

基于多模态特征融合的时空图卷积网络

杨安¹, 王伟⁺, 刘成²

云南大学 软件学院, 云南省昆明市 650504

+ 通信作者 E-mail: softwareanalytics@protonmail.com

摘 要: 交通流量数据有非常高的复杂性, 在空间维度上, 不同位置的交通状况相互影响程度不同。在时间维度上, 不同时间段的交通状况之间存在相关性, 不同情况下的相关性也不同。天气、事故等非时空因素对交通流量的影响也非常重要。针对非时空因素, 特别是天气因素对交通流量的影响研究成果相对比较少。为了解决这些问题, 基于图神经网络对历史交通流量、天气状况等多模态数据构建了 AFFGCN 交通流量预测模型: (1) 针对空间依赖, 基于图神经网络构建提取模型, 同时基于空间注意力计算每个交通节点与其邻居节点的相关系数, 计算结果对图卷积网络的邻接矩阵进行动态加权。(2) 使用 CNN 提取时间依赖, 基于时间注意力对原始数据进行重构。(3) 为了拟合天气因素与交通流量数据之间的耦合关系, 使用多头注意力实现天气特征与交通流量特征的动态耦合, 并在此基础上构建交通流量预测模型。实验表明, AFFGCN 对比当前较为先进的基线方法提高了 10%~45% 的准确率, 与未考虑天气因素的 AFFGCN 对比提高了 25%。

关键词: 交通流量预测; 天气因素; 图卷积网络; 注意力机制; 循环神经网络

文献标志码: A 中图分类号: TP***

Spatio-temporal graph convolution network based on multimodal feature fusion

YANG An¹, WANG Wei¹⁺, Liu Cheng²

College of Software, Yunnan University, City Kunming 650504, China

Abstract: Traffic flow data is very complex. In spatial dimension, traffic conditions at different locations have different degrees of interaction. On the time dimension, there is correlation between traffic conditions in different time periods, and the correlation is different in different situations. Non temporal and spatial factors such as weather and accidents are also very important for traffic flow. The research results on the impact of non space-time factors, especially weather factors on traffic flow are relatively few. To solve these problems, the AFFGCN traffic flow prediction model is built based on the historical traffic flow, weather conditions and other multimodal data based on the graph neural network: (1) For spatial dependence, the extraction model is built based on the graph neural network. At the same time, the correlation coefficient between each traffic node and its neighbor node is calculated based on spatial attention. The calculation results dynamically weight the adjacency matrix of the graph convolution network. (2) CNN is used to extract time dependence, and the original data is reconstructed based on time attention. (3) In order to fit the coupling relationship between weather factors and traffic flow data, multiple attention is used to realize the dynamic coupling of weather features and traffic flow features, and a traffic flow prediction model is built on this basis. The experiment shows that the accuracy of AFFGCN is improved by 10%~45% compared with the current advanced baseline method, and 25% compared with the AFFGCN without considering weather factors.

Key words: Traffic Flow Forecasting; Weather Condition; Graph Convolution Network; Attention

Mechanism; Recurrent Nerual Network

交通流量预测旨在以历史时刻观测数据（例如车速、车辆密度等）为基础，生成对未来交通流量趋势的预测^[1]。该研究是智能交通系统（Intelligent traffic systems, ITS）的重要组成部分，对提高交通管理效率^[2]、城市规划合理性^[3]、出行体验^[4]等方面具有重要意义^[5]。早期研究聚焦于路口、道路等微观节点的短期预测^{[6][7]}。由于面向区域或城域等宏观节点的中长期交通流量预测对市政管理、企业决策等具有重要的应用价值，逐渐成为研究的热点。统计模型^{[8][9]}、仿真模型^[10]和机器学习模型^{[11][12]}是早期交通流量预测的常用模型。随着研究不断深入，基于深度学习预测模型逐渐暴露出一些问题：

（1）众所周知，复杂的天气对交通路网状况有显著的影响，特别是暴雨天和雪天，道路通勤效率与天气因素有明显的关联。然而将天气因素纳入交通流量预测建模成果比较有限^[13]。

（2）在计算模型方面为了将深度学习应用于交通数据，通常需要将地图抽象为网格（通常按照经纬度进行划分）。网格划分破坏了城市功能布局的连续性，无法真实反映节点间的时空依赖。同时深度模型通常假设两时刻数据间具有独立性，这显然与物理事实相矛盾^[14]。

本文针对以上两个问题基于图神经网络和多模态数据融合技术构建了交通流量预测模型，有效提高了预测模型的性能，模型总体简要结构如图 1 所示：

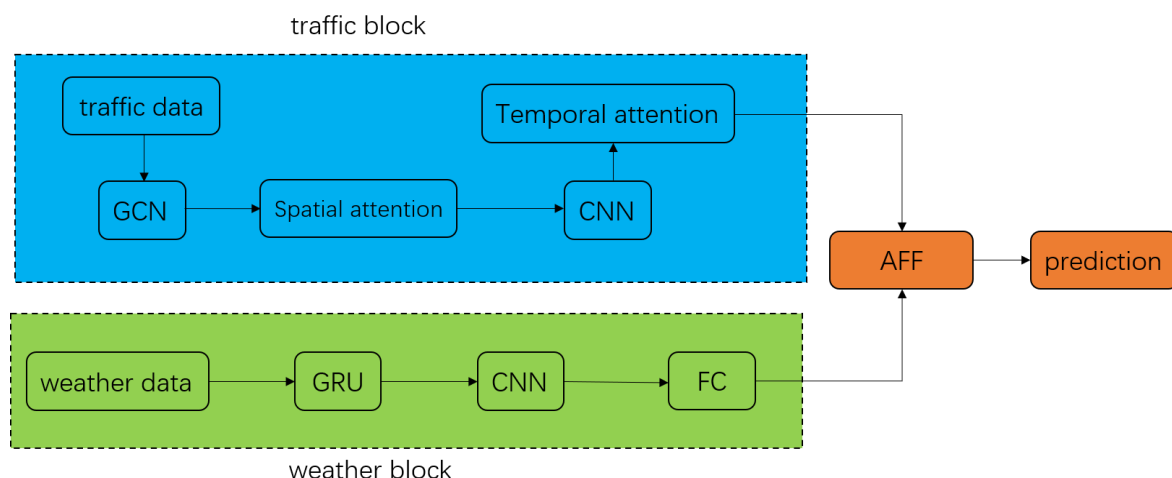


图 1 模型总体结构

Fig.1 Overall structure of the model

1 相关工作

1.1 基于图神经网络的交通流量预测研究

近年来深度学习应用方兴未艾，针对车辆轨迹、时间序列、栅格地图、视频监控等交通数据，研究人员使用 RNN^[15]、CNN^[16]、LSTM^[17]、GRU^[17]、Seq2Seq^[18]等深度学习算法建立了交通流量预测模型，提升了预测结果的准确性。深度学习在交通流量预测领域的成功应用有两个重要原因：（1）交通数据深度内嵌于时间、空间、非时空等多个维度。有效感知数据间的依赖对交通流量预测至关重要。深度学习算法具有强大的函数拟合与特征工程能力，能够有效感知交通数据中的依赖^[19]。（2）随着数据采集、传输和存储

技术的不断发展，交通数据的模态和数量也得到了极大丰富，为有效提高预测模型的性能提供了可能。

然而基于深度学习模型的交通流量预测模型逐渐暴露出一些问题：在计算模型方面为了将深度学习应用于交通数据，通常需要将地图抽象为网格（通常按照经纬度进行划分）。网格划分破坏了城市功能布局的连续性，无法真实反映节点间的时空依赖。同时深度模型通常假设两时刻数据间具有独立性，这显然与物理事实相矛盾。

近期研究结果表明使用图 $G=(V,E,A)$ 作为交通网络的抽象描述更具合理性^[20]。节点集合 V 表示路口，传感器，道路等交通设施， $|V|=N$ 表示

节点的数量, 边集合 E 表示节点之间的连通性, 邻接矩阵 $A=(a_{ij})_{N \times N} \in R^{N \times N}$ 刻画了图 G 的结构特征, a_{ij} 的取值可以是二进制, 也可以是实数。邻接矩阵从空间拓扑、城市功能、交通行为模式等角度反映了交通节点之间的关联性, 避免了由于网格化导致城市功能布局连续性被破坏的问题。

传统的深度学习无法使用图数据作为输入, 因此交通流量预测需要新的计算泛型。图神经网络 (Graph neural network, GNN) 以图数据作为输入, 不仅具有深度学习强大的函数拟合与特征工程能力, 同时也能利用交通网络的结构信息, 适合解决交通流量预测问题。

图卷积神经网络 (Graph convolution neural network, GCNN) 和图循环神经网络 (Recurrent graph neural network, RGNN) 是基于图神经网络的交通流量预测的两种主要方法^[20]。由于 GCNN 具有并行计算、梯度稳定和内存需求低的优点, 逐渐成为感知空间依赖的主要工具。根据图卷积定义方式的不同, 常用的 GCNN 有谱图网络 (Spectral graph network) 和空间图网络 (Spatial graph network) 两类。谱图网络基于邻接矩阵将图结构抽象为拉普拉斯矩阵 (Laplacian), 通过引入傅里叶变换把在图像、语音等网格数据上定义良好的卷积操作推广到图数据上^[21]。ChebNet 和 GCN 是交通流量预测领域使用最多的两种谱图网络。ChebNet 针对传统谱图网络计算复杂度高 $O(N^3)$ 的问题, 使用 Chebyshev 多项式对图卷积进行了 K 阶近似, 降低了计算复杂度。在 ChebNet 的基础上提出了 GCN, 将 Chebyshev 多项式的阶数 K 限制为 2, 每一个图卷积层仅对 1 跳邻居信息进行聚合, 进一步降低图卷积的计算复杂度^{[22][23]}。

