

TensorFlow configuration

This appendix describes step by step the installation of TensorFlow 1.15 as well as the packages needed to run the TimeGAN framework.

1. TensorFlow version

TensorFlow is an open source Python framework developed by Google, which is used to perform large-scale machine learning models calculations, whose first release dates back to 2015.

2. Development IDE

The Anaconda Navigator 2.3.0 IDE along with PyCharm Professional Edition 2022.2.1 were used for this work.

3. Environment configuration

The version of TensorFlow used in this work is 1.15, the same as the code base of the TimeGAN framework. Since the version of TensorFlow is not the latest (2.11 to date), it is required to create an exclusive environment so that TimeGAN does not interfere with other projects using current versions of this or other Python packages.

3.1 Python and TensorFlow versions

TensorFlow version 1.15 was used for this work, which uses the Python version 3.7.

3.2 Python environment

First of all, creating an environment exclusively for this project is recommended, so as not to alter the other existing configurations on the computer. To do this, the following command is executed:

```
conda create --name timegan python = 3.7 (1)
```

Table 1 show required packages to run TimeGAN generation processes.

Package name	Package version
numpy	1.18.4
easydict	1.9
pytz	2017.2
pillow	6.2.0
pyparsing	2.0.3
dm-tree	0.1.1
mpmath	1.2.1
protobuf	3.9.3
pandas	1.3.5
matplotlib	3.5.3
scikit-learn	1.0.2
seaborn	0.11.2

Table 1: Required packages

Then, the created environment must be activated with the following instruction:

```
conda activate timegan (2)
```

The installed version of Anaconda is updated with the command:

```
conda update -n base -c defaults conda (3)
```

It is also recommended to update the *pip* version, with the command:

```
python.exe -m pip install --upgrade pip (4)
```

3.3 Packages installation

The packages required to successfully run TimeGAN in this work are listed below.

First, the *jupyter* package is installed with the command:

```
pip install jupyter (5)
```

Any other code editor could also be used, however, the configuration shown here may require additional changes. Figure 1 shows the respective output after running the command (5).

```

C:\Windows\system32\cmd.exe
Using cached cffi-1.15.1-cp37-cp37m-win_amd64.whl (179 kB)
Collecting pycparser
Using cached pycparser-2.21-py2.py3-none-any.whl (118 kB)
Installing collected packages: webencodings, wcwidth, Send2Trash, pywin32, pickleshare, mistune, ipython-genutils, fastj
sonschema, backcall, zipp, widgetsnbextension, typing-extensions, traitlets, tornado, tinycss2, soupsieve, six, pyzmq, p
ywinpty, pyparsing, pygments, pycparser, psutil, prompt-toolkit, prometheus-client, pkgutil-resolve-name, parso, pandocfilters, nest-asyncio, lxml, jupyterlab-widgets, jupyterlab-pygments, Jinja2, entrypoints, defusedxml, decora
tor, debugpy, attrs, terminado, packaging, matplotlib-inline, jupyter-core, jedi, importlib-resources, importlib-metadata,
a, cffi, bleach, beautifulsoup4, qtpy, jupyter-client, jsonschema, ipython, argon2-cffi-bindings, nbformat, ipykernel, a
rgon2-cffi, qtconsole, nbclient, jupyter-console, ipywidgets, nbconvert, notebook, jupyter
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behavior
r is the source of the following dependency conflicts.
pandas 1.1.5 requires numpy>=1.15.4, which is not installed.
pandas 1.1.5 requires pytz>=2017.2, which is not installed.
matplotlib 3.3.2 requires numpy>=1.15, which is not installed.
matplotlib 3.3.2 requires pillow>=6.2.0, which is not installed.
keras-preprocessing 1.1.2 requires numpy>=1.9.1, which is not installed.
h5py 2.10.0 requires numpy>=1.7, which is not installed.
Successfully installed Send2Trash-1.8.0 argon2-cffi-21.3.0 argon2-cffi-bindings-21.2.0 attrs-22.1.0 backcall-0.2.0 beaut
ifulsoup4-4.11.1 bleach-5.0.1 cffi-1.15.1 debugpy-1.6.3 decorator-5.1.1 defusedxml-0.7.1 entrypoints-0.4 fastjsonschema-
2.16.1 importlib-metadata-4.12.0 importlib-resources-5.9.0 ipykernel-6.15.1 ipython-7.34.0 ipython-genutils-0.2.0 ipywid
gets-8.0.1 jedi-0.18.1 Jinja2-3.1.2 jsonschema-4.14.0 jupyter-1.0.0 jupyter-client-7.3.5 jupyter-console-6.4.4 jupyter-c
ore-4.11.1 jupyterlab-pygments-0.2.2 jupyterlab-widgets-3.0.2 lxml-4.9.1 matplotlib-inline-0.1.6 mistune-2.0.4 nbclient-
0.6.7 nbconvert-7.0.0 nbformat-5.4.0 nest-asyncio-1.5.5 notebook-6.4.12 packaging-21.3 pandocfilters-1.5.0 parso-0.8.3 p
ickleshare-0.7.5 pkgutil-resolve-name-1.3.10 prometheus-client-0.14.1 prompt-toolkit-3.0.30 psutil-5.9.1 pycparser-2.21
pygments-2.13.0 pyparsing-3.0.9 pyrsistent-0.18.1 pywin32-304 pywinpty-2.0.7 pyzmq-23.2.1 qtconsole-5.3.1 qtpy-2.2.0 six
-1.16.0 soupsieve-2.3.2.post1 terminado-0.15.0 tinycss2-1.1.1 tornado-6.2 traitlets-5.3.0 typing-extensions-4.3.0 wcwidt
h-0.2.5 webencodings-0.5.1 widgetsnbextension-4.0.2 zipp-3.8.1
(timegan) C:\Users\Diego Tamayo>

