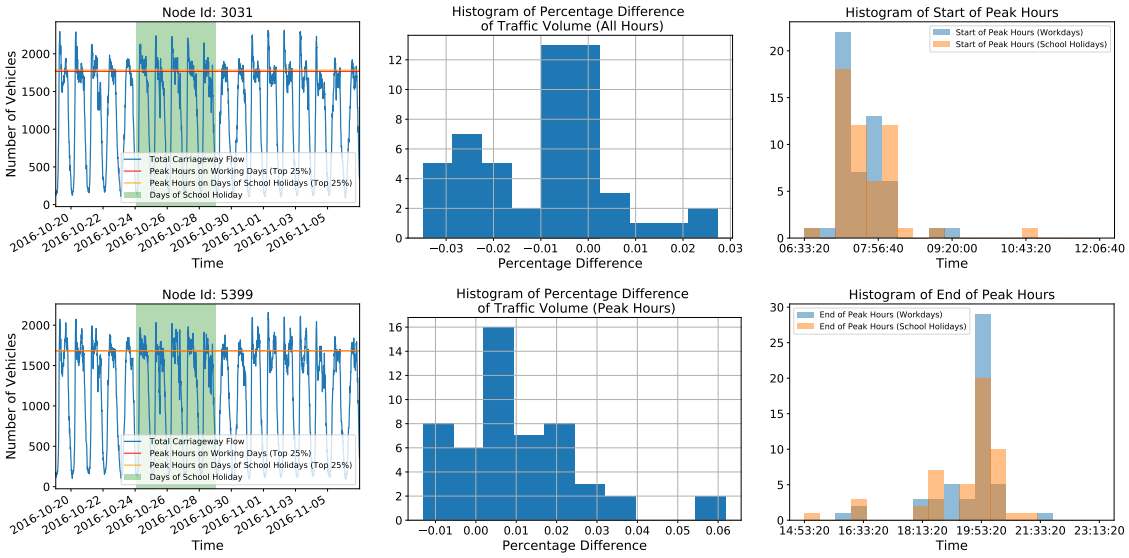


Table 3.7: Comparison between workdays and school holidays on all 52 nodes. There is no much difference between workdays and school holidays.

	Max	Min	Mean
Percentage difference (all hours)	0.0273	-0.0349	-0.0092
Percentage difference (peak hours)	0.0620	-0.0130	0.0102
Start Peak Hours (Workdays)	09:14	06:42	07:38
Start Peak Hours (School Holidays)	10:55	06:48	07:44
End Peak Hours (Workdays)	21:36	16:02	19:28
End Peak Hours (School Holidays)	21:19	14:34	19:17

Figure 3.11: (Left) Workdays v.s. Christmas period at Node 2933 and 3031. (Middle) Histogram of percentage difference on all 52 nodes, which indicates busier traffic during workdays. (Right) Histogram of start/end of peak hours on all 52 nodes. As shown, the Christmas period has much later start of peak hours but earlier ending of peak hours.

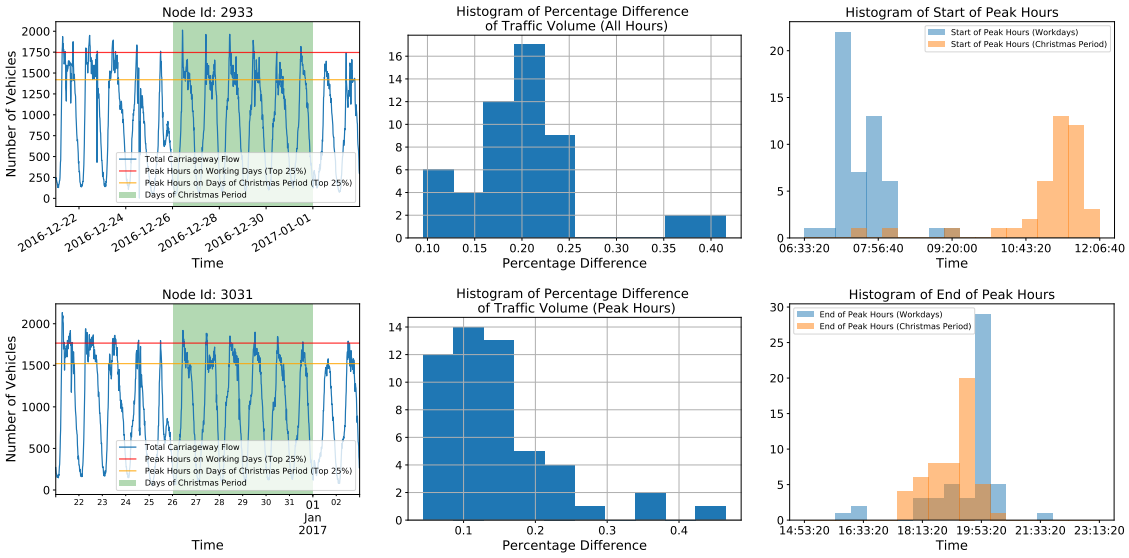


Table 3.8: Comparison between workdays and Christmas period on all 52 nodes. The Christmas period on average has an overall 20% less traffic and 14% less in peak hours than workdays. The average start of peak hours is further delayed to 11:26, which is nearly 4 hours later than workdays. However, the Christmas period has 24 minutes earlier ending of peak hours.

	Max	Min	Mean
Percentage difference (all hours)	0.4159	0.0955	0.2002
Percentage difference (peak hours)	0.4654	0.0438	0.1458
Start Peak Hours (Workdays)	09:14	06:42	07:38
Start Peak Hours (Christmas period)	12:42	07:34	11:26
End Peak Hours (Workdays)	21:36	16:02	19:28
End Peak Hours (Christmas period)	20:24	17:57	19:05

Figure 3.12: (Left) Workdays v.s. Christmas Day at Node 2933 and 3031. The highest volume on the Christmas Day is around the borderline of peak hours during workdays. (Middle) Histogram of percentage difference on all 52 nodes, which indicates significantly lower traffic on the Christmas Day. (Right) Histogram of start/end of peak hours on all 52 nodes. As shown, the Christmas Day has later start/end of peak hours.

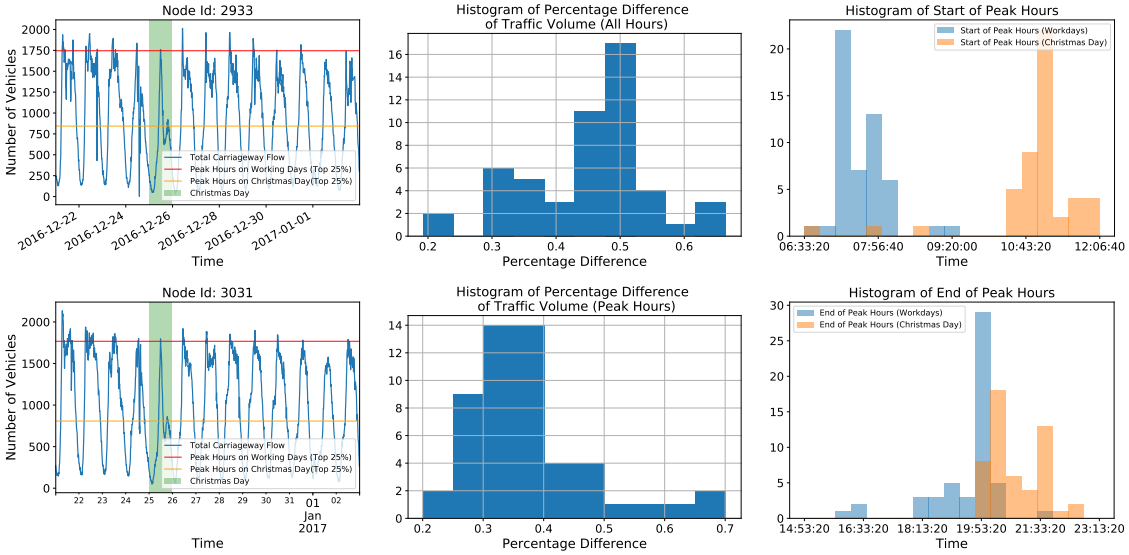


Table 3.9: Comparison between workdays and Christmas Day on 52 nodes. Christmas Day has significantly less traffic (45.06%) and later ending of peak hours than workdays.

	Max	Min	Mean
Percentage difference (all hours)	0.6652	0.1926	0.4506
Percentage difference (peak hours)	0.7014	0.2009	0.3737
Start Peak Hours (Workdays)	09:14	06:42	07:38
Start Peak Hours (Christmas Day)	13:09	06:44	10:59
End Peak Hours (Workdays)	21:36	16:02	19:28
End Peak Hours (Christmas Day)	22:29	19:44	20:48

Figure 3.13: (Left) Workdays v.s. New Year's Day at Node 2243 and 2598. Interestingly, there are midnight peak highlighted by green background. The minor peak is higher at Node 2598 than 2243, because the former node heads outwards central London while the latter one heads the opposite direction. (Middle) Histogram of percentage difference on all 52 nodes, which indicates lower traffic on the New Year's Day. (Right) Histogram of start/end of peak hours on all 52 nodes. As shown, the New Year's Day has significantly later start of peak hours around 12:30.

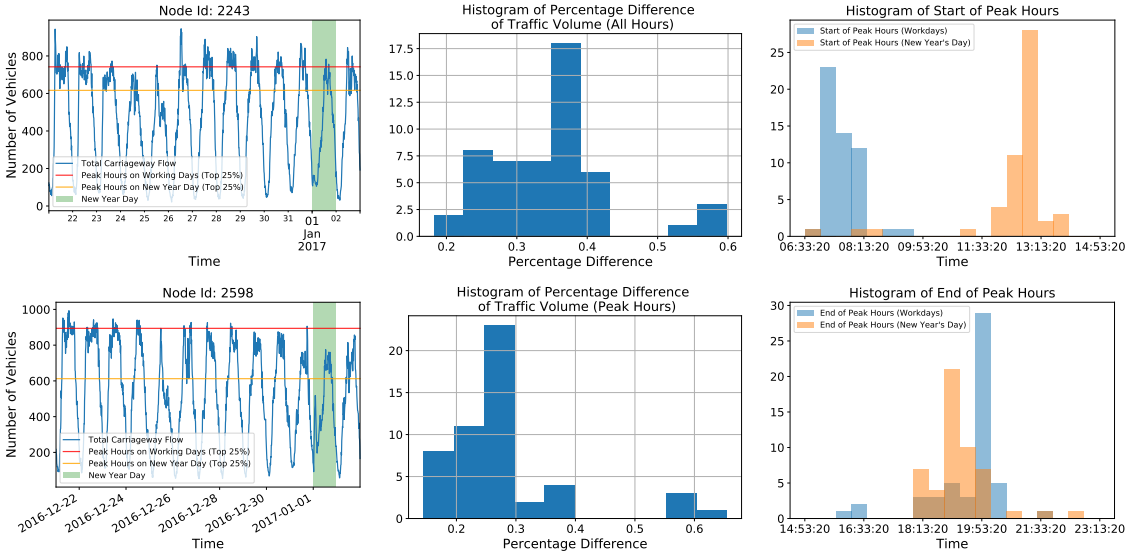


Table 3.10: Comparison between workdays and New Year's Day on all 52 nodes. New Year's Day has less traffic (35.14%) and rather later start of peak hours (nearly 5 hours) than workdays.

	Max	Min	Mean
Percentage difference (all hours)	0.5984	0.1824	0.3514
Percentage difference (peak hours)	0.6538	0.1450	0.2795
Start Peak Hours (Workdays)	09:14	06:42	07:38
Start Peak Hours (New Year's Day)	13:37	06:44	12:32
End Peak Hours (Workdays)	21:36	16:02	19:28
End Peak Hours (New Year's Day)	22:44	17:59	19:16

3.3 Spatial Correlation

Traffic volume at any time and location is correlated with not only previous observations and also observations on neighbouring places. Intuitively, in some busy routes, if a huge volume is observed at the entrances, the locations afterwards should expect and get ready for the volume soon. In the Highways England dataset, the 52 nodes, potentially correlated with each other, are located around Heathrow Airport, one of the busies airport worldwide.

Figure 3.14 illustrates the correlation coefficient matrix computed by VAR, one of the baselines. Some nodes have strong correlation with each other e.g. Node 1138, 1400 and 1449. However, some nodes on 'A' roads e.g. 6156-1, 6178-1 and 7044-1 have no clear spatial correlation with any others. In the following paragraphs of this section, we will discuss such spatial correlation in specific circumstances.

Same Direction Evidences indicate that nodes on the same direction of the same motorway have strong correlation. In Figure 3.15, we select five nodes on M4 westbound (Node 2598, 1449, 1138, 1400, 1017) and four nodes on M25 southbound (Node 5211, 496, 3031, 5399). Though the number of vehicles are different among nodes, the overall trend is highly correlated. We can notice the rise or drop of traffic volume happens with short delay or even simultaneously, which can be helpful for forecasting.

Opposite Direction Figure 3.16 selects four nodes on M4, two eastbound and two westbound. As depicted, both directions have similar peak hours, but the traffic eastbound has no direct impact on that westbound and vice versa. During the dramatic drop of traffic westbound, which might be caused by accidents, the traffic eastbound was not affected much. Therefore, the information from opposite direction for traffic prediction can be very limited.

Crossroads Circumstances on crossroads need to be discussed case by case. Firstly, we select nodes around the overpass in the northwest of Heathrow Airport, which is the intersection between M4 and M25. Additionally, the large roundabout in the north of Heathrow Airport on M4 is also a good case.

