

PYCON  
CHINA

2019

基于深度学习的多维时间序列  
预测在数据机房中的应用

浙江省邮电工程建设有限公司 大数据研究院

演讲人：杨赛赛 时间：10.19

# 目 录

1

背景介绍

2

研究目标

3

研究内容

4

后续工作

1.

## 背景介绍



# 数据机房面临的能耗问题

数据机房面临电量消耗巨大的问题

2017年，全国数据中心耗电量高达1200亿千瓦时，远超三峡电站全年发电量，且仍在急速攀升；高能耗给企业带来了沉重的运营负担，也造成了全社会能源的巨大浪费，部分电力负荷中心城市因此纷纷出台“限制”政策，禁止新建、扩建数据中心

## 数据中心怎成“能耗巨兽”

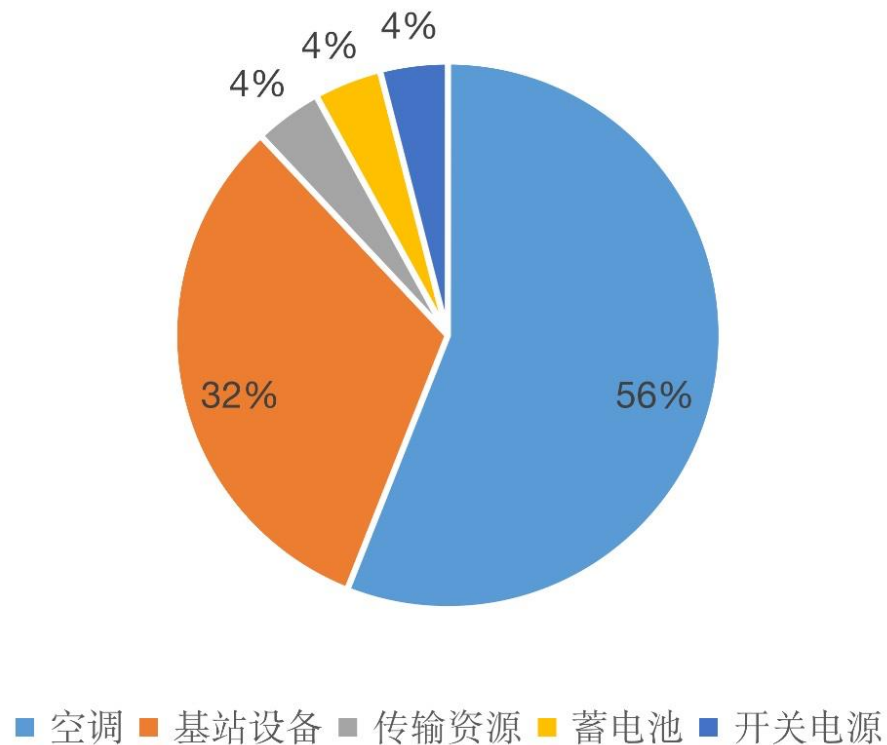
2019-07-24 10:34 · 来源：中国能源报-中国能源网 · 作者：朱妍 姚金楠 · 责编：刘澄谚  
数据中心;能耗;服务器

