

# Short-Term Forecasting of Taxi Demand Using Convolutional LSTM Network

## Abstract

打车需求的预测对城市的出租车管理是十分重要的,它能够帮助打车平台将空车从需求少的区域调度到需求大的区域并能够减少乘客的等待时间。打车需求预测需要考虑时间,空间以及节假日,气候等因素,由于这些因素的影响使预测非常具有挑战性。因此,我们提出一个基于 ConvLSTM 的深度学习方法来获取时空信息。更具体的是,我们利用一个基于 ConvLSTM 框架去模拟交通数据在时间上的临近性,周期性,趋势性等特征。对于每个性质,我们设计一个 ConvLSTM 单元的分支,每个分支模拟时间的不同性质并能够提取时空信息。本框架能够整合三个分支的输出,并根据输入数据自动调整每个分支的权重。输出也会进一步融合天气,日期等外部因素去预测打车需求。我们使用纽约的出租车数据评估我们的方法。实验结果显示我们的方法能够获取复杂的时空依赖关系并且性能超过其他的深度学习方法比如卷积神经网络。

## Introduction

最近几年,由于在线打车平台的出现,使人们的出行方式发生了重大的改变。乘客要想打车,只需要在 APP 上输入自己的上车位置和目的地,然后把请求发送给打车平台,平台就会自动为乘客分配乘坐车辆。相比于传统的运输系统,在线叫车更加方便和灵活。由于车辆的分配都是由打车平台来完成,因此,有效的车辆分配能够减少乘客和司机的互相等待时间。打车平台应该能够帮助司机决定未来应该去哪等待和接乘客。因此,我们需要一个能够预测整个城市未来打车需求的智能打车平台。

打车需求的预测是非常有挑战性的因为它和许多潜在的信息相关。打车需求的预测主要是解决一下三种依赖关系:

- (1) Temporal dependences: 打车需求有强烈的周期性,比如在早上和晚上车辆的需求比较大而在午夜车辆的需求量较少;并且打车需求是依赖最近的历史打车需求的变化。
- (2) Spatial dependences: 某个地区的打车需求不仅由这个地区的打车需求来决定也受附近区域的影响。通常来说,一个区域的打车需求受离这个区域较近的区域的影响较大。因此,一个能够捕捉局部空间信息的模型是被需要的。
- (3) External variable: 除了时空变量影响打车需求,一些外部变量,例如节假日,天气等因素也对打车需求有影响。

目前,尽管对打车需求预测的研究不是很多,但是对交通流量和交通速度的预测,以及降雨量的预测都能够为我们提供很好的参考。Zhang 等人[Zhang2016]提出一个深度的时空残差网络去同时预测一个城市某块区域的 inflow 和 outflow。为了捕捉时空依赖关系, Wu 等人[Wu 2016]提出使用 1D 卷积神经网络和 LSTM 进行交通流量预测。Shi 等人[Shi 2015]创新的提出将 CNN 和 LSTM 融合到一个端到端的深度学习结构中叫做 Conv-LSTM。Shi 等人用 Conv-LSTM 进行降雨量的预测,它为我们提供了一个新的解决时空序列预测问题的方法。

在本篇论文中,我们提出一个预测一个城市某个地区的打车需求预测方法。我们将一个城市划分成若干个更小的区域然后统计每个区域内一段时间内的打车量。通过这种方式,每个地区的历史打车数据变成了打车需求数量的数据序列。我们使用 ConvLSTM 训练历史数据获得时空依赖关系,并融合天气等外部因素到网络结构以提高预测的准确度。并且,我们引入了多任务学习的想法到我们的研究中,我们把多个任务整合在一起在模型的最顶层同时

进行训练,替代分别训练每个任务。多任务学习(MTL)能够充分利用网络中共享的权重去提供一个更好的预测结果。论文的贡献可以总结如下三个部分:

- 1) 我们提出一个基于 ConvLSTM 的深度学习框架去获取时空信息去进行打车需求预测。
- 2) 在交通系统中,同时学习多个任务能够获得更好的结果。我们提出一个融合多任务的框架。它显示出多任务预测结果好于单任务预测。
- 3) 实验结果显示,提出的模型性能好于已经存在的深度学习预测方法。

文章的剩余部分按如下方式组织,第二部分回顾打车预测任务中已经存在的方法和总结在相关问题中应用的深度学习方法。第三部分阐述研究的问题和简单解释自定义的变量。第四部分描述定义的网络结构,第五部分比较提出的方法和基准模型的性能,第六部分为总结和展望。