空间图网络基于节点及其邻居节点的连接关系, 将卷积操作定义为邻居节点的选择算法。该方法突破了谱图方法对卷积核 (拉普拉斯矩阵) 对称性的限制, 可以分析不同方向对交通流量的影响 (例如车流量的上下游问题)。同时, 由于空间方法不需要特征分解 (Eigen decomposition), 计算复杂度进一步降低。扩散卷积 (Diffusion convolution) 是在交通流量预测领域使用较广的一种空间图网络。该网络将邻接矩阵

归一化为概率转移 (Probability transition) 矩阵, 以刻画节点间关联性的强弱。将图卷积定义为基于概率转移矩阵和重启概率 (Restart probability) 的复合函数^{[24][25]}。

1.2 交通预测研究的特征

交通预测作为智慧交通系统重要的组成部分, 已经有几十年的历史。交通预测研究的特征对象主要集中在速度、车流量、密度等三类, 但是对天气、节假日、事故等研究的较少。

关于对交通速度预测的研究, 如: L Ge, H Li, J Liu, A Zhou 等提出了一种由时空分量和外部分量组成的时域图卷积网络 (GTCN) 来解决交通速度预测问题。实验表明, 提出的 GTCN 模型具有较高的精度^[26]。James J.Q.Yu 等提出了一种基于数据驱动的深度学习的城市交通速度预测方法。具体而言, 该方法利用图卷积和注意机制捕获几何交通数据相关性, 并在生成对抗学习框架中利用编码器和解码器架构提取和扩展时间数据相关性。利用真实城市道路网络和相关数据进行了全面的案例研究, 以评估其性能, 可以观察到基线方法的持续改进^[27]。

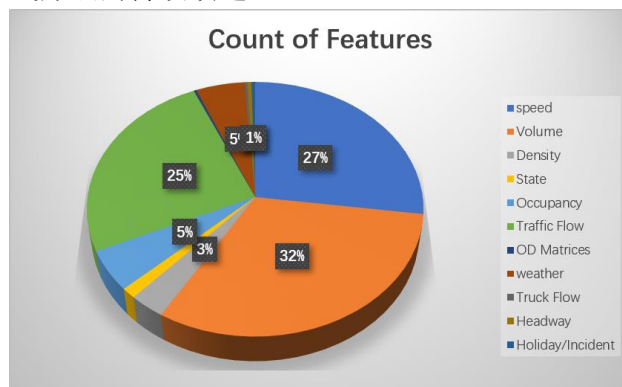


图2 交通预测的特征对象统计

Fig.2 Feature object statistics of traffic prediction

关于对交通车流量预测的研究, 如: Wein Zhang 等提出一种基于卷积神经网络 (CNN) 深度学习框架的交通流量预测模型。通过时空特征选择算法 (STFSA) 确定最佳输入数据的时间滞后和空间数据量, 从实际数据中提取选定的时空特征并转化为二维矩阵, 该方法在准确性方面优于基线模型^[28]。Sun J, Zhang J, Li Q 等通过扩展图卷积来处理空间信息, 提出了利用空间图卷积构建

多视图卷积网络(MVGCN)来处理车流量预测问题,不同的视图可以捕捉到不同的因素。广泛的实验结果表明优于最先进的方法的适应性^[29]。

关于对交通预测与天气、节假日、事故等的关系研究,如:Arief, Koesdwiady, Ridha 等提出了一种结合了用于交通和天气预测的深度信念网络和决策级数据融合方案,以提高利用天气条件进行预测的精度。实验结果验证了所提方法的有效性^[30]。D Zhang, MR Kabuka 等将门控循环神经网络应用于考虑天气条件的城市交通流预测。运行结果表明,降低了预测错误率^[31]。

综合来看,对于交通车流量的预测是最主流的研究,天气、节假日、事故等被看作为外部因素。

1.3 交通流量抽象方式

时间序列描述一个对象随着时间变化的情况,一个数据对应一个时间点,并且相邻数据之间的时间间隔相同,间隔可以是任意大小。在交通流量预测领域,天气数据、交通流量等都可以抽象为时间序列。

时间序列表示形式:

$$\mathbf{X}_t = [\mathbf{X}_t^1, \mathbf{X}_t^2, \mathbf{X}_t^3, \dots, \mathbf{X}_t^{N-1}, \mathbf{X}_t^N] \in \mathbf{R}^{N \times d} \quad (1)$$

其中, \mathbf{X}_t^i 表示 $t=1$ 时刻的数据,则表示所有时刻的数据。传统的统计模型如 ARIMA 会找出与 t 时刻数据的相关的项数,比如是 k 项,那么就使用 $\mathbf{X}_{t-k}, \mathbf{X}_{t-k+1}, \dots, \mathbf{X}_{t-1}$ 来预测 \mathbf{X}_t ^[32]。

SK Prasad, D Aghajarian 等调查了当前的技术状态,并且提出了科学界关于处理时空大数据集的新兴的研究挑战。大数据集的一个共同点就是它们都包含时间空间两个维度。各种机构、科学界以及整个社会越来越多地依赖于空间数据管理、分析和空间数据挖掘,以获得见解并产生可操作的计划。因此,需要可靠的软件基础设施生态系统来进行时空大数据管理和分析,这将成为解决不同科学和工程领域广泛的研究问题的关键工具,并为用户提供下一代工具^[33]。

随着研究人员的不断深入发现,交通流量数据深入内嵌时间空间两个维度。交通车流量需要同时考虑这两个维度才能进一步提高预测精度。有效感知数据间的时空依赖对交通流量预测至

关重要,因此交通流量数据抽象方式逐渐从时间序列转变为时空序列,不同时刻节点的交通流量可以使用特征矩阵进行抽象:

$$\mathbf{X}_t = [\mathbf{X}_t^1, \mathbf{X}_t^2, \mathbf{X}_t^3, \dots, \mathbf{X}_t^{N-1}, \mathbf{X}_t^N] \in \mathbf{R}^{N \times d} \quad (2)$$

\mathbf{X}_t^i 表示 t 时刻,第 i 个节点的特征,例如交通车流量、车速等, d 表示特征维度。。

1.4 时空注意力机制

所谓注意力机制,最早出现在自然语言处理领域,是在对词向量进行 embedding 过程中,要在每个单词(以英文为例)选取一组权重来表示这个单词对于整个句子要表达的意思的重要性。也就是说注意力机制实际上就是生成一组权重来对特征进行调整^[34]。

在空间维度上,不同地点的交通条件相互影响,且相互影响具有高度的动态性。使用注意力机制来自适应的捕捉空间维度节点之间的动态相关性。

时间维度上,使用空间注意力机制计算出来的权值矩阵对邻接矩阵进行加权,这个权值矩阵就一直是一个确定的矩阵,但是随时间变化,道路之间相互影响力也会发生变化,那么意味着这个权值矩阵应该也是随着时间发生变化的。通过参考空间注意力的计算方法,时间注意力机制也是计算两个时间点之间的相互关系来生成权值矩阵。

时空注意力机制就是空间注意力权值矩阵对 GCN 的邻接矩阵进行加权,时间注意力权值矩阵原始交通流量时间序列进行重构,形成时空注意力机制。利用空间注意力机制捕捉不同位置之间的动态空间相关性,利用时间注意力机制捕捉不同时间的动态时间相关性。时空注意力机制极大地扩大了感受野,可以非常好的抽象出交通网络的动态化属性。

交通路口相当于图数据结构的节点,每一个节点对每一个节点的交通流量的影响不一样,但是在下一个时刻,每一个节点对每一个节点的交通流量的影响又会不一样。同时每一个时间点对于要预测的时间的重要性也不一样,因此同时考虑时间和空间的注意力才能对交通时空序列的时空特性很好的计算出来。

2 交通流量的时空依赖建模

如图 1 所示, 本文分别针对交通流数据和天气数据分别提取特征, 并基于此构建交通流量预测模型。本章针对交通流数据利用图神经网络和循环神经网络网络提取其中的时空依赖。

2.1 问题描述

本文前面提到交通流量数据用时空序列进行抽象:

$$\mathbf{X}_t = [\mathbf{X}_t^1, \mathbf{X}_t^2, \mathbf{X}_t^3, \dots, \mathbf{X}_t^{N-1}, \mathbf{X}_t^N] \in \mathbb{R}^{N \times F} \quad (3)$$

其中 \mathbf{X}_t 表示, t 时刻所有节点的特征。 \mathbf{X}_t^1 表示, t 时刻, 节点 1 所有的特征, 定义:

$$\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \dots, \mathbf{X}_{\tau-1}, \mathbf{X}_\tau] \in \mathbb{R}^{N \times F \times \tau} \quad (4)$$

其中 \mathbf{X} 代表 τ 个时间片段所有节点所有时刻的特征。

交通流量数据深入内嵌时间、空间两个维度, 有效提取交通流量数据的时空依赖关系对交通流量预测至关重要。传统时间序列分析模型难以处理不稳定以及非线性数据, 传统机器学习模型用于处理复杂数据但是依赖与特征工程需要较多专家经验。深度学习模型通常假设交通流量的两时刻数据具有独立性, 这显然与事实相矛盾。使用时空序列对交通流量进行抽象, 并且使用图神经网络显然更合理, 但是传统图神经网络将交通网络看作一个静态结构, 无法对交通网络结构的动态变化进行建模。注意力机制是解决这个问题主要方法, 它生成的权值矩阵对邻居矩阵进行动态调整。时空注意力机制可以有效的捕捉交通网络的动态时空相关性^[35]。