中国能源报-中国能源网 | 位于冀、蒙交界处的张北，一个充满古韵的高原县城，如今已成为拥有60万台服务器的世界级数据中心栖息地——阿里巴巴云数据中心便落户于此。

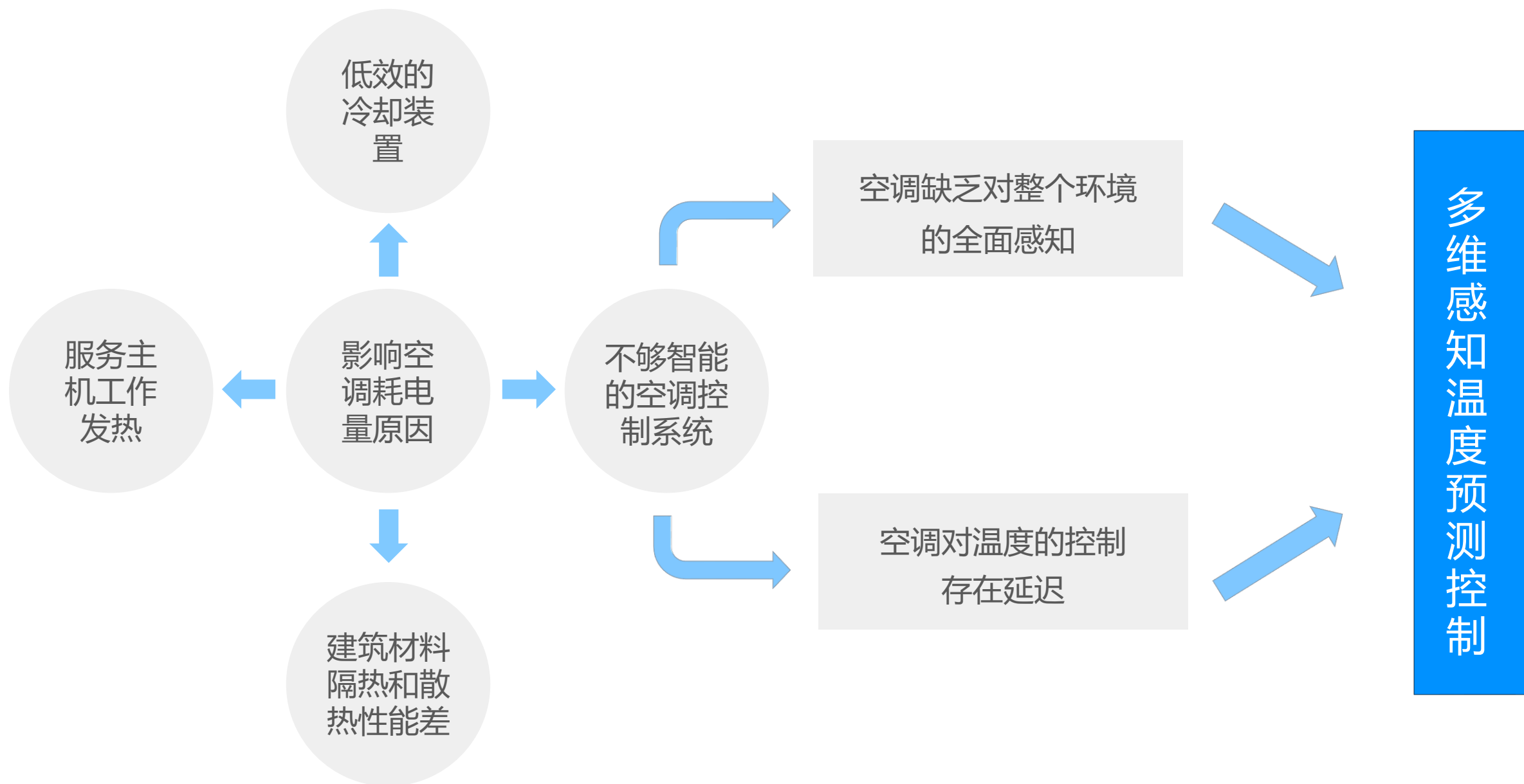
数千公里之外的贵州，群山峻岭之间，分布着苹果、华为、腾讯等多家公司的数据中心。无论身处何地，你手机云端的相片、朋友圈的视频、网盘里的资料，或许都存储于此。

