

大数据计算基础课程报告

题 目 ： 工业大数据—多维时间序列上的错列异常检测

专 业 数据科学与大数据技术专业

学 号 1170300707

姓 名 郑胜文

课 程 大数据计算基础

日 期 2019/12/20

# 摘 要

关键词：多维时间序列；错列异常检测；神经网络；预测；工业大数据

本次大作业的题目是工业大数据—多维时间序列上的错列异常检测。本次大作业基本思路如下：现在有个很大的数据集，由于电脑的计算力有限，所以一开始决定剪切数据集。对于工业大数据的多维时间序列数据来说，每一维的不仅与本维的上一时刻值有关，还与其他维的上一时刻值有关，所以想要预测一维的值要用上所有维的指标。根据这一想法使用LSTM把每一维都分别进行预测，然后把每个指标的预测值和他们的测试集的值进行处理，根据3σ原则来判断某个时间的数据是否出问题。由于是工业大数据的缘故，某一时刻如果只有一个指标出问题，那无法证明这一时刻的多维数据出现错列异常，可能是相关传感器突然出了某个问题但又恢复了，只有当某一时刻的全部指标都超出了3σ才能证明这个时刻出现了错列异常。本文的算法大致思路如上。

**目 录**

[摘 要 I](#_Toc27338278)

[第1章 绪 论 4](#_Toc27338279)

[1.1 研究问题的背景 4](#_Toc27338280)

[1.2 研究问题的挑战 4](#_Toc27338281)

[1.3 本文的工作要解决的问题以及方法 4](#_Toc27338283)

[1.4 本文的贡献 4](#_Toc27338284)

[第2章系统/方法框架 4](#_Toc27338286)

[2.1 系统框架 4](#_Toc27338287)

[2.2 各部分简介 4](#_Toc27338288)

[第3章技术一 4](#_Toc27338289)

[第4章技术二 4](#_Toc27338291)

[第5章实验 5](#_Toc27338293)

[5.1 实验设计 5](#_Toc27338294)

[5.2 对比实验 5](#_Toc27338295)

[5.3 实验结果 5](#_Toc27338296)

[5.4 实验受参数的影响 5](#_Toc27338297)

[第6章相关工作 5](#_Toc27338298)

[第7章结论 5](#_Toc27338299)

# 第1章 绪 论

## 1.1 研究问题的背景

工业大数据具有大体量、价值密度低和维度高等特点。产生的多维时间序列异常模式复杂、难以识别，给工业大数据分析带来挑战。工业智造化的快速发展对多维时间序列的异常处理需求也越来越强烈。由于这些异常数据的存在，就会增大其相应的分析误差，使一些已有的方法变得不准确，得不到我们要预测的结果，造成无法估计的损失。工业上，一组传感器数据往往通过类似于计算机网络中“包”的方式进行传输，而在传输过程中可能因为某种原因导致以某种顺序发送的多维时间序列到达后乱序导致异常模式，我们称这种情况为“错列”。因此，本实验的目标为设计并实现算法对多维时间序列进行错列异常检测。

## 1.2 研究问题的挑战

多维时间序列错列的异常模式定义不明确，在网络上找不到相关的研究，难以学习相关的方法，不知道该如何处理错列。

工业大数据的数据复杂，很难判断是错列还是某一时刻的传感器出了故障，不知道该如何分辨。

## 1.3 本文的工作要解决的问题以及方法

要解决的问题有两个：

首先是要解决如何去预测数值的问题，这里我采取的办法是把给的数据转换成监督学习数据，使用t-5,t-4,t-3,t-2,t-1时刻的所有数据分别去预测t时刻的诸多指标，使用LSTM，loss函数选取了MAE，优化算法选取了Adam算法；

第二点就是要如何判断某个时间点的数据是否有问题。有一些主流的方法，我这里采取了3σ原则来判断，但我认为如果只是单个指标在3σ之外的话，就不能判断这个点的数据有问题，这样会造成很多误判，所以我在算法中选择只有当某一时刻的全部指标都在3σ时才判断这个点有问题，当错误点的数量超过了一个，则有可能是错列异常。

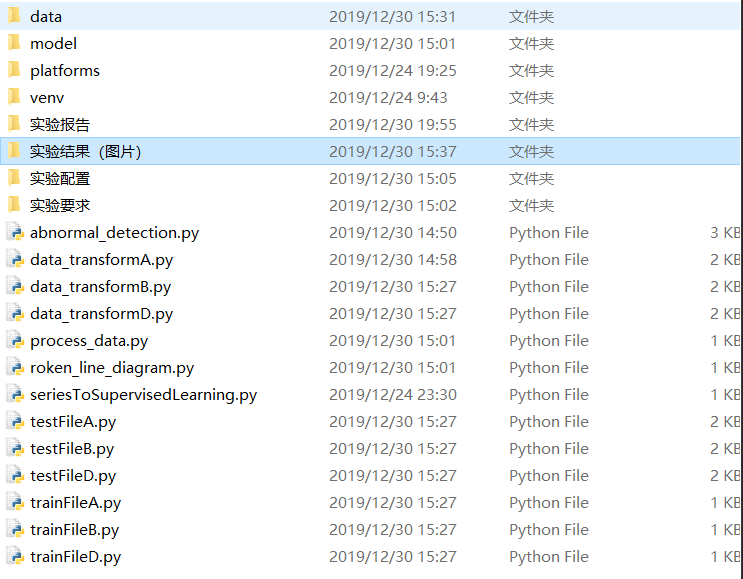
## 1.4 本文的贡献

在处理多维时间序列上更严谨，对于真正的误差判断更为精准，减少了误判率，有实用价值。

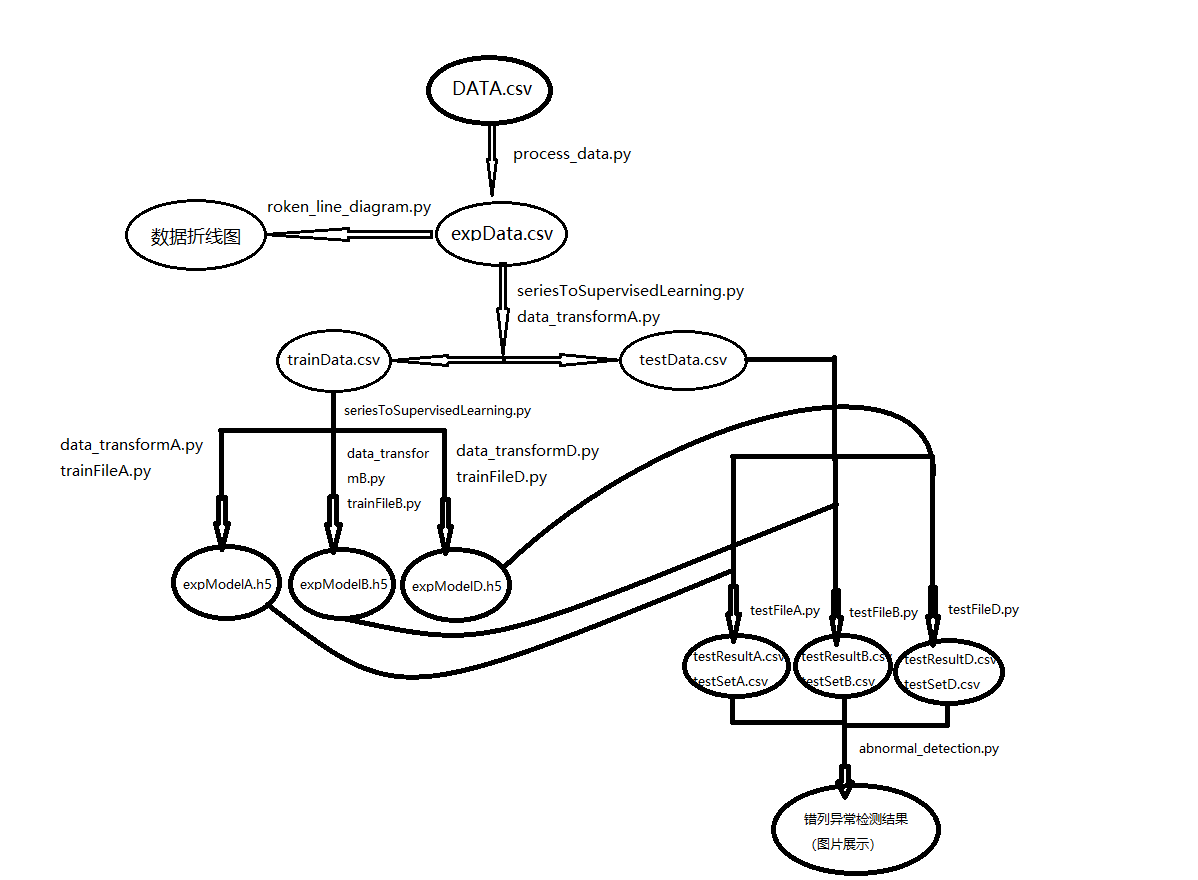
# 第2章系统/方法框架

## 2.1 系统框架

文件夹内容如图所示



代码和数据文件架构如下：



## 2.2 各部分简介

根据上一小节的结构图来介绍每一部分：

process\_data.py文件会把原始数据给切割出实验数据，使用pandas库和datetime库；

roken\_line\_diagram.py文件会根据实验数据画出几个指标的折线图；

seriesToSupervisedLearning.py文件可以把本来的实验数据结构转换成监督学习的数据结构；

data\_transformA.py（data\_transformB.py、data\_transformD.py）调用seriesToSupervisedLearning.py，把expData.csv切割成训练数据和测试数据，并且得出训练A、B和D的监督学习数据；