## Related Work

### A. Prediction Applications Using Past Taxi Data

研究人员在打车请求预测上做了一些研究。Zhao 等人[Zhao 2016]为打车需求预测提出了最大可预测性理论。他们通过展示历史打车需求序列的熵来证明打车需求是高度可预测的。他们使用了三个算法评估最大可预测性理论。相比于我们的方法,该方法只考虑了时间关系,并未考虑空间依赖关系。Davis 等人[Davis 2016]使用时间序列模型预测印度 Bengaluru 市的打车需求。需求信息被发送到司机的移动手机 APP 上以便于司机了解哪个地区的需求量较大。Zhang 等人[Zhang 2017]提出实时出租车分配的应用。在他们的系统中,两种类型的乘客被定义去模拟实时的打车需求:被遗留下来的乘客和即将到达的乘客。根据每辆出租车的 GPS 轨迹,遗留下来的乘客数量和即将到达的乘客数量通过自己设计的方法获得。它使用隐马尔科夫链模拟遗留下来的乘客和即将到达乘客之间的状态转换。同我们的方法相比,该方法并没有考虑天气,节假日等信息对打车需求的影响。Miao 等人[Miao 2016]提出一个车辆分配框架去平衡出租车的供应。Yuan 等人[Yuan 2013]提出一个能够帮助司机尽可能快速拉到乘客的算法。他们使用泊松分布去预测每个泊车地点能够拉到乘客的概率,选择概率最大的区域进行推荐。在他们的工作中,接乘客的位置都是固定的。我们的方法可以预测每个区域的打车请求。Wang 等人[Wang2015]研究向出租车司机推荐一组路线。他们使用人工构建的特征去训练单隐层神经网络。而我们的方法仅需要人为的构建少数几个特征。Moreira-Matias 等人[Moreira-Matias 2013]组合 Possion 模型和 ARMA 模型去预测每个出租车站点的打车需求。相比于我们的方法,他们使用统计模型预测打车请求,统计模型在处理时空非线性关系的数据时经常会失败。Chiang 等人[Chiang 2015]提出一个基于网格的高斯混合模型预测打车需求。在他们的方法中,总的订单数量由 Possion 模型提前确定。当实时的打车需求快速变化的时候,他们的方法会出现很大的误差。

### B. Deep Learning

最近,大量的研究人员开始使用深度学习方法去做预测问题[18]-[21]。Xu 等人[Xu 2017]提出使用 LSTM 去做一个长期的打车需求预测。但是他们的方法并没有很好的考虑空间依赖关系。Laptev 等人[Laptev 2017]等人提出使用 LSTM 预测极端情况下 Uber 的打车量。Lv 等人[Lv 2015]使用深度神经网络预测交通流。他们采用堆叠的自编码器逐层训练网络。方法显示他们的模型对比模型更准确。Wang 等人[Wang 2017]提出一个深度学习框架去预测 supply-demand。该方法能够自动的学习复杂的 supply-demand 模式并且仅需要构建少量的特征。但是,方法仅能够做短期的预测,不能进行长期预测。Huang 等人[Huang 2014]提出一个两层的深度学习结构进行交通流的预测。方法第一层是一个深度信念网络(Deep Belief Network, DBN),顶部是一个多任务回归层。相比于我们的方法,这个方法并没有考虑时间依赖关系。在交通预测领域,需要考虑时空依赖关系。因此,研究人员最近开始使用 CNN

和 LSTM 进行交通预测。Zhang 等人[Zhang 2016]提出一个 DeepST 的结构去预测交通流量。他们的模型能够通过卷积神经网络去学习时空关系。同时 Zhang 等人[Zhang 2016]提出一个基于残差网络的深度学习框架预测交通流量。他们的工作是基于经纬度将城市划分成一个  $I \times J$  的网格地图，每个网格代表一个区域。Shi 等人[Shi 2015]提出 Conv-LSTM 网络，他们将 CNN 和 LSTM 融合到一个网络结构中进行降雨量的预测。他们的实验结果显示，Conv-LSTM 性能超过了 fully-connected LSTM，因为 Conv-LSTM 能够学习到复杂的时空特性。

在我们的论文中，考虑到打车需求的预测不仅依赖时空关系也依赖其他变量的影响，我们对 Conv-LSTM 进行扩展，使他能够处理时空关系的同时也能够处理外部因素。

## Preliminary

在这个部分，我们会给出文章中变量的定义和打车需求预测问题的详细定义。

定义 1(地区和时间划分)：在我们的方法中，我们根据经纬度将城市的某块区域均匀的划分成  $I \times J$  个网格，每个网代表一块区域。同时，我们以 5 分钟为间隔统计一次打车请求数量。根据定义 1，我们定义若干个变量：

### (1) Taxi Demand

在第  $t$  个 timeslot 下，位于网格  $(I, J)$  内的打车请求量为在一个时间间隔内网格内订单数量，定义为  $d_t^{i,j}$ 。在第  $t$  个 timeslot 下，所有网格内的打车需求定义为矩阵  $D_t \in R^{I \times J}$ ,  $R$  代表真实的数据集。

我们收集了天气数据和 week of day, time of day 数据作为外部变量

### (2) Week of Day

因为在周末和工作日，打车需求会有不同的变化，因为我们定义了变量  $w_t$  表示 day-of-week.

$$w_t = \begin{cases} 0, & \text{if } t \text{ belongs to weekdays} \\ 1, & \text{if } t \text{ belongs to weekends} \end{cases}$$

### (3) Time of Day

通过对纽约出租车 24 小时的打车数据分析发现，在一天的不同时间打车需求会呈现不同的变化趋势，因此，我们将一天大致分为三个时间段，即 00:00-07:00, 7:00-17:00, 17:00-24:00，分别定义为 dawn time, daylight time, night time

$$v_t = \begin{cases} 0, & \text{if } t \text{ belong to dawn time,} \\ 1, & \text{if } t \text{ belong to daylight time,} \\ 2, & \text{if } t \text{ belong to night time} \end{cases}$$

### (4) Weather

我们考虑两种天气变量，包括 sky conditions 和 weather type, 其中 sky conditions 包括 CLR(clear sky), FEW(few clouds), SCT(scattered clouds), BKN(broken clouds), OVC(overcast) 等，weather type 包括 rain, snow, fog, thunderstorm。论文中，sky condidions 和 weather type 在  $t$ th time interval 表示为  $sc_t$  和  $wt_t$ 。