空调是数据机房中电量消耗最大的设备

数据中心耗电量统计



# 空调为什么那么耗电？怎么优化节能？



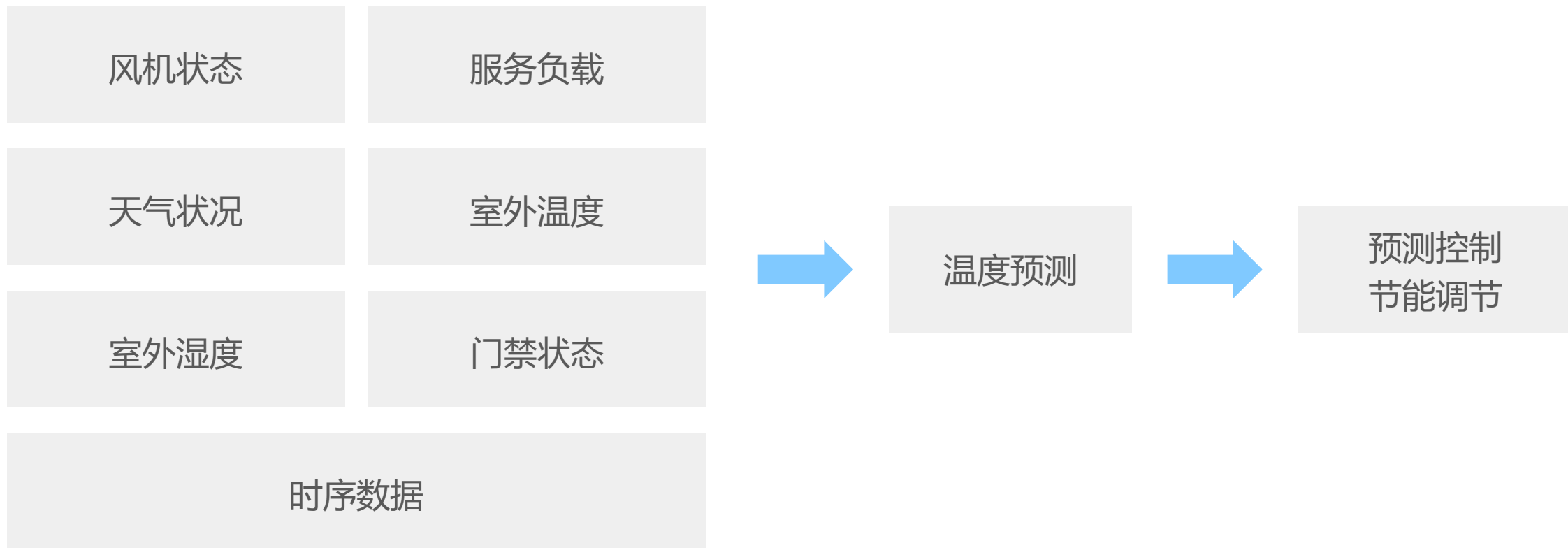
2.