```

Figure 1: *jupyter* package installation

Next, the *numpy* package is installed with the following command:

```
pip install numpy==1.18.4 (6)
```

Figure 2 shows the respective output after running the command (6).

```

C:\Windows\system32\cmd.exe - conda update -n base -c defaults conda - conda install spyder - conda install jupyter
nbclassic pkgs/main/noarch::nbclassic-0.3.5-pyhd3eb1b0_0
notebook pkgs/main/win-64::notebook-6.4.12-py37h9a95532_0
prometheus_client pkgs/main/win-64::prometheus_client-0.14.1-py37h9a95532_0
prompt_toolkit pkgs/main/noarch::prompt_toolkit-3.0.20-hd3eb1b0_0
pywinpty pkgs/main/win-64::pywinpty-2.0.2-py37h5da7b33_0
send2trash pkgs/main/noarch::send2trash-1.8.0-pyhd3eb1b0_1
sniffio pkgs/main/win-64::sniffio-1.2.0-py37h9a95532_1
terminado pkgs/main/win-64::terminado-0.13.1-py37h9a95532_0
websocket-client pkgs/main/win-64::websocket-client-0.58.0-py37h9a95532_4
widgetsnbextension pkgs/main/win-64::widgetsnbextension-3.5.2-py37h9a95532_0
winpty pkgs/main/win-64::winpty-0.4.3-4

Proceed ([y]/n)? y
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Retrieving notices: ...working... done

(timegan) C:\Users\Diego Tamayo>pip install numpy==1.18.4
Collecting numpy==1.18.4
Using cached numpy-1.18.4-cp37-cp37m-win_amd64.whl (12.8 MB)
Installing collected packages: numpy
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behavior
r is the source of the following dependency conflicts.
ydata-synthetic 0.2.0 requires easydict==1.9, which is not installed.
ydata-synthetic 0.2.0 requires scikit-learn==0.22.*, which is not installed.
ydata-synthetic 0.2.0 requires tensorflow==2.3.*, which is not installed.
matplotlib 3.3.2 requires pillow>=6.2.0, which is not installed.

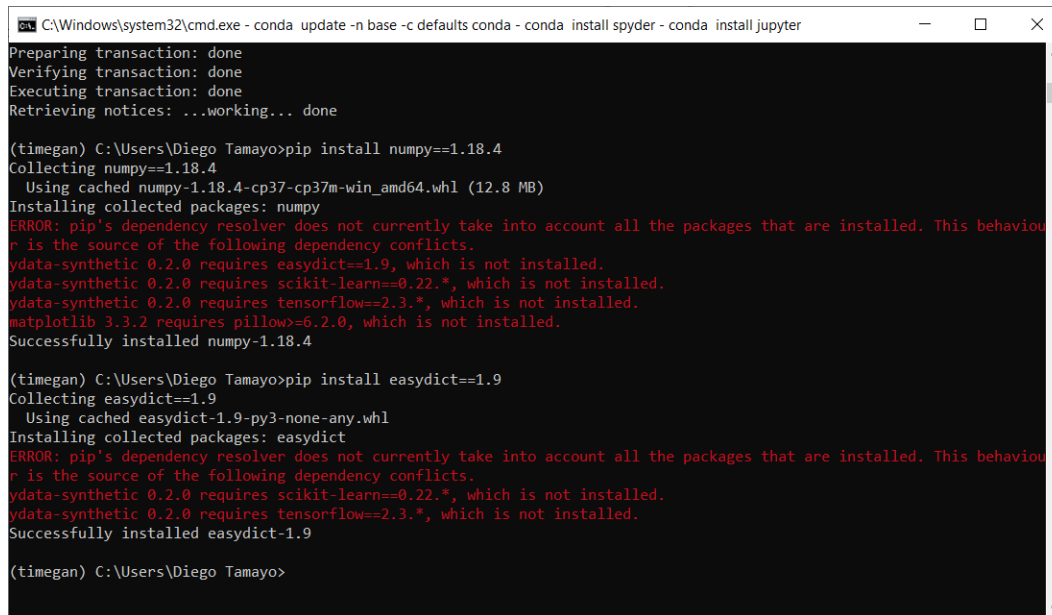
```