trainFileA.py（trainFileB.py、trainFileD.py）构建LSTM神经网络模型并训练得到模型；

testFileA.py（testFileB.py、testFileD.py）调用模型在测试数据上做预测，得到预测的数据文件和测试集；

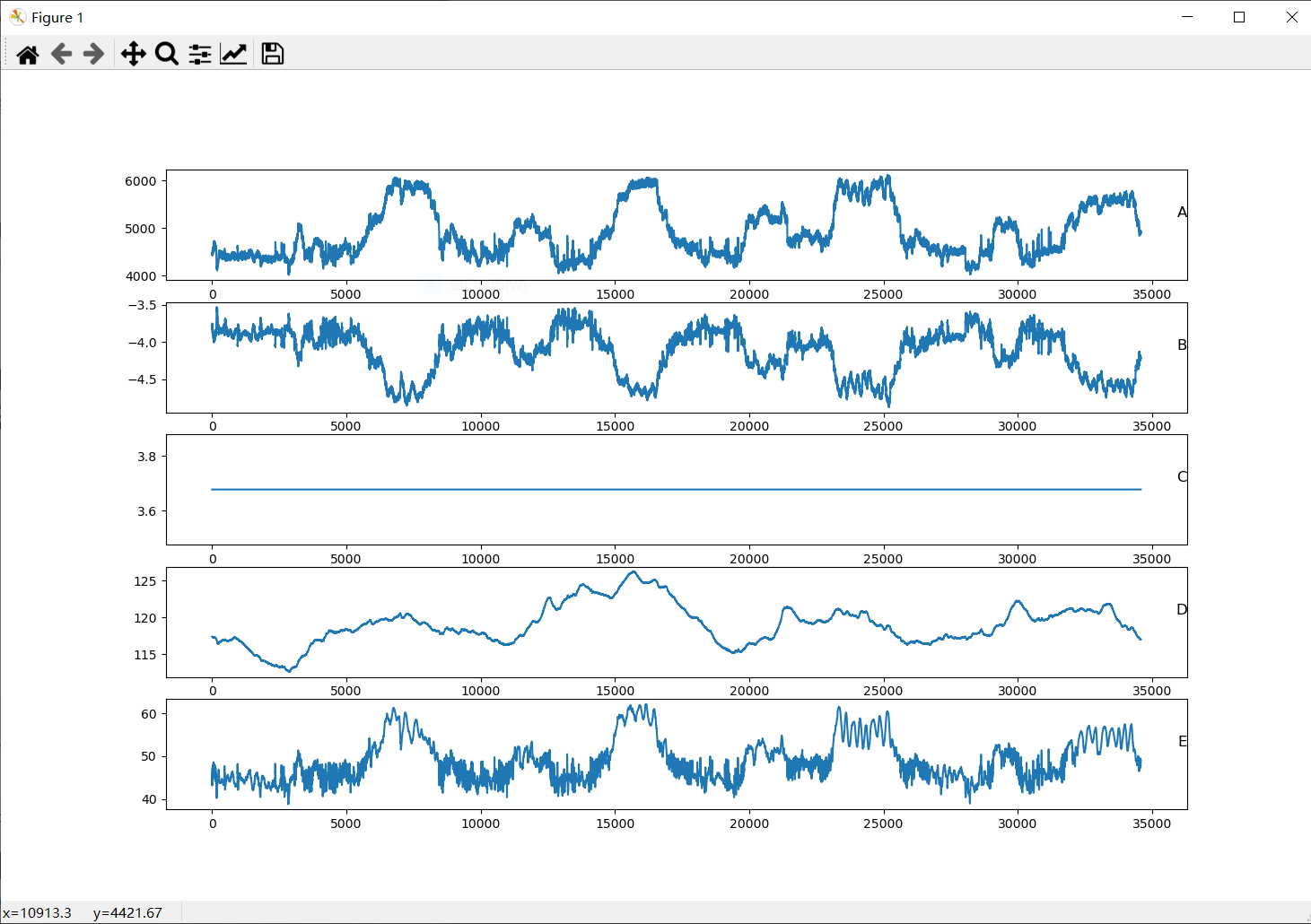
abnormal\_detection.py文件根据三个指标的预测集和测试集、依照3σ原则，得到错列异常检测结果。

# 第3章技术一

## 使用LSTM实现数据预测

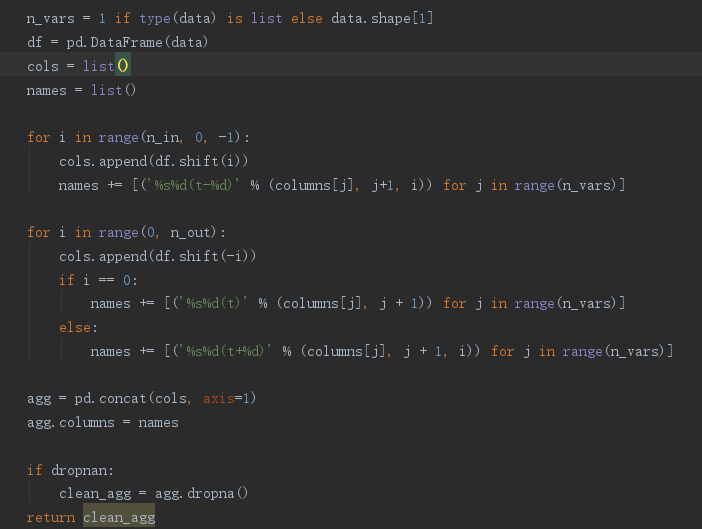
对于工业大数据来说，时间序列上数据的相关性很高，所以要想更好地预测数据，要使神经网络有记忆性，所以LSTM神经网络模型也就成了很好的选择。

原始数据分布：



观察数据可以看到指标C保持不变，所以决定把C指标去掉，变为四个指标，也就是D->C,E->D，此时有A,B,C,D四个指标，而A.B,D相关性很高，所以决定**通过A,B,D三个指标进行错列异常检测**。数据的传输周期设置为每天。

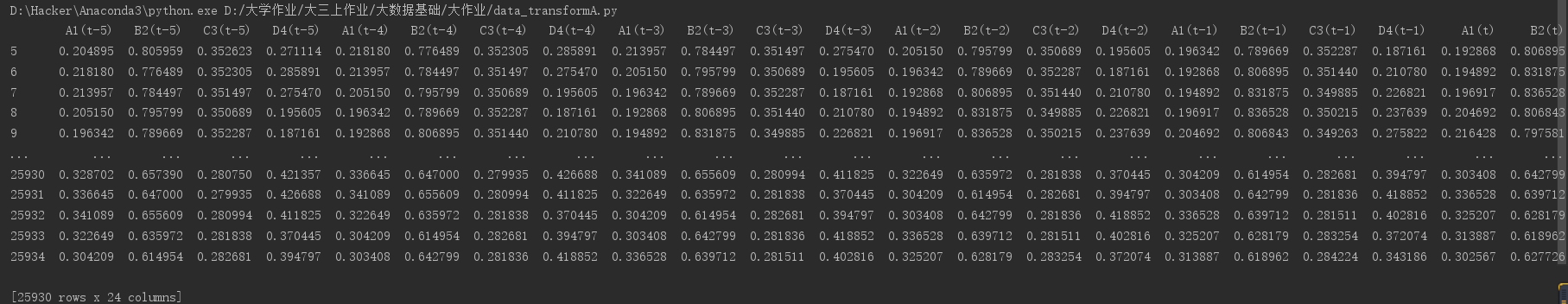
在介绍模型之前，我们需要先处理数据，将其变为监督学习数据。我的代码如下：



我们要把数据转化成输入、输出的模式。默认输入的步长为1，输出的步长为1，然而由于时间序列的特殊性，我在后来，决定把输入步长变为5，即使用t-5、t-4、t-3、t-2、t-1的全部数据去预测t时刻的数据。