## 研究目标



## 对数据机房的温度进行预测

- 根据机房的历史运行数据变化预测未来 XX 分钟机房的温度值，从而实现空调的预测控制。



**3.**

## 研究内容





# 多维时间序列预测方法解决机房温度预测

## • 时间序列预测方法的比较

### 传统时间序列预测

- 对单个维度历史信息进行预测
- 捕获简单线性关系，模型简单
- 代表算法有AR, ARIMA

### 基于深度学习的 时间序列预测

- 利用多维时间序列之间的信息
- 对变周期序列，多维空间依赖序列预测较弱
- 代表算法有RNN, LSTM

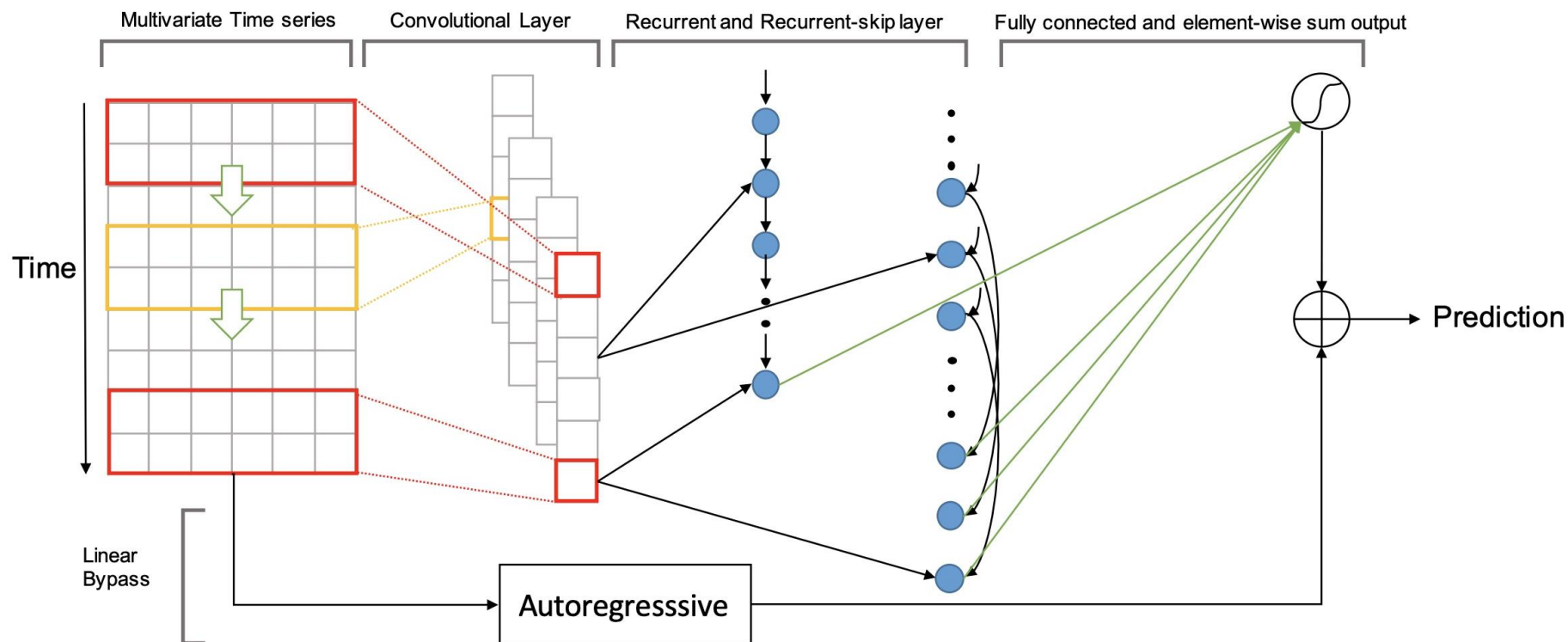
### 混合多维时间序列预测

- 提取多维序列之间更加复杂的关系
- 提取维度之间空间依赖关系，长短期依赖关系
- 算法有LSTNet, TPA-LSTM

对数据包含的信息提取能力越来越强



## 选择 LSTNet 作为温度预测建模算法



- Convolutional Layer 捕捉时间维度上的短期依赖和维度之间的空间依赖关系
- Recurrent and Recurrent-skip layer 捕捉长期宏观依赖和周期性信息
- Autoregressive 叠加线性比例关系

Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks

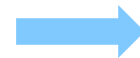
[Guokun Lai](#), [Wei-Cheng Chang](#), [Yiming Yang](#), [Hanxiao Liu](#)