问题描述: 给定历史观测值  $\{D_s | s = 0, \dots, t-1\}$ , 预测  $D_t$ 。

## Method

文章中，我们提出一个创新的深度学习框架去捕捉影响打车预测结果的时空因素以及其他环境因素。

### Convolutional LSTM

ConvLSTM 组合了 CNN 和 LSTM，因此 ConvLSTM 网络能够提取时间信息和空间信息，并且 ConvLSTM 在降雨量预测上已经取得了非常好的效果。ConvLSTM 的核心思想是将所有的 inputs, memory cell values, hidden states 和不同的 gates 转化为 3D tensors。ConvLSTM 的主要等式如下：

$$\begin{aligned} i_t &= \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i) \\ f_t &= \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f) \\ C_t &= f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \\ o_t &= \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o) \\ H_t &= o_t \circ \tanh(C_t) \end{aligned}$$

其中， $*$  代表 convolution operator,  $\circ$  代表 Hadamard product. Input tensors, hidden tensors, memory cell tensors, input gate tensors, output gate tensors 和 forget gate tensors 分别表示为  $X_t, H_t, C_t, i_t, o_t, f_t \in R^{M \times N \times L}$ ,  $M, N$  表示网格的行数和列数。

通过  $T$  次迭代，每个 ConvLSTM 层可以把一个输入的张量序列  $X = (X_1, X_2, \dots, X_T)$  映射为隐藏的状态序列  $H = (H_1, H_2, \dots, H_T)$ 。本文中，每个 ConvLSTM 单元可以表示成一个函数  $F: R^{T \times M \times N \times L} \rightarrow R^{T \times M \times N \times L'}$ ，其中  $T$  代表时间序列长度， $M$  和  $N$  表示网格的行数和列数， $L$  代表通道数。我们通过堆叠多个 ConvLSTM 单元构成一个深度 ConvLSTM 网络。

### Deep Spatio-Temporal Convolutional Network

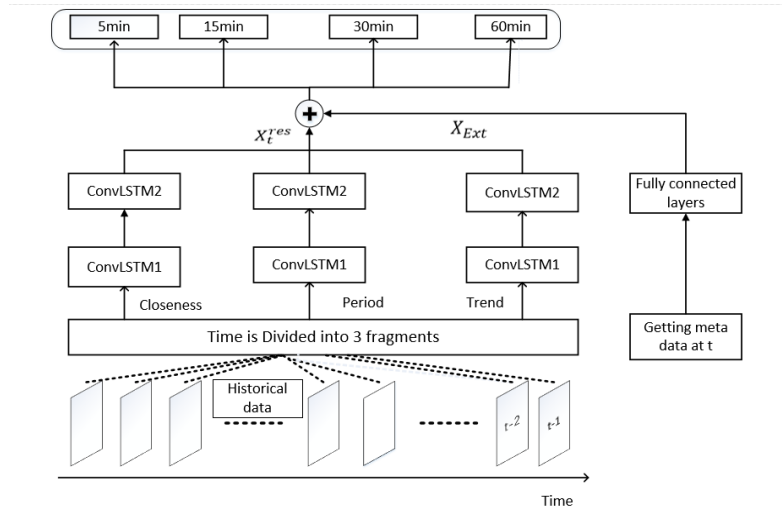


Figure 1: Framework of the proposed ConvLSTM approach

图中展示了 ST-ConvLSTM 的网络结构，网络由 temporal closeness, trend, period 和 external variable 四个模块构成。因为在一个 time interval 内 taxi demand 是一个 2 维矩阵，因此应用转换函数  $F^t: R^{M \times N} \rightarrow R^{M \times N \times 1}$  将输入的 2 维矩阵转为类似于图片的单通道 3 维张量。根据时间的性质，不同时间戳的数据被选择输入到网络中去模拟 closeness, period, trend。前三个模块共享相同的 convolutional LSTM 网络结构。这样的网络结构能够捕捉到时间和空间的依赖关系。在环境模块，我们首先获得  $t$  时刻的原始数据例如天气状况，工作日，然后把原始数据转换为二进制向量，最后将数据输入到一个两层的全连接网络中。前三个模块的输出经过早期融合为  $X_{res}$ ，然后  $X_{res}$  和外部模块的输出  $X_{ext}$  融合。因为随着时间的推移，打车请求是一个多任务的问题，不同的时间点对应不同的问题，因此，顶层是一个多任务学习层。

## Structure of the First Three Components

根据调研，时间序列一般具有下列三种性质：1) temporal closeness; 2) period; 3) trend. 基于这点, 我们从给定的历史记录中获得 recent, near 和 distant 时间戳的记录去模拟 temporal closeness, period, trend 三种性质。首先 closeness 部分被表示为  $[X_{t-l_c}, \dots, X_{t-1}]$ ,  $l_c$  代表时间戳的数量。然后，period 和 trend 部分被表示为  $[X_{t-l_p \cdot p}, X_{t-(l_p-1) \cdot p}, \dots, X_{t-p}]$  和  $[X_{t-l_s \cdot s}, X_{t-(l_s-1) \cdot s}, \dots, X_{t-s}]$ , 其中  $p$  表示 1 天,  $s$  表示 1 周。将三组数据放入不同的模块，每个模块都由堆叠的 conv-LSTM 层构成，其中  $L_d$  代表卷积层的数量，每个模块的输出分别表示为：