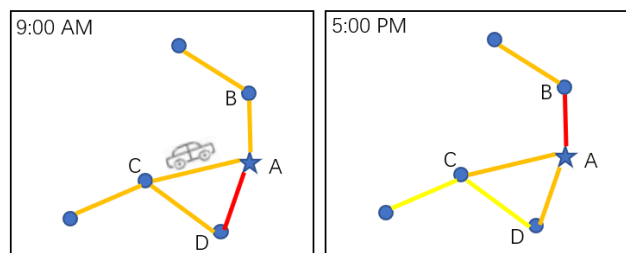


图 3 空间角度的时空相关性

Fig.3 Spatiotemporal correlation of spatial angle

如图所示 3, 从空间角度分析, 说明不同路段对于 A 点影响不同, 且随着时间推移, 各个路

段之间的强度也会发生变化。

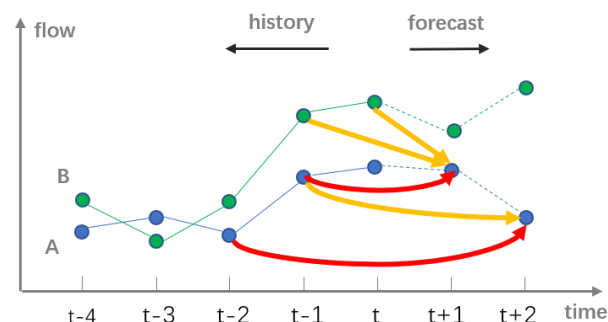


图 4 时间角度的时空相关性

Fig.4 Spatiotemporal correlation of temporal angle

图 4 所示, 从时间角度分析, 对于 A 路段的交通流量来说, 不同位置的历史观测值, 对 A 路段未来流量有不同影响。

由此可见交通流量数据时间相关性与空间相关性深入交错内嵌在一起。注意力机制是对邻接矩阵进行调整, 目的是为了更有效的利用信息。

本文提出一种基于时空注意力机制的图卷积网络, 分别使用 GCN 提取空间特征, 使用空间注意力机制计算空间依赖, 使用 CNN 提取时间特征, 使用时间注意力机制对 CNN 的输出结果进行调整。时空注意力机制结合时空卷积有效刻画了交通数据的时空动态演化属性, 这为后续预测的准确性提供了良好的基础。其模型架构图如图 5 所示。

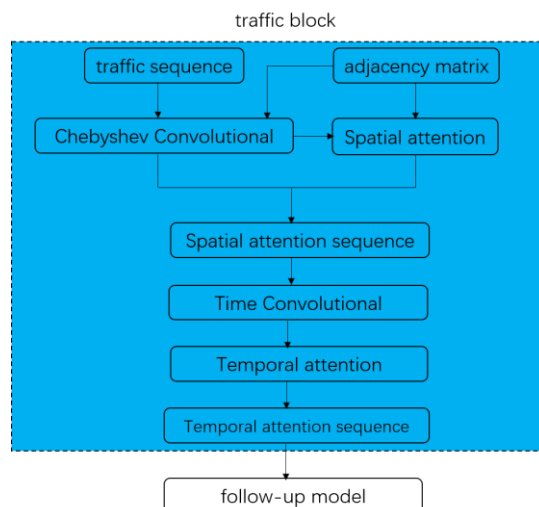


图 5 交通模块结构

Fig.5 Traffic block structure

2.2 交通流量数据的空间依赖

使用 GCN+CNN 分别对交通数据进行空间和时间依赖的捕获,使用时空注意力机制实现动态赋权,可以有效计算交通数据的时空动态化属性。本节内容将研究怎样使用 GCN 以及使用怎样的 GCN 结合空间注意力机制获取数据的空间依赖。

2.2.1 构建拉普拉斯矩阵

常用的 GCN 有谱图网络和空间图网络,谱图网络基于邻接矩阵将图结构抽象为拉普拉斯矩阵 (Laplacian)。

给定一个节点数为 n 的简单图 G , D 是 G 的度矩阵, A 是 G 的邻接矩阵。而度矩阵 D 为,存放每个节点的邻居节点的个数的对角矩阵,它的求解公式为:

$$D = \text{diag}(\text{sum}(A, 1)) \quad (5)$$

其中 $\text{sum}(\square)$ 操作表示将矩阵沿着某一个维度进行累加,这里的 1 即为第二个维度,所以公式(5) $\text{sum}(\square)$ 操作的结果是一个一维数组,而 $\text{diag}(\square)$ 操作可以将一个一维数据映射为一个对角矩阵,那么 G 的拉普拉斯矩阵可以表示为:

$$L = D - A \quad (6)$$

拉普拉斯矩阵也需要进行归一化处理,在此之前先求出它的最大特征值:

$$\lambda_{\max} = \max(\text{eigs}(L, 1)) \quad (7)$$

而其中 $\text{eigs}(\square)$ 操作则表示求解矩阵的特征值,参数 1 表示求出所有的特征值并且返回的是一个由特征值组成的向量, $\max(\square)$ 操作则求解出最大特征值。

对拉普拉斯矩阵进行归一化操作:

$$\tilde{L} = \frac{2}{\lambda_{\max}} L - I_N \quad (8)$$

\tilde{L} 则表示归一化后的拉普拉斯矩阵, I_N 表示该拉普拉斯对应维度的单位矩阵。

2.2.2 GCN 与切比雪夫多项式

本文采用切比雪夫多项式近似求解进行图卷积,切比雪夫多项式原本的近似求解过程:

$$T_0(x) = 1 \quad (9)$$

$$T_1(x) = x \quad (10)$$

$$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x) \quad (11)$$

这里的 x 则表示一个变量,公式(11)即为切比雪夫多项式的递归公式,代入 2.2.1 中的归一化拉普拉斯矩阵则有:

$$T_k(\tilde{L}) = 2\tilde{L}T_{k-1}(\tilde{L}) - T_{k-2}(\tilde{L}) \quad (12)$$

结合本文的任务,将公式(12)改写为图卷积操作:

$$g_{\theta} *_G x = g_{\theta}(L)x = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{L})x \quad (13)$$

其中 g_{θ} 表示卷积核, $*_G$ 表示图的形式的乘积操作, x 则表示输入数据。所以实际上,公式(13)是利用卷积核 g_{θ} 提取 0 到 $(k-1)$ 阶邻居的信息。

经过了时空注意力和若干个图卷积的过程,使得图结构种的每一个点都提取了全局点的特征,但是由于空间注意力的机制,使得不会由于图卷积的堆叠使得节点特征趋于过平滑。

2.2.3 空间注意力机制

在 GCN 每一层图卷积中所有节点都会同时吸收它的邻居节点的特征,再进行一次图卷积所有节点都会同时吸收它的跳邻居节点的特征,经过多次图卷积(需要进行多少次图卷积操作取决于输入的图结构),每一个节点都会有所有节点的特征,即全局特征。通过图卷积堆叠,虽然可以显著的扩大感受野,但是该方法也会导致过平滑现象,使网络节点特征趋同(也就是说图卷积堆叠过多可能导致每一个的节点的特征非常接近),预测准确性下降。

一个图节点在进行图卷积的过程中,它需要结合自己的特征与所有邻居节点的特征,加入空间注意力机制,在吸收自己与所有邻居节点时加入一个权重向量,也就是给图结构中的边加上一

组参数, 这个权值的大小代表目标节点与邻居节点以及目标节点自身的重要程度, 也就是待更新的目标节点更多的考虑哪个节点的特征, 更少的考虑哪个节点的特征。

公式(3)与(4)表示了交通流量的抽象方式, 定义公式(14)表示向量 \mathbf{X}^i 代表节点 i 的 τ 个时间段交通流量特征。 \mathbf{X}_1^i 代表 $t=1$ 时刻, 节点 i 的特征表示为:

$$\mathbf{X}^i = [\mathbf{X}_1^i, \mathbf{X}_2^i, \mathbf{X}_3^i, \dots, \mathbf{X}_{\tau-1}^i, \mathbf{X}_{\tau}^i] \in \mathbb{R}^{F \times \tau} \quad (14)$$

在本文 2.2.2 中提到使用了切比雪夫作为图卷积的公式, 提取了顶点 i 的空间特征, 在此处将输出结果看作为 \mathbf{X}^i 的隐藏特征 \mathbf{h}^i :

$$\mathbf{h}_i = \text{GCN}(\mathbf{x}^i) \quad (15)$$

其他的顶点也对应它的相关隐藏特征, 对于顶点 i , 逐个计算它的邻居节点 ($j \in N_i$) 和它自己之间的相关系数, 目前计算相关系数的方法特别多, 比如内积等, 本文采用对隐藏特征拼接操作, 最后把拼接后的高维特征映射到一个实数上, 用公式表示如下:

$$S_{ij} = \text{attention}([\mathbf{h}_i \parallel \mathbf{h}_j]), j \in N_i \quad (16)$$

其中 S_{ij} 表示节点 i 与节点 j 之间的相关系数, $[\parallel]$ 表示对节点 i 和 j 变换后的特征进行了拼接操作, $\text{attention}(\cdot)$ 表示将特征映射到一个实数上。

使用 softmax 进行归一化处理:

$$S'_{ij} = \frac{\exp(\text{Leaky ReLU}(e_{ij}))}{\sum_{k \in N_i} \exp(\text{Leaky ReLU}(e_{ik}))} \quad (17)$$

S'_{ij} 即代表节点 i 与 j 的注意力系数, 依次使用上述操作处理, 对所有的节点的邻居节点计算注意力系数。计算的结果形成注意力系数矩阵 \mathbf{S} , 它作为 GCN 的邻接矩阵 \mathbf{A} 的注意力矩阵。

将空间注意力机制应用到该图卷积过程中即为:

$$g_{\theta} *_{\mathbf{G}} x = g_{\theta}(L)x = \sum_{k=0}^{K-1} \theta_k(T_k(\tilde{L}) \parallel \mathbf{S})x \quad (18)$$