# LSTNet Python 代码实现

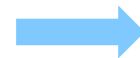
```
def LSTNetModel(init, input_shape):  
    # m is the number of time-series  
    m = input_shape[2]  
  
    # Get tensor shape except batchsize  
    tensor_shape = input_shape[1:]  
  
    X = Input(shape=tensor_shape)  
  
    # CNN  
    # Add an extra dimension of size 1 which is the channel dimension in Conv2D  
    C = Reshape((input_shape[1], input_shape[2], 1))(X)  
    # Apply a Conv2D that will transform it into data of dimensions (batchsize, time, 1, NumofFilters)  
    C = Conv2D(filters=init.CNNFilters, kernel_size=(init.CNNKernel, m), kernel_initializer=init.initialiser)(C)  
    C = Dropout(init.dropout)(C)  
    # Adjust data dimensions by removing axis=2 which is always equal to 1  
    c_shape = K.int_shape(C)  
    C = Reshape((c_shape[1], c_shape[3]))(C)  
  
    # GRU  
    # Apply a GRU layer (with activation set to 'relu' as per the paper) and take the returned states as result  
    R = GRU(init.GRUUnits, activation="relu", return_sequences=False, return_state=True)(C)  
    R = Dropout(init.dropout)(R)  
  
    # SkipGRU  
    # Calculate the number of values to use which is equal to the window divided by how many time values to skip  
    pt = int(init.window / init.skip)  
  
    S = PreSkipTrans(pt, int((init.window - init.CNNKernel + 1) / pt))(C)  
    _, S = GRU(init.SkipGRUUnits, activation="relu", return_sequences=False, return_state=True)(S)  
    S = PostSkipTrans(int((init.window - init.CNNKernel + 1) / pt))([S, X])  
  
    # Concatenate the outputs of GRU and SkipGRU  
    R = Concatenate(axis=1)([R, S])  
  
    # Dense layer  
    Y = Flatten()(R)  
    Y = Dense(1)(Y)  
  
    # AR  
    Z = PreARTrans(init.highway)(X)  
    Z = PostARTrans(m)([Z, X])  
  
    # Generate output as the summation of the Dense layer output and the AR one  
    Y = Add()([Y, Z])  
  
    # Generate Model  
    model = Model(inputs=X, outputs=Y)  
  
    return model
```



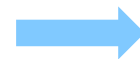
Convolutional Layer



Recurrent and Recurrent-skip layer



Fully Connected Layer



Autoregressive

# LSTNet 模型的训练

## 优化算法: Adam

同时具有动量更新和自适应调整学习速率, 占用内存少。

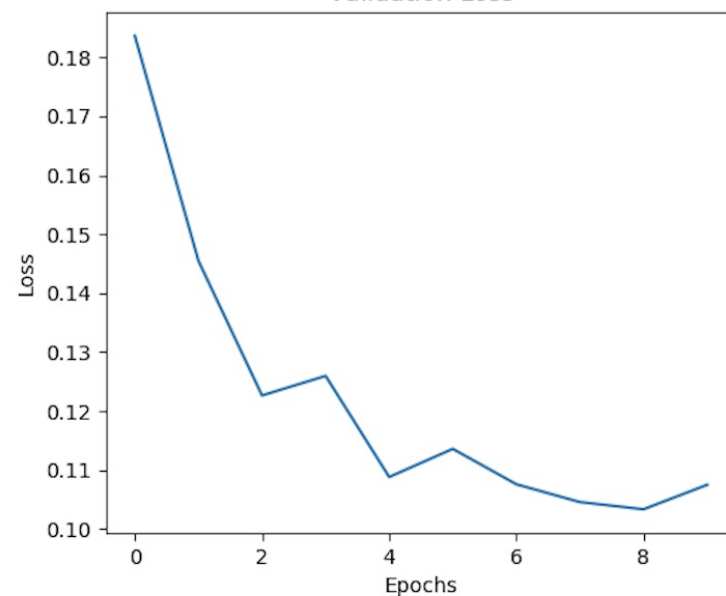
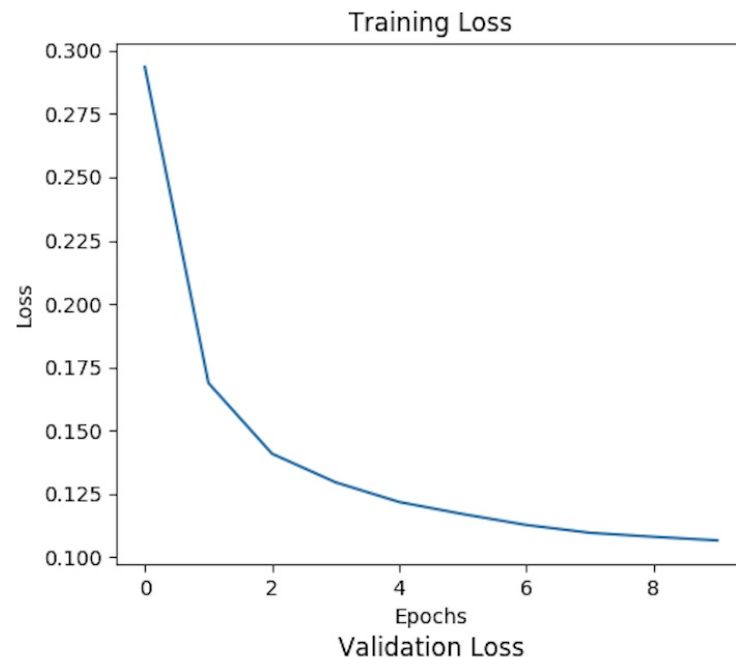
## 损失函数: Logcosh