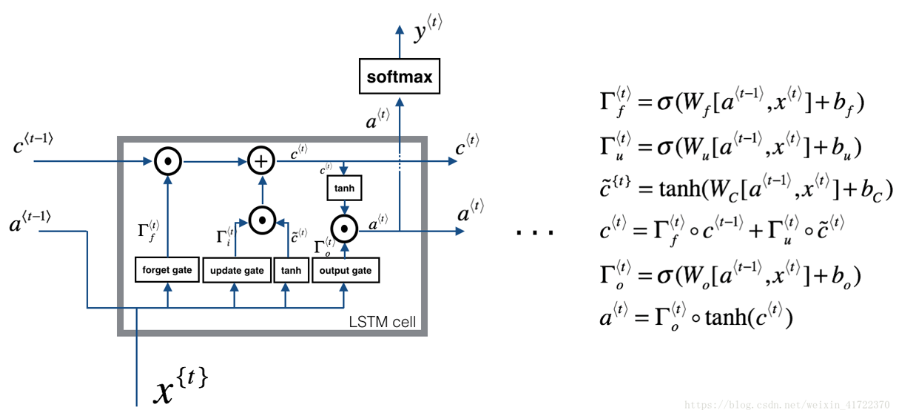
举个例子，现在有一个数组，为[1,2,3,4,5]，把输入步长设为3，则经过函数处理，监督学习数据变为：input：[[1,2,3][2,3,4]] , output：[4,5]。

我的数据经过处理后大致如下：



下面介绍一下LSTM：

“某地开设大量工厂，空气污染十分严重……天空都变成了灰色”，如果我们的模型要试着预测这句话的最后一个词“灰色”时，仅仅根据短期依赖无法完成，因为如果不看上文“空气污染十分严重”我们是无法知道天空是“蓝色”还是“灰色”。因此，当前预测位置和相关信息之间的文本间隔就可能变的很大，当这个间隔变得很大时，简单的循环神经网络就会丧失学习到如此远的信息的能力。LSTM就是用来解决这样的问题。 LSTM网络是一个拥有三个“门”的特殊网络结构，依次为“遗忘门”、“输入门”、“输出门”。下图为LSTM的网络结构和公式，其中c为记忆细胞状态，x为输入，a为各层输出。



下面来分别解释一下这三个门，理解这三个门的作用也是理解LSTM的关键。1.遗忘门：

作用于：记忆细胞状态

作用效果：选择性遗忘记忆细胞中的信息

例子：“She is busy today……I am”当预测“am”的时候我们要选择性的忘记之前的主语“She”，否则将出现语法错误。

2.输入门：

作用于：记忆细胞状态

作用效果：将新的信息选择性的记录到新的细胞状态中

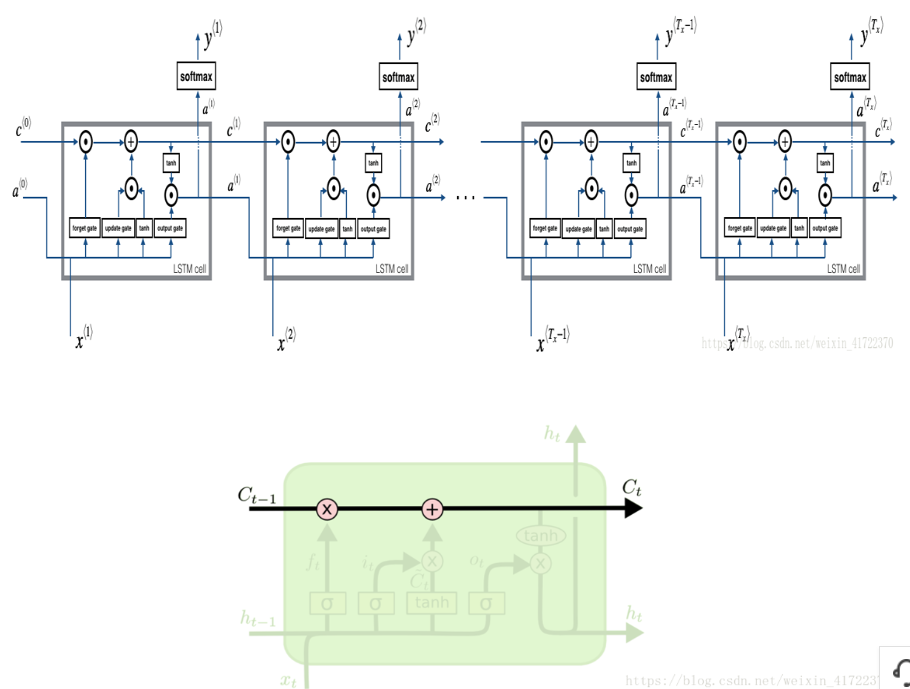
例子：上面那个句子我们会根据“I”将这个主语信息更新到细胞状态中，所以最后会预测出“am”。

3.输出门：

作用于：输入和隐层输出

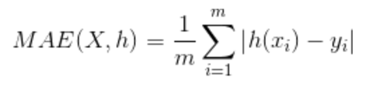
作用效果：使得最后输出即包括细胞状态又包括输入，将其结果更新到下一个下一个隐层。

通过这三个门，LSTM能更加有效的决定哪些信息被遗忘，哪些信息被保留，通过LSTM的前向传播图我们可以看到一个细胞状态可以很容易传到很远距离影响输出，所以LSTM可以解决远距离的信息的学习。



下面介绍MAE损失函数：

MAE（Mean Absolute Error）是平均绝对误差，公式如下：



它是一种常用的回归损失函数，它是目标值与预测值之差绝对值的和，表示了预测值的平均误差幅度，而不需要考虑误差的方向，范围是0到∞，这对于数据预测很适合，因为不考虑误差方向不影响预测的效果。同时，他是绝对误差的平均值，可以更好地反映预测值误差的实际情况。

下面说明Adam优化算法的优点：

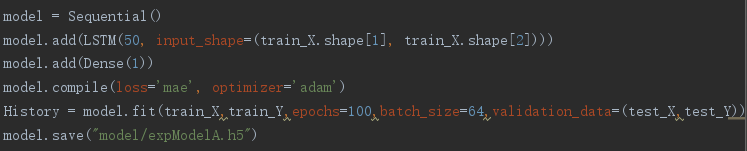
Adam 优化算法是随机梯度下降算法的扩展式，是一种可以替代传统随机梯度下降过程的一阶优化算法，近来其广泛用于深度学习应用中，尤其是计算机视觉和自然语言处理等任务。

它有以下优势：

①直截了当地实现；②高效的计算；③所需内存少；④梯度对角缩放的不变性；⑤适合解决含大规模数据和参数的优化问题；⑥适用于非稳态（non-stationary）目标；⑦适用于解决包含很高噪声或稀疏梯度的问题；⑧超参数可以很直观地解释，并且基本上只需极少量的调参。

由以上的优点可以看出非常适合去处理工业大数据的数据。

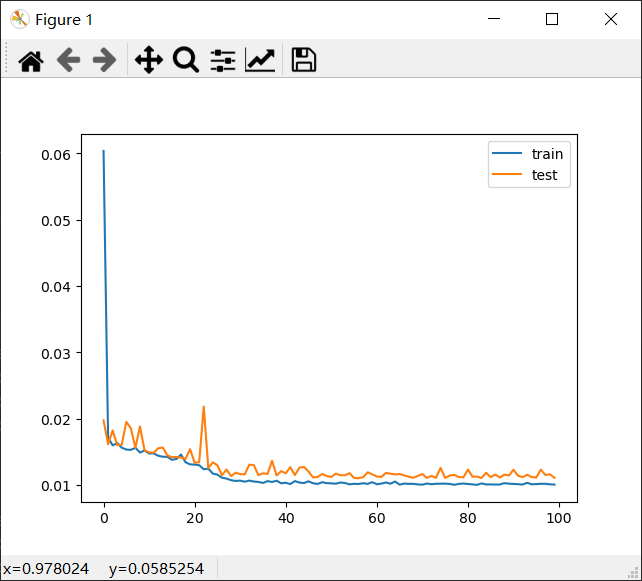
**综上，开始搭建LSTM神经网络：**

****

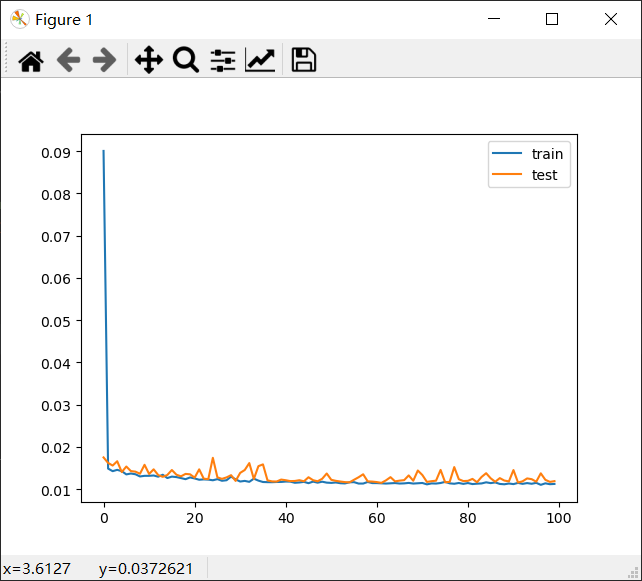
神经网络先采用了五十层的LSTM网络，然后采取了一个Dense层，之后loss函数采用了MAE，至于优化算法采用了Adam算法。然后训练轮数为100，批量大小为64，然后把测试集作为验证集，最后得到模型以h5格式存储起来。