2.3 交通流量数据的时空依赖

在本文 2.2 中使用 GCN 提取空间特征, 对 GCN 输出的特征使用空间注意力机制计算它的

空间注意力矩阵进而重构邻接矩阵和 GCN 的输出结果。2.1 中提到, 交通网络的空间特征具有动态演化属性。在本节中, 对 GCN 重构的输出结果使用 CNN 提取时间特征, 再使用时间注意力机制重构 CNN 的输出结果, 时空注意力的结合再加上时空卷积提取出交通数据的两个维度的特征。

使用 CNN 提取时间特征的公式为:

$$\mathbf{h}_i = \Psi(g_{\theta} *_{\mathbf{G}} \mathbf{x}) \quad (19)$$

其中 Ψ 表示卷积操作, \mathbf{h}_i 代表 i 时刻经过卷积后新的交通特征序列, 对新的序列将使用时间注意力机制重构。

研究两个时间片段的相关性, 对于时空序列用下列这个公式进行抽象:

$$\mathbf{X}_i = [\mathbf{X}_i^1, \mathbf{X}_i^2, \mathbf{X}_i^3, \dots, \mathbf{X}_i^{N-1}, \mathbf{X}_i^N] \in \mathbb{R}^{N \times F} \quad (20)$$

\mathbf{X}_i^1 表示 i 时刻节点 1 对于的交通流量特征,

\mathbf{X}_i 则表示 i 时刻所有节点的交通流量特征。

将 \mathbf{X}_i 经过普通卷积的结果 \mathbf{h}_i 看做 \mathbf{X}_i 的隐藏特征, 其他的时间点也对应它的相关隐藏特征, 对于 i 时刻, 逐个计算它与其他时间片段之间的相关系数, 采用对隐藏特征拼接操作, 最后把拼接后的高维特征映射到一个实数上, 用公式表示如下:

$$T_{ij} = \text{attention}([\mathbf{h}_i \parallel \mathbf{h}_j]), j \in N_i \quad (21)$$

其中 T_{ij} 表示 i 时刻与 j 时刻之间的相关系数, $[\parallel]$ 表示对 i 时刻与 j 时刻变换后的特征进行了拼接, $\text{attention}(\cdot)$ 表示将特征映射到一个实数上。

学习 i 时刻与 j 时刻之间的相关性, 就是通过可学习的参数 \mathbf{w} 与 $\text{attention}(\cdot)$ 映射完成的。

最后再进行一下特征归一化, 本文使用 softmax 操作:

$$T'_{ij} = \frac{\exp(\text{Leaky ReLU}(e_{ij}))}{\sum_{k \in N_i} \exp(\text{Leaky ReLU}(e_{ik}))} \quad (22)$$

T'_{ij} 即代表节点 i 与 j 的注意力系数, 依次使用上述操作处理, 对所有的时间点计算注意力系数。计算的结果形成注意力系数矩阵 \mathbf{T} , 得到的注意力矩阵 \mathbf{T} 对 CNN 的输出结果 \mathbf{h} 相乘得到最

终输入, 用公式表示如下:

$$\hat{\mathbf{h}} = (\hat{\mathbf{h}}_1, \hat{\mathbf{h}}_2, \hat{\mathbf{h}}_3, \dots, \hat{\mathbf{h}}_{\tau-1}, \hat{\mathbf{h}}_{\tau}) = (\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3, \dots, \mathbf{h}_{\tau-1}, \mathbf{h}_{\tau}) \cdot \mathbf{T} \quad (23)$$

3 结合天气因素的交通预测

3.1 问题描述

众所周知, 天气对于出行体验以及交通效率都有显著的影响。那么天气因素对于交通流量的预测应该被当作一个外部因素。在过去的研究中, 相对来说比较少的考虑了天气、假日以及事故等外部因素对交通流量预测的影响。但是显而易见, 考虑这些外部因素能够提高交通流量预测的准确性^[36]。

本文前面提到交通网络结构被抽象成图结构, 这里所有节点都共享相同的天气数据, 换言之天气数据不需要考虑空间特征。本文的天气数据来源于 mesonet 网站, 从该网站找出对应交通数据的时间和地点的天气数据。天气数据可以分为两类, 一类是一般的数值数据比如气温、风速等, 还有一类是分类特征, 比如天气代码, 它代表这个时刻的天气评价类别。分类特征使用 onehot 处理成多维向量, 一般的数值数据经过正常的归一化处理, 然后与 onehot 的处理结果合并, 合并的结果为最终的天气数据。

很显然天气数据对交通流量的预测是一个辅助作用, 所以天气数据首先应该匹配交通数据的时间戳。本文使用两个小时的数据预测一个小时的数据。本文使用 GRU 结合 CNN 对天气数据进行预测, 预测的结果与天气数据做损失, 对应的交通模块也与交通做损失。GRU+CNN 处理后的天气特征和第 2 章中的 GCN+CNN 处理的交通特征, 使用多头注意力机制进行特征融合, 融合后特征生成最终的预测结果。

本文前面提到交通数据时空序列进行抽象:

$$\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \dots, \mathbf{X}_{\tau-1}, \mathbf{X}_{\tau}] \in \mathbb{R}^{N \times F \times \tau} \quad (24)$$

其中代表 τ 个时间片段所有节点所有时刻的所有特征, 以 t 时刻为例:

$$\mathbf{X}_t = [\mathbf{X}_t^1, \mathbf{X}_t^2, \mathbf{X}_t^3, \dots, \mathbf{X}_t^{N-1}, \mathbf{X}_t^N] \in \mathbb{R}^{N \times F} \quad (25)$$

其中 \mathbf{X}_t 表示, t 时刻所有节点的所有特征。 \mathbf{X}_t^1

表示, t 时刻节点 1 所有的特征。交通数据时空序列经过交通模块的处理, 生成交通特征时空序列, 可以表示为: \mathbf{X} 经过函数 f_{tf} 生成 \mathbf{h}_{tf} , 即:

$$\mathbf{h}_{tf} = f_{tf}(\mathbf{X}) = [\mathbf{h}_{t+1}^{tf}, \mathbf{h}_{t+2}^{tf}, \mathbf{h}_{t+3}^{tf}, \dots, \mathbf{h}_{t+n-1}^{tf}, \mathbf{h}_{t+n}^{tf}] \quad (26)$$

其中 n 代表最后的预测时间步长, 而天气数据没有空间特征, 所以天气数据是一个多特征的时间序列, 将天气数据抽象成下面形式的时间序列:

$$\mathbf{M} = [\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3, \dots, \mathbf{M}_{\tau-1}, \mathbf{M}_{\tau}] \in \mathbb{R}^{F \times \tau} \quad (27)$$

其中 \mathbf{M} 表示天气数据的所有时刻所有特征, \mathbf{M}_1 即表示 $t=1$ 时刻天气数据的所有特征。同样的道理, 天气数据时间序列经过天气模块的处理, 生成天气特征时间序列, 可以表示为: \mathbf{M} 经过函数 f_{wx} , 生成特征 \mathbf{h}_{wx} , 即:

$$\mathbf{h}_{wx} = f_{wx}(\mathbf{M}) = [\mathbf{h}_{t+1}^{wx}, \mathbf{h}_{t+2}^{wx}, \mathbf{h}_{t+3}^{wx}, \dots, \mathbf{h}_{t+n-1}^{wx}, \mathbf{h}_{t+n}^{wx}] \quad (28)$$

交通模块与天气模块的输出, 本文提出一种用注意力特征融合机制 (Attention Feature Fusion, AFF), 对两种特征进行融合, 即两种序列经过函数 f_{fusion} , 再结合反向传播算法生成特征 \mathbf{Y} , \mathbf{Y} 就是本文的预测结果序列, 用公式表示为:

$$\mathbf{Y} = f_{fusion}(\mathbf{h}_{wx}, \mathbf{h}_{tf}) = [\mathbf{y}_{t+1}, \mathbf{y}_{t+2}, \mathbf{y}_{t+3}, \dots, \mathbf{y}_{t+n-1}, \mathbf{y}_{t+n}] \quad (29)$$

本章内容使用的方法其模型架构如图 6 所示:

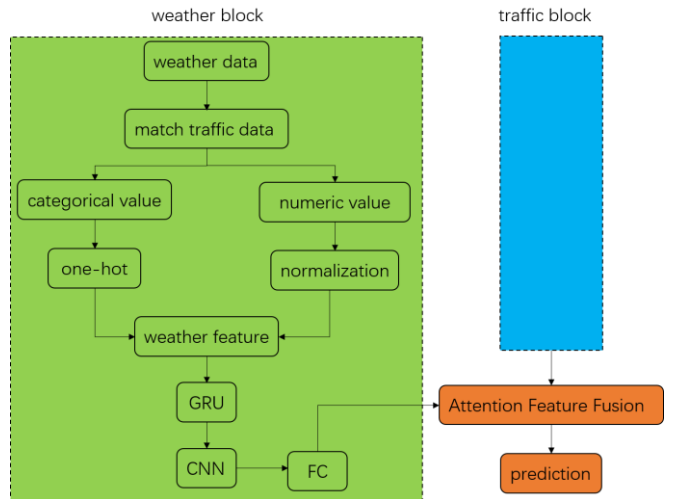


图 6 天气模块结构

Fig.6 weather block structure

3.2 天气数据的处理

表 1 天气特征列表

Table 1 List of weather characteristics

编号	名称	含义
1	valid	观察的时间戳
2	tmpf	2 米高度的气温
3	dwpf	露点温度
4	relh	相对湿度 (%)
5	drct	度数为单位的风向
...
24	peak_wind_gust	高峰阵风风值
25	peak_wind_drct	高峰阵风方向
26	peak_wind_time	高峰阵风时刻
27	feel	表观温度 (热指数)

为了匹配与交通数据匹配的天气数据, 根据交通数据集的时间以及地点, 从 mesonet 网站 (美国爱荷华州州立大学的爱荷华州大气观测站网) 中下载对应的天气数据。首先时间戳对应天气时间序列的顺序, 不予处理, 特征 15、19、24、25、26 没有采集数据不予选择, 21、22、23 由于交通数据集对应的时间和地点根本没有积冰现象也不予选择, 最终的天气数据由时间戳与 17 个天气特征。

