



Subway vibration signal Anomaly Detection

Based on Deep Convolutional Neural Networks

by

Mimi Zhang

Supervisor: Dr Tim Bashford

Project submitted as part of the requirements for the award of
MSc Software Engineering (Web Development)

November 2020

Code Description

1 File structure

The general function of each file is as follows:

- **Identify_anomaly**: The file contains python code for predicting abnormal data.
- **Matlab_code**: The file contains matlab code to extract data and prepare for prediction.
- **Code_description_English.pdf**: This file is used to explain the specific code flow and usage of the competition in English.

Next, both files of **Identify_anomaly** and **Matlab_code** will be described in detail and how to use them.

2 Matlab_code file description

This file contains 2 files, as shown in Figure 1.

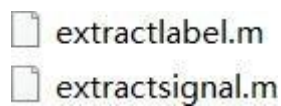


Figure 1

The function of this file is to extract and store all the data according to the sensor number

before identifying the abnormal data. At the same time, it also extracts the labels corresponding to the data collected by sensors at different times to facilitate the calculation of model prediction accuracy.

The functions of the two matlab files in Figure 1 are as follows:

- **extractlabel.m**: Used to extract the labels corresponding to the sensor
- **extractsignal.m**: Used to extract data corresponding to the sensor

In the **extractsignal.m** file, there are two lines of code, as shown in Figure 2, used to assign values to the variables **data_save_dir** and **file_dirname**. The variable **data_save_dir** is the storage path after data extraction, and **file_dirname** is the original file storage path. Users can change the values of these two variables according to actual needs.

```
data_save_dir="E:\研究生学习\SHG\IPC-SHM-P2\chuli\base_on_class"; %signal extraction storage directory  
file_dirname="E:\研究生学习\SHG\IPC-SHM-P2\data"; %directory of original signal
```

Figure 2

In the **extractlabel.m** file, the values of two variables can be modified according to requirements. The variable **path** is the data label directory, and the variable **saverdir** is the label directory corresponding to the data collected by each sensor at different times, as shown in Figure 3.

```
path='E:\研究生学习\SHG\IPC-SHM-P2\label.mat';%label data directoty  
saverdir='E:\研究生学习\SHG\IPC-SHM-P2\chuli\分类标签';%extracted label storage directory
```

Figure 3

3 Identify_anomaly file description

The file information contained in this file is shown in Figure 4.

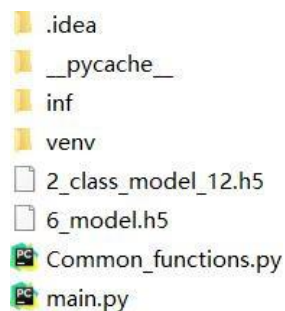


Figure 4

The main files in Figure 4 are explained as follows:

- **main.py**: Main function for input data path
- **Common_functions.py**: Used to process data, such as data cleaning and statistical feature extraction
- **2_class_model_12.h5**: Two-class model file
- **6_model.h5**: Six classification model files
- **inf**: Used to save prediction data information

Python code operating environment description: Tensorflow 2.0, scikit-learn 0.20.

When running the **main.py** file, the three files **Common_functions.py**, **2_class_model_12.h5**, and **6_model.h5** should be in the same directory, as well as the **inf** folder. In the run function of **main.py**, the variables **path** and **path_2** respectively correspond to the directories storing sensor data and corresponding labels, as shown in Figure 5.

```
path='E:/研究生学习/SHG/IPC-SHM-P2/chuli/base_on_class/%s'%c #Sensor data storage directory  
path_2='E:/研究生学习/SHG/IPC-SHM-P2/chuli/分类标签/%s.mat'%c #Label storage directory
```

Figure 5

If the abnormal data identification is carried out for multiple sensors, only the input parameters need to be modified. For example, to identify sensors 1~7 (data from 7 sensors). Just assign the variable **sensor_num_range** to **range(1, 8)**(Python built-in function) at the entrance of the program in the **main.py** file, as shown in Figure 6.

```
if __name__ == '__main__':  
    c = []  
    sensor_num_range = range(1, 8)  
    for i in sensor_num_range:  
        c.append('sensor_' + str(i).zfill(2))
```

Figure 6

4 Usage example

This content takes an example to fully illustrate the code usage process.

1. First, extract the data according to the sensor number, and extract the labels

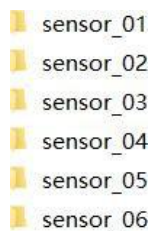
corresponding to the data collected by sensors at different times.