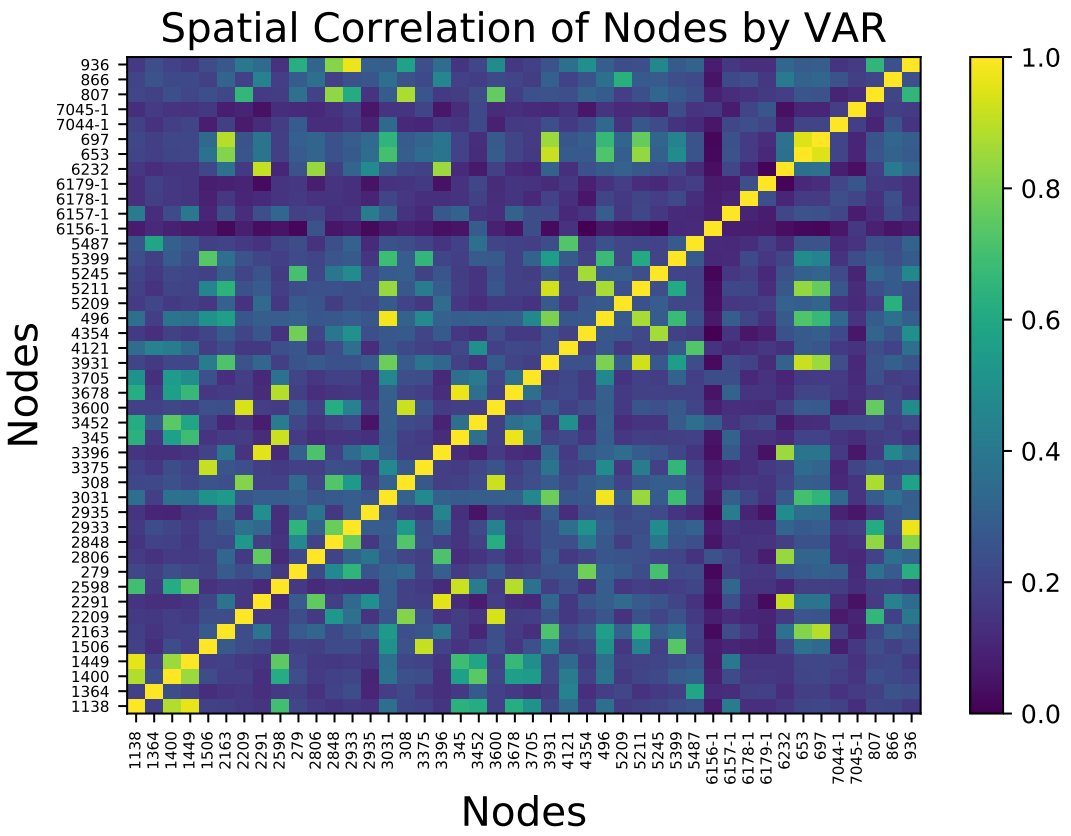
(1) One inwards multiple outwards. We select Node 3931 as input and three outwards nodes (Node 5211, 4121, 2291). Node 3931 and 5211 are both located on M25 southbound, while Node 4121 heads westbound and 2291 eastbound on M4. The top three figures in Figure 3.17 have shown that severe change of traffic volume on the input node does affect that on all the three outputs but the influence at Node 5211 is more significant. In most of the times, Node 3931 has no dominant impact at Node 4121 and 2291, but Node 5211 is clearly correlated with Node 3931.

(2) Multiple inwards one outwards. Node 3931, 1017, 121 are chosen as inputs and Node 3031 as outputs. Both Node 3931 and 3031 are located both on M25

southbound, while Node 1017 heads westbound and Node 121 eastbound on M4. The three figures in the middle of Figure 3.17 have shown that the output has highest traffic volume in most of the times and is directly affect by Node 3931. Minor noises at Node 1017 and 121 do not affect Node 3031 much. Moreover, traffic at Node 3031 dropped significantly on the Christmas Eve, as Node 3931 was severely blocked.

(3) Roundabout north of Heathrow Airport. We select Node 1138, 1017, 6156-1 and 6157-1 around the large roundabout. As shown in the map (bottom right of Figure 3.17), traffic flow from Node 1138 can go either straight to Node 1017 or left turn to Node 6157; and vehicles counted by Node 1017 come from either Node 1138 or Node 6156-1 [3]. The figures at the bottom of Figure 3.17 illustrate that most vehicles from Node 1138 head toward straight to Node 1017 on workdays, but during Christmas holidays, a large ratio of vehicles from Node 1138 turn left onto Node 6157-1 towards Heathrow Airport in the morning.

Figure 3.14: Spatial correlation matrix. It is a symmetric matrix whose diagonal values are higher than those off-diagonal. Some nodes e.g. Node 1138, 1400 and 1449 have significant correlation with one another, while other nodes e.g. 6156-1, 6178-1 and 7044-1 have no clear correlation with any others.



[3]Data of traffic going/coming from north is not available thus not taken into consideration. The conclusions following neither relies on the traffic from/to north.

Figure 3.15: Spatial correlation with same direction. (Top) Traffic volume of five nodes on M4 westbound (Node 2598, 1449, 1138, 1400, 1017). We can notice significant decreasing of traffic and midnight peak on New Year's Eve happened almost simultaneously. (Bottom) Traffic volume of four nodes on M25 southbound (Node 5211, 496, 3031, 5399). Similar traffic trend on M25 can be found and such correlations stand on both workdays and special days.

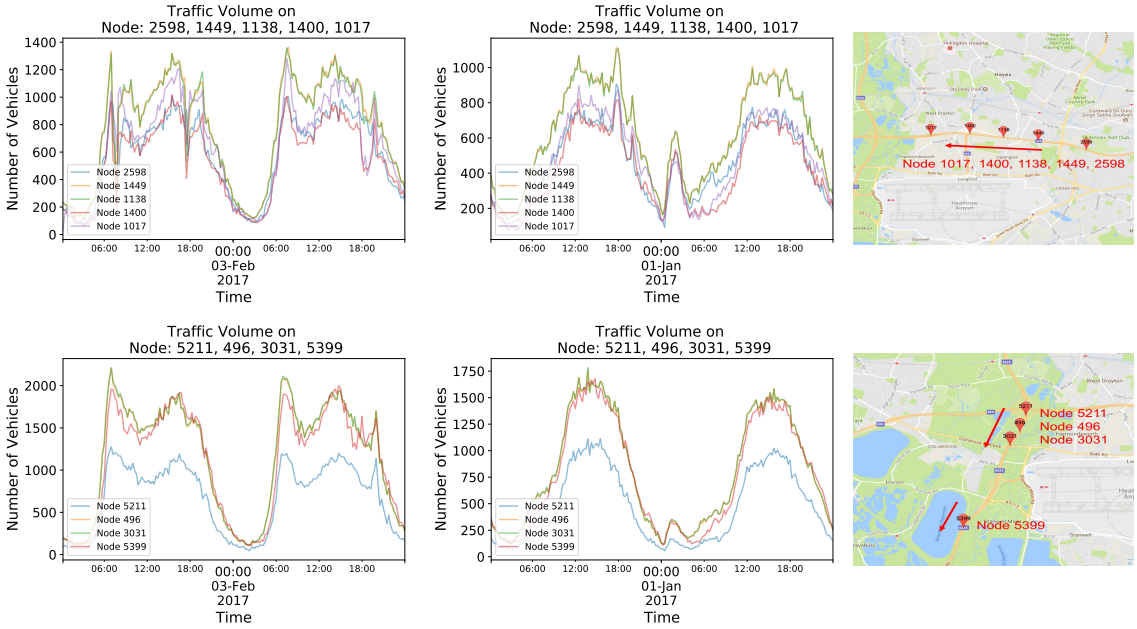


Figure 3.16: Spatial correlation with opposite directions. Two nodes westbound (Node 1449, 1138) and two nodes eastbound (Node 2806, 3470) are selected. Though nodes with on the same side have strong traffic similarity, correlation between opposite directions is not significant.

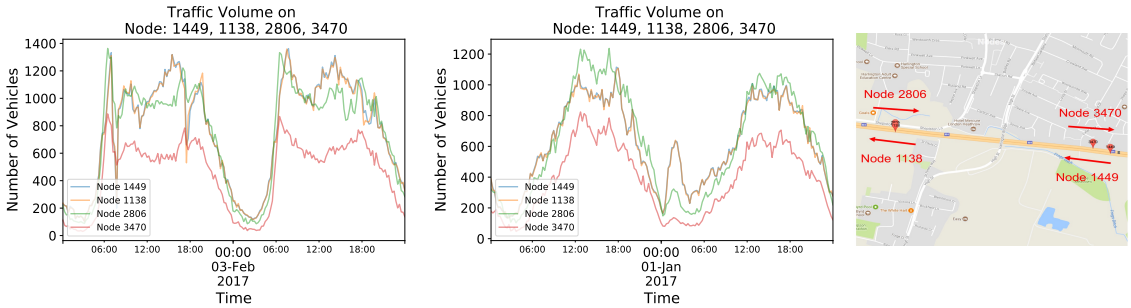
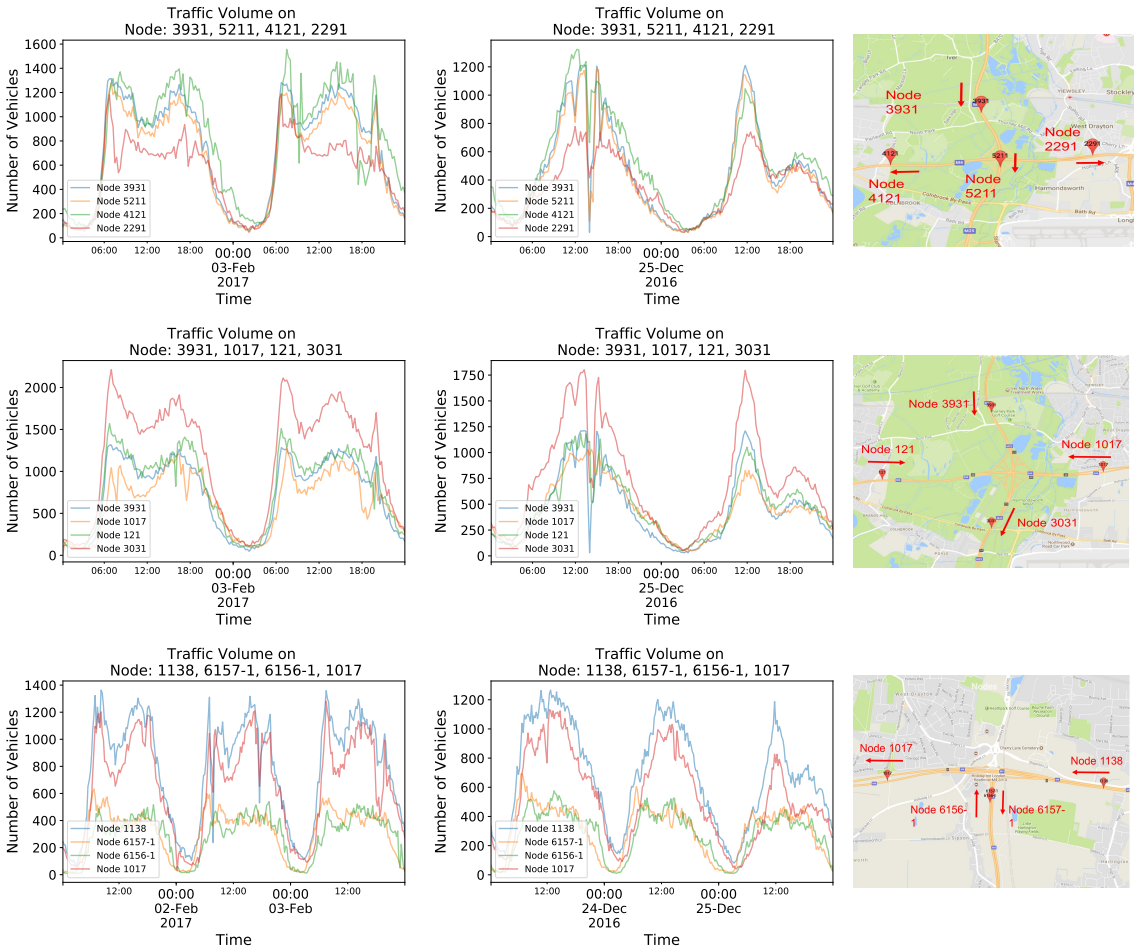


Figure 3.17: Spatial correlation on crossroads. (Top) One inwards (Node 3931) and three outwards (Node 5211 4121, 2291). The input has direct and dominant influence at Node 5211 which shares the same direction. Node 4121 (left turn) and 2291 (right turn) can be affected if the input traffic volume is severely blocked. (Middle) Multiple inwards (Node 3931, 1017, 121) and one outwards (Node 3031). Node 3031 has the highest volume mostly and is directly affect by Node 3931. Minor noises at Node 1017 and 121 do not affect Node 3031 much but significant changes at Node 3931 can influence all of them. (Bottom) Roundabout around Node 1138, 1017, 6156-1 and 6157-1. On workdays, most vehicles from Node 1138 head straight to Node 1017. However, during Christmas holidays, a large ratio of vehicles from Node 1138 turn left onto Node 6157-1 towards Heathrow Airport in the morning.



3.4 Data Refinery

With combination of multiple sources of raw data collection, the Highways England dataset achieves overall outstanding quality. However, outliers and error points are still inevitable. Besides, in order to make models training easier and quicker, it is necessary to prepare a cleaner and more compact dataset. In this section, we will introduce basic steps we followed in data refinery, which is implemented based on Pandas [4].

Deduplication In data deduplication, we eliminate duplicate copies of repeating data points. In the Highway England dataset, traffic data is sampled for each 15 minutes, which means there is supposed to be one and only one data point in each 15-minute time slice. Thus, data points are firstly grouped into each corresponding slice. Then for those groups that have multiple data points, we merge the data points into one with average values.

Validation Many reasons such as unexpected accidents, planned maintenance events or breaking down of sensors can result in invalid data points. If there are too many not-a-number or missing values (more than 10%) in a single day, we will drop all data points on that day, which is thought to be unreliable and unsuitable for models. Other invalid or missing data points are filled by interpolation.

Clipping Traffic has lowest volume in midnight during which there can be 10 vehicles per 15 minutes or even fewer in some 'A' roads. Such conditions can be a disaster for models, because 5-vehicle prediction error can lead to more than 50% error rate. Therefore, for better regularisation, we set a lower bound which is the lowest 2% and all values below the lower bound are replaced by the bound value.

Feature Selection The KDDCUP17 dataset provides several factors for traffic prediction, but not all of them have clear correlation with traffic volume. Based on the holiday information provided, we firstly group the dates into three categories, workdays, weekends and public holidays. We finally select traffic volume, type of days, precipitation as features for prediction.

[4] An open sourced Python library for data structures and data analysis. Pandas: `http://pandas.pydata.org/`, accessed 2017-08-22

Chapter 4

Methodology and Implementation

This chapter will firstly give a formal definition of short-term traffic prediction and analyse limitations of the baselines. We will secondly propose two deep learning models, CNN-LSTM with attention mechanism (CNN-LSTM-Attn) and Temporal-Spatial-LSTM (TS-LSTM). The CNN-LSTM-Attn combines LSTM with CNN embedding and incorporates global attention mechanism to model temporal dependencies. The second model TS-LSTM is built upon CNN-LSTM-Attn but utilises a relational module for spatial dependencies. We will finally introduce the implementation and design pattern of this project which make agile development and efficient model training feasible.