许多机器学习算法和几乎所有深度学习架构都无法处理原始形式的纯文本。这意味着他们对算法的输入必须是数字的, 才能解决分类或回归问题。因此, 有必要使用编码技术将这些分类变量编码为数值^[37]。17 个天气特征中 12、13、14、20 这 4 个为分类特征, 使用 one-hot 处理成向量。其中 12、13、14 分别代表天空 1 至 3 级的覆盖范围, 它们的特征都是由 6 个不同的代码组成, 因此将它们对应转化为 6 个不同的 6 维向量, 如下表所示:

表 2 one-hot 向量表 1

Table 2 One pot Vector Table-1

特征代码	One-hot 向量
CLR	[1, 0, 0, 0, 0, 0]
FEW	[0, 1, 0, 0, 0, 0]
VV	[0, 0, 1, 0, 0, 0]

SCT	[0, 0, 0, 1, 0, 0]
BKN	[0, 0, 0, 0, 1, 0]
OVC	[0, 0, 0, 0, 0, 1]

特征 20 代表当前天气代码 (当前天气的综合评价), 它的特征由 7 个不同的代码组成, 将它们对应转化为 7 个不同的 7 维向量, 如下表所示:

表 3 one-hot 向量表 2

Table 3 One pot Vector Table-2

特征代码	One-hot 向量
HZ	[1, 0, 0, 0, 0, 0, 0]
RA	[0, 1, 0, 0, 0, 0, 0]
BR	[0, 0, 1, 0, 0, 0, 0]
RA BR	[0, 0, 0, 1, 0, 0, 0]
BR	[0, 0, 0, 0, 1, 0, 0]
BCFG	[0, 0, 0, 0, 0, 1, 0]
BCFG	[0, 0, 0, 0, 0, 0, 1]

4 个分类特征处理成了 25 维的 onehot 向量, 剩下的 13 个特征为正常的数值特征。其中特征 4 为百分数, 直接转化为对应的小数。剩下的 12 数值特征, 本文使用下式进行归一化处理:

$$S = \frac{x - \min(x)}{[\max(x) - \min(x)]} \quad (30)$$

公式 33 中, 将原始数据减去该类型数据的最大值与该类型数据的最大值与最小值的差作比值。经过公式 32 的处理, 12 种类型的天气数据都被归一化到[0, 1]的范围。

经过 onehot 对分类特征的处理后, 原本 12、13、14、20 这 4 个分类特征, 分别变成了 6 维、6 维、6 维、7 维的 onehot 向量, 也就是一共 25 维的数据。加上这里的 13 维正常数值归一化后的数据, 最终的天气数据为 38 个维度的特征。

3.3 提取时间特征

在 3.2 中已经完成了天气数据的最终构建, 使用 GRU+CNN 提取天气数据的时间依赖。单个 GRU 的计算, 它会接受上一个 GRU 的隐藏特征输出 \mathbf{o}_{t-1} 和当前时刻的输入数据 \mathbf{x}_t , 也就是说单个 GRU 接收的输入是一个时间点的两个特征向量。隐藏特征和当前输入数据经过 GRU 的处理, 会输出当前 GRU 的隐藏输出 \mathbf{o}_t ^[38]。

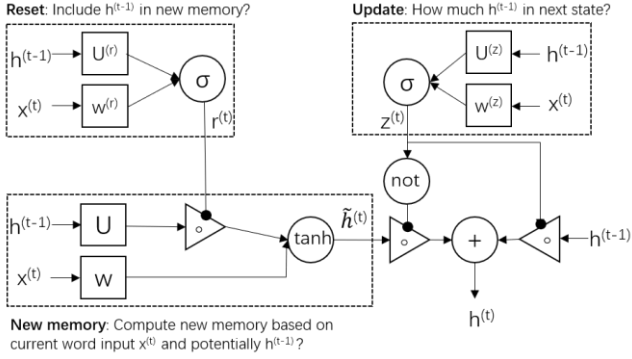


图7 GRU 内部结构

Fig.7 GRU internal structure

对于本文的任务, t 时刻的 GRU, 它的输入数据应该为:

$$\mathbf{x}_t = \mathbf{h}_t^{\text{wx}} = [h_1^{\text{wx}}, h_2^{\text{wx}}, h_3^{\text{wx}}, \dots, h_{37}^{\text{wx}}, h_{38}^{\text{wx}}] \quad (31)$$

以及上一个 GRU 的隐藏特征输出 \mathbf{o}_{t-1} :

$$\mathbf{o}_{t-1} = [o_1, o_2, o_3, \dots, o_{c-1}, o_c] \quad (32)$$

其中 c 为 hidden_size, 本文的实验设置 input_size=hidden=38, 将输入数据 \mathbf{h}_t^{wx} 以及上一个 GRU 的隐藏特征输出 \mathbf{o}_{t-1} 分别乘上三组可训练参数, 分别用于更新门、重置门以及记忆单元的计算, 得到重置门和更新门的输出:

$$\mathbf{r}_t = \sigma(\mathbf{x}\mathbf{r}'_t + \mathbf{h}\mathbf{r}'_t + \mathbf{b}_r) \quad (33)$$

$$\mathbf{z}_t = \sigma(\mathbf{x}\mathbf{z}'_t + \mathbf{h}\mathbf{z}'_t + \mathbf{b}_z) \quad (34)$$

结合重置门的输出计算记忆单元:

$$\tilde{\mathbf{o}}_t = \tanh(\mathbf{x}\mathbf{o}'_t + \mathbf{r}_t \square \mathbf{h}\mathbf{o}'_t) \quad (35)$$

其中 $\mathbf{x}\mathbf{r}'_t$ 、 $\mathbf{x}\mathbf{z}'_t$ 、 $\mathbf{x}\mathbf{o}'_t$ 为当前输入数据乘上三组可训练参数的结果, 而 $\mathbf{h}\mathbf{r}'_t$ 、 $\mathbf{h}\mathbf{z}'_t$ 、 $\mathbf{h}\mathbf{o}'_t$ 为上一个隐藏特征输出乘上三组可训练参数的结果, 根据记忆单元和更新门得到最终输出:

$$\mathbf{o}_t = (1 - \mathbf{z}_t) \square \tilde{\mathbf{o}}_t + \mathbf{z}_t \square \mathbf{o}_{t-1} \quad (36)$$

其中 σ 为 sigmoid 激活函数, 由于可能设置多层的 GRU, 所以当前时刻的输入可能为上一层 GRU 的输出, 以上就是本文对单个 GRU 的计算过程, 如果 $t=1$, 则不计算上一个 GRU 的隐藏特征, 即将 $\mathbf{o}_{t=0}$ 设置为全为 0 的矩阵。

天气数据经过处理变成一个时间序列, 每一

个时间点包含 38 维的数据, 本文使用两层 GRU 的叠加提取它们的时间特征, 使用 CNN 将时间步长从 24 缩放到 12, 最后 CNN 的处理结果就是预测的天气数据, CNN 的输出结果将与天气的正确值做损失。

3.4 特征融合生成预测

交通模块与天气模块的各自的输出经过本注意力特征融合机制生成融合特征, 但是在此之前分别计算出两者对应的损失。交通模块的输出结果实际上是对交通流量的预测, 天气模块实际上是对天气数据的预测, 两个模块的输出分别与相对应的标签数据做损失计算, 两个模块的输出使用维度变换和注意力特征融合生成最终预测。最后训练集、验证集、测试集对交通模块和天气模块分别做损失的公式为:

$$\text{Loss} = \alpha \mathbf{L}_{\text{tf}} + (1 - \alpha) \mathbf{L}_{\text{wx}} \quad (37)$$

其中 Loss 为总的损失结果, α 为超参数, \mathbf{L}_{tf} 为交通模块的损失结果, \mathbf{L}_{wx} 为天气模块的损失结果。 α 是一个取值范围为 $[0, 1]$ 的值, 它的大小根据实验效果调整, 它的取值决定了最终的预测结果天气特征所占的重要性。由于天气对交通预测的效果是一个辅助作用, 所以很显然 α 的值会比 $(1 - \alpha)$ 的值大。

由于是回归任务, 使用 MSE 函数计算损失, \mathbf{L}_{tf} 、 \mathbf{L}_{wx} , 各自的计算公式为:

$$\mathbf{L}_{\text{tf}} = \text{MSE}(\text{out}_{\text{tf}}, \text{Label}_{\text{tf}}) \quad (38)$$

$$\mathbf{L}_{\text{wx}} = \text{MSE}(\text{out}_{\text{wx}}, \text{Label}_{\text{wx}}) \quad (39)$$

out_{wx} 、 out_{tf} 分别为交通与天气模块的输出, Label_{wx} 、 Label_{wx} 为对应的标签。

经过天气模块处理后的天气特征是包含 38 个特征的时间序列 (数据维度为 $38 \times T$, T 为时间步长), 经过交通模块处理后的交通特征是包含 N 个节点, 每个节点都对应一个特征的时空数据 (数据维度为 $N \times T$, T 为时间步长)。使用 FC 层对天气特征进行维度映射, 输入通道为 38, 输出通道为 N , 处理后的结果天气特征和交通特征的维度保持一致 (数据维度为 $N \times T$, T 为时间步长):

$$\mathbf{h}'_{\text{wx}} = \text{FC}(\mathbf{h}_{\text{wx}}) \quad (40)$$

将交通特征乘上一组可学习参数得到 \mathbf{Q} 向

量, 天气特征分别乘上两组不同的可学习参数得到 \mathbf{K} 和 \mathbf{V} 向量:

$$\mathbf{Q} = \mathbf{W}_q \mathbf{h}_{\text{tr}} \quad (41)$$

$$\mathbf{K} = \mathbf{W}_k \mathbf{h}'_{\text{wx}} \quad (42)$$

$$\mathbf{V} = \mathbf{W}_v \mathbf{h}'_{\text{wx}} \quad (43)$$

引入注意力机制的目的是得出一组 $N \times N$ 的注意力系数矩阵, 这个矩阵代表 \mathbf{h}'_{wx} 的 N 维特征对于交通流量预测作用的重要性。首先需要设置一个打分函数 f_{score} , 以节点 1 为例, 计算节点 1 与其他所有特征的注意力分数向量:

$$\text{score}_1 = ([q_1 \square k_1], [q_1 \square k_2], [q_1 \square k_3], \dots, [q_1 \square k_N]) \quad (44)$$