A,B,D三个指标的loss值如图所示：

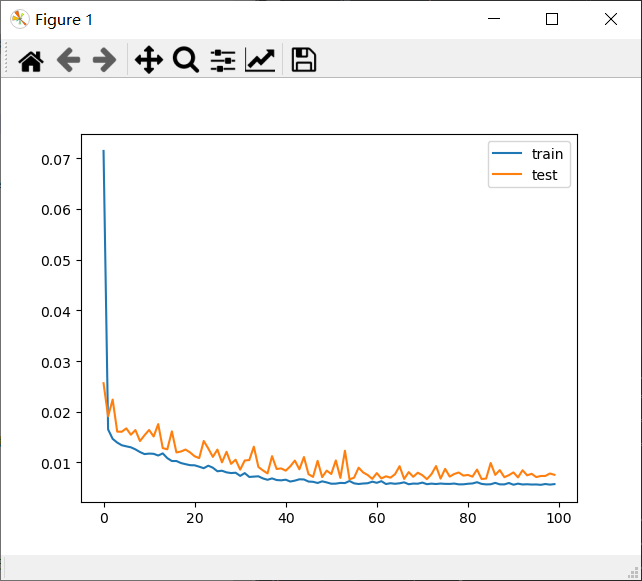
A：



B:



D:



# 第4章技术二

多维时间序列的错列异常检测

工业大数据上的多维时间序列数据有明显的特征：数据相关性高，噪音较大等等。

我决定这样处理多维时间序列：使用多维序列去预测单维序列，然后再把每个单维联合在一起处理，进行错列异常检测。

单维异常检测时采取拉依达准则( PauTa Criterion 或 3σ准则) 处理异常数据。

拉依达准则( PauTa Criterion 或 3σ准则)原理：

拉依达准则是指先假设一组检测数据只含有随机误差，对其进行计算处理得到标准偏差，按一定概率确定一个区间，认为凡超过这个区间的误差，就不属于随机误差而是粗大误差，含有该误差的数据应予以剔除。

这种判别处理原理及方法仅局限于对正态或近似正态分布的样本数据处理，它是以测量次数充分大为前提（样本>10），当测量次数少的情形用准则剔除粗大误差是不够可靠的。

3σ法则为：

数值分布在（μ-σ,μ+σ)中的概率为0.6827

数值分布在（μ-2σ,μ+2σ)中的概率为0.9545

数值分布在（μ-3σ,μ+3σ)中的概率为0.9973

可以认为，Y 的取值几乎全部集中在（μ-3σ,μ+3σ)区间内，超出这个范围的可能性仅占不到0.3%.

**3σ算法实现：**

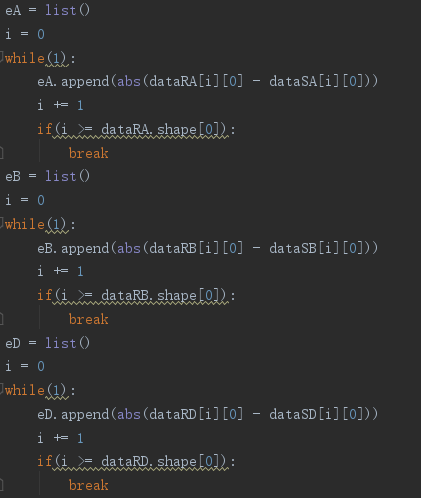
首先需要保证需要检验的数据列大致上服从正态分布；

然后计算需要检验的数据列的标准差；

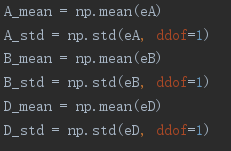
最后比较数据列的每个值，是否大于标准差的3倍；

大于3倍标准差的即判断该样本有异常。

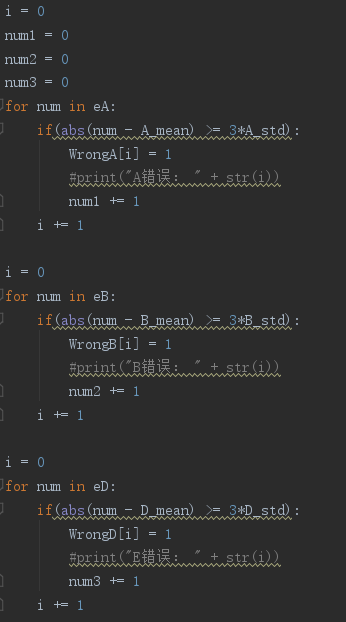
在abnormal\_detection.py文件中我实现了错列异常检错：

我首先读取A,B,D的预测集和原值集，然后分别做绝对值差：

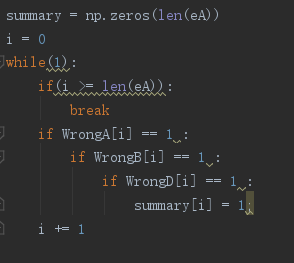
然后根据绝对差值list去分别计算A,B,D指标的预测值和原值的绝对差值的标准差和均值：



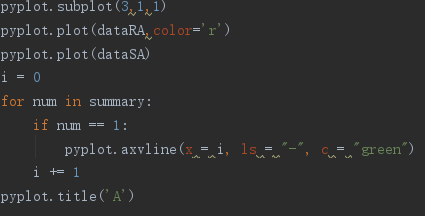
然后根据这个值和绝对差值list去找出A,B,D指标的超过3σ的样本点，并记录这些样本点：



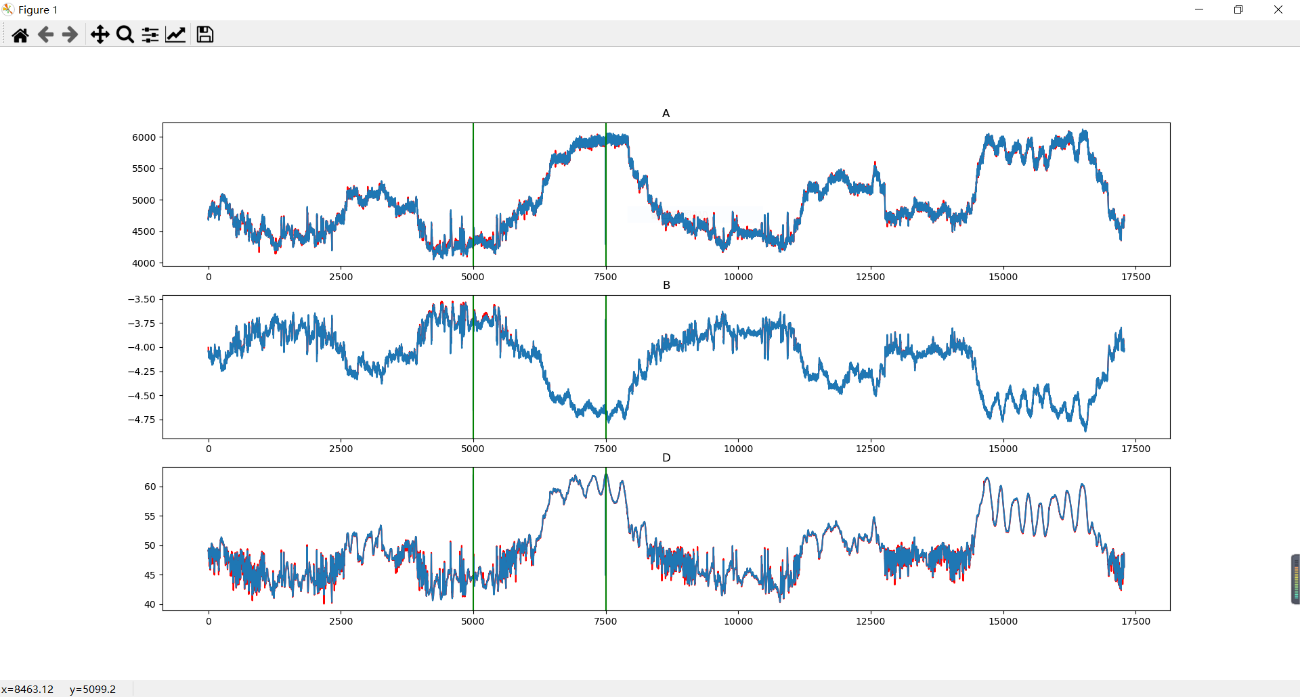
鉴于是工业大数据，维度很高，若只是很少的维度产生了异常值，那并不证明这一时刻发生了错列异常，而又可能只是单纯的小故障，我认为只有当某一时刻的所有样本值产生了问题才证明这一时刻可能发生错列异常：