4.1 Short-Term Traffic Prediction

In general, given a sequence of observations $\{x_1, x_2, ..., x_t\}$ at a given location, modeling temporal relations only with model $\mathcal{M}$ and its parameters θ, the goal is to predict observations in the near future $\{x_{t+1}, x_{t+2}, ..., x_{t+H}\}$, which can be mathematically defined as:

$$\hat{x}_{t+1}, \hat{x}_{t+2}, ..., \hat{x}_{t+H} = \underset{x_{t+1}, x_{t+2}, ..., x_{t+H}}{\arg\max} P(x_{t+1}, x_{t+2}, ..., x_{t+H} | x_1, x_2, ..., x_t; \mathcal{M}; \theta) \quad (4.1)$$

Considering spatial correlation, we can further define a $K \times T$ matrix $\{X_1, X_2, ..., X_T\}$ which include observations at K nodes. The traffic prediction problem with both

temporal and spatial information can be rewritten as:

$$\hat{X}_{t+1}, \hat{X}_{t+2}, ..., \hat{X}_{t+H} = \underset{X_{t+1}, X_{t+2}, ..., X_{t+H}}{\arg\max} P(X_{t+1}, X_{t+2}, ..., X_{t+H} | X_1, X_2, ..., X_t; \mathcal{M}; \theta) \tag{4.2}$$

Most previous literature demonstrate traffic prediction within 30 minutes in the future. However, our goal is to predict traffic volume in the next 2 hours, which is longer and more challenging than most of short-term traffic prediction tasks. In the Highways England dataset, with 15-minute intervals, 8 points of observations are needed to predict in 2 hours. In the KDDCUP17 dataset, we need to predict 6 observations with 20-minute intervals.

4.2 Limitation of Baselines

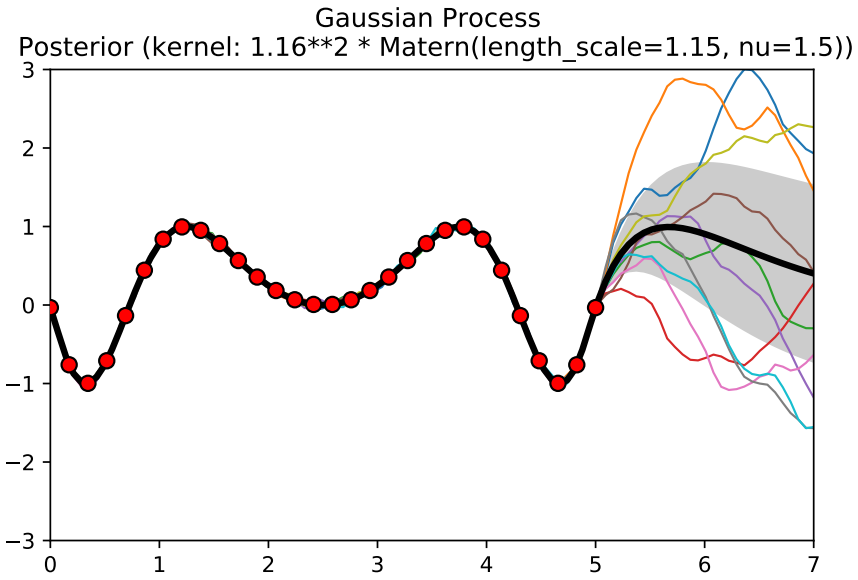
In the Highways England dataset, four traditional models are chosen as baselines, including SARIMA, VAR, Gaussian Process and Support Vector Regression. Previous literature have demonstrated their performance on traffic prediction tasks, but their limitation is also obvious.

SARIMA SARIMA integrates daily seasonality with autoregressive and moving average, which models the repeating daily pattern effectively and efficiently. (1) However, firstly, SARIMA with daily seasonality is insensitive to special conditions such as special days. Special conditions are not taken into consideration mathematically in SARIMA, so the traffic volume predicted on special days is almost same as workdays, which leads to high error rate. (2) In addition, not only on special days, traffic may have minor variance during workdays, but SARIMA cannot react to such variance properly. (3) Thirdly, SARIMA is not a general model, which means a SARIMA trained by a given sequence (at location A) cannot be applied on another sequence (at location B), if the two sequences are different. In the Highways England dataset, unless the two locations are very closed to each other on the same direction of the same road, the two traffic sequences at the two locations respectively are different with each other in most cases. Therefore, as we have 52 nodes in the Highways England dataset, 52 SARIMA models are needed.

VAR In the baselines, VAR is the only model that takes correlation among nodes into consideration. VAR performs well in very short-term forecasting (lag 1 or 15 minutes), but lacks stability and accuracy in longer prediction. Without moving average, the predicted sequence has higher variance and can be easily affected by minor noises of previous observations. Additionally, VAR is not sufficiently robust, as outliers in the whole transportation network can also influence prediction significantly.

Gaussian Process (GP) Firstly, Gaussian Process can approximate interpolation of given data points with good accuracy and high confidence, but the variance may go outrage as shown in Figure 4.1 if we want to predict observations further right. Secondly, inevitable minor noises make traffic observations look zigzag, which makes it difficult to choose proper kernels and may lead to overfitting.

Figure 4.1: The variance of Gaussian Process may go outrage on the right.



Support Vector Regression (SVR) has similar limitations as the above three baseline models. Performance of SVR relies on choice of kernels. RBF has been proved with outstanding performance in previous literature, but RBF may easily result in overfitting if the sequence is not smooth enough. The linear kernel is more stable for very short-term prediction but fails to achieve satisfying accuracy in longer forecasting.

Experimental support of these arguments and quantitative comparison between baselines and our proposed methodologies will be covered in details in Section 5.3.

4.3 Temporal Dependencies

An essential and fundamental assumption that lead to feasibility of short-term traffic prediction is the consistent seasonality and strong temporal patterns. RNN is born to process sequential data and designed to encode accumulated observations. With leverage of LSTM, RNN is able to connect relative information between data points with a large gap and handle long-term dependencies.

RNN with LSTM has been widely applied in natural language processing. However,

compared with text applications, which have a sequence length of tens or 250 at most (Karpathy et al. (2015); Sutskever et al. (2011); Yu et al. (2017)), traffic prediction have rather longer sequential observations. With 15-minute intervals in the Highways England dataset, there are 96 observations a day and 672 a week. Both intuition and experiments make us believe that more previous observations (longer input sequence) can make the forecasting more robust if modelling properly, while on the other hand, prediction is prone to bias in previous data points with short observations.

Therefore, we propose **CNN-LSTM-Attn** model, which applies CNN for long sequence embedding and global attention mechanism for seasonality. We claim that CNN-LSTM-Attn is capable to leverage both longer observations and consistent seasonal dependencies for traffic volume prediction. Figure 4.2 demonstrates the network structure of CNN-LSTM-Attn.

Reshaping Input The model uses one-week observations as input which is reshaped into image-like matrix. The new shape of the input is (7, 96, 2), which refers to 7 days a week, 96 data points a day, and 2 features each point (traffic flow + type of the day).

CNN A 2-layer 2D CNN with ReLU and max pooling is applied to reduce the length of sequence and extract more features as representation. CNN is able to capture patterns in local region and processed sequence is then fed into the RNN with LSTM.

Global Attention Normally, the target hidden state h_t would be given to the FC decoder directly for outputs. However, the seasonality has shown that short-term prediction is highly related with observations at the same time days before. For example, prediction at 8-10am should be similar with previous observations at 8-10am but has trivial relations with traffic in midnight. Thus, we introduce global attention mechanism proposed by Luong et al. (2015) which considers all source hidden states of the RNN and derive an attentional hidden state $\hat{h_t}$.

The alignment vector α_t is defined to measure relations between each source hidden state h_s and the target hidden state h_t.

$$\alpha_t(s) = \text{softmax}(h_s, h_t) = \frac{\exp(\text{score}(h_s, h_t))}{\sum_{s'} \exp(\text{score}(h_{s'}, h_t)))} \tag{4.3}$$

where score is a content-based linear function which is defined as:

$$\text{score}(h_s, h_t) = h_s^T W_\alpha h_t \tag{4.4}$$

The alignment vector α_t will then be used to derive a context vector c_t, which is

a weighted linear combination of all source hidden states and their corresponding alignment vectors.

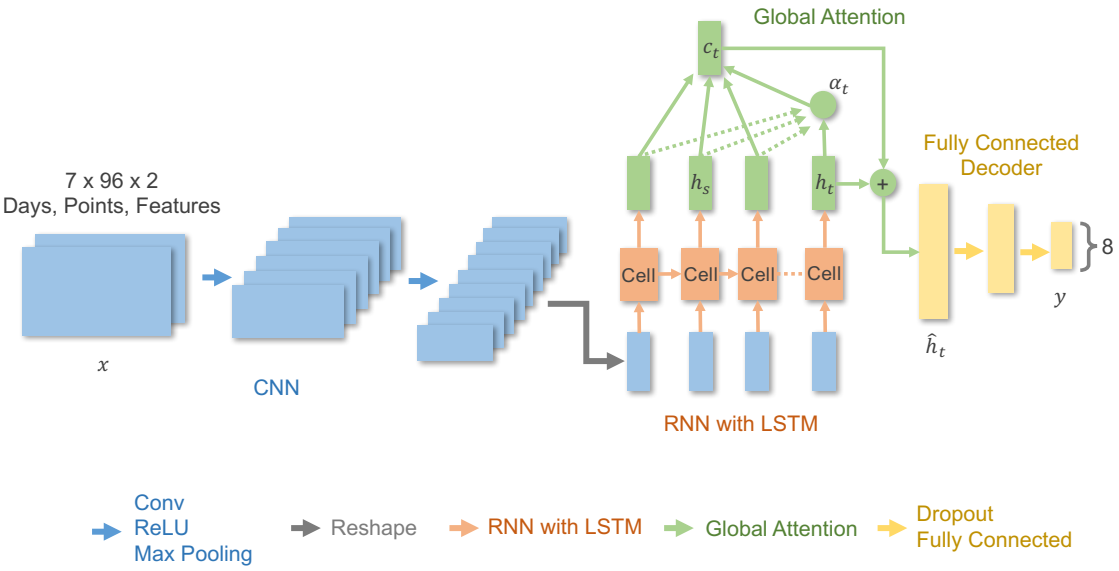
$$c_t = \sum_s \alpha_t(s) h_s \tag{4.5}$$

The attentional hidden state $\hat{h_t}$ is the concatenation of c_t and h_t, and will be transmitted to the FC decoder.

FC Decoder The decoder is composed of dropout and fully connected hidden layers. The dropout is applied as an efficient regularisation to prevent overfitting (Srivastava et al. (2014)). The decoder has eight outputs which are the predicted values of eight observations in the next 2 hours. The eight values are predicted at the same rather than one by one, because we believe that all information needed to predict future values is inside the input observations. If the representation learned by CNN-LSTM-Attn is good enough, the decoder should be able to forecast with good accuracy. As the next 2-hour values are predicted together, there is no need to deal with accumulation of mistakes (Pascanu et al. (2013)).

The structural settings of CNN-LSTM-Attn, such as 1D or 2D CNN, and number of units in the FC decoder, rely on the size of input sequence and vary if we use one-day rather than one-week observations as input. But the idea and overall structure of CNN-LSTM-Attn remain the same and can potentially be applied to other sequential data.

Figure 4.2: CNN-LSTM-Attn network structure. The CNN-LSTM-Attn consists of four sub-networks, CNN, RNN with LSTM, global attention mechanism and fully connected decoder.



4.4 Spatial Dependencies

Temporal dependencies provide fundamental clues for short-term traffic prediction, and spatial dependencies can do the same. Section 3.3 has demonstrated clear spatial correlation among nodes, especially nodes with same direction and those around crossroads. Therefore, modelling and encoding spatial dependencies among nodes could be useful for prediction.

Santoro et al. (2017) proposes a simple module for relational reasoning in CNN. In the visual question answering scenario, the relational module works together with CNN and RNN to combine information from images and text, and understand relation between objects. For example, given an image of 3D-rendered objects, the deep neural network with relational module is able to answer whether there are any rubber objects that have same size as the yellow metallic cylinder. The relational module is composed of two separated FC networks and considers relations across all pairs of objects, which are extracted by CNN.

Besides, Li et al. (2017) introduces an spatial attention mechanism in RNN. The idea is to use a content-based function to measure relations between two nodes and derive an spatial attentional hidden state which is a weighted combination of target hidden states of all nodes.

Based on the relational module and the spatial attention mechanism, we propose **Temporal-Spatial-LSTM (TS-LSTM)**, which is able to incorporate both temporal dependencies and spatial correlation. Figure 4.3 illustrate the network structure of TS-LSTM.

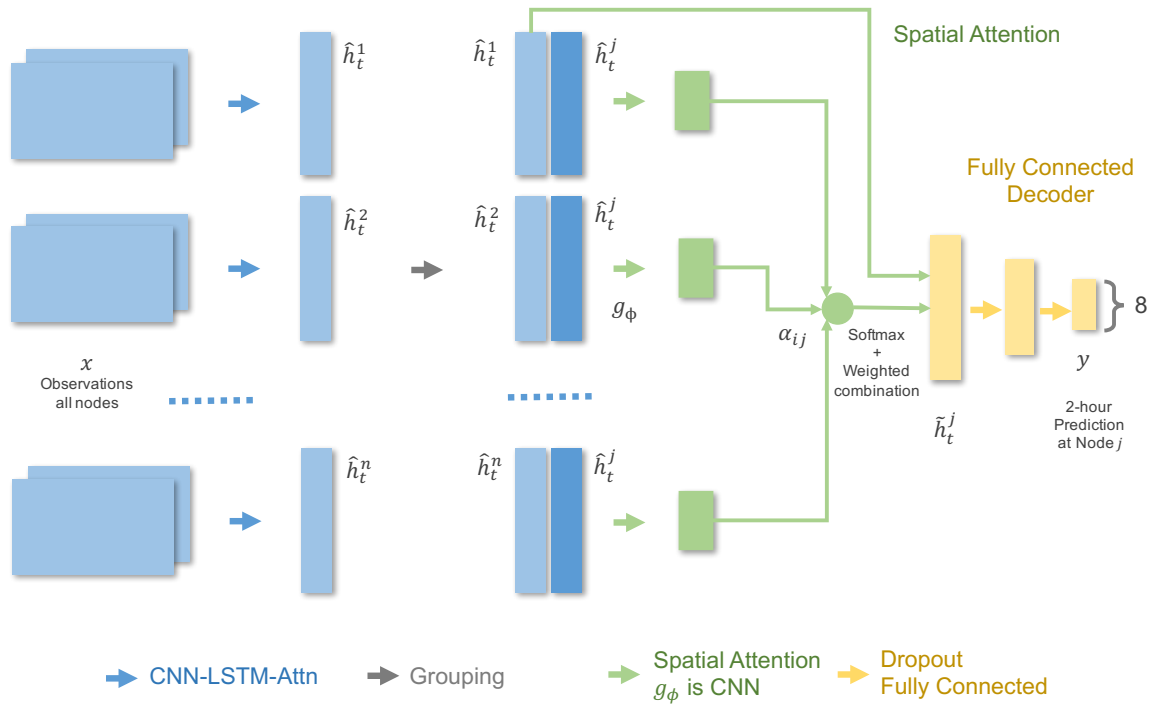
The TS-LSTM firstly uses the attentional hidden states $\hat{h}_t^i$ computed by CNN-LSTM-Attn at each node as input for the spatial attention module, which shares the same idea as the relational module. Given a node j we want to predict, the spatial attention module computes relation across pairs between all nodes and the node j by a content-based function g_ϕ, which is a CNN. The alignment vector α_{ij} is the softmax of g_ϕ outputs. The spatially attentional hidden states $\widetilde{h}_t^j$ is defined as weighted linear combination of $\hat{h}_t^i$ and its corresponding alignment vector α_{ij}, which is then fed into the FC decoder and makes the prediction.

$$\begin{aligned} \alpha_{ij} &= \text{softmax}(h_t^i, h_t^j) \\ &= \frac{\exp(g_\phi(h_t^i, h_t^j))}{\sum_{i'} \exp(g_\phi(h_t^{i'}, h_t^j)))} \\ \widetilde{h}_t^j &= \sum_i \alpha_{ij} \hat{h}_t^i \end{aligned} \tag{4.6}$$

During training stage, for a faster convergence, the CNN-LSTM-Attn is pre-trained

while other parameters are initialized randomly. We can simply understand the design of TS-LSTM that CNN-LSTM-Attn is trained to extract temporal dependencies of each node and the spatial attention module learns to infer spatial dependencies across pairs of nodes. The spatial attention module is not necessarily privy to which relations between nodes actually exist, nor the actual meaning of any particular relation. Thus, the spatial attention module must learn to unearth the implication and existence of spatial relations without specification and supervision. Moreover, as the spatial relational module is rather flexible, nodes with trivial spatial correlation implied by prior knowledge can be dropped.