$$\begin{aligned} (H_{t-l_c}^{(L_d)}, H_{t-l_c+1}^{(L_d)}, \dots, H_{t-1}^{(L_d)}) &= F_{L_d}^c \dots F_l^c \dots F_1^c(X_{t-l_c}, X_{t-l_c+1}, \dots, X_{t-1}) \\ X_t^c &= f(H_{t-1}^{(L_d)}) \\ (H_{t-l_p \cdot p}^{(L_d)}, H_{t-(l_p-1) \cdot p}^{(L_d)}, \dots, H_{t-p}^{(L_d)}) &= F_{L_d}^p \dots F_l^p \dots F_1^p(X_{t-l_p \cdot p}, X_{t-(l_p-1) \cdot p}, \dots, X_{t-p}) \\ X_t^p &= f(H_{t-p}^{(L_d)}) \\ (H_{t-l_s \cdot s}^{(L_d)}, H_{t-(l_s-1) \cdot s}^{(L_d)}, \dots, H_{t-s}^{(L_d)}) &= F_{L_d}^t \dots F_l^t \dots F_1^t(X_{t-l_s \cdot s}, X_{t-(l_s-1) \cdot s}, \dots, X_{t-s}) \\ X_t^s &= f(H_{t-s}^{(L_d)}) \end{aligned}$$

其中  $F$  代表 ConvLSTM 操作,  $f$  表示  $relu$  激活函数,  $f(z) = \max(0, z)$ ,  $X_t^c, X_t^p, X_t^s$  分别为 closeness, period, trend 模块顶层的输出。

## Structure for no-spatial time-series variables

打车请求数量受到许多复杂的外部因素的影响，例如，天气和时间。Figure(a)显示了一天的打车需求变化，我们发现打车需求量大的主要有两个时间段，即 7:00-17:00 和下午 19:00-24:00，其中，7:00-17:00 显示乘客白天的正常打车需求，19:00-24:00 这个时间段属于下班时间，所以打车请求达到一个峰值。Figure(b) 展示了一周的打车需求变化，图中能够看到工作日和周末的打车需求有非常大的区别。Figure(c)展示了异常天气对打车需求的影响。由于大雨的出现，那个时间段的打车量同平常比较多。因此，我们定义  $E_t$  表示在预测的 time interval  $t$  时的外部变量，我们主要考虑 day of week, time of day, sky conditions 和 weather type。然而，预测的时段  $t$  的天气我们是不知道的，因此我们用  $t-1$  时间段的天气代替  $t$  时段的天气。外部模块主要由两个全连接层构成，全连接层的作用是将输入由低维空间映射到高维空间使输出的形状和  $X_t$  相同，输出表示为  $X_{Ext}$ 。

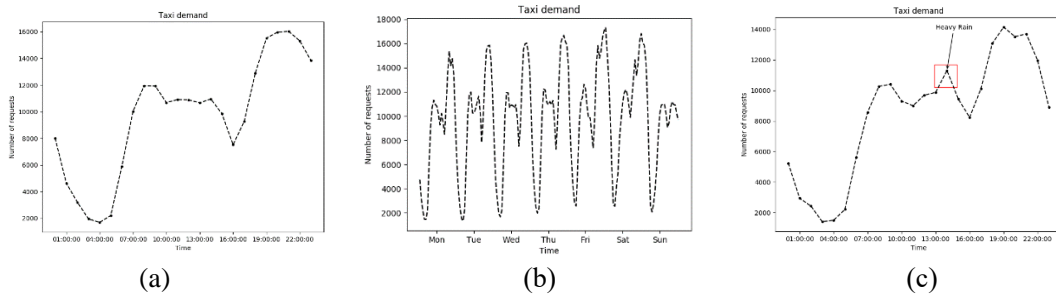


Figure 2: Temporal Dependences

## Fushion

根据融合的时间，我们的模型分为早期融合和后期融。早期融合时基于参数矩阵的方法对 closeness, period, trend 三个模块进行融合，融合方式如下：

$$X_t^{res} = W_c \circ X_t^c + W_p \circ X_t^p + W_s \circ X_t^s$$

其中， $\circ$  代表 Hadamard product,  $W_c, W_p, W_s$  表示为三个模块分配的不同权重。

后期融合是将三个模块的输出和外部模块的输出进行融合，最后在  $t$ th 时间段的融合后的输出表示为：

$$\widehat{X}_t = \tanh(X_t^{res} + X_{Ext})$$

其中,  $\tanh$  一个双曲正切函数, 保证输出的值在-1 到 1 之间。

## Top Layers with Multi-task learning

多任务学习是基于共享表示, 把多个任务放在一起学习的机器学习方法。多任务学习涉及多个相关的任务同时并行学习, 梯度同时反向传播, 多个任务通过底层的共享表示来互相帮助学习, 提高泛化性能。多任务学习第一次被(Huang et al 2014)引入进行交通预测。随着时间的推移, 未来交通状况的预测是一个多任务的问题, 同理打车请求也是一个多任务预测问题, 在不同的时间点对应不同的任务。在我们的深度学习结构中, 我们把相关的多任务预测放在顶部的回归层。这些任务一起训练并通过后向传播调节参数, 因此我们相信每个任务会得到一个好的结果。我们选择均方误差作为损失函数。