Figure 2: *numpy* package installation

Then, the *easydict* package is installed with the following command:

```
pip install easydict==1.9 (7)
```

Figure 3 shows the respective output after running the command (7).



```
C:\Windows\system32\cmd.exe - conda update -n base -c defaults conda - conda install spyder - conda install jupyter
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Retrieving notices: ...working... done

(timegan) C:\Users\Diego Tamayo>pip install numpy==1.18.4
Collecting numpy==1.18.4
  Using cached numpy-1.18.4-cp37-cp37m-win_amd64.whl (12.8 MB)
Installing collected packages: numpy
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behavior
is the source of the following dependency conflicts.
ydata-synthetic 0.2.0 requires easydict==1.9, which is not installed.
ydata-synthetic 0.2.0 requires scikit-learn==0.22.*, which is not installed.
ydata-synthetic 0.2.0 requires tensorflow==2.3.*, which is not installed.
matplotlib 3.3.2 requires pillow>=6.2.0, which is not installed.
Successfully installed numpy-1.18.4

(timegan) C:\Users\Diego Tamayo>pip install easydict==1.9
Collecting easydict==1.9
  Using cached easydict-1.9-py3-none-any.whl
Installing collected packages: easydict
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behavior
is the source of the following dependency conflicts.
ydata-synthetic 0.2.0 requires scikit-learn==0.22.*, which is not installed.
ydata-synthetic 0.2.0 requires tensorflow==2.3.*, which is not installed.
Successfully installed easydict-1.9

(timegan) C:\Users\Diego Tamayo>
```

Figure 3: *easydict* package installation

Then, it is installed the *pytz* package with the following command:

```
pip install pytz==2017.2 (8)
```

Figure 4 shows the respective output after running the command (8).

Next, the *pillow* package is installed with the following command:

```
pip install pillow==6.2.0 (9)
```

Figure 5 shows the respective output after running the command (9).

Next, the *pyparsing* package is installed with the following command:

```
pip install pyparsing==2.0.3 (10)
```

```
C:\Windows\system32\cmd.exe - conda update -n base -c defaults conda - conda install spyder - conda install jupyter
Installing collected packages: numpy
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behavior
r is the source of the following dependency conflicts.
ydata-synthetic 0.2.0 requires easydict==1.9, which is not installed.
ydata-synthetic 0.2.0 requires scikit-learn==0.22.*, which is not installed.
ydata-synthetic 0.2.0 requires tensorflow==2.3.*, which is not installed.
matplotlib 3.3.2 requires pillow>=6.2.0, which is not installed.
Successfully installed numpy-1.18.4

(timegan) C:\Users\Diego Tamayo>pip install easydict==1.9
Collecting easydict==1.9
  Using cached easydict-1.9-py3-none-any.whl
Installing collected packages: easydict
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behavior
r is the source of the following dependency conflicts.
ydata-synthetic 0.2.0 requires scikit-learn==0.22.*, which is not installed.
ydata-synthetic 0.2.0 requires tensorflow==2.3.*, which is not installed.
Successfully installed easydict-1.9

(timegan) C:\Users\Diego Tamayo>pip install pytz==2017.2
Collecting pytz==2017.2
  Using cached pytz-2017.2-py2.py3-none-any.whl (484 kB)
Installing collected packages: pytz
  Attempting uninstall: pytz
    Found existing installation: pytz 2022.1
    Uninstalling pytz-2022.1:
      Successfully uninstalled pytz-2022.1
Successfully installed pytz-2017.2

(timegan) C:\Users\Diego Tamayo>
```