Figure 4.3: TS-LSTM network structure.



4.5 Implementation

The implementation of deep neural networks is based on TensorLayer (Dong et al. (2017a)), which is a high-level library of TensorFlow (Abadi et al. (2015)). Our design pattern takes advantage of functional programming and object-oriented programming and provides agile project development with simple structure, friendly interfaces, minimum redundancy and sufficient flexibility.

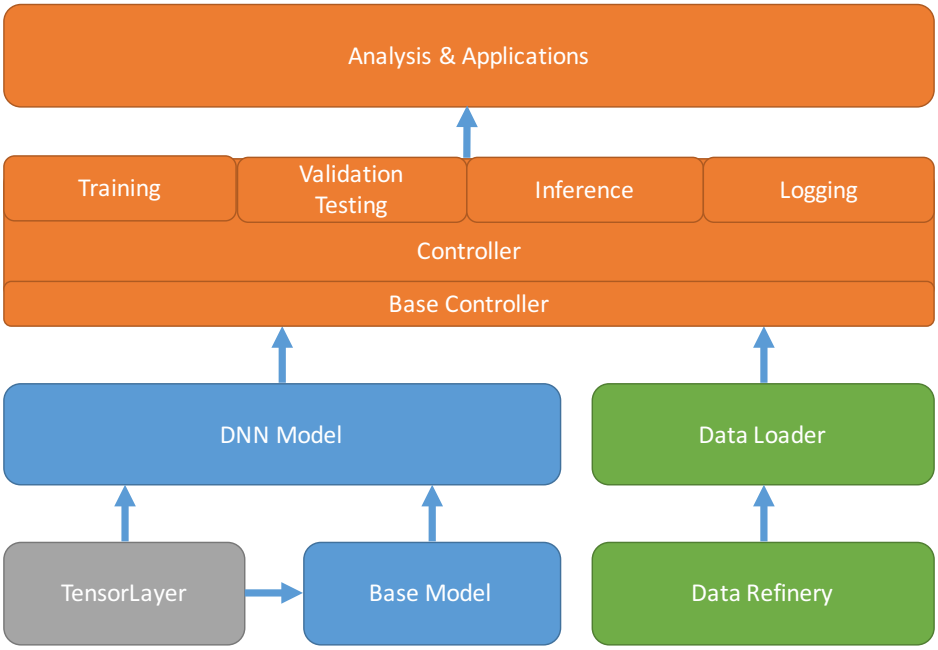
Figure 4.4 demonstrates the structure of our project.

1. The green section supports data refinery, data analysis and data loading for

controllers. Results of data analysis and methods of data refinery have been discussed in Chapter 3.

2. In the blue section, we firstly defines a base model, which integrates several fundamental interfaces including placeholders, training operations, loss functions, network structure and models (Source Code 4.1). New deep neural networks inherit the base model and acquire all features of the superclass. The only modification needed is to update the network structure. Besides, methods can be overrode if any other interfaces need updating. For example, we can firstly build a simple RNN for the prediction problem, and then other models can share the same definitions of input, output, loss functions and optimisers, but only a new network structure.
3. The orange section implements controllers and applications. Similarly, the base controller defines the training, testing and inference process (Source code 4.2). Some functionalities such as model saving/loading can be also implemented within the base controller. Generally, there should be one controller corresponds with each DNN model. Most controllers share exactly same process for training/testing/inference, so we only need to update some basic settings e.g. path of directory for logging.

Figure 4.4: Project structure. All deep neural network models are implemented based on TensorLayer and the base model. Given the model and refined data, the controller defines the process for training, validation, testing and inference. Applications including analysis and visualisation is built upon prediction results.



Source Code 4.1: An example of implementation of the base model, which will be inherited and overrode by other DNN models. As most models share same definition of input, output, loss functions and optimisers, only minor changes needed in child classes.

```
import tensorflow as tf
import tensorlayer as tl

# Base model
class Model():

    # Initialisation of the model
    def __init__(self, model_name):
        # model name will be used as scope of variables
        self.model_name = model_name

    # Definition of placeholders for input, output
    # And other placeholders if necessary
    def __create_placeholders__(self):
        # x = tf.placeholders
        # y = tf.placeholders
        pass

    # Deep neural network structure
    # Needs to be updated for different models
    def __get_network__(self, x, is_train, reuse):
        pass

    # Definition of train_net for training
    # Definition of test_net for validation, testing and inference
    def __create_model__(self):
        # train_net = self.__get_network__(x, is_train=True, reuse=False)
        # test_net = self.__get_network__(x, is_train=False, reuse=True)
        pass

    # Definition of loss functions
    def __create_loss__(self):
        pass

    # Definition of optimisers, learning rate decay policy
    # And other training operations if necessary
    def __create_training_op__(self):
        pass
```

Source Code 4.2: An example of implementation of the base controller, which will be inherited and overrode by other controllers. As some models may share exactly the same training, testing and inference process, we only need to modify some basic settings e.g. path of directory for logging.

```
import tensorflow as tf
import tensorlayer as tl

# Base controller
class Controller():

    # Initialisation of controller
    def __init__(self, model):
        # the deep neural network model
        self.model = model

    # Definition of inference
    def __test_sequence__(self, epoch, test_batch_x, test_batch_y):
        pass

    # Definition of validation and testing
    def __valid_test__(self, epoch, test_batch_x, test_batch_y):
        pass

    # Definition of training
    def __train__(self, epoch, train_batch_x, train_batch_y):
        pass

    # Controller for inference
    def controller_test_sequence(self):
        # load testing data node by node
        # then call self.__test_sequence__()
        # get prediction results and save
        pass

    # Controller for testing
    def controller_test(self):
        # load testing set then call self.__valid_test__()
        pass

    # Controller for training
    def controller_train(self):
        # load training set then call self.__train__()
        # meanwhile, load validation set and call self.__valid_test__()
        pass
```

Chapter 5

Experiments and Evaluation

We finished two tasks to demonstrate the outstanding performance of our proposed methodologies. Task one is to predict traffic volume on the Highways England dataset and compare our models with baselines quantitatively. The second task is exactly same as KDDCUP 2017 traffic volume prediction and we will demonstrate that our model outperforms the model that got 1st place in the competition. In this chapter, we will firstly introduce some basic information and settings of experiments and models. Then we will quantitatively compare our models with baselines in the two tasks, and analyse the experimental results. Thirdly, based on the forecasting results, we will demonstrate some interesting discoveries and potential applications. Finally, the visualisation system and demonstration will be covered at the end of this chapter.

5.1 Basic Information

5.1.1 Tasks

Task 1: Traffic volume prediction on the Highways England dataset. Given traffic data of the past day (96 observations) or week (672 observations), the goal is to predict number of vehicles in the next 2 hours with 15-minute time intervals (8 observations). This task tries to demonstrate outstanding stability and accuracy of our proposed methodologies.

Task 2: Traffic volume prediction on the KDDCUP 2017 dataset. Similarly, given traffic data of the past week, 2-hour traffic data prior testing period and environment factors, the goal is to predict number of vehicle in the 2 hours specified as illustrated by Figure 3.2. This task tries to show that our model outperforms than the model

which got first place in the competition.

5.1.2 Datasets Partition

An essential and fundamental rule of traffic prediction problem is that we should not use observations in the future to predict those in the past. Therefore, when splitting training set and testing set, we strictly follow the rule and guarantee that the testing set is after the training set.

The Highways England dataset is divided into three parts. We use data from 01/01/2016 to 30/09/2016 as training set, 01/10/2016 to 31/12/2016 as validation set and data from 01/01/2017 to 30/06/2017 as testing set.

For the KDDCUP 2017 dataset, as described in Section 3.1.2 and Figure 3.2, we use data of traffic volume from 19/09/2016 to 10/10/2016 as training set, the week 11/10/2016 to 17/10/2016 as validation set, and 18/10/2016 to 24/10/2016 as testing set. In the testing set, we only need to predict traffic volume during 8-10am and 17-19pm each day, and data 2-hour prior the testing slices is given as a reference.

During training period, validation set is firstly used as a indicator for early stopping to prevent overfitting, and then fed back into training (Goodfellow et al. (2016)) before the model is applied on the testing set.

Table 5.1: Datasets partition in experiments.

		Start	End	Full Batch Size
Highways England dataset	Train	01/01/2016	30/09/2016	1,136,290
	Valid	01/10/2016	31/12/2016	387,860
	Test	01/01/2017	30/06/2017	814,292
KDDCUP 17 dataset	Train	19/09/2016	10/10/2016	3456
	Valid	11/10/2016	17/10/2016	
	Test	18/10/2016	24/10/2016	14

5.1.3 Evaluation Metrics

MAPE (Mean Absolute Percentage Error) is applied as the final evaluation metrics and also used as the loss function to update parameters of deep neural networks.

Mathematically, MAPE is defined as:

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_i - t_i|}{t_i} \tag{5.1}$$

where t_i is ground truth and y_i is prediction.

In the early stage of training, we use MSE (Mean Square Error) to speed up parameters optimisation, as MSE is more sensitive to error with large absolute value, which can help to fit data with large values like traffic during peak hours more quickly.

$$MSE = \sum_{i=1}^{N}(y_i - t_i)^2 \tag{5.2}$$

5.2 Models Settings

5.2.1 Baselines

SARIMA uses order (1, 0, 1) for ARIMA and seasonal order (1, 1, 0, 96). **VAR** uses order 1 for autoregressive process. 44 nodes, rather than all 52, are used for prediction due to missing and invalid observations in some nodes. Both SARIMA and VAR are implemented by the StatsModels [1].**Gaussian Process (GP)** uses Matérn covariance function as the kernel with length scale $l = 8$ and $\nu = 0.5$. **Support Vector Regression (SVR)** uses linear kernel with default penalty parameter $C = 1$ of the error term. Both GP and SVR are implemented by the scikit-learn [2]. All the four baseline models are tested on the Highways England dataset only.

5.2.2 DNN

Table 5.2 summarises the structure of the CNN-LSTM-Attn and details of some training settings for Task 1. The structure of TS-LSTM is included in Table 5.3. Only one CNN-LSTM-Attn and one TS-LSTM will be trained for the whole transportation network.

Only CNN-LSTM-Attn has been tested on the KDDCUP 17 dataset because prior knowledge for spatial correlation is unclear. The settings of CNN-LSTM-Attn for KDDCUP17 dataset is mostly same as Table 5.2. In the testing set of KDDCUP17,

[1]StatsModels: `http://www.statsmodels.org/dev/index.html`, accessed: 2017-09-02.
[2]scikit-learn: `http://scikit-learn.org/stable/`, accessed: 2017-09-02.

we only need to predict traffic volume at 8-10am and 5-7pm given ground truth at 6-8am and 3-5pm as shown in Figure 3.2. Therefore, during training and inference, we use historical data in highly related time period as input, illustrated in Figure 5.1, rather that all observations in the previous week.

By default, all the baseline and DNN models use previous one-week observations as input for training and inference.

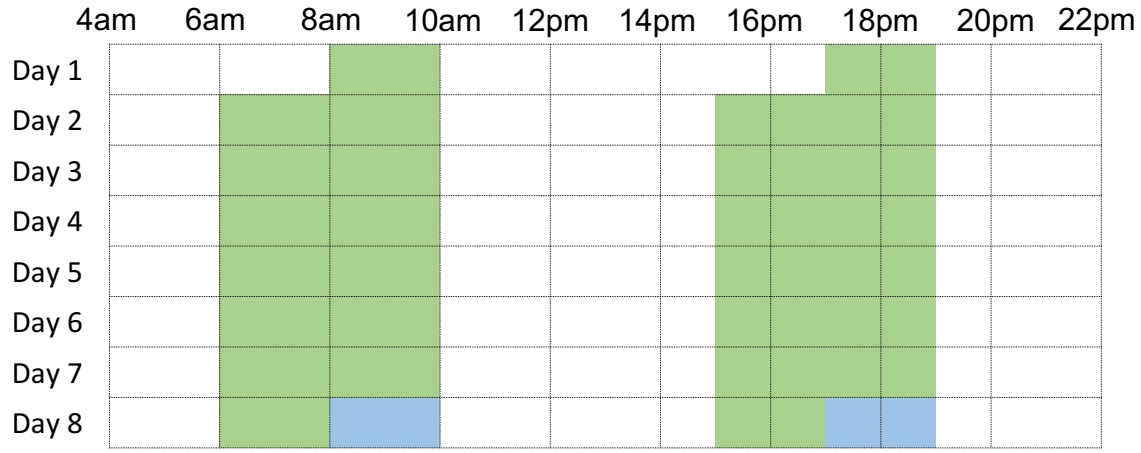
Table 5.2: Settings of CNN-LSTM-Attn for Highways England dataset.