Logcosh是预测误差的双曲余弦的对数。不会受到偶尔出现的极端不正确预测的强烈影响, 同时收敛速度快。

## 评价指标: RMSE和 $R^2$

RMSE: 预测值与真实值的误差平方根的均值

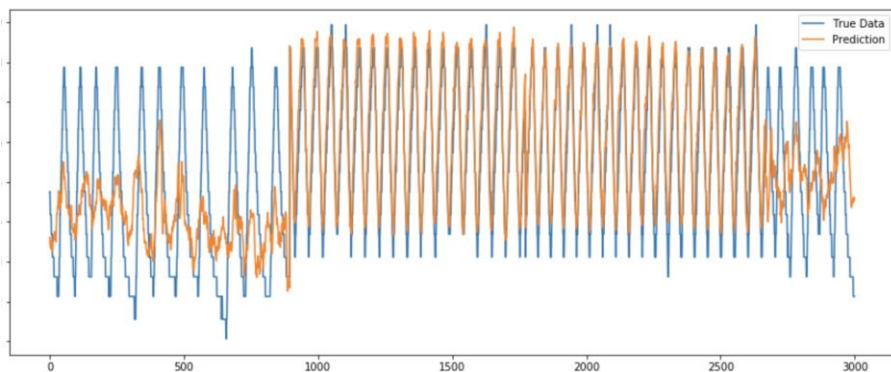
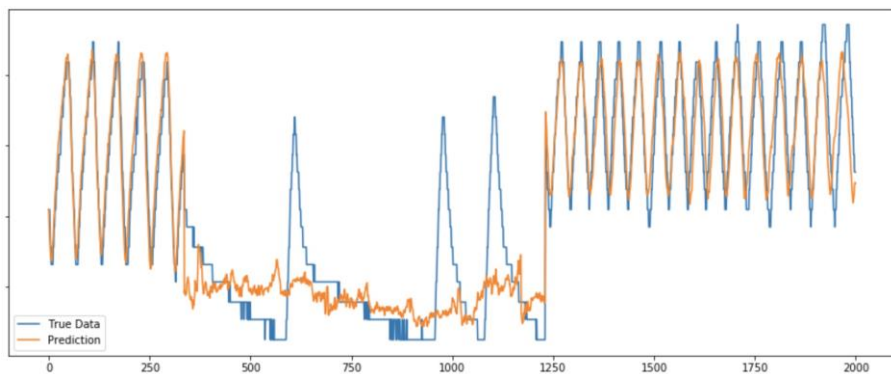
$R^2$ : 范围为0-1, 越接近1, 表明这个模型对数据拟合能力越好。