## Experiments

本部分我们使用纽约的出租车数据评估提出的 ConvLSTM-MTL 模型。此外, 我们把提出的模型和其他的 5 个基础模型进行对比。

### A. Experimental Setup

我们使用纽约的出租车数据集[TaxiNY]评估我们提出的网络模型的性能。在纽约有两种出租车, 分别为黄色出租车和绿色出租车。黄色出租车主要在曼哈顿区搭载乘客, 绿色出租车的营运地点主要在郊区。数据集包含 2009 年 1 月到 2016 年 6 月黄色出租车和绿色出租车的 GPS 轨迹。每条轨迹都包含上车地点和时间。在我们论文中, 我们使用的是 2015 年 1 月到 2015 年 6 月的黄色出租车的打车数据, 包含 6000 万条数据。我们选择最后一周的数据作为测试数据, 那之前的数据作为训练数据。取样的时间间隔为 5 分钟, 网格的大小为 10\*10。Baselines:我们把 ConvLSTM-MTL 和下列的 5 个基准模型对比:

LSTM[23]: 长短记忆网络, 是一种特殊的 RNN, 它有能力学习长期的依赖信息。其中, 网络的层数为 4 层, 每个模块的序列长度分别为  $l_c = 12, l_p = 1, l_s = 1$ 。

GRU[24]:GRU 是一种的新的 RNN, 也是被用来解决长期依赖问题,同样采用和 LSTM 相同的设置。

CNN+LSTM: CNN[25]是卷积神经网络, 用来提取空间信息, LSTM 用来提取时间信息, 将二者组合在一起提取时空信息, 网络的层数为 4 层, 卷积层卷积的尺寸为 3\*3, LSTM 层的输入序列长度分别设置为  $l_c = 12, l_p = 1, l_s = 1$ 。

DeepST[16]:一种基于深度神经网络的时空数据预测模型, 在交通流预测上展示了较好的性能。

网络的层数为 2 层, 卷积核的尺寸为 3\*3

ST-ResNet[1]:一种基于深度残差网络的时空数据预测模型, 在交通流预测任务上展示了好的性能。网络的层数为 2 层, 残差单元的数量为 12。卷积核的尺寸为 3\*3。

**Preprocessing:**在我们的模型输出上, 我们选择  $\tanh$  作为最后的激活函数,  $\tanh$  的取值范围是[-1,1]。我们使用 Min-Max 的方法规范化数据, 规范化后的数据取值范围是[-1,1]。在最后的实验评估中, 我们将数据恢复到原来的标准, 然后和真实的值进行比较。对于外部变量, 我们采用 one-hot 的编码方式将 time of day, day of week, weather type 和 sky conditions 转换为二进制向量。

**Hyperparameters:**网络中可学习的参数使用 Keras 默认的均匀分布对参数进行初始化。ConvLSTM1 的卷积核尺寸是 3\*3, 卷积核的数量是 64。ConvLSTM2 的卷积核尺寸是 3\*3, 卷积核的数量是 1。Adam[]被选作优化算法, batchsize 是 24。在 ConvLSTM-MTL 模型中有 5 个超参数, 其中 p 和 s 被设为一天和一周。对于三个独立的序列长度, 设置如下:  $l_c \in \{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15\}$ ,  $l_p \in \{1,2,3,4,5\}$ ,  $l_s \in \{1,2,3,4,5\}$ 。我们选择 90%的数据作为训练集, 剩余 10%的数据作为验证集, 然后我们根据验证集的最好评分来决定提前停止训

练。最后，我们设定固定数量的 epochs 来训练数据，在这里 epoch 数量设置为 500。

Layer name	Output size	Closeness	Period	Trend
ConvLSTM1	$\{l_c, l_p, l_s\} \times 10 \times 10$	3×3, 64	3×3, 64	3×3, 64
ConvLSTM2	10 × 10	3×3, 1	3×3, 1	3×3, 1

Table 1 : Details of ConvLSTM

Evaluation Metric:我们选择 Root Mean Square Error 作为评估标准：

$$\text{RMSE} = \sqrt{\frac{1}{m} \sum_i (x_i - \hat{x}_i)^2}$$

其中，x是真实值， $\hat{x}$ 是预测值，m 是数据的数量。

## Comparison with Other Methods

本部分我们使用历史记录预测在 time interval t 的打车请求。表 1 显示在纽约出租车数据集上所有方法的 RMSE。从表中能够看到，ConvLSTM-MTL 性能超过了其他的方法。我们的方法的实现了多任务预测，在 5 min 的任务预测上，相比于其他方法，性能提升 13%-25 左右，15 min 的任务预测上，性能提升 20%-30%左右，30min 任务上性能提升 22%-35%，60 min 的任务预测上性能提升 20%-35%左右。从表中能够看到，CNN 和 LSTM 的组合模型性能要好于其他模型，说明 CNN 和 LSTM 的组合既能提取空间信息又能提取时间信息。而其他四个模型只能获得时间信息或者空间信息，因此预测的误差大于 CNN 和 LSTM 的组合。我们的方法好于 CNN 和 LSTM 的组合是因为 ConvLSTM 能够提取到更复杂的时空信息，因此性能由于 CNN 和 LSTM 的组合。图 3 表示对模型的结果进行了排序得到的柱状图。

Model	RMSE			
	5 min	15min	30min	60 min
<b>CNN+LSTM</b>	3.76	9.02	16.26	32.44
<b>GRU</b>	4.39	9.82	18.19	36.10
<b>LSTM</b>	3.98	9.62	17.78	34.89
<b>DeepST</b>	4.22	10.47	19.67	39.19
<b>STResNet</b>	4.16	10.34	19.29	38.12
<b>ConvLSTM-MTL</b>	3.30	7.21	12.69	25.54
<b>ConvLSTM-MTL-noEXT</b>	3.32	7.28	12.71	25.79