Layers	Settings
Input	One-week observations by default. The size (7, 96, 2) refers to 7 days a week, 96 data points a day and 2 features per each data point (traffic volume + type of day). We use mini-batch with batch size 128.
CNN	Two-layer 2D CNN with ReLU and max pooling. The first convolutional layer has 16 filters and each filter is 7×8 with strides 1×1. The first max pooling layer has a filter size 1×2 with strides 1×2. The second convolutional layer has 32 filters and each filter is 7×4 with strides 1×1. The second max pooling layer has a filter size 7×2 with strides 7×2.
RNN	Single-layer LSTM as cell with 256 units of the hidden state.
FC Decoder	Two fully connected layers with two dropout layers. The first FC layer has 128 units and the second has 8 units. Both FC layers use identity activation function. Both dropout layers keep 80% parameters when training.
Notes	Parameters are all initialized randomly by uniform distribution between [-0.05, 0.05]. Adam optimiser is applied for parameter optimisation (Kingma and Ba (2014)). MSE is used as loss function in early training stage, then MAPE. The learning rate is initially set as 0.002 and exponentially decays with rate 0.8 every two epochs.

Table 5.3: Settings of TS-LSTM for the Highways England dataset.

Layers	Settings
CNN-LSTM-Attn	Same as the settings in Table 5.2. 44 nodes are used and mini-batch with batch size 32.
g_ϕ	Three-layer 2D CNN with ReLU and max pooling, and three fully connected layers with two dropout layers. The output has only one unit which is α_{ij}.
Notes	Parameters are all initialized randomly by uniform distribution between [-0.05, 0.05]. Adam optimiser is applied for updating. CNN-LSTM-Attn is pre-trained.

Figure 5.1: Input of CNN-LSTM-Attn for KDDCUP17 dataset. Instead of using all observations in the previous week, only the highly related historical data is selected as input. For example, if traffic volume between 8-10am on the 8th day (left blue block) needs to be predicted, the historical observations between 6-10am (highlighted by green on the left) will be used as input. Input sequence for forecasting 17-19pm is selected by the same rule.



5.3 Experimental Results

Task 1: On the Highways England dataset, we quantitatively compare performance of the baselines with DNNs. Table 5.4 has shown that CNN-LSTM-Attn achieves best performance with overall 9.26% error rate on the testing set, lowest compared with all other models. Besides, in longer term prediction, CNN-LSTM-Attn is more stable and accurate.

However, TS-LSTM is less accurate than CNN-LSTM-Attn and only demonstrates comparable performance with LSTM. Many reasons as cited below may lead to underfitting, which can be solved if more time is allowed for this project in the future.

1. Local region: The nodes selected in the Highways England dataset are distributed in a relative small region with high average speed, which means the average journey time in the region is short (12 minutes estimated by Google Map on M4 from Brentford to Slough, London), even shorter than the sampling interval (15 minutes). Section 3.3 demonstrates strong correlation among specific nodes but the influence happened with minor delay almost simultaneously. Therefore, prediction improvement, especially for longer term, with information from other nodes can be very limited.
2. Complexity: As prior knowledge is not necessarily needed for the spatial attention mechanism, the module must learn to extract relations across every pair of nodes. Given N nodes, there are N^2 pairs that the module needs to learn. Therefore, with more nodes, the complexity of TS-LSTM grows explo-

sively and more parameters are necessary to prevent underfitting. Moreover, the depth of TS-LSTM is doubled than CNN-LSTM-Attn, so the efficiency of updating parameters is also a concern.

Table 5.5 quantitatively compares performance among different DNNs. Both global attention mechanism and CNN embedding can effectively encode temporal dependencies in traffic prediction and have achieved improvement with lower MAPE than the raw LSTM. Besides, CNN is more efficient than the attention mechanism especially for longer term prediction.

Table 5.6 demonstrates longer observations can lead to higher accuracy. One-day observations are used as input of CNN-LSTM-Attn and compared with the CNN-LSTM-Attn fed by one-week observations. Experiments have shown that the latter one achieves lower MAPE. Intuitively, with only one-day observations, the model is more vulnerable and prone to variance of traffic data on that day, but CNN-LSTM-Attn with one-week observations is more robust as more historical data is taken into consideration.

Table 5.4: MAPE (%) on the Highways England testing set. CNN-LSTM-Attn achieves best overall performance with lowest error rate 9.26%.

Prediction	SARIMA	VAR	GP	SVR	LSTM	CNN-LSTM-Attn	TS-LSTM
15-min		8.99	12.15	9.48	7.73	**7.54**	8.53
30-min		12.06	19.16	13.23	8.77	**8.35**	8.50
45-min		15.09	25.47	17.01	9.51	**8.88**	9.17
60-min		18.37	31.17	20.94	10.02	**9.25**	9.86
75-min		21.93	36.27	25.14	10.42	**9.60**	10.33
90-min		25.74	40.87	29.47	10.74	**9.89**	10.59
105-min		29.77	45.03	33.97	11.04	**10.12**	11.05
120-min		33.93	48.82	38.64	11.54	**10.44**	12.07
Overall	22.00	20.73	32.37	23.48	9.97	**9.26**	10.01

Table 5.5: Comparison among different DNNs on the Highways England testing set.

Prediction	LSTM	LSTM-Attn	CNN-LSTM	CNN-LSTM-Attn	TS-LSTM
15-min	7.73	7.66	**7.52**	7.54	8.53
30-min	8.77	8.66	**8.35**	**8.35**	8.50
45-min	9.51	9.35	8.96	**8.88**	9.17
60-min	10.02	9.84	9.36	**9.25**	9.86
75-min	10.42	10.21	9.70	**9.60**	10.33
90-min	10.74	10.48	9.98	**9.89**	10.59
105-min	11.04	10.78	10.20	**10.12**	11.05
120-min	11.54	11.33	10.55	**10.44**	12.07
Overall	9.97	9.79	9.33	**9.26**	10.01

Table 5.6: Comparison between different length of input on the Highways England testing set.

Prediction	CNN-LSTM-Attn[1]	CNN-LSTM-Attn[2]
15-min	7.58	**7.54**
30-min	8.40	**8.35**
45-min	8.98	**8.88**
60-min	9.39	**9.25**
75-min	9.71	**9.60**
90-min	9.97	**9.89**
105-min	10.23	**10.12**
120-min	10.54	**10.44**
Overall	9.35	**9.26**

[1] Previous one-day observations as input (96 data points).
[2] Previous one-week observations as input (672 data points).

Figure 5.2 shows the learning curves of DNNs. Though CNN-LSTM-Attn with one-week observations has higher MAPE on training set in the very early stage, the error rate drops dramatically later. Because TS-LSTM uses pre-trained CNN-LSTM-Attn as part of its encoder, the learning curves of TS-LSTM drops quicker.

Figure 5.2: Learning curves of training on the Highways England dataset.

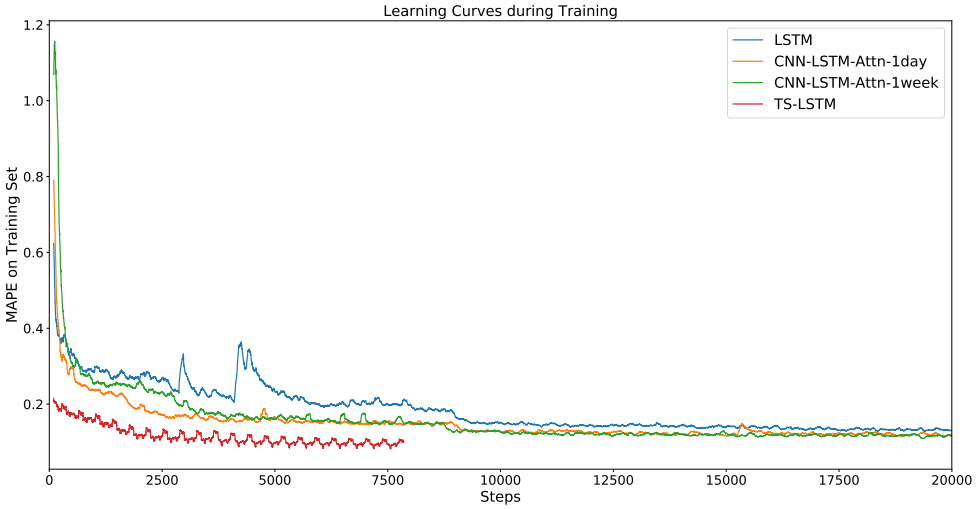


Figure 5.3 shows MAPE of CNN-LSTM-Attn at each node. The red line indicates the average overall MAPE 9.26%. As illustrated, CNN-LSTM-Attn achieves overall good prediction accuracy at most nodes. However, the prediction accuracy on 'A' roads including Node 6156-1, 6157-1, 6178-1, 6179-1, 7044-1 and 7045-1 is not as good as that on motorways. Because the average traffic volume on 'A' roads less than one third of that on motorways (Table 3.4), prediction on 'A' roads is more challenging.

Figure 5.3: MAPE at each node (CNN-LSTM-Attn) on the Highways England dataset.

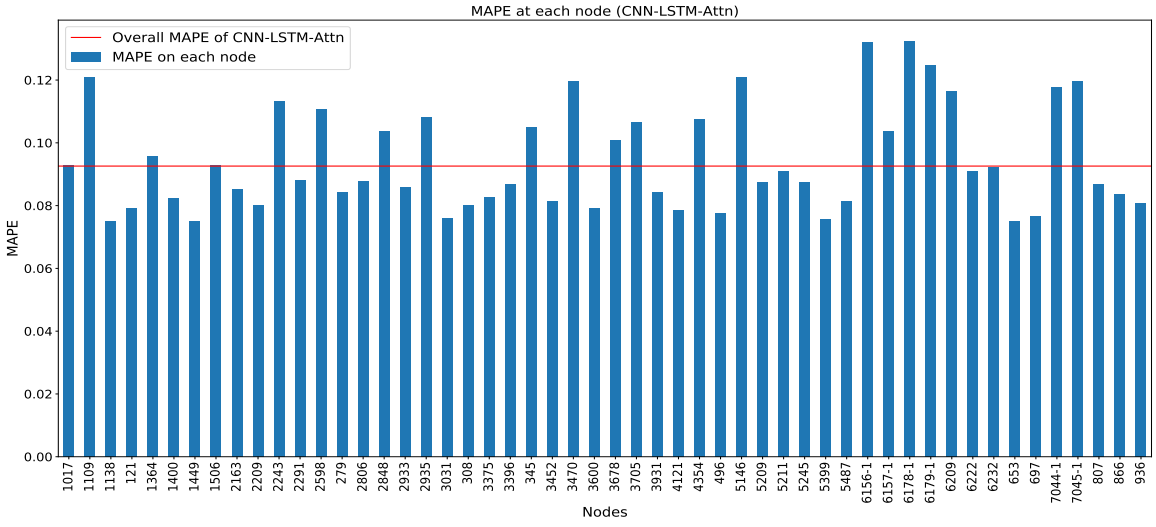
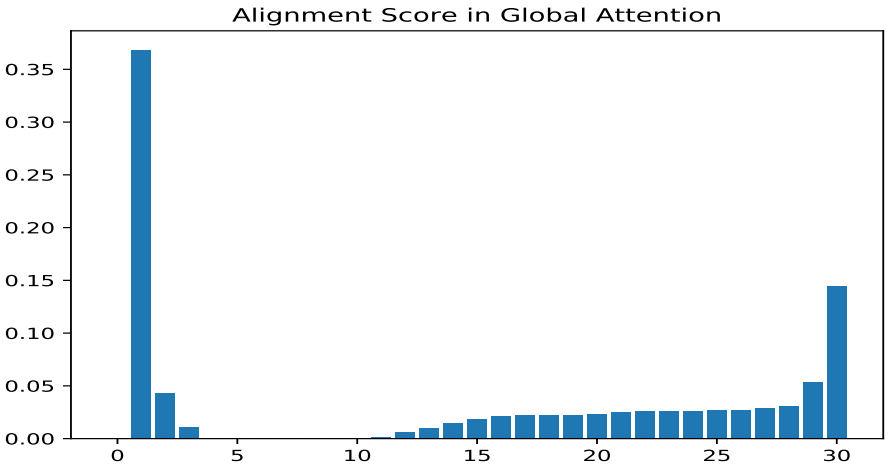


Figure 5.4 visualises the average alignment score α_t of LSTM-Attn using one-day observations. The attentional hidden state is highly related with the leftmost and rightmost source hidden states but has trivial or even zero alignment score in other points. The distribution of the alignment score can be easily explained by the daily seasonality and smoothness of traffic series, because the leftmost source hidden state

corresponds with the traffic data one day before and the rightmost source hidden state incorporates the latest data point. Therefore, the global attention module is able to infer the temporal seasonality without supervision.

Figure 5.4: The alignment score of the global attention (LSTM-Attn using one-day observations).



Task 2: On the KDDCUP 17 dataset, we compare the CNN-LSTM-Attn with one-week observations, which achieves lowest MAPE in the Task 1, and the model that got first place in the competition, which is built upon FC network and decision tree. Table 5.7 shows that CNN-LSTM-Attn achieves lower overall MAPE 10.48 and performs better on the testing set than the model that got first place.

Table 5.7: MAPE (%) on the KDDCUP 2017 testing set.

	CNN-LSTM-Attn	The-Model-Got-1st-Place
MAPE	**10.48**	12.03

Figure 5.5: Prediction at Node 4121 by CNN-LSTM-Attn (15-minute to 1-hour ahead).