最后把A,B,D三个指标的原值（蓝色）和预测值（红色）画出来，并在异常值样本处画一条垂直于x轴的绿线：



效果如图：



# 第5章实验

## 5.1 实验设计

实验设计分为五步：

1. 准备数据并处理数据
2. 搭建LSTM神经网络并进行训练，得到模型
3. 根据模型进行预测，得到预测结果，并计算RMSE衡量预测效果
4. 处理预测数据，进行错列异常检测
5. 设计对比实验再次进行实验

**以上实验步骤的具体步骤和结果截图见下文的5.3实验结果部分。**

## 

## 5.2 对比实验

改变原测试集的某些值，进行对调，得到新的测试集，然后用模型去处理新的测试集，得到预测数据与新的测试集进行异常处理，完成错列异常检测。（**效果见下文5.3实验结果**）

## 5.3 实验结果

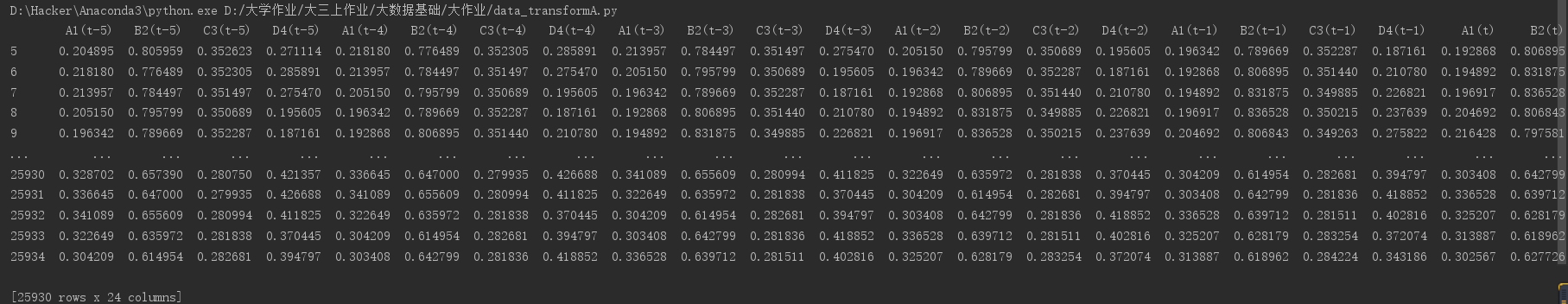
5.1说到实验设计为五步：

1. 准备数据并处理数据
2. 搭建LSTM神经网络并进行训练，得到模型
3. 根据模型进行预测，得到预测结果，并计算RMSE衡量预测效果
4. 处理预测数据，进行错列异常检测
5. 设计对比实验再次进行实验