# LSTNet 预测效果展示

## LSTNet 算法实验结果及与 LSTM 算法的对比

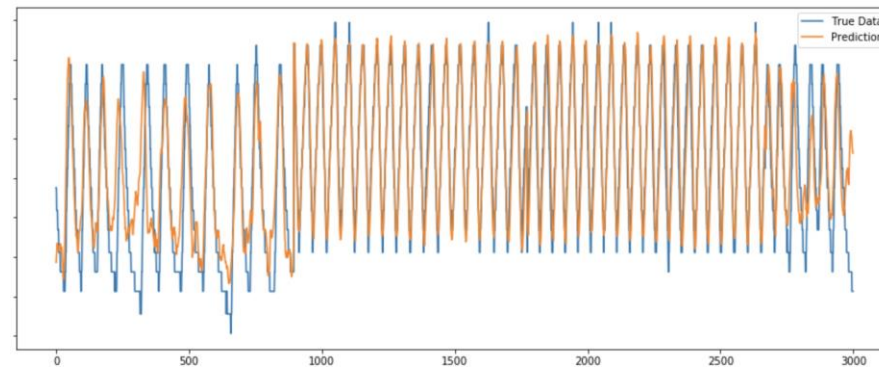
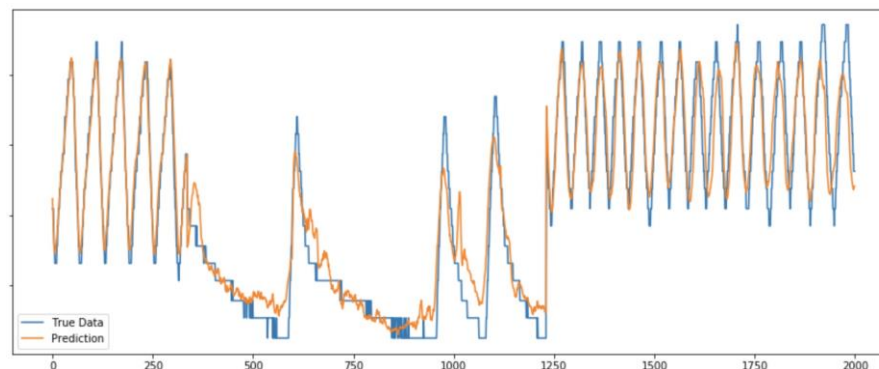
测试结果: RMSE: 0.3度  $R^2$ : 90%