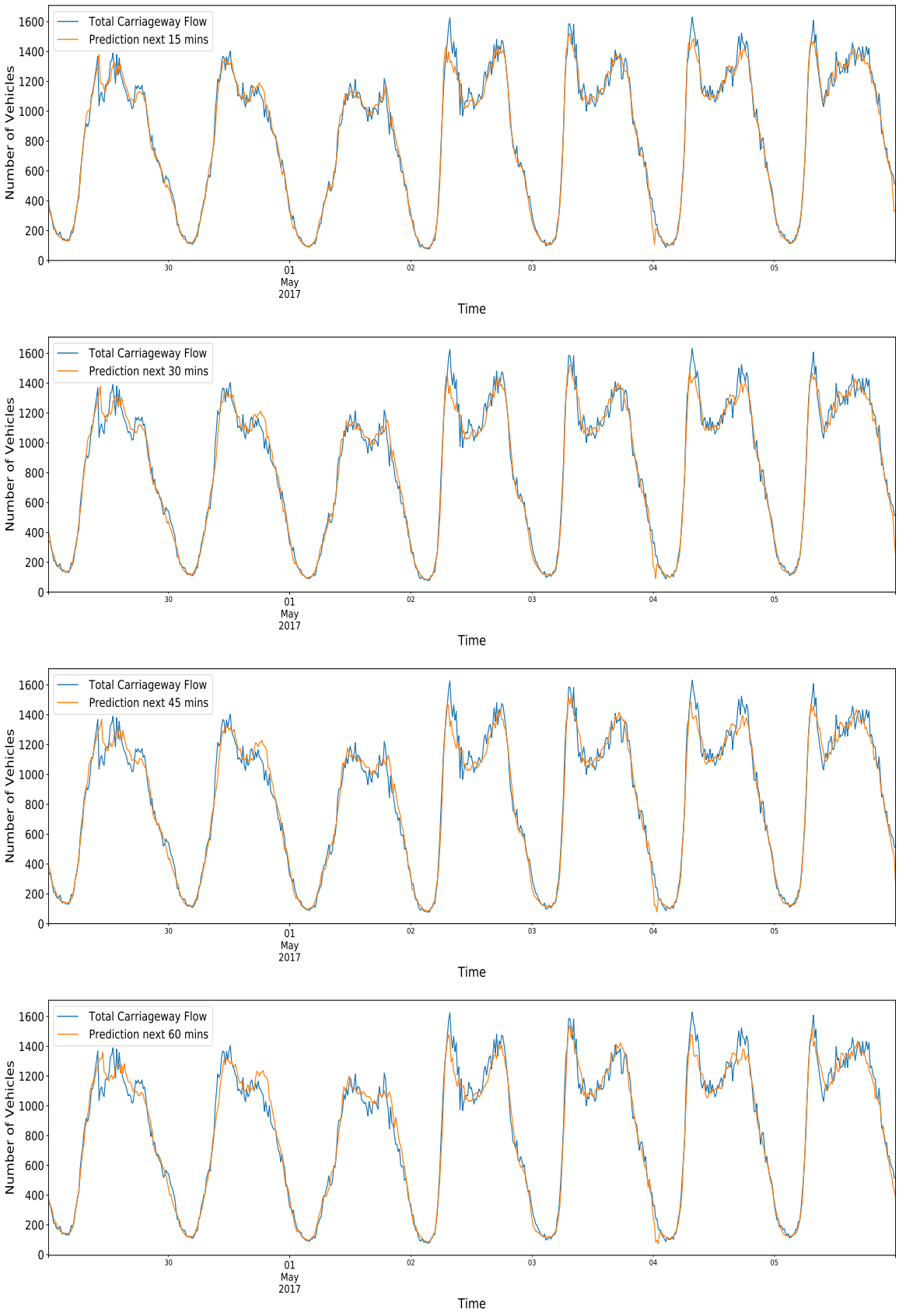
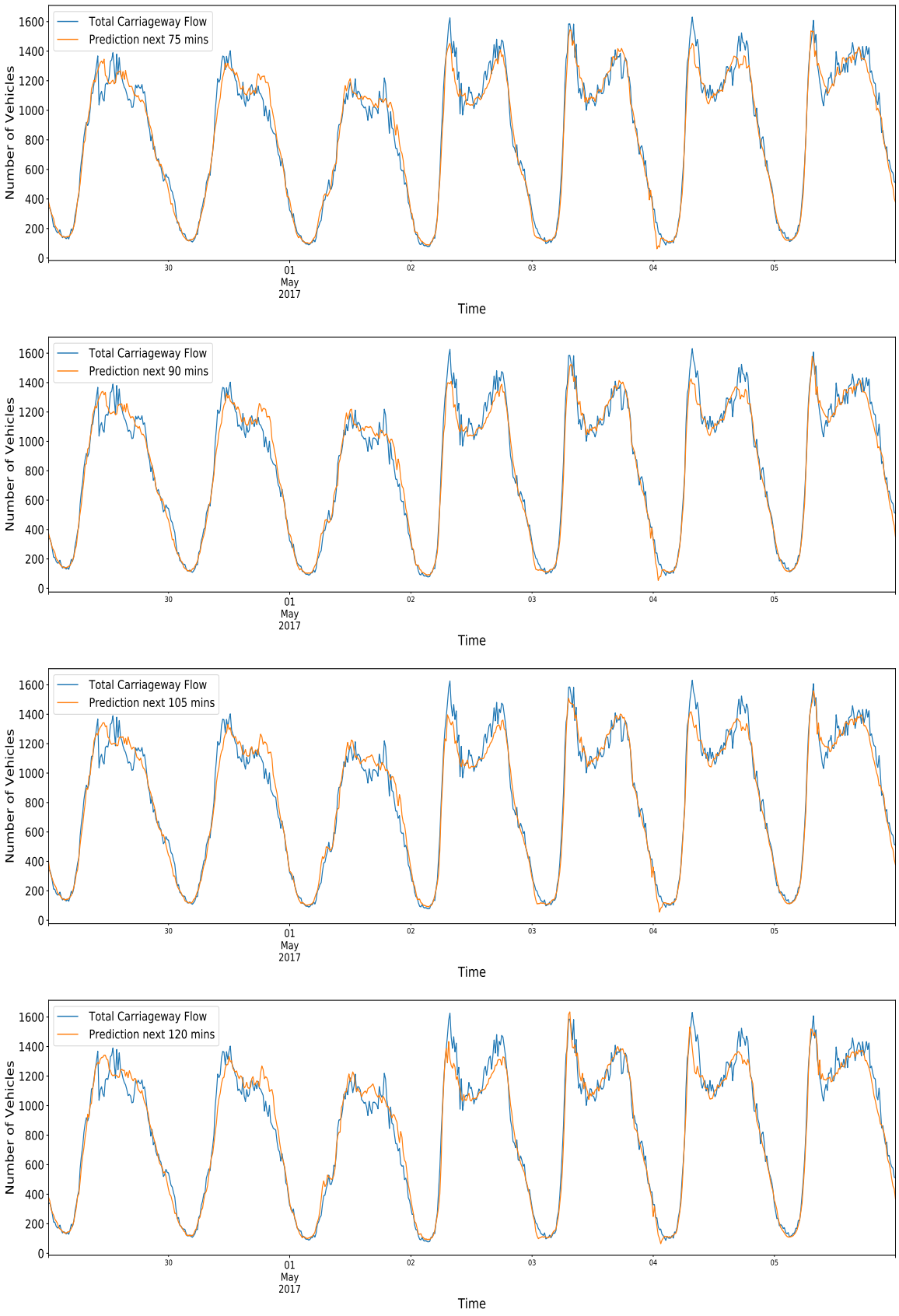


Figure 5.6: Prediction at Node 4121 by CNN-LSTM-Attn (75-minute to 2-hour ahead).

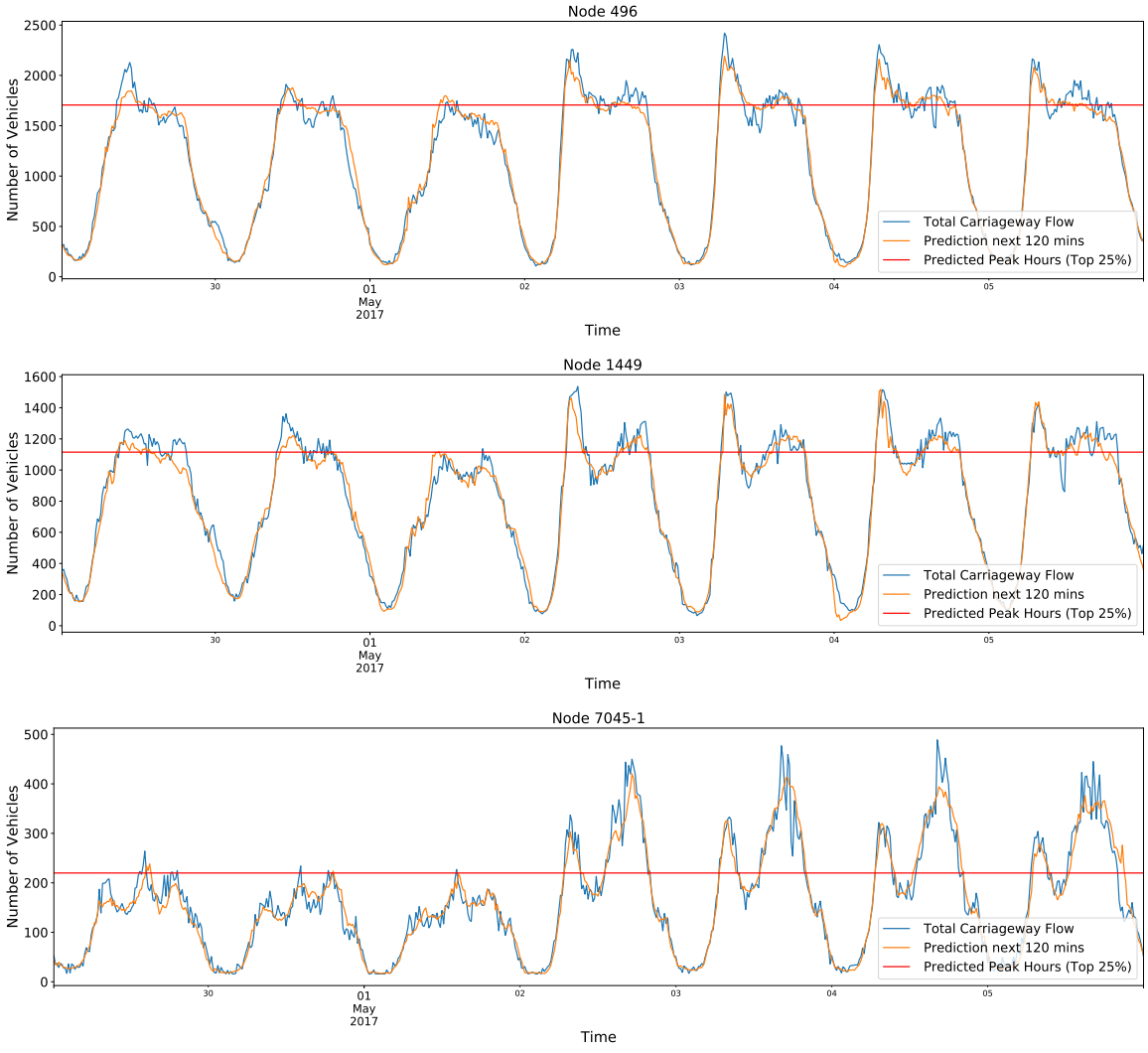


5.4 Analysis and Discovery

5.4.1 Peak Volume Alert

With help from our prediction model CNN-LSTM-Attn, the regulators of motorways can forecast the start of peak hours more precisely and how long it may last 2 hours in advance. The model achieves outstanding accuracy not only on workdays and also weekends and the bank holiday (1st May 2017). Figure 5.7 highlights the borderline of peak hours by the red lines and demonstrates our model can potentially be applied for early alert of peak traffic volume.

Figure 5.7: High volume alert based on 2-hour prediction by CNN-LSTM-Attn.



5.4.2 Abnormal Detection

The traffic can be blocked or delayed by unexpected accidents or events, and such abnormal conditions are also reflected by abnormal traffic volume during specific hours. Figure 5.8, 5.9 and 5.10 demonstrate significant outliers which lead to high error rate of our prediction model and slightly affect forecasting in the following hours. Each outlier corresponds with a piece of news, which includes causes of unusual conditions on M4 and M25 and is screen captured from Twitter [3].

Figure 5.8: Abnormal traffic condition on M25 on 16th Jun 2017.

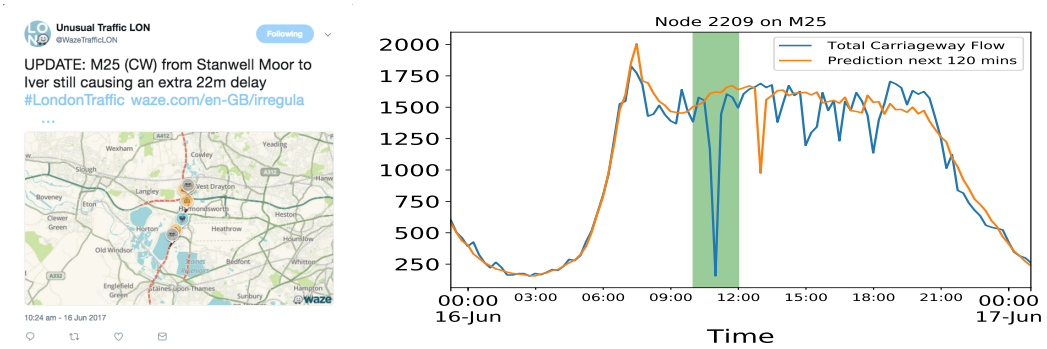


Figure 5.9: Abnormal traffic condition on M25 on 21st May 2017.

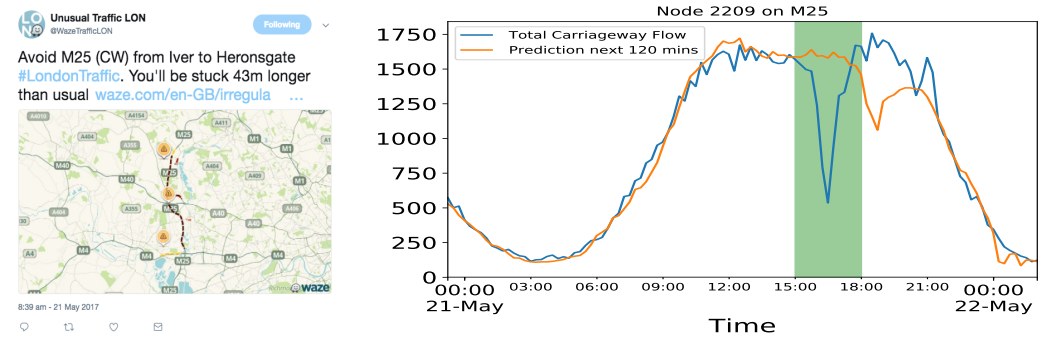
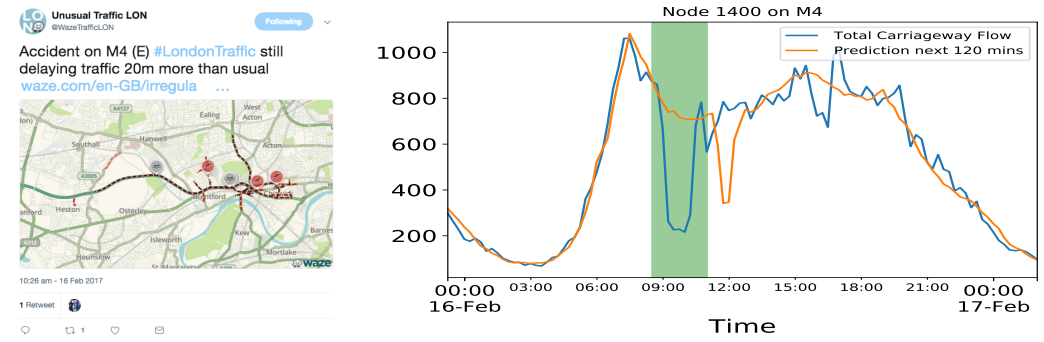


Figure 5.10: Abnormal traffic condition on M4 on 16th Feb 2017.



[3] Twitter @WazeTrafficLON: `https://twitter.com/wazetrafficlon`, accessed: 2017-09-04.