Table 2: Models performance comparison at various future time points

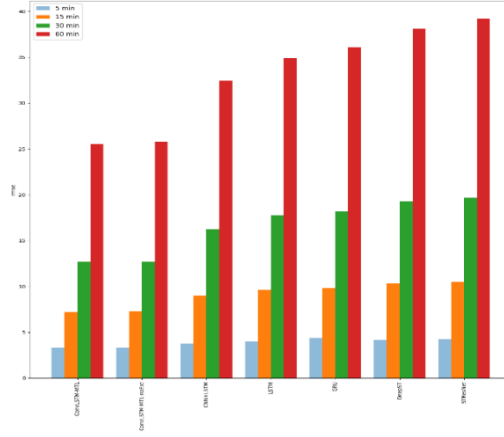


Figure 3: Model ranking on TaxiNY. The smaller the better

Figure 4 展示了本方法和使用 CNN 和 LSTM 组合的方法预测曲线，因为 CNN+LSTM 是对比模型中性能最好的一个模型，因此绘制了 CNN+LSTM 模型和我们方法对需求的预测曲线。其中，我们对自己划分的网格区域进行了编号，然后绘制了 6 号区域和 15 号区域的预测曲线。其中 Figure 4(a)和 Figure 4(b) 展示了 6 号区域和 15 号区域的预测窗口大小为 30min 时的需求变化曲线。Figure 4 (c)和 Figure 4 (d)展示了 6 号区域和 15 号区域的预测窗口大小为 60min 时的需求变化曲线。首先，从图中我们能够得到 6 号区域和 15 号区域打车需求有不同的变化趋势。我们方法的预测曲线基本能和真实变化曲线重合，这说明我们的方法对需求的变化趋势能够很好的预测出来并且预测的准确度高于 CNN+LSTM 方法。

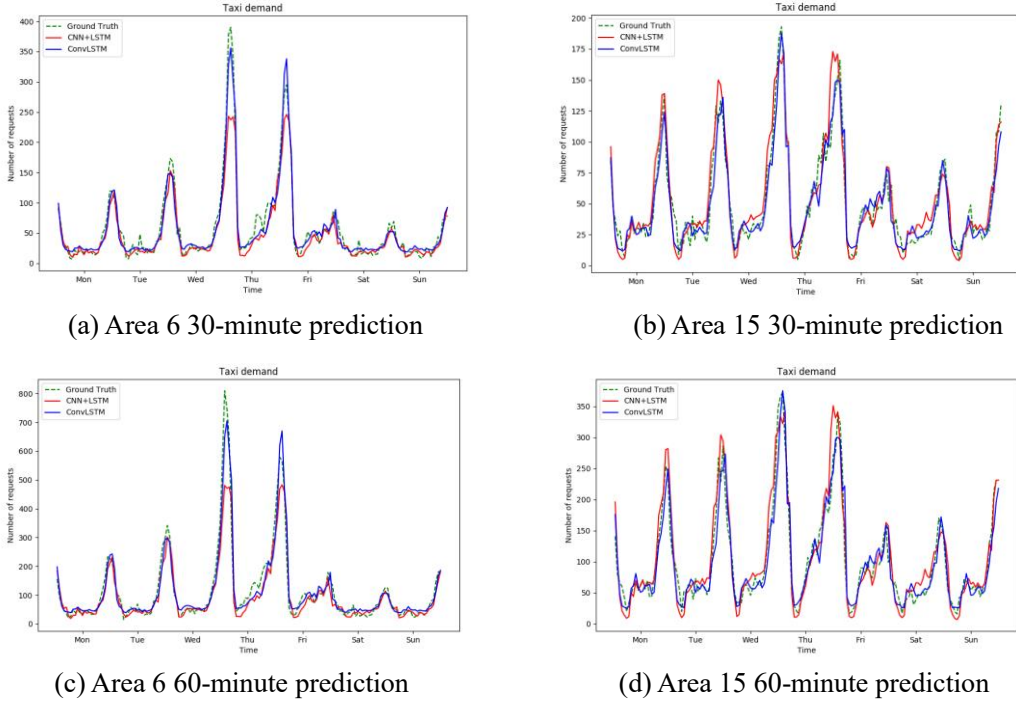


Figure 4: Comparson of ConvLSTM and CNN+LSTM

### Impact of temporal closeness, period, trend

我们使用出租车数据集验证 closeness, period, trend 三个模块对实验结果的影响，见图[]。图 1 (a)展示，当固定 $l_p=1$  和 $l_s=1$  时，改变 $l_c$ ,closeness 模块对实验结果的影响。从图中看到，



随着 $l_c$ 长度的增加, 对 5min, 15min, 30min 任务的影响很小, 但是对 60min 任务的影响较大, 当 $l_c=12$  时, 性能最好。这是因为对于长期的预测任务需要学习较多的信息才能对预测结果有帮助, 而短期预测任务仅需要距离当前时刻较近历史记录就能取得好的预测结果。同理, 图 4(b)表示的是固定 $l_c = 12$ 和 $l_s=1$ , 改变 $l_p$ 。图中看到, 当 $l_p=1$  时, 有最好的性能, 当随着 $l_p$ 继续增加, RMSE 几乎没变化。这意味着短距离的周期模块对模型的性能有帮助, 长距离的周期模块对模型没有帮助。图 4 (c)表示的是固定 $l_c = 12$ 和 $l_p=1$ , 改变 $l_s$ 。同样, 在 $l_s=1$  时取得最好的结果, 然后随着 $l_s$ 的增加, RMSE 几乎无变化。