下面展示实验结果：

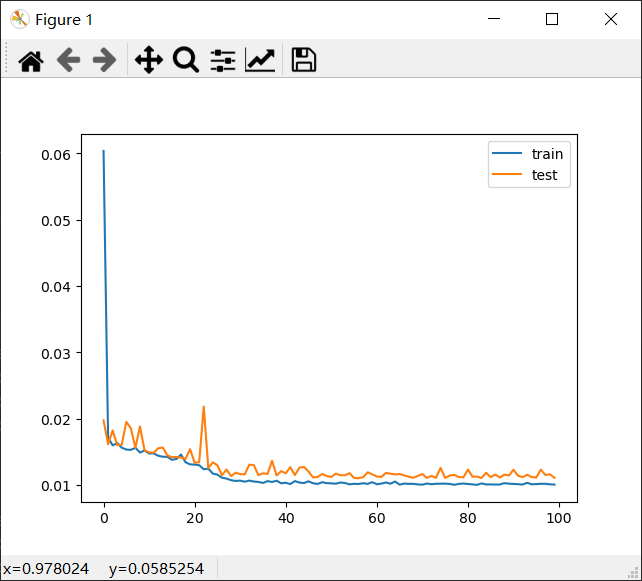
（1）

把源数据转化成监督学习数据，格式如下所示：

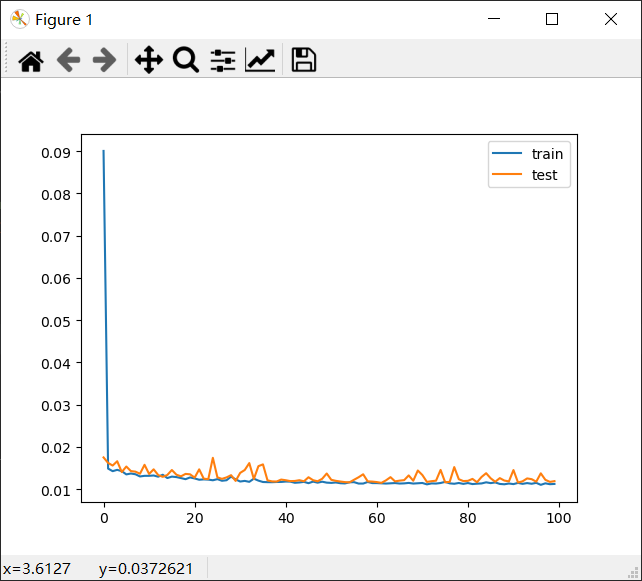


（2）模型的loss值随训练轮数变化效果如下所示：

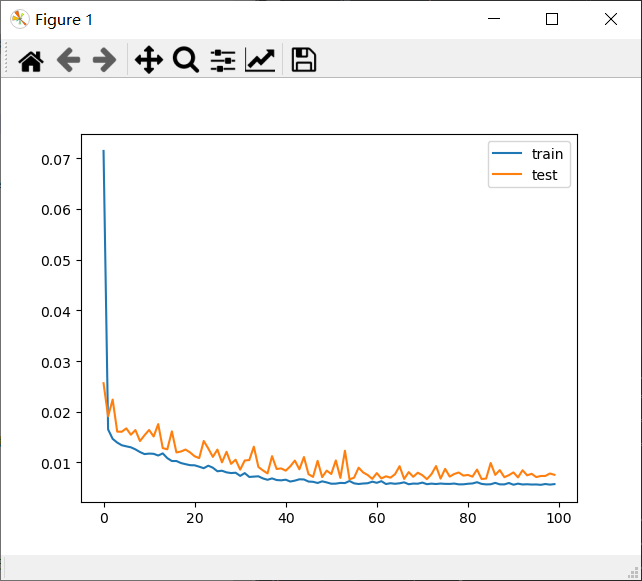
指标A：



指标B：



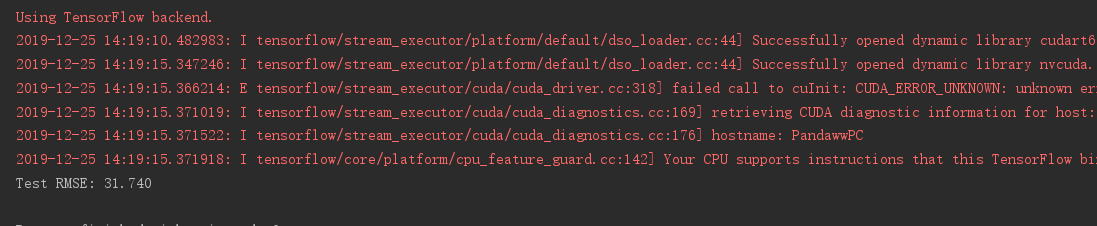
指标D：



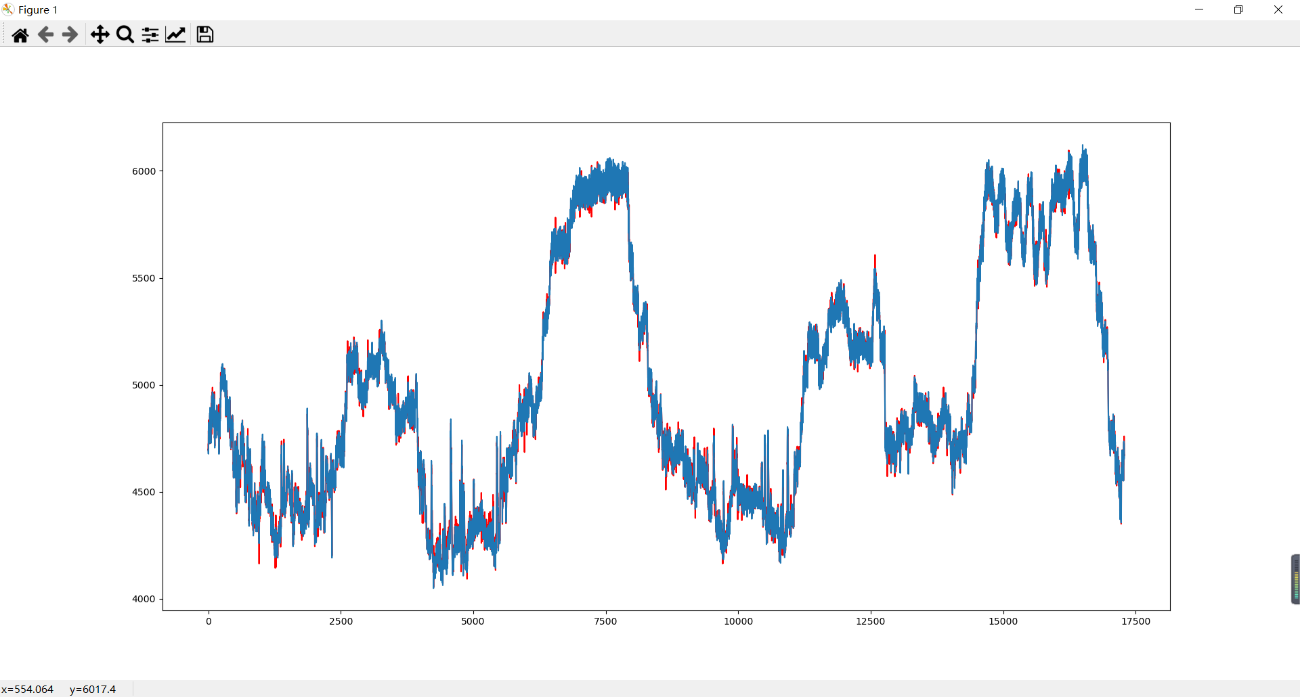
（3）下面展示模型的预测效果（蓝色为原数据，红色为预测值），以及测试的RMSE值，可以看到预测的效果都很好：