Figure 4: *pytz* package installation

```
C:\Windows\system32\cmd.exe - conda update -n base -c defaults conda - conda install spyder - conda install jupyter
matplotlib 3.3.2 requires pillow>=6.2.0, which is not installed.
Successfully installed numpy-1.18.4

(timegan) C:\Users\Diego Tamayo>pip install easydict==1.9
Collecting easydict==1.9
  Using cached easydict-1.9-py3-none-any.whl
Installing collected packages: easydict
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behavior
r is the source of the following dependency conflicts.
ydata-synthetic 0.2.0 requires scikit-learn==0.22.*, which is not installed.
ydata-synthetic 0.2.0 requires tensorflow==2.3.*, which is not installed.
Successfully installed easydict-1.9

(timegan) C:\Users\Diego Tamayo>pip install pytz==2017.2
Collecting pytz==2017.2
  Using cached pytz-2017.2-py2.py3-none-any.whl (484 kB)
Installing collected packages: pytz
  Attempting uninstall: pytz
    Found existing installation: pytz 2022.1
    Uninstalling pytz-2022.1:
      Successfully uninstalled pytz-2022.1
Successfully installed pytz-2017.2

(timegan) C:\Users\Diego Tamayo>pip install pillow==6.2.0
Collecting pillow==6.2.0
  Using cached Pillow-6.2.0-cp37-cp37m-win_amd64.whl (2.0 MB)
Installing collected packages: pillow
Successfully installed pillow-6.2.0

(timegan) C:\Users\Diego Tamayo>
```

Figure 5: *pillow* package installation

Figure 6 shows the respective output after running the command (10).

The next step is to install the *dm-tree* package with the following command:

```
pip install dm-tree==0.1.1 (11)
```

```
C:\Windows\system32\cmd.exe - conda update -n base -c defaults conda - conda install spyder - conda install jupyter
ydata-synthetic 0.2.0 requires tensorflow==2.3.*, which is not installed.
Successfully installed easydict-1.9

(timegan) C:\Users\Diego Tamayo>pip install pytz==2017.2
Collecting pytz==2017.2
  Using cached pytz-2017.2-py2.py3-none-any.whl (484 kB)
Installing collected packages: pytz
  Attempting uninstall: pytz
    Found existing installation: pytz 2022.1
    Uninstalling pytz-2022.1:
      Successfully uninstalled pytz-2022.1
Successfully installed pytz-2017.2

(timegan) C:\Users\Diego Tamayo>pip install pillow==6.2.0
Collecting pillow==6.2.0
  Using cached Pillow-6.2.0-cp37-cp37m-win_amd64.whl (2.0 MB)
Installing collected packages: pillow
Successfully installed pillow-6.2.0

(timegan) C:\Users\Diego Tamayo>pip install pyparsing==2.0.3
Collecting pyparsing==2.0.3
  Using cached pyparsing-2.0.3-py2.py3-none-any.whl (37 kB)
Installing collected packages: pyparsing
  Attempting uninstall: pyparsing
    Found existing installation: pyparsing 3.0.4
    Uninstalling pyparsing-3.0.4:
      Successfully uninstalled pyparsing-3.0.4
Successfully installed pyparsing-2.0.3

(timegan) C:\Users\Diego Tamayo>
```