而其他 $(N-1)$ 个节点的注意力的分数向量则为:

$$\text{score}_2 = ([q_2 \square k_1], [q_2 \square k_2], [q_2 \square k_3], \dots, [q_2 \square k_N]) \quad (45)$$

$$\text{score}_3 = ([q_3 \square k_1], [q_3 \square k_2], [q_3 \square k_3], \dots, [q_3 \square k_N]) \quad (46)$$

$$\text{score}_N = ([q_N \square k_1], [q_N \square k_2], [q_N \square k_3], \dots, [q_N \square k_N]) \quad (47)$$

其中, q_1, q_2, \dots, q_N , 是从 \mathbf{Q} 中分解出的向量, 每个向量的维度为 $T \times 1$, T 为时间步长, k_1, k_2, \dots, k_N 同理, $[a \square b]$ 操作表示向量 \mathbf{a} 与 \mathbf{b} 的内积, 而 $\text{score}_1, \text{score}_2, \dots, \text{score}_N$ 则表示节点 1, 2, \dots, N 注意力分数向量。

N 个节点的注意力分数向量形成注意力分数矩阵 score , 它包含 N 个节点与 N 个特征的 $N \times N$ 个注意力分数, 使用 Softmax 进行归一化处理生成注意力系数矩阵:

$$\text{Att}_{ij} = \frac{\exp(\text{Leaky ReLU}(e_{ij}))}{\sum_{k \in N_i} \exp(\text{Leaky ReLU}(e_{ik}))} \quad (48)$$

Att_{ij} 表示节点 i 与特征 j 之间的注意力系数, 计算的结果生成注意力系数矩阵 \mathbf{Att} , 最后的输出结果计算公式为:

$$\text{out} = \mathbf{h}_f + \mathbf{h}'_{\text{wx}} \mathbf{Att} \quad (49)$$

经过一个全连接层即生成最终的预测:

$$\mathbf{Y} = \text{FC}(\text{out}) \quad (50)$$

此处的预测结果包含 12 个点即 60 分钟的预测值, 其中输入数据为 24 个点即 120 分钟的数据, 15 分钟和 30 分钟的预测结果则取 60 分钟预测结果的前 3 个点和前 6 个点的值。

4 实验

本章对本文提出的注意力特征融合机制 (Attention Feature Fusion, AFF) 结合时空注意力的 GCN (Spatial-Temporal GCN) 对交通流量预测的模型 (AFFGCN) 进行实验验证。

4.1 实验数据集

表 4 数据集情况

Table 4 Data set situation

数据集	节点	特征数	时长	时间窗口
PeMSD4	307	3	59 天	5min
PeMSD8	170	3	61 天	5min

(1) PeMSD4 交通数据集

该交通数据集来源于 PeMS 网站^[39], 包含旧金山湾区 29 条高速公路 3848 个探测器, 2018 年 1 月 1 日至 2018 年 2 月 28 日这 2 个月的数据。这些传感器每 5 分钟收集一次数据, 包含 3848 个所有的传感器每 5 分钟经过的车辆数。

此外本数据集还包含一个 3×307 的邻接矩阵文件, 该数据表示了 307 个路口之间的相邻情况(即连通性) 以及节点之间的距离。图 8 显示了旧金山湾区高速公路的传感器分布图。

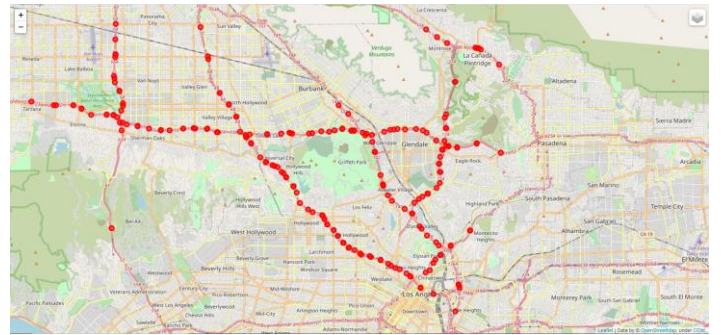


图 8 PeMSD4 传感器分布

Fig.8 PeMSD4 Sensor Distribution

(2) PeMSD8 交通数据集

该交通数据集来源于 PeMS 网站^[39], 包含圣贝纳迪诺市 (美国加利福尼亚州南部一座城市) 8 条高速公路 1979 个探测器, 2016 年 7 月 1 日至 2016 年 8 月 31 日这 2 个月的数据。这些传感器每 5 分钟收集一次数据, 包含 1979 个所有的传感器每 5 分钟经过的车辆数。

此外本数据集还包含一个 3*170 的邻接矩阵文件, 该数据表示了 170 个路口之间的相邻情况以及节点之间的距离。图 9 显示了圣贝纳迪诺市的传感器分布图。

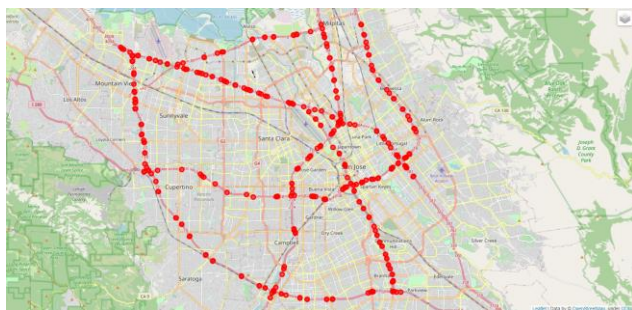


图 9 PeMSD8 传感器分布

Fig.9PeMSD8 Sensor Distribution

(3)PeMSD4 天气数据集

本数据集来源于 mesonet 网站 (美国爱荷华州立大学的爱荷华州大气观测站网) [40], 将和 PeMSD4 交通数据集搭配使用, 这个数据集记录了加州洛杉矶地区 2012 年 3 月 1 日至 2012 年 6 月 30 日 4 个月的天气数据, 具体包括气温、降雨量、风速等 32 个天气特征。该数据集每小时收集一次数据, 共包含 3427 条数据, 其中包含若干条异常时间戳数据。

(4)PeMSD8 天气数据集

本数据集来源于 mesonet 网站 (美国爱荷华州立大学的爱荷华州大气观测站网) [24], 将和 PeMSD8 交通数据集搭配使用, 这个数据集记录了加州旧金山湾区 2017 年 1 月 1 日至 2017 年 5 月 31 日 5 个月的天气数据, 具体包括气温、降雨量、风速等 32 个天气特征。该数据集每五分钟收集一次数据, 共包含 46560 条数据, 其中包含若干条异常时间戳数据。

4.2 数据集处理

使用公开数据集可以显著节约成本, 但是绝大多数公开数据集也会存在各种各样的问题, 比

如缺失值、异常值、异常时间戳等, 对于这些问题需要对其进行一定的处理。

(1) 缺失值、0 值、异常值

有时候由于交通管制或者传感器失灵等会有部分数据没有记录, 这里使用同类均值插补的方法进行填充。有时候由于某些原因出现记录错误, 比如出现 0 值或者出现显著异常的值也使用均值插补的方法对异常值进行替换。

(2) 异常时间戳

本文的交通数据集和天气数据集都是采用 5min 间隔的时间戳数据, 但是有时候传感器在一个时间间隔记录了两次数据。对于这种情况如果是累加的数据比如是交通流量, 将两个数据合并来替换, 但是如果是状态的数据比如温度就用均值替换。

(3) 数据合并

本文中的数据集, 其中交通数据集都是道路传感器 5 分钟收集一次, 天气数据集是天气观测站 5 分钟收集一次。本文将所有 5 分钟的数据合并成 30 分钟的数据。其中累加的数据依次累加起来, 而状态数据如温度使用均值替换。

(4) 数据集划分

在训练之前, 按照常规的处理情况, 60% 的数据作为训练集, 20% 的数据作为验证集, 20% 的数据作为测试集。

4.3 实验设置

4.3.1 实验环境

本文使用交通数据结合天气数据预测交通车流量, 交通数据集分别包含 170 和 307 个节点, 每个节点包含 3 个特征 (车流量、速度、时间占有率), 天气数据集分别包含 38 个和 27 个特征 (其中包含 25 维、16 维 onehot 特征)。数据采样每 5 分钟一次, 使用两个小时数据, 分别预测 15 分钟、30 分钟、1 个小时的交通流量。

本文选择 pytorch 作为深度学习框架, 使用 Adam(Adaptive Moment Estimation)作为优化器, MAE 作为损失函数, 5.3.3 中损失函数公式的 α 设置为 0.9, 最大 epoch 为 300, 初始学习率设为 0.001, 从第 50 个 epoch 开始, 每 10 个 epoch 学

习率衰减为原来的 1/10, 当验证机 MAE 大于 10 且连续 100 个 epoch 没有显著下降时, 将终止训练, 最后将保留最优结果和最后一次训练的结果。其中具体的开发环境如表 5 所示:

表 5 实验环境

Table 5 Experimental environment

	名称	型号/版本
硬件环境	CPU	Intel(R)Core(TM)i7-11800H
	GPU	GeForce RTX 3080 Ti
	内存	16GB
软件环境	操作系统	Windows 10 家庭中文版
	编程环境	Python 3.8.12
	深度学习框架	CUDA 11.1
		Pytorch 1.8.1 GPU
		PyTorch geometric 1.6.1

4.3.2 评价指标

评价指标实际就是要量化出预测值和实际值的差异, 与此相同的是神经网络在进行正向传播和反向传播的过程也需要计算预测值和实际值的差异, 二者的目的都是相同的。本文使用 MAE、RMSE、MAPE 这三个指标来分析评价实验结果, 其中 MAE 同时作为深度学习中的损失函数。使用 x_i 代表标签值, y_i 代表预测值, 其中 $i \in [1, N]$, N 表示预测值的个数。

(1) 平均绝对误差 (Mean Absolute Error, MAE):

$$MAE(x_i, y_i) = \frac{1}{N} \sum_{i=1}^N |x_i - y_i| \quad (52)$$

(2) 均方根误差 (Root Mean Squared Error, RMSE):

$$RMAE(x_i, y_i) = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2} \quad (53)$$

(3) 平均绝对百分比误差 (Mean Absolute Percentage, MAPE):

$$MAPE(x_i, y_i) = \frac{1}{N} \sum_{i=1}^N \left| \frac{x_i - y_i}{x_i} \right| \quad (54)$$

4.3.3 对比方法

由于机器学习方法、统计方法以及其他普通深度学习方法在交通预测的性能已经显著低于图神经网络, 本文所有的对比实验方法全部使用图神经网络相关的变体。

(1) GRCN: Graph-Refined Convolutional Network, 结构-精细图卷积网络, 原本是为了解决推荐系统中基于 GCN 解决隐式反馈问题的一种模型, 被改进用于交通预测^[41]。

(2) Gate_STGCN: 门控时空图卷积网络, 将 RNN 和 GCN 结合在一起的神经网络。

(3) DGCN: Dynamic Graph Convolution Network, 动态图卷积网络, 引入了一种潜在网络以提取时空特征, 以自适应地构建动态道路网图邻接矩阵^[42]。

(4) DGCN_MASK: 只使用掩码拉普拉斯矩阵的模型的修改版的动态图卷积网络。

(5) DGCN_recent: 只有邻近数据的动态图卷积网络

(6) DGCN_Res: 只使用残差拉普拉斯矩阵的模型修改版的动态图卷积网络。

(7) DGCN_GAT: 该方法是将 DGCN 模型的空间特征层 GTCL 替换为 GAT。

(8) ST-GCN: 本文使用方法 (没有使用天气数据), 使用时空注意力机制计算交通数据的时空相关性, 使用 GCN 提取空间特征, 使用 CNN 提取时间特征生成预测。

(9) AFFGCN: 本文使用方法, 使用时空注意力机制计算交通数据的时空相关性, 使用 GCN 提取空间特征, 使用 CNN 提取时间特征, 使用 GRU+CNN 提取天气数据特征, 使用注意力特征融合机制合并天气与交通数据生成预测。

4.4 实验结果分析

如表 6 所示, 在 PEMS04 数据集中, 本文提出的 AFFGCN 模型对比 GRCN 模型、Gated_STGCN 模型、DGCN_GAT 模型等 7 个模型在 15 分钟的交通流量预测 (共 3 个时间点) 分别提升了 35%、33%、30%、23%、18%、19%、10% (MAE、RMSE、MAPE 的均值, 未取小数) 的准确率。而本文的 AFFGCN 模型与本文没有考

考虑的天气因素的 ST-GCN 模型提升了在 MSE 评价指标上提高了 24% 的准确率, RMSE 提高了 28%, MAPE 提高了 24%。

AFFGCN 模型对比 GRCN 模型、Gated_STGCN 模型、DGCN_GAT 模型等 7 个模型在 30 分钟的交通流量预测(共 6 个时间点)分别提升了 36%、37%、29%、24%、19%、20%、11% 的准确率。而本文的 AFFGCN 模型与本文没有考虑的天气因素的 ST-GCN 模型提升了在 MSE 评价指标上提高了 26% 的准确率, RMSE 提高了

27%, MAPE 提高了 24%。

AFFGCN 模型对比 GRCN 模型、Gated_STGCN 模型、DGCN_GAT 模型等 7 个模型在 60 分钟的交通流量预测(共 12 个时间点),表现出更加卓越的性能,分别提升了 42%、45%、30%、27%、22%、23%、14% 的准确率。而本文的 AFFGCN 模型与本文没有考虑的天气因素的 ST-GCN 模型提升了在 MSE 评价指标上提高了 29% 的准确率, RMSE 提高了 31%, MAPE 提高了 28%。

表 6 PEMS04 实验结果

Table 6 PEMS04 experimental result

模型	15min			30min			60min		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
GRCN	23.18	35.2	18.18	24.06	36.48	18.82	26.54	39.97	21.24
Gated_STGCN	22.12	33.15	18.62	23.87	35.7	20.09	27.6	41.1	23.54
DGCN_GAT	21.81	34.68	15.91	21.9	34.9	15.83	22.21	35.4	15.86
DGCN_MASK	19.88	31.06	15.23	20.31	31.86	15.39	20.96	33.07	15.65
DGCN_recent	19.11	29.93	13.23	19.54	30.74	13.44	20.25	31.98	13.9
DGCN_Res	19.04	30.54	13.65	19.45	31.31	13.92	20.11	32.46	14.27
DGCN	17.14	27.49	12.29	17.51	28.18	12.53	18.10	29.21	12.84
ST-GCN	20.76	32.5	14.83	21.14	33.29	14.88	22.19	35.16	15.6
AFFGCN	15.26	23.54	11.23	15.68	24.31	11.38	15.71	24.36	11.24

如表 7 所示,在 PEMS08 数据集中,本文提出的 AFFGCN 模型对比 GRCN 模型、Gated_STGCN 模型、DGCN_GAT 模型等 7 个模型在 15 分钟的交通流量预测分别提升了 30%、27%、20%、19%、16%、16%、12% 的准确率。而本文的 AFFGCN 模型与本文没有考虑的天气因素的 ST-GCN 模型提升了在 MSE 评价指标上提高了 22% 的准确率, RMSE 提高了 23%, MAPE 提高了 26%。

AFFGCN 模型对比 GRCN 模型、Gated_STGCN 模型、DGCN_GAT 模型等 7 个模型在 30 分钟的交通流量预测分别提升了 31%、31%、19%、19%、16%、16%、12% 的准确率。而本文的 AFFGCN 模型与本文没有考虑的天气因素的 ST-GCN 模型提升了在 MSE 评价指标上提高了 21% 的准确率, RMSE 提高了 23%, MAPE 提高了 23%。

AFFGCN 模型对比 GRCN 模型、Gated_STGCN 模型、DGCN_GAT 模型等 7 个模型在 60 分钟的交通流量预测分别提升了 36%、39%、20%、20%、19%、18%、14% 的准确率。而本文的 AFFGCN 模型与本文没有考虑的天气因素的 ST-GCN 模型提升了在 MSE 评价指标上提高了 23% 的准确率, RMSE 提高了 25%, MAPE 提高了 24%。

表 7 PEMS08 实验结果

Table 7 PEMS08 experimental result

模型	15min	30min	60min
----	-------	-------	-------

	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
GRCN	18.16	26.81	14.04	18.77	27.89	14.3	20.3	30.38	15.47
Gated_STGCN	17.41	25.91	12.86	18.68	28.19	13.94	21.36	32.59	16.23
DGCN_GAT	15.93	24.44	11.35	16.1	24.89	11.39	16.48	25.73	11.53
DGCN_MASK	15.69	23.8	11.57	15.98	24.49	11.64	16.44	25.5	11.83
DGCN_recent	15.5	23.47	10.53	15.87	24.23	10.7	16.6	25.51	11.1
DGCN_Res	15.27	23.44	10.9	15.59	24.14	11.08	16.06	25.16	11.36
DGCN	14.73	22.86	10.17	15	23.51	10.3	15.48	24.5	10.61
ST-GCN	17	25.37	12.1	17.07	25.7	11.91	17.66	26.84	12.2
AFFGCN	13.2	19.42	8.96	13.51	19.88	9.15	13.64	20.07	9.23

总的来看,本文所提出的预测模型,在两组真实数据集(天气+交通)上都取得了卓越的预测效果,对比本文没有考虑天气因素的方法都提升了20%以上的性能,这非常好的证明了天气因素对于交通预测有非常积极的促进作用。

由此可见,在同样的预测方法中,考虑了天气因素的交通流量预测,对比没有考虑天气因素的交通流量预测至少有20%提高空间。天气对于交通预测的影响之大已经被逐步发现,相信在后续的研究中会出现越来越多将天气作为外部因素进行交通预测。

4 总结与展望

本文首先分析研究背景与研究工作的实际意义,接着分析了早期使用的预测技术与当前主要使用的方法,在此基础上本文提出了:1)使用GCN提取交通数据的空间特征,使用空间注意力机制调整GCN的输出获得空间依赖2)使用CNN提取时间特征,使用时间注意力机制调整CNN的输出获得交通数据的时空依赖3)分析了多种天气特征,使用归一化公式结合onehot对其做归一化处理,使用GRU+CNN对天气数据提取时间特征4)使用注意力特征融合机制将天气数据和交通数据进行融合生成预测。实验表明,本文所提出的方法显著提高了预测的准确性,并且证明了天气因素对于交通流量预测的有效性。

本文主要考虑了交通网络的动态演化属性与天气状况的外部因素两个方向进行交通流量的预测。在后续工作中,将从以下几个方面进一步深入研究:(1)在交通网络的动态演化属性研究中考考虑将长程节点纳入邻接矩阵,也就是当把交通网络抽象成图数据结构过程中,节点之间的边也用数据进行量化,也就是说每条路的长度是不一样的,继续使用图神经网络对边也提取特征,进一步提高模型的感受野。(2)从节点、区域和社