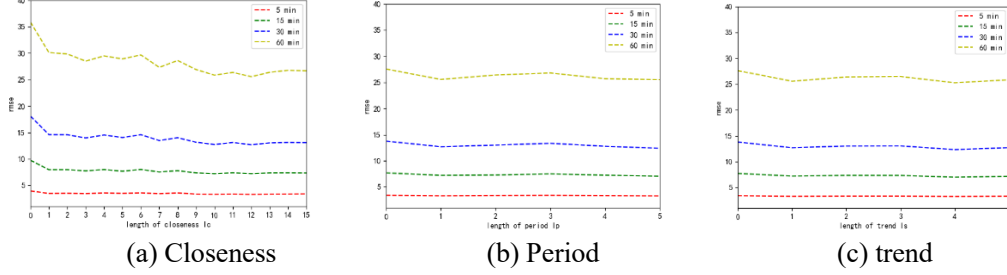


Figure 5: Impact of temporal closeness, period, trend

### Impact of filter size and number

卷积的感知域是由使用的卷积核的尺寸决定的。我们将卷积核的尺寸从  $2 \times 2$  变到  $5 \times 5$ 。图 5 (a) 显示卷积和尺寸越大, RMSE 越小, 这是因为大的感知域有更强的能力去模拟空间依赖关系。图 5(b)展示卷积核数量越多 RMSE 越小。

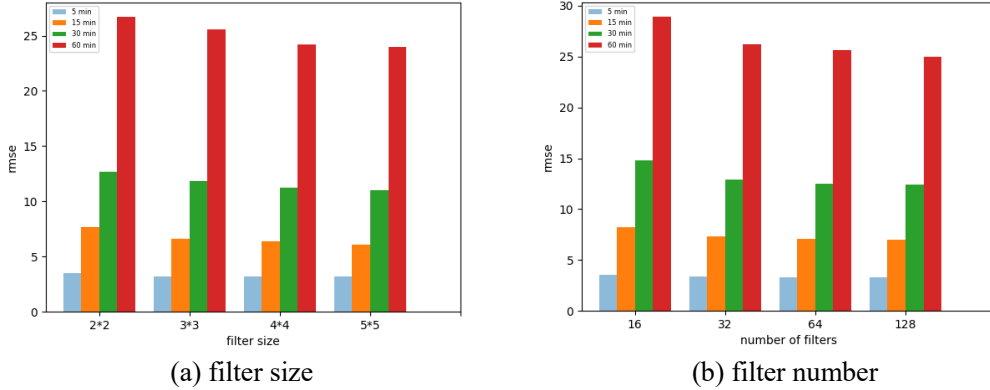


Figure 6: Impact of filter size and number

### Comparison between Multi task Learning and Single task Learning

在这个部分, 我们评估多任务学习 (MTL) 的有效性。在同一个深度学习模型框架下, 做单任务学习和多任务学习的对比。

首先, 在深度学习网络顶层训练所有任务, 然后和分别训练每个任务得到的结果进行对比。从表 3 中, 我们能够看到, 同时训练所有任务能够降低所有任务的误差。相比于单任务, 5min 预测的误差降低了 8%左右, 15min 预测的误差降低了 2%左右, 30min 预测的误差降低 7%左右, 60min 预测上误差降低了 18.5%, 因此我们得到多任务预测对长期预测的效果更好。同时训练所有任务是最方便快捷的方法因为它只需要训练一次而不需要分别对每个任务进行训练。在这里, 我们要预测的未来 5min, 15min, 30 min, 60 min 的打车请求, 这些任务之间是相关的, 因此对打车请求的预测结果是有促进作用的。图 7 是单任务和多任务的对比柱状图, 从中可以清晰的看到多任务的性能由于单任务。

Task	RMSE			
	5min	15min	30min	60min
STL	3.60	7.21	13.51	31.51
MTL	3.30	7.06	12.50	25.67

Table 3: performance comparison between STL and MTL

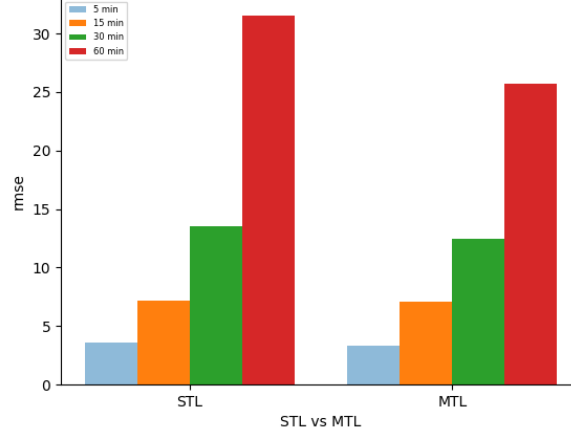


Figure 7: performance comparison between STL and MTL

## Conclusions

在论文中，我们提出了一个基于 ConvLSTM 的深度学习方法进行打车需求的预测。我们提出的框架由多个 Conv-LSTM 层构成并且融合了天气，节假日等外部变量。在网络的顶层加入了多任务回归层实现多任务学习。纽约 6 个月的出租车数据被用来训练我们的模型。实验结果显示，我们提出的方法性能超过了 5 个对比模型，并且引入了多任务学习后的性能也超过单独训练每个模型的性能。