LSTM

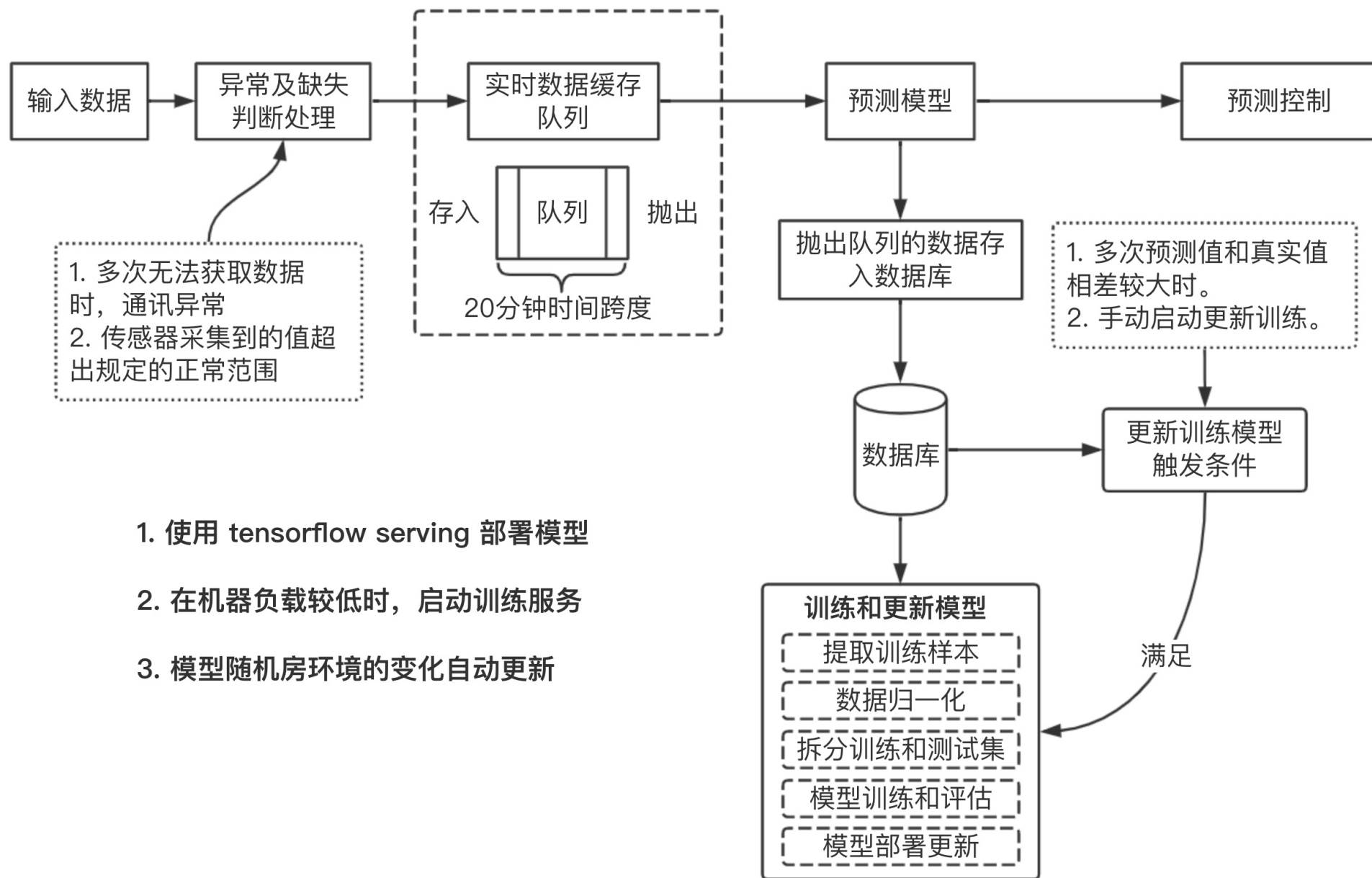
- 橙色线代表模型预测温度值
- 蓝色线代表真实温度值

测试结果: RMSE: 0.25度  $R^2$ : 97%



LSTNet

## 模型部署和自动更新



1. 使用 tensorflow serving 部署模型
2. 在机器负载较低时，启动训练服务
3. 模型随机房环境的变化自动更新

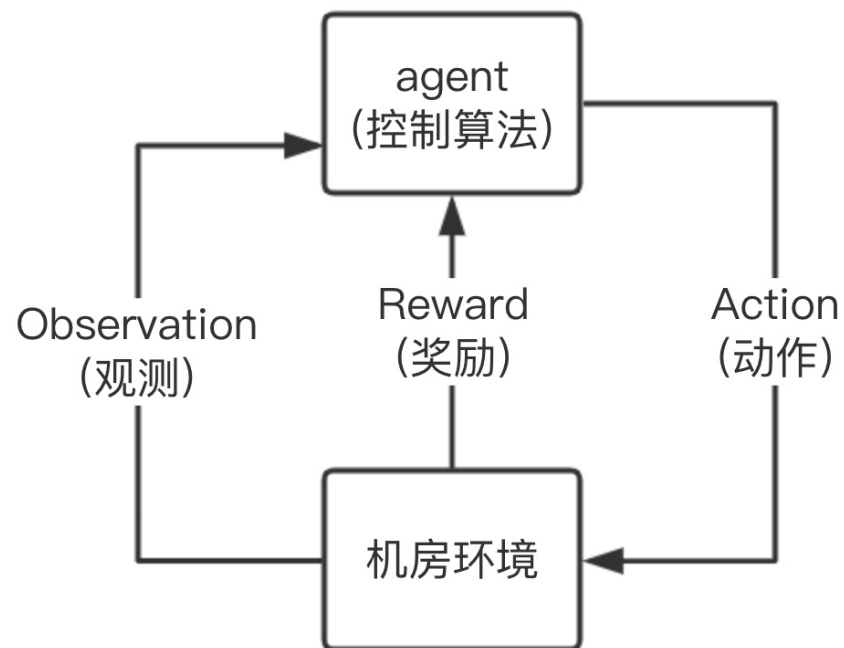
4.

## 后续工作



## 结合温度预测模型对空调进行节能控制

- 利用温度预测模型实现强化学习节能控制
  - 强化学习探索策略的制定
  - 强化学习模拟实验环境





THANK  
YOU 2019

项目数据及源代码地址:

<http://uee.me/cu9GV>

momodel.ai