Figure 6: *pyparsing* package installation

Figure 7 shows the respective output after running the command (11).

```
C:\Windows\system32\cmd.exe - conda update -n base -c defaults conda - conda install spyder - conda install jupyter

(timegan) C:\Users\Diego Tamayo>pip install pillow==6.2.0
Collecting pillow==6.2.0
  Using cached Pillow-6.2.0-cp37-cp37m-win_amd64.whl (2.0 MB)
Installing collected packages: pillow
Successfully installed pillow-6.2.0

(timegan) C:\Users\Diego Tamayo>pip install pyparsing==2.0.3
Collecting pyparsing==2.0.3
  Using cached pyparsing-2.0.3-py2.py3-none-any.whl (37 kB)
Installing collected packages: pyparsing
  Attempting uninstall: pyparsing
    Found existing installation: pyparsing 3.0.4
    Uninstalling pyparsing-3.0.4:
      Successfully uninstalled pyparsing-3.0.4
Successfully installed pyparsing-2.0.3

(timegan) C:\Users\Diego Tamayo>pip install dm-tree==0.1.1
Collecting dm-tree==0.1.1
  Using cached dm_tree-0.1.1-cp37-cp37m-win_amd64.whl (84 kB)
Requirement already satisfied: six>=1.12.0 in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (from dm-tree==0.1.1) (1.16.0)
Installing collected packages: dm-tree
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.
tensorflow-privacy 0.5.1 requires mpmath, which is not installed.
tensorflow-privacy 0.5.1 requires tensorflow-estimator>=2.3.0, which is not installed.
Successfully installed dm-tree-0.1.1

(timegan) C:\Users\Diego Tamayo>
```

Figure 7: *dm-tree* package installation

The next step is to install the *mpmath* package with the following command:

```
pip install mpmath (12)
```

Figure 8 shows the respective output after running the command (12).

```

C:\Windows\system32\cmd.exe - conda update -n base -c defaults conda - conda install spyder - conda install jupyter

(timegan) C:\Users\Diego Tamayo>pip install pillow==6.2.0
Collecting pillow==6.2.0
  Using cached Pillow-6.2.0-cp37-cp37m-win_amd64.whl (2.0 MB)
Installing collected packages: pillow
Successfully installed pillow-6.2.0

(timegan) C:\Users\Diego Tamayo>pip install pyparsing==2.0.3
Collecting pyparsing==2.0.3
  Using cached pyparsing-2.0.3-py2.py3-none-any.whl (37 kB)
Installing collected packages: pyparsing
  Attempting uninstall: pyparsing
    Found existing installation: pyparsing 3.0.4
    Uninstalling pyparsing-3.0.4:
      Successfully uninstalled pyparsing-3.0.4
Successfully installed pyparsing-2.0.3

(timegan) C:\Users\Diego Tamayo>pip install dm-tree==0.1.1
Collecting dm-tree==0.1.1
  Using cached dm_tree-0.1.1-cp37m-win_amd64.whl (84 kB)
Requirement already satisfied: six>=1.12.0 in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (from dm-tree==0.1.1) (1.16.0)
Installing collected packages: dm-tree
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.
tensorflow-privacy 0.5.1 requires mpmath, which is not installed.
tensorflow-privacy 0.5.1 requires tensorflow-estimator>=2.3.0, which is not installed.
Successfully installed dm-tree-0.1.1

(timegan) C:\Users\Diego Tamayo>

```

Figure 8: *mpmath* package installation

Once the necessary packages have been installed, the *tensorflow* framework is installed with the following command:

```
pip install tensorflow==1.15 (13)
```

Figure 9 shows the respective output after *tensorflow* has been installed.

```

C:\Windows\system32\cmd.exe - conda update -n base -c defaults conda - conda install spyder - conda install jupyter - conda inst...

oauthlib-3.2.0 | 130 KB | ##### | 100%
tensorflow-estimator | 267 KB | ##### | 100%
tensorboard-data-server | 17 KB | ##### | 100%
icc_rt-2019.0.0 | 6.0 MB | ##### | 100%
google-pasta-0.2.0 | 46 KB | ##### | 100%
protobuf-3.20.1 | 229 KB | ##### | 100%
opt_einsum-3.3.0 | 57 KB | ##### | 100%
grpcio-1.42.0 | 1.8 MB | ##### | 100%
termcolor-1.1.0 | 9 KB | ##### | 100%
pyjwt-2.4.0 | 38 KB | ##### | 100%
tensorflow-base-1.15 | 34.3 MB | ##### | 100%
absl-py-0.15.0 | 103 KB | ##### | 100%
yarl-1.8.1 | 80 KB | ##### | 100%
keras-preprocessing | 35 KB | ##### | 100%
markdown-3.3.4 | 144 KB | ##### | 100%
_tflow_select-2.2.0 | 3 KB | ##### | 100%
tensorboard-plugin-w | 671 KB | ##### | 100%
tensorflow-1.15.0 | 4 KB | ##### | 100%
requests-oauthlib-1. | 23 KB | ##### | 100%
h5py-3.7.0 | 800 KB | ##### | 100%
blinker-1.4 | 23 KB | ##### | 100%
keras-applications-1 | 29 KB | ##### | 100%
werkzeug-0.16.1 | 258 KB | ##### | 100%
scipy-1.7.3 | 13.8 MB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Retrieving notices: ...working... done

(timegan) C:\Users\Diego Tamayo>