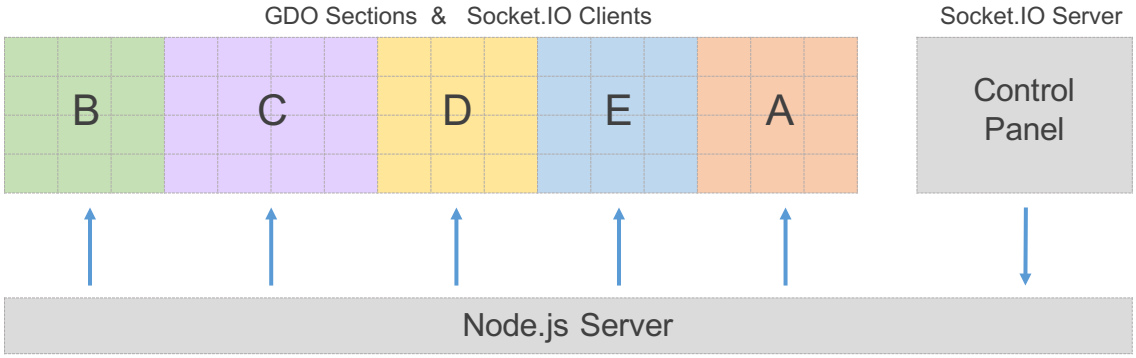
5.5 Visualisation

GDO provides a straightforward, efficient and attractive visualisation experience to demonstrate results of discovery and evaluation of the forecasting model. In this project, we visualise analytic and experimental results of the Highways England dataset.

The visualisation system is built upon the Section Mode of GDO framework. In Section Mode, the screen wall is split into five sections and each section is an individual web browser. Besides, there is a separated control panel, which is also a web browser.

The communication between six browsers is supported by Node.js and Socket.IO as shown in Figure 5.11. The commands will be sent out from the control panel and firstly received by the server. Then the server will distribute the command to related sections and the browsers that receive the command will react accordingly. The control panel is the only way for users to interactive with the GDO, so there is no need to reverse the message transmission from the sections back to the control panel.

Figure 5.11: Visualisation system structure. There are six browser-based screen, including five sections on the GDO and one control panel. The Node.js server distributes messages/commands coming from the control panel to the GDO and the five sections will react accordingly. The communication channel is built by Socket.IO.



The visualisation on each screen is summarised as below. You can find the panoramic view of visualisation on GDO in Figure 5.18.

1. Section B: Temporal analysis. As shown in Figure 5.12, the first row shows the decreasing of traffic volume on special days, including weekends, Christmas Day and New Year's Day, and the nine figures below compare different patterns, including start and end of peak hours, between workdays and special days.
2. Section C: Spatial correlation. Based on the coefficient correlation matrix com-

puted by the VAR, an animated heatmap built by Google Map API [4] highlights spatial correlation between a given node (labelled by a flag) and others (Figure 5.13). The region with reddish colour and relatively large radius indicates stronger correlation between those nodes and the given node. The animation is controlled by the control panel and the given node varies in each frame.

3. Section D: Animated heatmap of traffic volume (ground truth) at each node in the transportation network (Figure 5.14). The reddish region indicates heavy traffic volume and the greenish nodes have fewer vehicles. The animation is controlled by the control panel.
4. Section E: Animated heatmap of traffic volume (2-hour prediction) at each node in the transportation network (Figure 5.15). Similarly, the reddish region indicates large number of vehicles and the greenish nodes are relatively quieter. The animation is also controlled by the control panel and the simultaneousness between Section D and E can help us evaluate accuracy of our forecasting model intuitively.
5. Section A: Prediction accuracy ((Figure 5.16)). The line charts compares ground truth and our prediction. The numbers on the right are the MAPE of our forecasting model (CNN-LSTM-Attn).
6. Control panel: It only has a start button and stop button to control the animations in Section C, D, E ((Figure 5.17)).

Figure 5.12: Visualisation on Section B of GDO: temporal analysis. Comparison of traffic volume and patterns between workdays and special days.

[4] Google Map API: `https://developers.google.com/maps/`, accessed 2017-08-31.

Figure 5.13: Visualisation on Section C of GDO: spatial correlation. An animated heatmap based on coefficient matrix computed by the VAR.

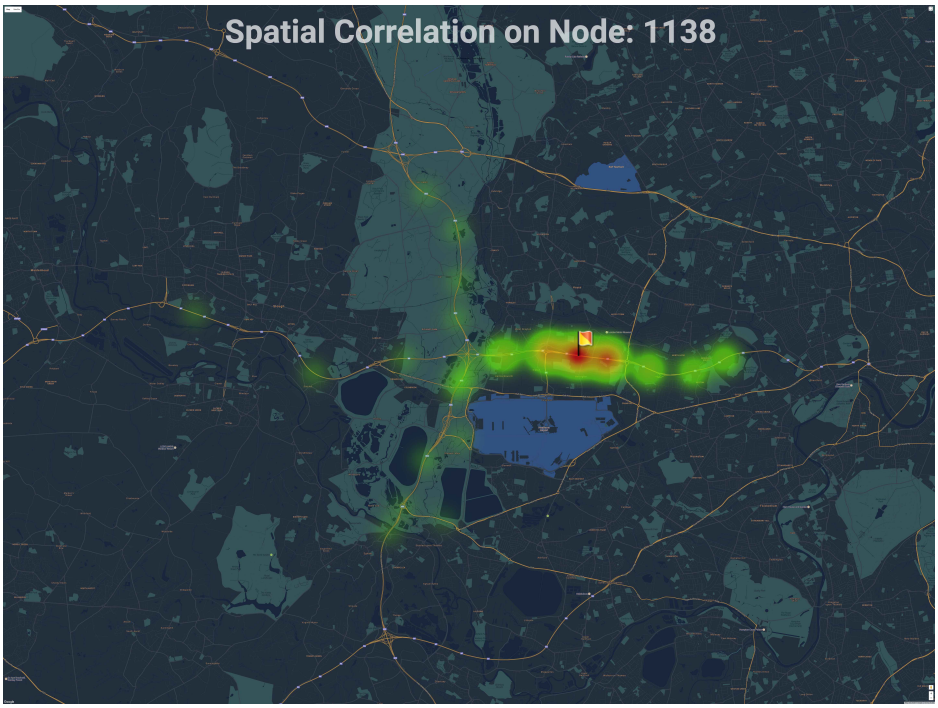


Figure 5.14: Visualisation on Section D of GDO: animated heatmap of traffic volume in the network (ground truth).

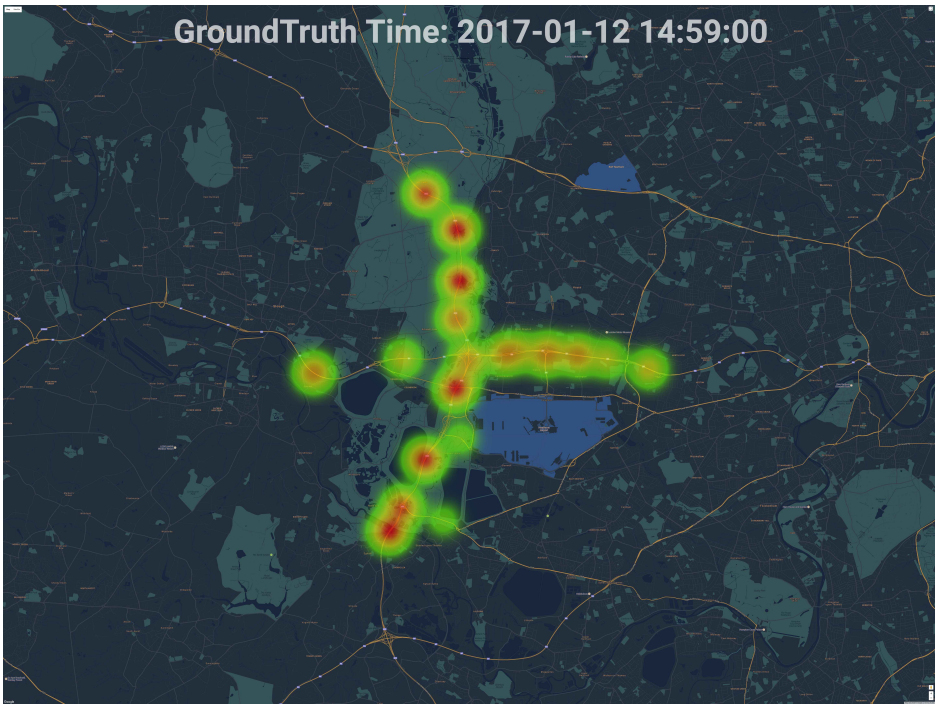


Figure 5.15: Visualisation on Section E of GDO: animated heatmap of traffic volume in the network (2-hour prediction).

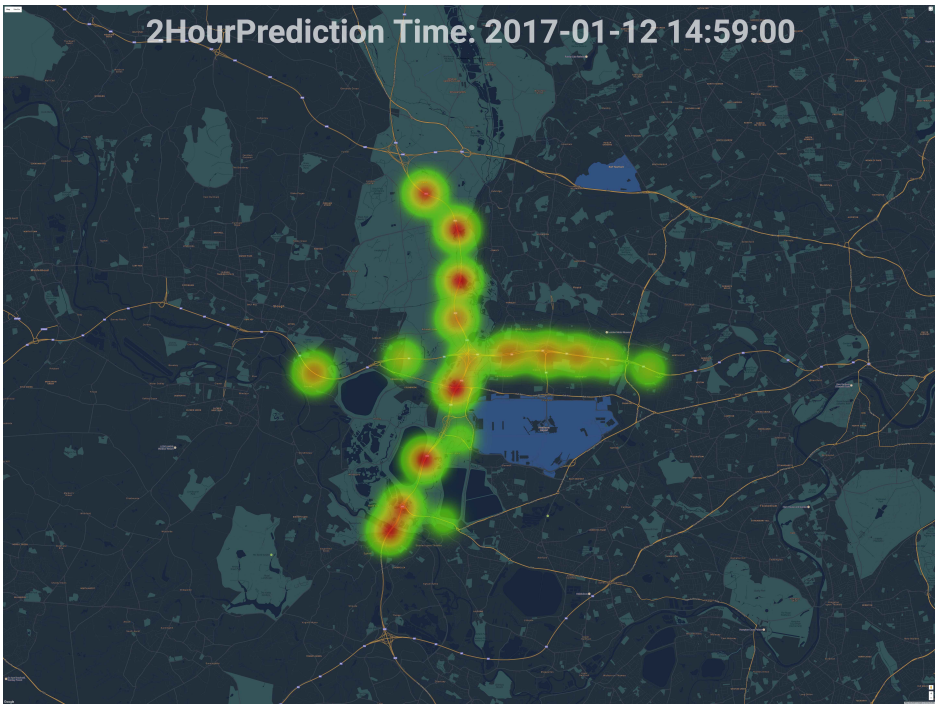


Figure 5.16: Visualisation on Section A of GDO: prediction accuracy of the CNN-LSTM-Attn model.

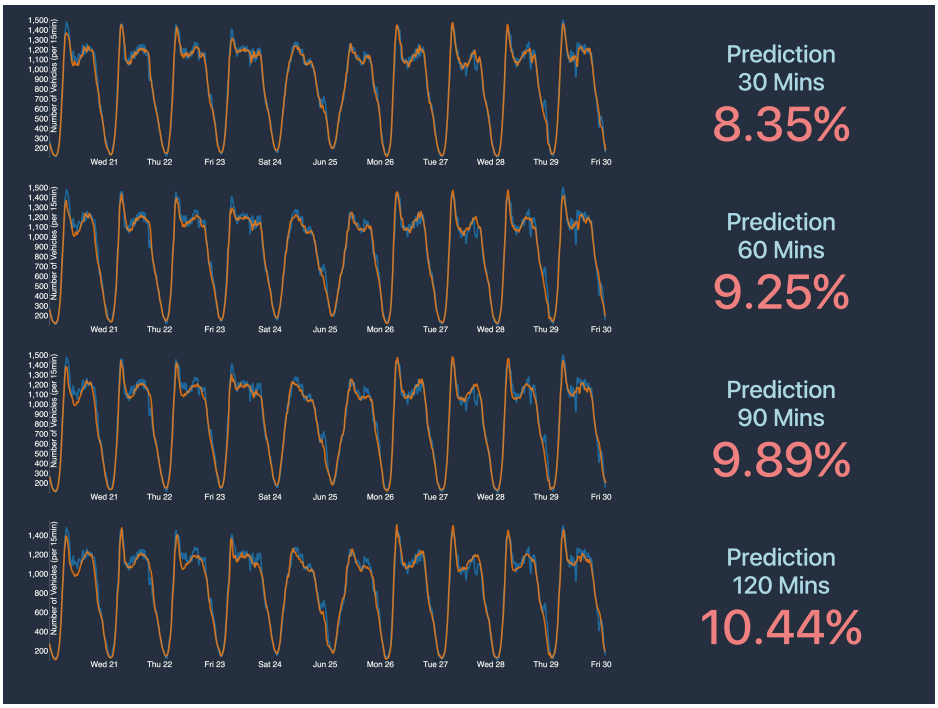


Figure 5.17: Control panel. Controlling the start/stop of animations in Section C, D, E.

Figure 5.18: The panoramic view of visualisation on GDO.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this project, we propose two deep learning models, CNN-LSTM-Attn and TS-LSTM, to model temporal and spatial features of traffic volume for short-term prediction (2-hour ahead). Our research includes two traffic datasets, the Highways England dataset and the KDDCUP17 dataset. We firstly analyse temporal pattern and spatial correlation on the Highways England dataset and statistics lead to conclusions:

1. Traffic volume on motorways triple that on 'A' roads. The peak hours on workdays start 7-8am and end 19-20pm on average. Abnormal traffic volume can be detected but corresponding reasons are unavailable in our datasets.
2. Traffic volume has strong and continuous daily seasonality, and clear first-order autoregressive pattern, both of which are fundamental and essential for short-term traffic volume prediction.
3. Special days including weekends, bank holidays, Christmas holidays and New Year's Day have significantly different traffic patterns than workdays. The differences include lower traffic volume, later start of peak hours and various end of peak hours.
4. Evidences indicate that traffic on the same direction of the same motorways is highly correlated and has stronger influence on following nodes around crossroads. However, information from opposite direction is very limited. Severe block or delay can influence traffic in corresponding local region, especially following nodes on the same direction.

In this project, SARIMA, VAR, Gaussian Process and Support Vector Regression are used as baselines, and the limitations of the baselines are also obvious.

1. SARIMA integrates repeating daily traffic pattern effectively but is not sensitive to special or recent conditions. Besides, multiple SARIMAs are needed for multiple locations, because one SAIRMA cannot fit difference sequences.

2. VAR is able to incorporate correlation among nodes, but can easily affected by noises in previous observations.

3. Gaussian Process can approximate interpolation of given observations but the variance of prediction in further future may go outrage.