指标A：

RMSE与数据的均值比例约为0.6%，可以看出预测效果很好：

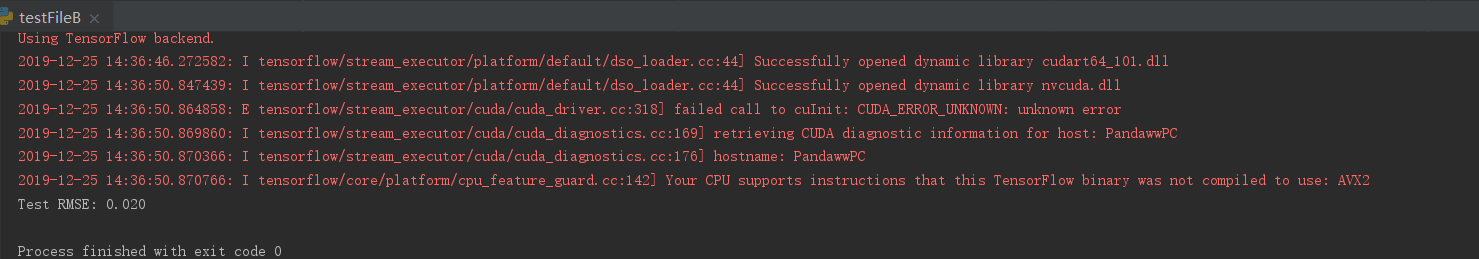


根据折线图也可看出拟合度也很高：

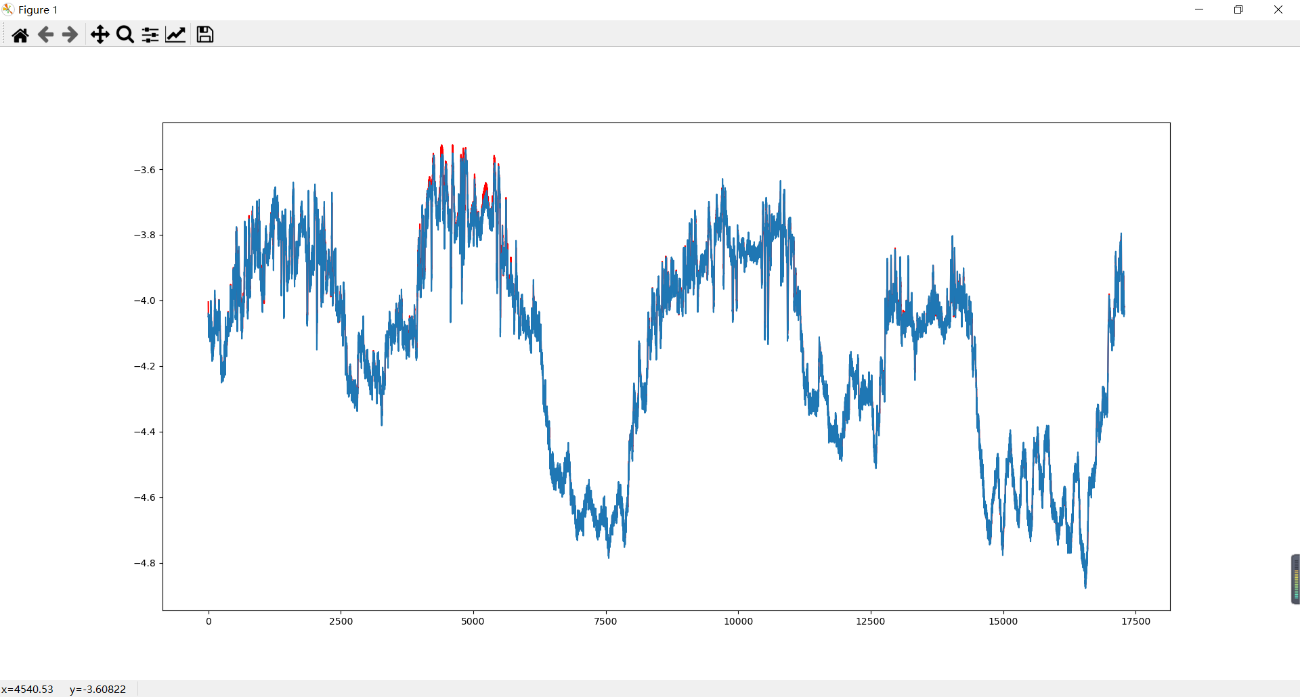


指标B：

RMSE与数据的均值比例约为0.5%，可以看出预测效果很好：

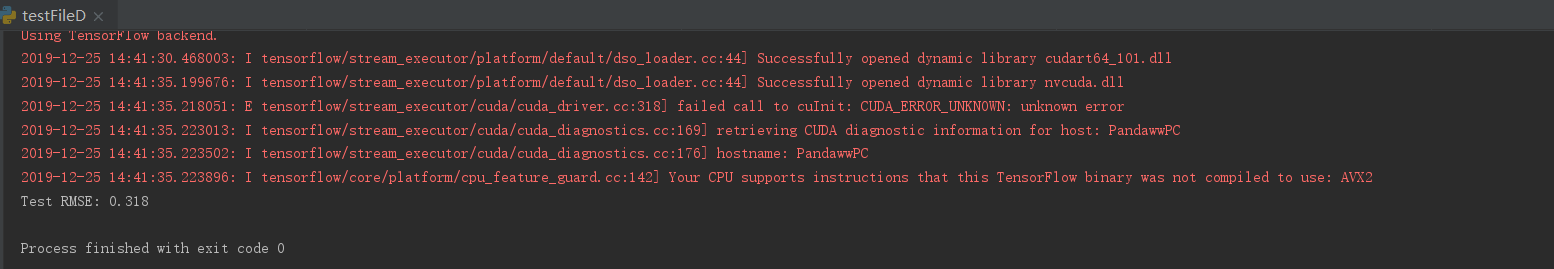


根据折线图也可看出拟合度也很高：

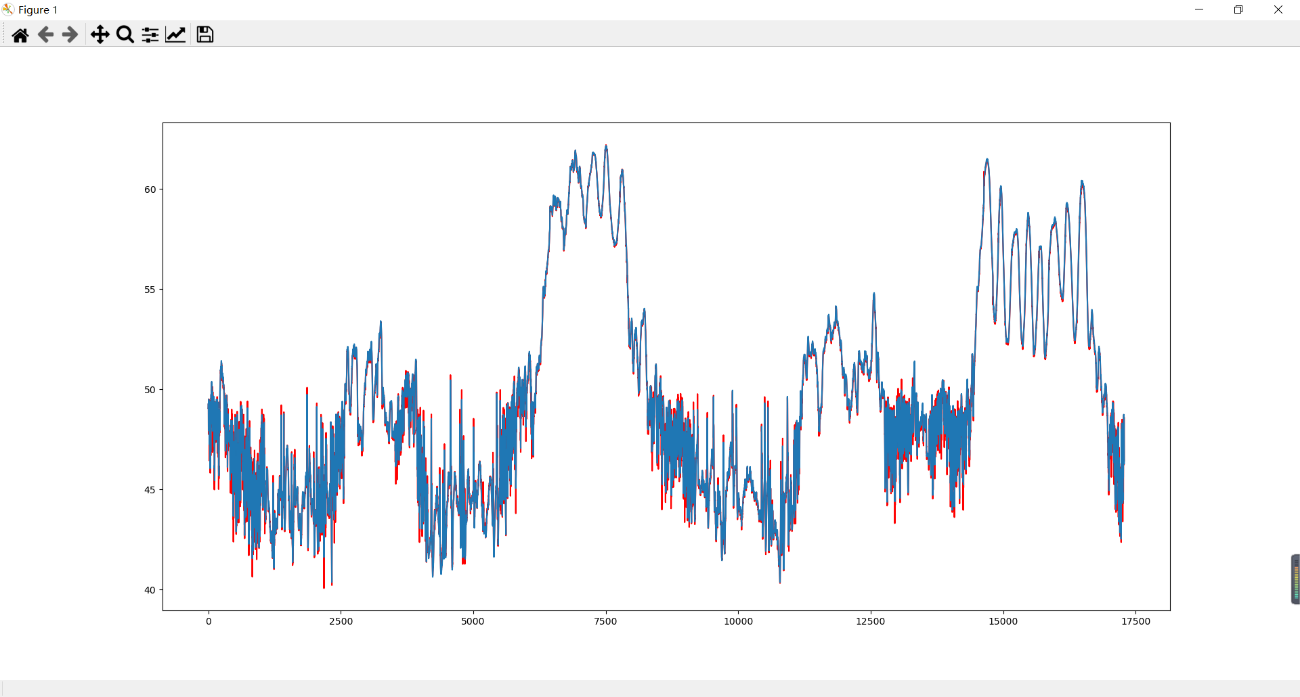


指标D：

RMSE与数据的均值比例约为0.6%，可以看出预测效果很好：



根据折线图也可看出拟合度也很高：

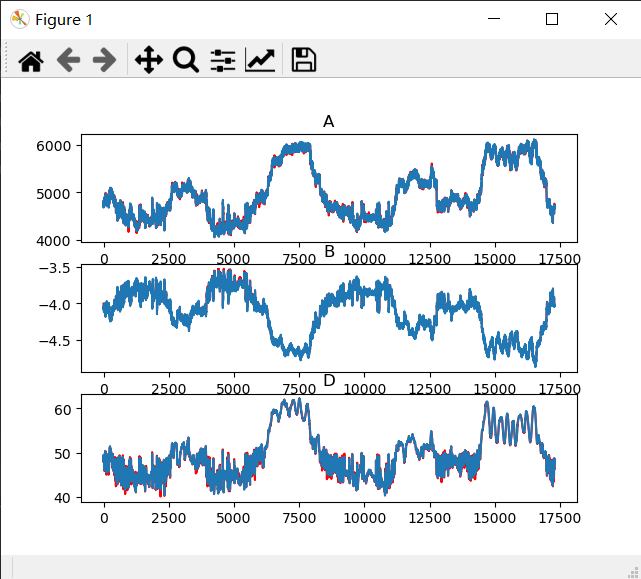


（4）

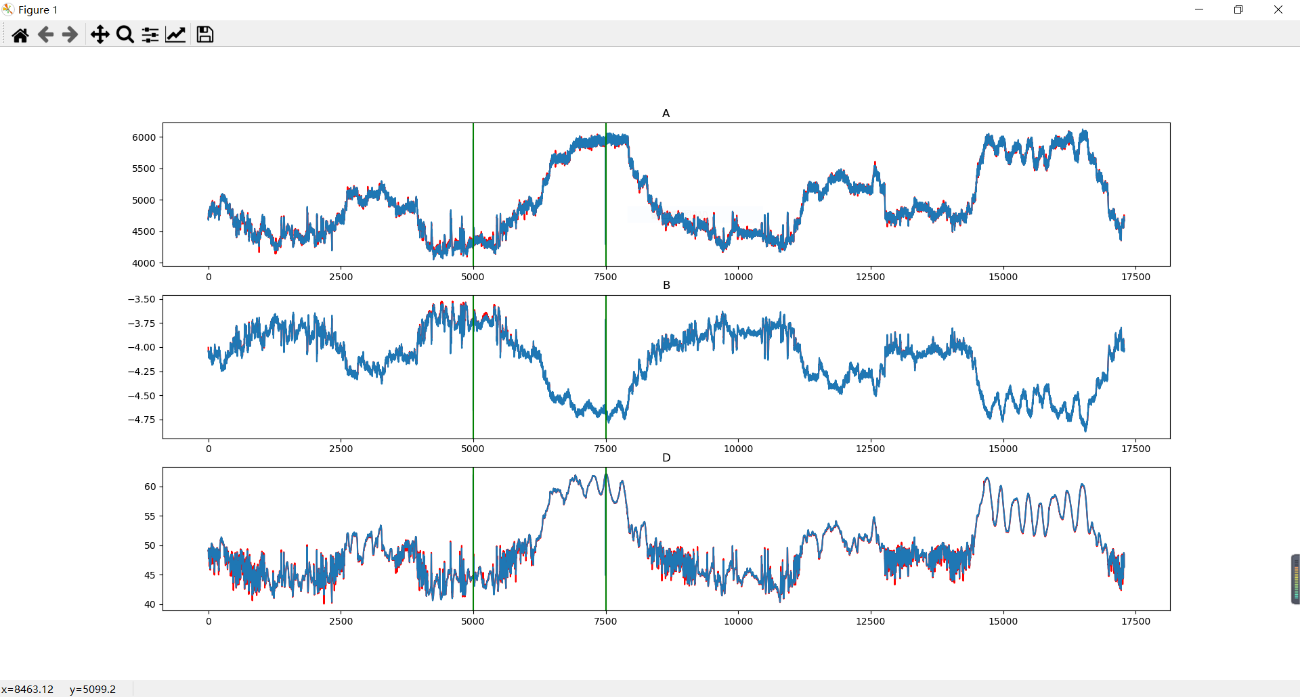
根据3σ原则，可以看出测试集A,B,D的预测值分别有331、180、529个样本点的值超过3σ有异常，然而这只是单维有异常，还需要将他们联合分析。数据分别有17825个样本点，可以看出预测效果很不错



对A,B,D指标进行联合分析，若一个样本点的A,B,D指标同时有异常，则这一样本点才有可能有错列异常，若是出现异常点，则会在相应的指标处画一个垂直于x轴的绿色线，经过分析，发现原数据并没有出现错列异常检测，所以没有出现绿色线：



（5）设计对比实验，即人工创造错列数据，手动把原数据的5000和7500点的数据进行了对调，5200和9000点的数据进行了对调，然后运行程序发现了5000点和7500点处的错列异常，而没有发现5200和9000点的，原因在于5000和7500点的数据三个指标差别都很大，而5200点和9000点处只有B指标差别很大，无法判断究竟是错列异常还是单纯的传感器出故障，所以未进行报错，效果图如下，可以看到异常点有绿线：