将来，我们可以添加更多的信息到网络中，比如，商业区，饭店，商店等信息融合到网络中。并且我们的网络也可以迁移到其他时空预测领域，比如交通流预测，通勤时间预测等等。

## References

- [1][Zhang 2016] J. Zhang, Z. Yu and D. Qi. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction, Association for the Advance of Artificial Intelligence, 2017.
- [2][Wu 2016] Wu, Y., Tan, H., 2016. Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework. arXiv preprint arXiv:1612.01022.
- [3][Shi 2015] Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c., 2015. Convolutional lstm network: A machine learning approach for precipitation nowcasting, in: Advances in Neural Information Processing Systems, pp. 802–810.
- [4][Zhao 2016] K. Zhao, D. Khryashchev, J. Freire, C. Silva, and H. Vo, “Predicting taxi demand at high spatial resolution: Approaching the limit of predictability,” in Proc. IEEE BigData, Dec. 2016, pp. 833–842.
- [5][Davis 2016] N. Davis, G. Raina, and K. Jagannathan, “A multi-level clustering approach for forecasting taxi travel demand,” in Proc. IEEE ITSC, Dec. 2016, pp. 223–228.
- [6][Zhang 2017] D. Zhang, T. He, S. Lin, S. Munir, and J. A. Stankovic, “Taxi-passenger demand modeling based on big data from a roving sensor network,” IEEE Trans. Big Data, vol. 3, no. 1, pp.

362–374, Sep. 2017.

[7][Miao 2016] F. Miao et al., “Taxi dispatch with real-time sensing data in metropolitan areas: A receding horizon control approach,” *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 463–478, Apr. 2016.

[8][Yuan 2013] N. J. Yuan, Y. Zheng, L. Zhang, and X. Xie, “T-finder: A recommender system for finding passengers and vacant taxis,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 10, pp. 2390–2403, 2013.

[9][Wang 2015] R. Wang, C.-Y. Chow, Y. Lyu, V. Lee, S. Kwong, Y. Li, and J. Zeng, “Taxirec: recommending road clusters to taxi drivers using ranking-based extreme learning machines,” in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2015, p. 53.

[10][Moreira-Matias 2013] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas, “Predicting taxi–passenger demand using streaming data,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1393–1402, 2013.

[11][Chiang 2015] M. F. Chiang, T. A. Hoang, and E. P. Lim, “Where are the passengers? : a grid-based gaussian mixture model for taxi bookings,” in *Sigspatial International Conference on Advances in Geographic Information Systems*, 2015.

[12][Xu 2017] Jun Xu, Rouhollah Rahmatizadeh, Ladislau Bölöni. Real-Time Prediction of Taxi Demand Using Recurrent Neural Networks, *IEEE Transactions on Intelligent Transportation Systems*, 2017

[13][Laptev 2017] N. Laptev, Jason Yosinski, Li Erran Li, Slawek Smyl, Time-series Extreme Event Forecasting with Neural Networks at Uber, *International Conference on Machine Learning*, 2017.

[14][Wang 2017] Dong Wang, Wei Cao, Jian Li, Jieping Ye, DeepSD: supply-demand prediction for online car-hailing services using deep neural networks, *International Conference on Data Engineering*, 2017.

[15][Huang 2014] Huang, W., Song, G., Hong, H., Xie, K., 2014. Deep architecture for traffic flow prediction: Deep belief networks with multitask learning. *IEEE Transactions on Intelligent Transportation Systems* 15, 2191–2201.

[16][Zhang 2016] Zhang, J.; Zheng, Y.; Qi, D.; Li, R.;and Yi, X. . Dnn-based prediction model for spatiotemporal data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 92. ACM.

[17][NY dataset] NYC Taxi & Limousine Commission. Taxi and Limousine Commission (TLC) Trip Record Data. Accessed: Dec. 2016. [Online]. Available:

[http://www.nyc.gov/html/tlc/html/about/trip\\_record\\_data.shtml](http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml)

[18][Grover 2015] A. Grover, A. Kapoor, and E. Horvitz, “A deep hybrid model for weather forecasting,” in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 379–386.

[19][Wu 2016] S. Wu, W. Ren, C. Yu, G. Chen, D. Zhang, and J. Zhu, “Personal recommendation using deep recurrent neural networks in netease,” in *32nd IEEE International Conference on Data Engineering, ICDE 2016, Helsinki, Finland, May 16-20, 2016*, 2016, pp. 1218–1229.

[20][Ding 2015] X. Ding, Y. Zhang, T. Liu, and J. Duan, “Deep learning for eventdriven stock prediction,” in *Proceedings of the 24th International Joint Conference on Artificial Intelligence (ICJAI15)*, 2015, pp. 2327–2333.

[21][Mocanu 2016] E. Mocanu, P. H. Nguyen, M. Gibescu, E. M. Larsen, and P. Pinson, “Demand forecasting at low aggregation levels using factored conditional restricted boltzmann machine,” in *2016 Power Systems Computation Conference (PSCC)*, June 2016, pp. 1–7.

[22][Lv 2015] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, “Traffic flow prediction with big data: a deep learning approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

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

[24][Cho 2014] K.Cho,B. van Merriënboer,C. Gulcehre, D.Bahdanau, F.Bougares, H.Schwenk, Y. Bengio, Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation(2014):1724-1734.

[25][LeCun 1995] LeCun, Y.; Bengio, Y.; et al. 1995. Convolutional networks for images, speech,

and time series. The handbook of brain theory and neural networks 3361(10):1995.