A Simple User Manual for the Source Code

In the ‘AQI_Prediction_MTMC’ pack, ‘Air Quality Prediction’ folder is included. In it, a user manual (‘User Manual.PDF’), three runnable python files (‘test_MTMC.py’, ‘test_MTMC_load&evaluate.py’, ‘test_MTMC_other_models.py’) and several related folders are included.

The source code was tested with a Windows 10 64bit computer equipped with Intel i7-8700K CPU and NVIDIA GeForce GTX 1080 GPU, and it should be compatible with any other machines as long as the system requirements are fulfilled.

1 System Requirements

In order to run the code, the following software tools are required to be installed in a 64bit computer:

- Python 3.6.6 or above.

Other vital site-packages for python are also needed:

- Keras (version 2.3.1 recommended);
- TensorFlow (version 1.12.0 recommended);
- NumPy;
- SciPy;
- Matplotlib;
- Pandas;
- CSV;
- PyWavelets;
- sklearn;
- pyhht.
- ……

If any problem appears while running the program, please check whether these packages or any other basic packages are properly installed.

2 Air Quality Prediction

In ‘AQI_Predction_MTMC’ folder there are:

- test_MTMC.py: This file is the main function file for multi-task multi-channel (MTMC) model. This runnable python file contains all it needs to conduct a full experiment using MTMC model. Several parameters can be set in it to modify the settings of the training & forecasting process. The result and evaluation of the experiment is saved in ‘saved/MTMC’ folder and can be loaded and visualized by running ‘test_MTMC_load&evaluate.py’ file.
- test_MTMC_other_models.py: This file is the main function file for other state-of-art time series forecasting models. This runnable python file contains all it needs to conduct a full experiment using other model. Several parameters can be set in it to modify the settings of the training & forecasting process and the AQI variable to predict. The result and evaluation of the experiment is saved in ‘saved/MTMC’ folder and can be loaded and visualized by running ‘test_MTMC_load&evaluate.py’.
- test_MTMC_load&evaluate.py: This file contains several functions to evaluate and compare the performance of different methods of time series forecasting. This file is only to load the results and visualize them, thus it should be run after the experiment fully completed by running ‘test_MTMC.py’ file and ‘test_MTMC_other_models.py’ file.
- ‘dataset’ folder: This folder contains the used AQI dataset of 12 observing stations. The data main function needs to train and predict is read from the CSV files in here.
- ‘Part’ folder: This folder contains two python files, which are ‘part_data_preprocessing.py’ and ‘part_evaluate.py’. ‘part_data_preprocessing.py’ file stores several functions the main functions need to process the data read from the CSV files. ‘part_evaluate.py’ file contains several functions for evaluating the forecasting performance.
- ‘Support’ folder: This folder contains several supportive files, which are related to wavelet transform function, NLSTM neural network, VMD and PCC calculation, respectively.
- ‘Model’ folder: This folder contains ‘model_major.py’ file, which contains built deep learning model in it.
- ‘result’ folder: This folder contains CSV files created after running ‘test_MTMC_load&evaluate.py’ file, in which the results of evaluation to the experiment are recorded.

- ‘saved’ folder: This folder contains several NPY files. These files are generated after conducting the experiments to store the results and evaluations, and can be read by running ‘test_MTMC_load&evaluate.py’ file.

3 test_MTMC.py

File ‘test_MTMC.py’ contains two models for AQI time series prediction. It is a runnable file in this project and provides the result of the contained methods.

Its ‘neo_prediction’ function starts on line 664 as the main function in this file. The multivariate AQI record data is loaded by function ‘load_data_general’ on line 676.

Lines 704-716 are functions of the two prediction models. These functions contains training, testing and evaluating processes, thus the prediction results and the evaluations are obtained by running them. The prediction results are returned respectively by these functions, the evaluations are recorded in two global variables ‘eva_output’ and ‘result_all’. The functions include:

- Function ‘MTMC_wvlt’ on line 704: the method of Multi-Task Multi-Channel WNLSTM framework.
- Function ‘MT’ on line 711: the method of Multi-Task NLSTM framework.

4 test_MTMC_other_models.py

File ‘test_MTMC_other_models.py’ is the main function in this project, it is contains several models for AQI time series prediction. This file contains all the functions including those defined in file ‘test_MTMC.py’. Therefore, running ‘test_MTMC_other_models.py’ provides all the result the user needs.

On line 1134, the function ‘neo_prediction’ is defined in file ‘test_MTMC.py’, and contains two models. On lines 1135-1140, function ‘main’ is called multiple times to perform prediction on different AQI variables.

‘main’ function starts on line 984, the size of training set and testing set is set on lines 1019-1020. The AQI record data is loaded and preprocessed on lines 1026-1048.

- Function ‘load_data_ts’ on line 1045 provides regular training set and testing set.
- Function ‘load_data_emd’ on line 1046 provides training set and testing set based on data decomposed using empirical mode decomposition (EMD).

- Function ‘load_data_wvlt’ on line 1047 provides training set and testing set based on data decomposed using wavelet transform decomposition (WT).
- Function ‘load_data_VMD’ on line 1048 provides training set and testing set based on data decomposed using variational mode decomposition (VMD).

Lines 1068-1102 are functions of the prediction models. These functions contains training, testing and evaluating processes, thus the prediction results and the evaluations are obtained by running them. The prediction results are returned respectively by these functions, the evaluations are recorded in two global variables ‘eva_output’ and ‘result_all’. The functions include:

- Function ‘Decide_Tree’ on line 1068: the method of Decision Tree.
- Function ‘Random_forest’ on line 1070: the method of Random forest.
- Function ‘SVR’ on line 1072: the method of Support Vector Machine Regression.
- Function ‘MLP’ on line 1074: the method of Multilayer Perception.
- Function ‘LSTM’ on line 1077: the method of LSTM.
- Function ‘EMD_LSTM’ on line 1079: the hybrid method of LSTM and EMD.
- Function ‘Wavelet_LSTM’ on line 1081: the hybrid method of LSTM and WT.
- Function ‘VMD_LSTM’ on line 1083: the hybrid method of LSTM and VMD.
- Function ‘Nested_LSTM’ on line 1086: the method of NLSTM.
- Function ‘EMD_NLSTM’ on line 1088: the hybrid method of NLSTM and EMD.
- Function ‘Wavelet_NLSTM’ on line 1090: the hybrid method of NLSTM and WT.
- Function ‘VMD_NLSTM’ on line 1092: the hybrid method of NLSTM and VMD.
- Function ‘Stacked_LSTM’ on line 1095: the method of SLSTM.
- Function ‘EMD_SLSTM’ on line 1097: the hybrid method of SLSTM and EMD.
- Function ‘Wavelet_SLSTM’ on line 1099: the hybrid method of SLSTM and WT.
- Function ‘VMD_SLSTM’ on line 1101: the hybrid method of SLSTM and VMD.

After running this file, the evaluation is printed and saved in CSV files in ‘result’ folder.