## 5.4 实验受参数的影响

有几个关键参数，那就是进行监督学习数据处理时的输入步长，还有LSTM层数，训练轮数，批大小。

对于输入步长来说，5是比较好的大小，经过实验，不管是大于5还是小于5，loss下降的都比较慢而且值比步长为5时大。

LSTM层数超过50层时对loss下降速率影响不大。

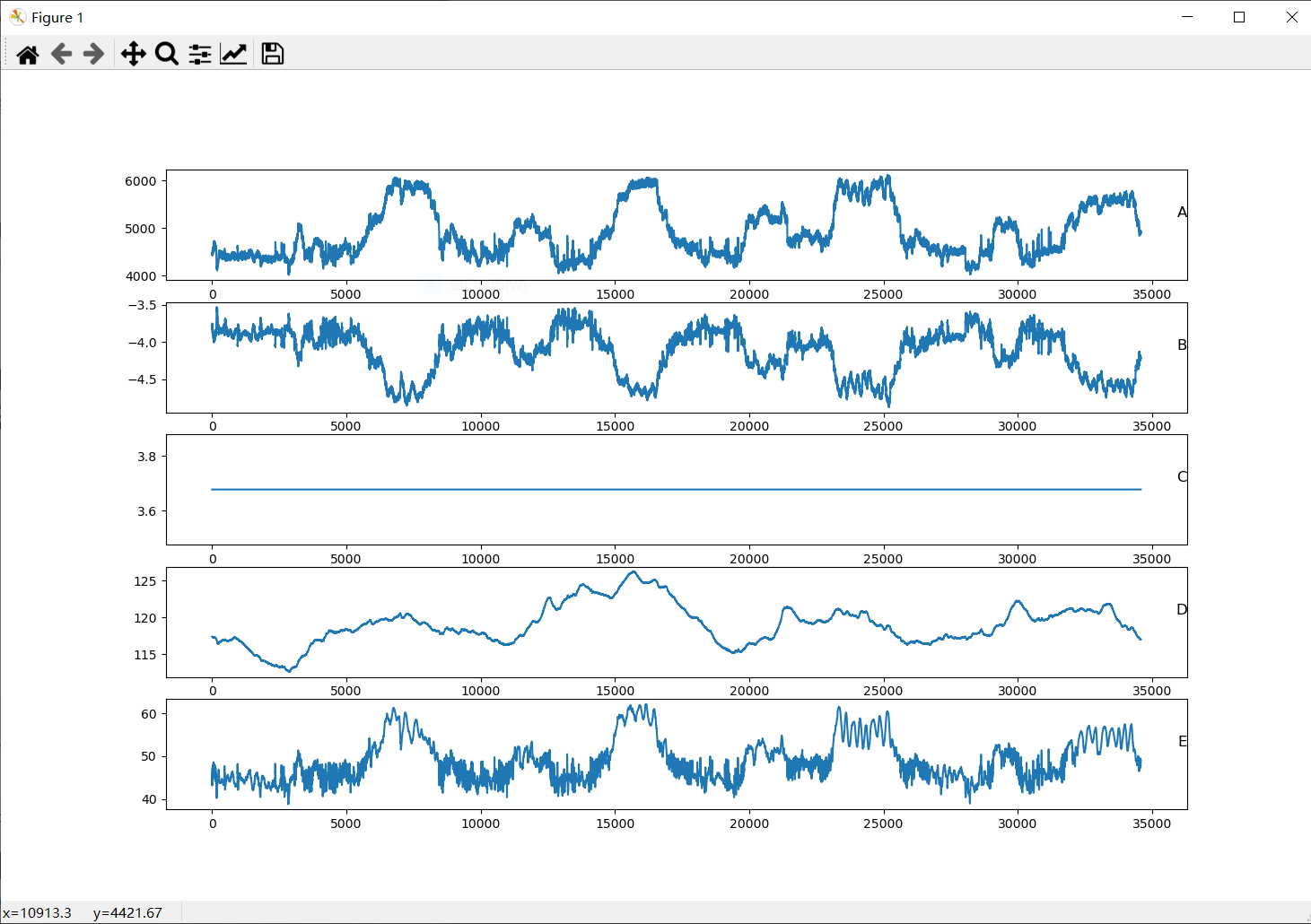
训练轮数超过30之后，对于loss值的影响就不大了。

批大小设定为64时有不错的表现，若是过大或者过小，则loss下降很慢。

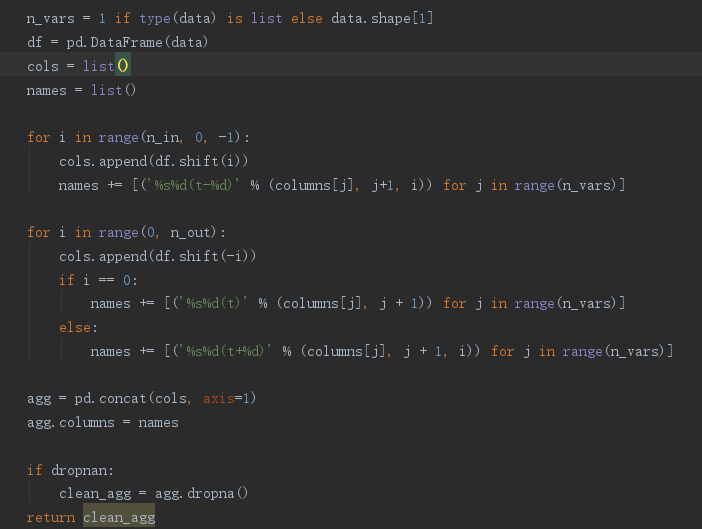
# 第6章相关工作

数据处理（把原数据转化为监督学习数据）：

原始数据分布：



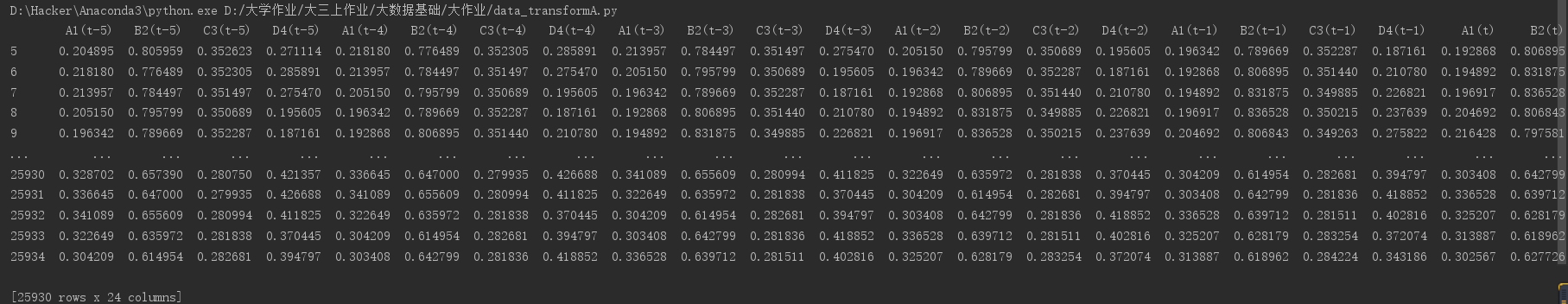
在介绍模型之前，我们需要先处理数据，将其变为监督学习数据。我的代码如下：



我们要把数据转化成输入、输出的模式。默认输入的步长为1，输出的步长为1，然而由于时间序列的特殊性，我在后来，决定把输入步长变为5，即使用t-5、t-4、t-3、t-2、t-1的全部数据去预测t时刻的数据。

举个例子，现在有一个数组，为[1,2,3,4,5]，把输入步长设为3，则经过函数处理，监督学习数据变为：input：[[1,2,3][2,3,4]] , output：[4,5]。

我的数据经过处理后大致如下：



# 

# 第7章结论

**多维时间序列错列的异常模式定义**：某几个时刻的数据由于某种原因而乱序到达接收端。样本点数据值与附近数据值差别较大，而与其他某处数据值很契合，则这样的样本点就是错列异常点

。

工业大数据上的错列异常检测是很艰难的，因为观察给出的原始数据可以明显的观察到，很多时候数据是没有规律性可言的，变换的幅度很大，噪声很大，这为错列异常造成了很大的阻挠。

当某一点发现该点数据值与周围点的数据值差别较大，你很难判断到底是错列异常还是单纯的发生了传感器出故障，数据出现较大波动。为了解决这一问题我则提出了上述方法，因为要同时那么多维的传感器都出现问题是可能性很小的事，所以这种情况下极有可能是发生了错列异常。

这次实验让我深深感受到了工业大数据的复杂之处，也让我对于LSTM神经网络有了更深层次的认识。明白了数据科学中的异常检测部分问题，在这一方面流派很多，没有特别主流的做法，感觉还需要更深刻的学习。