Assign values to the variables `data_save_dir` and `file_dirname` in the `extractsignal.m` file, as shown in Figure 7.

```
data_save_dir="E:\研究生学习\SHG\IPC-SHM-P2\chuli\base_on_class"; %signal extraction storage directory  
file_dirname="E:\研究生学习\SHG\IPC-SHM-P2\data"; %directory of original signal
```

Figure 7

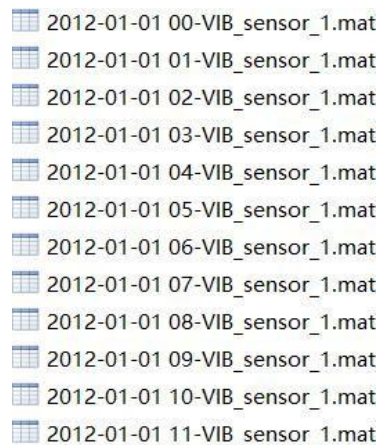
The extracted data files according to the sensor number are shown in Figure 8. Only part of extraction results are displayed in Figure 8.



sensor_01
sensor_02
sensor_03
sensor_04
sensor_05
sensor_06

Figure 8

The results in each of the above sensor files retain time information and sensor information, which are reflected in the way of data naming. An example of the results of the sensor numbered 1 is shown in Figure 9.



2012-01-01 00-VIB_sensor_1.mat
2012-01-01 01-VIB_sensor_1.mat
2012-01-01 02-VIB_sensor_1.mat
2012-01-01 03-VIB_sensor_1.mat
2012-01-01 04-VIB_sensor_1.mat
2012-01-01 05-VIB_sensor_1.mat
2012-01-01 06-VIB_sensor_1.mat
2012-01-01 07-VIB_sensor_1.mat
2012-01-01 08-VIB_sensor_1.mat
2012-01-01 09-VIB_sensor_1.mat
2012-01-01 10-VIB_sensor_1.mat
2012-01-01 11-VIB_sensor_1.mat

Figure 9

The extraction of label information corresponding to the sensor uses the code of `extractlabel.m`. Assign values to the variables `path` and `saverdir` in the code, as shown in Figure 10.

```
path='E:\研究生学习\SHG\IPC-SHM-P2\label.mat';%label data directoty  
saverdir='E:\研究生学习\SHG\IPC-SHM-P2\chuli\分类标签';%extracted label storage directory
```

Figure 10

The results of the final label files are shown in Figure 11.

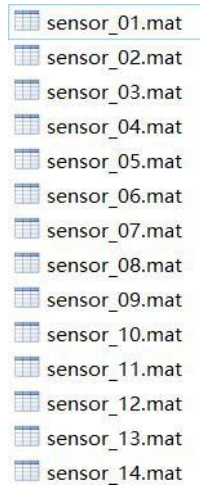


Figure 11

2. Make predictions on the previously processed data

As shown in Figure 12, assign the directory where the data is stored according to the sensor number and the directory where the corresponding label is located to the variables **path** and **path_2** in **main.py**.

```
path='E:/研究生学习/SHG/IPC-SHM-P2/chuli/base_on_class/%s'%c #Sensor data storage directory  
path_2='E:/研究生学习/SHG/IPC-SHM-P2/chuli/分类标签/%s.mat'%c #Label storage directory
```

Figure 12

Besides, a reproduction example is performed based on 7 sensors with numbers 1~7. As shown in Figure 13, in the **main.py** file, at the program entry, assign the variable **sensor_num_range** to **range(1, 8)**(Python built-in function).

```
if __name__ == '__main__':  
    c = []  
    sensor_num_range = range(1, 8)  
    for i in sensor_num_range:  
        c.append('sensor_' + str(i).zfill(2))
```

Figure 13

After running, the results of generated txt files are shown in Figure 14.



Figure 14

Each txt file contains the information regarding the time and number of the sensor and the overall prediction accuracy. For instance, Figure 15 illustrates the information of sensor_01.

```
The accuracy of the prediction on sensor_01 is 96.24%
sensor_01 2012-01-01 00 outlier
sensor_01 2012-01-01 01 outlier
sensor_01 2012-01-01 02 minor
sensor_01 2012-01-01 03 minor
sensor_01 2012-01-01 04 outlier
sensor_01 2012-01-01 05 minor
sensor_01 2012-01-01 06 minor
sensor_01 2012-01-01 07 minor
sensor_01 2012-01-01 08 minor
sensor_01 2012-01-01 09 minor
sensor_01 2012-01-01 10 minor
sensor_01 2012-01-01 11 minor
sensor_01 2012-01-01 12 minor
sensor_01 2012-01-01 13 minor
```

Figure 15