区三个层次探索交通网络粗化策略,将宏观、微观节点对交通流量的影响一体纳入预测,进一步扩大模型的感受野。(3)寻求更加有效的多任务学习方法,比如让天气数据和交通数据在训练过程中能够分开计算损失和梯度,让天气结合交通预测形成更加成熟有效的体系。(4)研究图神经网络体系结构自适应设计方法,解决由于演化性和层次性纳入建模范畴后导致模型体系结构设计复杂度增加的问题。(5)研究交通流量的长期预测,在交通预测的长期领域的预测效果始终不够理想,尝试引入Transformer体系解决长期预测的问题。

参考文献:

- [1] D. A. Tedjopurnomo, Z. Bao, B. Zheng, F. Choudhury, and A. K. Qin, "A Survey on Modern Deep Neural Network for Traffic Prediction: Trends, Methods and Challenges," IEEE Transactions on Knowledge and Data Engineering, vol. 14, no. 8, pp. 1-1, 2020.
- [2] Q. Chen, X. Song, H. Yamada, and R. Shibasaki, "Learning deep representation from big and heterogeneous data for traffic accident inference," in the AAAI Conference on Artificial Intelligence, 2016, vol. 30, no. 1, pp. 338-344.
- [3] K. Jayarajah, A. Tan, and A. Misra, "Understanding the interdependency of land use and mobility for urban planning," in ACM International Joint Conference and Symposium on Pervasive and Ubiquitous Computing and Weara

ble Computers, 2018, pp. 1079–1087.

[4] J. Kim, J.-D. Schmöcker, T. Nakamura, N. Uno, and T. Iwamoto, “Integrated impacts of public transport travel and travel satisfaction on quality of life of older people,” *Transportation research part A: policy and practice*, vol. 138, pp. 15–27, 2020.

[5] Ge L, Li H, Liu J, et al. Temporal Graph Convolutional Networks for Traffic Speed Prediction Considering External Factors[C]// 2019 20th IEEE International Conference on Mobile Data Management (MDM). IEEE, 2019.

[6] D. Nikovski, N. Nishiuma, Y. Goto, and H. Kumazawa, “Univariate short-term prediction of road travel times,” in *IEEE Intelligent Transportation Systems*, 2005, pp. 1074–1079.

[7] M. Lippi, M. Bertini, and P. Frasconi, “Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 871–882, 2013.

[8] B. M. Williams and L. A. Hoel, “Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results,” *Journal of transportation engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[9] J. Guo, W. Huang, and B. M. Williams, “Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification,” *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 50–64, 2014.

[10] B. Yu, H. Yin, and Z. Zhu, “Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting,” *International Joint Conference on Artificial Intelligence*, pp. 3634–3640, 2018.

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

[12] X. Feng, X. Ling, H. Zheng, Z. Chen, and Y. Xu, “Adaptive multi-kernel SVM with spatial-temporal correlation for short-term traffic flow prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2001–2013, 2018.

[13] Sun S, Wu H, Xiang L. City-Wide Traffic Flow Forecasting Using a Deep Convolutional Neural Network[J]. *Sensors*, 2020, 20(2):421.

[14] Abid Ee N Z U. The Deep 3D Convolutional Multi-Branching Spatial-Temporal-Based Unit Predicting Citywide Traffic Flow[J]. *Applied Sciences*, 2020, 10(21):7778.

[15] Z. Lv, J. Xu, K. Zheng, H. Yin, P. Zhao, and X. Zhou, “Lc-rnn: A deep learning model for traffic speed prediction,” in *International Joint Conference on Artificial Intelligence*, 2018, pp. 3470–3476.

[16] W. Zhang, Y. Yu, Y. Qi, F. Shu, and Y. Wang, “Short-term traffic flow prediction based on spatio-temporal analysis and CNN deep learning,” *Transportmetrica A: Transport Science*, vol. 15, no. 2, pp. 1688–1711, 2019.

[17] R. Fu, Z. Zhang, and L. Li, “Using LSTM and GRU neural network methods for traffic flow prediction,” in *The 31st Youth Academic Annual Conference of Chinese Association of Automation*, 2016, pp. 324–328.

[18] B. Liao et al., “Deep sequence learning with auxiliary information for traffic prediction,” in *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 537–546.

[19] S. Wang, J. Cao, and P. S. Yu, “Deep learning for spatio-temporal data mining: a survey,” *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2020.

[20] J. Ye, J. Zhao, K. Ye, and C. Xu, “How to Build a Graph-Based Deep Learning Architecture in Traffic Domain: A Survey,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–21, 2020.

[21] W. Chen, L. Chen, Y. Xie, W. Cao, Y. Gao, and X. Feng, “Multi-range attentive bicomponent graph convolutional network for traffic forecasting,” in the *AAAI Conference on Artificial Intelligence*, 2020, pp. 3529–3536.

[22] Z. Wu et al., “A Comprehensive Survey on Graph Neural Networks,” *IEEE transactions on neural networks and learning systems*, pp. 1–1, 2020.

[23] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering,” *arXiv preprint arXiv:1606.09375*, 2016.

[24] W. Jiang and J. Luo, “Graph Neural Network for Traffic Forecasting: A Survey,” *arXiv preprint arXiv:2101.11174*, 2021.

[25] Y. Xie, Y. Xiong, and Y. Zhu, “ISTD-GCN: Iterative Spatial-Temporal Diffusion Graph Convolutional Network for Traffic Speed Forecasting,” *arXiv preprint arXiv:2008.03970*, 2020.

[26] Ge L, Li H, Liu J, et al. Temporal Graph Convolutional Networks for Traffic Speed Prediction Considering External Factors[C]// 2019 20th IEEE International Conference on Mobile Data Management (MDM). IEEE, 2019.

[27] James J Q. Citywide traffic speed prediction: A geometric deep learning approach[J]. *Knowledge-Based Systems*, 2021, 212: 106592.

[28] Zhang W, Yu Y, Qi Y, et al. Short-term traffic flow prediction based on spatio-temporal analysis and CNN deep learning[J]. *Transportmetrica*, 2019, 15(2):1688-1711.

[29] Sun J, Zhang J, Li Q, et al. Predicting Citywide Crowd Flows in Irregular Regions Using Multi-View Graph Convolutional Networks[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2020, PP(99):1-1.

[30] Koesdwi Ad Y A, Soua R, Karray F. Improving Traffic Flow Prediction With Weather Information in Connected Cars: A Deep Learning Approach[J]. *IEEE Transactions on Vehicular Technology*, 2016, 65(12):9508-9517.

[31] Zhang D, Kabuka M R. Combining Weather Condition Data to Predict Traffic Flow: A GRU Based Deep Learning Approach[J]. *IET Intelligent Transport Systems*, 2018, 12(7):578-585.

[32] Keogh E, Kasetty S. On the Need for Time Series

es Data Mining Benchmarks: A Survey and Empirical Demonstration[J]. Data Mining & Knowledge Discovery, 2003, 7(4):349-371.

[33] Prasad S K, Aghajarian D, Mcdermott M, et al. Parallel Processing over Spatial-Temporal Datasets from Geo, Bio, Climate and Social Science Communities: A Research Roadmap[C]// 2017 IEEE International Congress on Big Data (BigData Congress). IEEE, 2017.

[34] Guo S, Lin Y, Feng N, et al. Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2019, 33:922-929.

[35] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[36] Barros J, Araujo M, Rossetti R J F. Short-term real-time traffic prediction methods: A survey[C]//2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS). IEEE, 2015: 132-139.

[37] Dahouda M K, Joe I. A deep-learned embedding technique for categorical features encoding[J]. IEEE Access, 2021, 9: 114381-114391.

[38] Cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[J]. arXiv preprint arXiv:1406.1078, 2014.

[39] PeMS 交通数据集来源: <https://archive.ics.uci.edu/ml/datasets/PEMS-SF>

[40] 天气数据集来源: <http://mesonet.agron.iastate.edu/ASOS/>

[41] Wei Y, Wang X, Nie L, et al. Graph-refined convolutional network for multimedia recommendation with implicit feedback[C]//Proceedings of the 28th ACM international conference on multimedia. 2020: 3541-3549.

[42] Guo K, Hu Y, Qian Z, et al. Dynamic graph convolution network for traffic forecasting based on latent network of laplace matrix estimation[J]. IEEE Transactions on Intelligent Transportation Systems, 2020.



杨安 (1996—), 男, 湖北黄冈人, 硕士研究生, 主要研究方向为机器学习、交通流量预测、时间序列等。

YANG An, born in 1996, M.S. candidate. His research interests include Machine learning, traffic flow prediction, time series etc.



王炜 (1979—), 男, 云南昆明人, 博士, 教授, 硕士生导师, 主要研究方向为机器学习、软件工程等。

WANG Wei, born in 1979, Ph.D., professor, M.S. supervisor. His research interests include machine learning, software engineering, etc.



刘成 (1999—), 男, 云南玉溪人, 硕士研究生, 主要研究方向为交通流量预测, 时间序列等。

Liu Cheng, born in 1999, M.S. candidate. His research interests include traffic flow prediction, time series etc.

1、杨安 (第一作者), 广东省东莞市凤岗镇翠湖豪苑, 523000, 1833330229@qq.com, 18824595271。

2、王炜 (通讯作者), 云南省昆明市云南大学呈贡校区软件学院, 650000, wangwei@ynu.edu.cn, 13354980100

参考文献类型

文献类型	图书	会议录	报纸	期刊	汇编
标志代码	M	C	N	J	G

文献类型	学位论文	报告	标准	专利	其他
标志代码	D	R	S	P	Z

载体类型标识

电子文献类型	数据库	计算机程序	电子公告	数据集
电子文献标志代码	DB	CP	EB	DS