4. Support Vector Regression with RBF kernel may result in overfitting if the sequence is not smooth enough and linear kernel fails to achieve good accuracy for longer-ahead forecasting.

We proposed CNN-LSTM-Attn to model temporal dependencies, and TS-LSTM which is built upon CNN-LSTM-Attn with spatial relation module to extract spatial correlation. Both CNN-LSTM-Attn and TS-LSTM outperform the baselines on the 2-hour traffic volume prediction on the Highways England dataset with overall MAPE 9.26% and 10.01% respectively. Not only on workdays, prediction on weekends and bank holidays are also highly accurate. Both the CNN embedding and global attention mechanism can effectively encode temporal dependencies and improve prediction accuracy. In addition, using previous one-week rather one-day observations as input also leads to higher accuracy. Besides, the CNN-LSTM-Attn achieves overall outstanding prediction accuracy at most nodes but performance on 'A' roads is not as good as that on motorways. Moreover, TS-LSTM fails to achieve comparable performance with CNN-LSTM-Attn due to lack of significant long term spatial correlation and complexity of the model, which we believe can be solved if more time is allowed for this project.

On the KDDCUP17 dataset, the CNN-LSTM-Attn achieves better accuracy with lower MAPE 10.48% than that 12.03% of the model which got first place in the competition.

With accurate 2-hour traffic forecasting, which is longer than previous literature (30-minute), we believe that our models can potentially applied for peak volume alert and abnormal traffic detection. Unusual traffic news from Twitter have shown that severe delay can be reflected by large error rate of the prediction.

The DNN models are implemented based on TensorLayer, and our design pattern takes advantages of functional programming and object-oriented programming, which provides agile project development with simple structure, friendly interfaces, minimum redundancy and sufficient flexibility. Furthermore, visualisation and animation on the GDO Section Mode help us demonstrate results of discovery and evaluation of the forecasting model.

6.2 Future Work

If more time is allowed for this project, there are some potentially interesting works that can be explored.

1. Improvement of the CNN-LSTM-Attn especially at the nodes that prediction is not accurate enough.
2. Simplifying the TS-LSTM with prior knowledge on the transportation network. Scenario in a larger region and investigation on spatial correlation during holidays among popular locations for tourism would be needed and attractive. Further research on how abnormal volume (extremely high or low) may affect neighbouring locations quantitatively can be useful for motorways regulators.
3. Visualisation of the features extracted by CNN, RNN and the attention mechanism can help us understand how deep neural networks work on time series analysis.
4. The Highways England dataset includes 52 monitoring nodes around Heathrow Airport and Uxbridge. More nodes in a larger region can be used to train a more general and powerful DNN model.
5. So far, only traffic data on motorways and 'A' roads in suburban area has been used. If tick-level traffic data in urban area is available, the same idea can be applied to use DNNs modelling temporal and spatial features. As traffic can be affected by traffic lights in urban area, further research on optimisation of length of the traffic lights based on forecasting by deep learning can be helpful to prevent severe traffic congestion.
6. Accident data is not available in current datasets but can be collected for analysis on how unexpected accidents may affect local traffic and how to make detection of such abnormal conditions more efficiently with leverage from deep learning.
7. Future research on relation between traffic volume and air pollution especially how traffic volume prediction may help to improve air quality monitoring and forecasting can be useful for both residents and governors.

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org. pages 46

Bishop, C. M. (2006). Pattern recognition. *Machine Learning*, 128:1–58. pages 10

Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons. pages 2, 4, 7

Chen, Y., Zhao, Y., and Yan, P. (2016). Daily etc traffic flow time series prediction based on k-nn and bp neural network. In *International Conference of Young Computer Scientists, Engineers and Educators*, pages 135–146. Springer. pages 2

Dong, H., Supratak, A., Mai, L., Liu, F., Oehmichen, A., Yu, S., and Guo, Y. (2017a). TensorLayer: A Versatile Library for Efficient Deep Learning Development. *ACM Multimedia*. pages 46

Dong, H., Zhang, J., McIlwraith, D., and Guo, Y. (2017b). I2t2i: Learning text to image synthesis with textual data augmentation. *arXiv preprint arXiv:1703.06676*. pages 15

Duan, Y., Lv, Y., and Wang, F.-Y. (2016). Travel time prediction with lstm neural network. In *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*, pages 1053–1058. IEEE. pages 2, 11, 15

Dumoulin, V. and Visin, F. (2016). A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*. pages 14

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT Press. pages 11, 12, 15, 51

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680. pages 11

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. pages 2

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780. pages 15

Hubel, D. H. and Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3):574–591. pages 12

Hubel, D. H. and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106–154. pages 12

Hubel, D. H. and Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243. pages 12

Kamarianakis, Y. and Prastacos, P. (2003). Forecasting traffic flow conditions in an urban network: Comparison of multivariate and univariate approaches. *Transportation Research Record: Journal of the Transportation Research Board*, (1857):74–84. pages 2, 8

Kamarianakis, Y. and Prastacos, P. (2005). Space–time modeling of traffic flow. *Computers & Geosciences*, 31(2):119–133. pages 2, 8

Karpathy, A., Johnson, J., and Fei-Fei, L. (2015). Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*. pages 43

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. pages 53

Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images. pages 11

LeCun, Y. et al. (1989). Generalization and network design strategies. *Connectionism in perspective*, pages 143–155. pages 12

Li, Y., Yu, R., Shahabi, C., and Liu, Y. (2017). Graph convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*. pages 2, 15, 45

Lippi, M., Bertini, M., and Frasconi, P. (2013). Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. *IEEE Transactions on Intelligent Transportation Systems*, 14(2):871–882. pages 2, 4, 5, 10

Luong, M.-T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*. pages 43

Lütkepohl, H. (2005). *New introduction to multiple time series analysis*. Springer Science & Business Media. pages 8

Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., and Wang, Y. (2017). Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*, 17(4):818. pages 11, 12, 13

Moorthy, C. and Ratcliffe, B. (1988). Short term traffic forecasting using time series methods. *Transportation planning and technology*, 12(1):45–56. pages 6

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. pages 2

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318. pages 44

Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*. pages 11

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. (2016). Generative adversarial text to image synthesis. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 3. pages 15

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1988). Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1. pages 11

Santoro, A., Raposo, D., Barrett, D. G., Malinowski, M., Pascanu, R., Battaglia, P., and Lillicrap, T. (2017). A simple neural network module for relational reasoning. *arXiv preprint arXiv:1706.01427*. pages 45

Shahsavari, B. and Abbeel, P. (2015). Short-term traffic forecasting: Modeling and learning spatio-temporal relations in transportation networks using graph neural networks. pages 2

Srivastava, N., Hinton, G. E., 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. pages 44

Sun, S. and Chen, Q. (2008). Kernel regression with a mahalanobis metric for short-term traffic flow forecasting. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 9–16. Springer. pages 5

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. pages 2

Sutskever, I., Martens, J., and Hinton, G. E. (2011). Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1017–1024. pages 43

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112. pages 15

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *CoRR abs/1609.03499*. pages 11, 15

Wang, F., Tan, G., and Fang, Y. (2009). Multiscale wavelet support vector regression for traffic flow prediction. In *Intelligent Information Technology Application, 2009. IITA 2009. Third International Symposium on*, volume 3, pages 319–322. IEEE. pages 5

Williams, C. K. and Rasmussen, C. E. (2006). Gaussian processes for machine learning. *the MIT Press*, 2(3):4. pages 10

Wu, C.-H., Ho, J.-M., and Lee, D.-T. (2004). Travel-time prediction with support vector regression. *IEEE transactions on intelligent transportation systems*, 5(4):276–281. pages 5, 10

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. pages 2

Xie, Y., Zhao, K., Sun, Y., and Chen, D. (2010). Gaussian processes for short-term traffic volume forecasting. *Transportation Research Record: Journal of the Transportation Research Board*, (2165):69–78. pages 10

Xue, J. and Shi, Z. (2008). Short-time traffic flow prediction based on chaos time series theory. *Journal of Transportation Systems Engineering and Information Technology*, 8(5):68–72. pages 2

Yang, Z., Bing, Q., Lin, C., Yang, N., and Mei, D. (2014). Research on short-term traffic flow prediction method based on similarity search of time series. *Mathematical Problems in Engineering*, 2014. pages 2

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., and Lipson, H. (2015). Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*. pages 13

Yu, L., Zhang, W., Wang, J., and Yu, Y. (2017). Seqgan: sequence generative adversarial nets with policy gradient. In *Thirty-First AAAI Conference on Artificial Intelligence*. pages 11, 43

Zhang, J., Zheng, Y., and Qi, D. (2017). Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Thirty-First AAAI Conference on Artificial Intelligence*. pages 11

Zhao, J. and Sun, S. (2016). High-order gaussian process dynamical models for traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, 17(7):2014–2019. pages 10

Chapter 7

Appendices

7.1 Dataset

Table 7.1: Detailed information and locations of the 52 nodes selected in the Highways England dataset. Distribution of the nodes on map can be found in Figure 3.1.

Node	Site Name and Descriptions	GPS (Lat, Lng)
1017	MIDAS site at M4/2259A priority 1 on link 199000301, Westbound	51.4948, -0.4695
1109	MIDAS site at M4/2178B priority 1 on link 125039801, Eastbound	51.4932, -0.3588
1138	MIDAS site at M4/2240A priority 1 on link 199127301, Westbound	51.4941, -0.4365
121	MIDAS site at M4/2302B priority 3 on link 199003201, Eastbound	51.4933, -0.5305
1364	MIDAS site at M4/2388A priority 1 on link 105008401, Westbound	51.5080, -0.6393
1400	MIDAS site at M4/2248A priority 1 on link 199000201, Westbound	51.4955, -0.4518
1449	MIDAS site at M4/2227A priority 1 on link 199127301, Westbound	51.4930, -0.4207
1506	MIDAS site at M25/4888B priority 1 on link 199132201, Anti-clockwise	51.4438, -0.5297

2163	MIDAS site at M25/5010B priority 1 on link 125023801, Anti-clockwise	51.5479, -0.5177
2209	MIDAS site at M25/5010A priority 8 on link 108032801, Clockwise	51.5475, -0.5172
2243	MIDAS site at M4/2210B priority 1 on link 123012101, Eastbound	51.4899, -0.3989
2291	MIDAS site at M4/2257B priority 1 on link 199000101, Eastbound	51.4956, -0.4652
2598	MIDAS site at M4/2210A priority 1 on link 199128802, Westbound	51.4897, -0.3989
279	MIDAS site at M25/4909A priority 1 on link 200045638, Clockwise	51.4598, -0.5178
2806	MIDAS site at M4/2240B priority 1 on link 199128901, Eastbound	51.4943, -0.4364
2848	MIDAS site at M25/4949A priority 1 on link 199004101, Clockwise	51.4927, -0.4947
2933	MIDAS site at M25/4936A priority 1 on link 200045641, Clockwise	51.4826, -0.5019
2935	MIDAS site at M4/2262L priority 1 on link 199000101, Eastbound	51.4952, -0.4710
3031	MIDAS site at M25/4938B priority 1 on link 200045646, Anti-clockwise	51.4839, -0.5003
308	MIDAS site at M25/4976A priority 5 on link 108032801, Clockwise	51.5188, -0.4992
3375	MIDAS site at M25/4879B priority 1 on link 103049601, Anti-clockwise	51.4360, -0.5361
3396	MIDAS site at M4/2262B priority 1 on link 199000101, Eastbound	51.4952, -0.4710
345	MIDAS site at M4/2193A priority 3 on link 123012302, Westbound	51.4890, -0.3753
3452	MIDAS site at M4/2264A priority 1 on link 199000301, Westbound	51.4944, -0.4766
3470	MIDAS site at M4/2227B priority 1 on link 199128901, Eastbound	51.4932, -0.4220

3600	MIDAS site at M25/4993A priority 2 on link 108032801, Clockwise	51.5357, -0.5005
3678	MIDAS site at M4/2178A priority 1 on link 123012302, Westbound	51.4931, -0.3586
3705	MIDAS site at M4/2264M priority 1 on link 199000301, Westbound	51.4943, -0.4766
3931	MIDAS site at M25/4963B priority 1 on link 199003403, Anti-clockwise	51.5063, -0.5003
4121	MIDAS site at M4/2301A priority 4 on link 105009001, Westbound	51.4932, -0.5291
4354	MIDAS site at M25/4887A priority 1 on link 199131002, Clockwise	51.4444, -0.5295
496	MIDAS site at M25/4943B priority 1 on link 200045646, Anti-clockwise	51.4878, -0.4965
5146	MIDAS site at M4/2193B priority 3 on link 125039801, Eastbound	51.4892, -0.3755
5209	MIDAS site at M4/2388B priority 1 on link 105007801, Eastbound	51.5082, -0.6397
5211	MIDAS site at M25/4949B priority 1 on link 199000502, Anti-clockwise	51.4926, -0.4944
5245	MIDAS site at M25/4879A priority 1 on link 199132301, Clockwise	51.4361, -0.5366
5399	MIDAS site at M25/4909B priority 1 on link 200045651, Anti-clockwise	51.4597, -0.5174
5487	MIDAS site at M4/2336A priority 1 on link 105008001, Westbound	51.4883, -0.5764
6156-1	TMU Site 6156/1 on link M4 spur (Heathrow) northbound between J4A and M4, Northbound	51.4927, -0.4538
6157-1	TMU Site 6157/1 on link M4 spur (Heathrow) southbound between M4 and J4A, Westbound	51.4932, -0.4535
6178-1	TMU Site 6178/1 on link A30 eastbound between M25 and A308, Eastbound	51.4398, -0.5079

6179-1	TMU Site 6179/1 on link A30 westbound between A308 and M25, Westbound	51.4396, -0.5073
6209	MIDAS site at M4/2156B priority 1 on link 199128403, Eastbound	51.4928, -0.3252
6222	MIDAS site at M4/2156A priority 1 on link 125039601, Westbound	51.4927, -0.3253
6232	MIDAS site at M4/2248B priority 1 on link 199128101, Eastbound	51.4957, -0.4518
653	MIDAS site at M25/4976B priority 5 on link 199003403, Anti-clockwise	51.5188, -0.4989
697	MIDAS site at M25/4993B priority 1 on link 199003403, Anti-clockwise	51.5358, -0.5003
7044-1	TMU Site 7044/1 on A3113 eastbound between M25 and A3044, Eastbound	51.4669, -0.4986
7045-1	TMU Site 7045/1 on A3113 westbound between A3044 and M25, Westbound	51.4668, -0.4987
807	MIDAS site at M25/4963A priority 1 on link 108032801, Clockwise	51.5062, -0.5006
866	MIDAS site at M4/2336B priority 1 on link 105008901, Eastbound	51.4886, -0.5765
936	MIDAS site at M25/4941A priority 1 on link 200045641, Clockwise	51.4864, -0.4981