```

Figure 9: *tensorflow* framework installation

To avoid future problems, the existing version of the *protobuf* package must be uninstalled, which is done with the following command:

```
pip uninstall protobuf (14)
```

Figure 10 shows the respective output after running the command (14).

```

C:\Windows\system32\cmd.exe - conda update -n base -c defaults conda - conda install spyder - conda install jupyter - conda inst...
termcolor-1.1.0 | 9 KB | ##### 100%
pyjwt-2.4.0 | 38 KB | ##### 100%
tensorflow-base-1.15 | 34.3 MB | ##### 100%
absl-py-0.15.0 | 103 KB | ##### 100%
yarl-1.8.1 | 80 KB | ##### 100%
keras-preprocessing | 35 KB | ##### 100%
markdown-3.3.4 | 144 KB | ##### 100%
_tflow_select-2.2.0 | 3 KB | ##### 100%
tensorboard-plugin-w | 671 KB | ##### 100%
tensorflow-1.15.0 | 4 KB | ##### 100%
requests-oauthlib-1. | 23 KB | ##### 100%
h5py-3.7.0 | 800 KB | ##### 100%
blinker-1.4 | 23 KB | ##### 100%
keras-applications-1 | 29 KB | ##### 100%
werkzeug-0.16.1 | 258 KB | ##### 100%
scipy-1.7.3 | 13.8 MB | ##### 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Retrieving notices: ...working... done

(timegan) C:\Users\Diego Tamayo>pip uninstall protobuf
Found existing installation: protobuf 3.20.1
Uninstalling protobuf-3.20.1:
  Would remove:
    c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages\protobuf-3.20.1-py3.7.egg-info
Proceed (Y/n)? y
Successfully uninstalled protobuf-3.20.1
(timegan) C:\Users\Diego Tamayo>

```

Figure 10: *protobuf* package uninstallation

The appropriate version of the *protobuf* package must now be installed with the following command:

```
pip install protobuf==3.9.0 (15)
```

Figure 11 shows the respective output after running the command (15).


```
C:\Windows\system32\cmd.exe - conda update -n base -c defaults conda - conda install spyder - conda install jupyter - conda inst...
werkzeug-0.16.1 | 258 KB | ##### | 100%
scipy-1.7.3 | 13.8 MB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Retrieving notices: ...working... done

(timegan) C:\Users\Diego Tamayo>pip uninstall protobuf
Found existing installation: protobuf 3.20.1
Uninstalling protobuf-3.20.1:
  Would remove:
    c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages\protobuf-3.20.1-py3.7.egg-info
Proceed (Y/n)? y
  Successfully uninstalled protobuf-3.20.1

(timegan) C:\Users\Diego Tamayo>pip install protobuf==3.9.0
Collecting protobuf==3.9.0
  Using cached protobuf-3.9.0-cp37-cp37m-win_amd64.whl (1.0 MB)
Requirement already satisfied: setuptools in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (from protob
uf==3.9.0) (63.4.1)
Requirement already satisfied: six>=1.9 in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (from protobuf
==3.9.0) (1.16.0)
Installing collected packages: protobuf
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviou
r is the source of the following dependency conflicts.
tensorflow 1.15.0 requires tensorboard<1.16.0,>=1.15.0, but you have tensorboard 2.8.0 which is incompatible.
tensorflow 1.15.0 requires tensorflow-estimator==1.15.1, but you have tensorflow-estimator 2.6.0 which is incompatible.
Successfully installed protobuf-3.9.0

(timegan) C:\Users\Diego Tamayo>
```

Figure 11: *protobuf* package installation

To find out which version of TensorFlow is installed, use the following command:

```
python -m pip show tensorflow (16)
```

Figure 12 shows the respective output after running the command (16).

```
C:\Windows\system32\cmd.exe - conda update -n base -c defaults conda - conda install spyder - conda install jupyter - conda inst...
Successfully uninstalled protobuf-3.20.1

(timegan) C:\Users\Diego Tamayo>pip install protobuf==3.9.0
Collecting protobuf==3.9.0
  Using cached protobuf-3.9.0-cp37-cp37m-win_amd64.whl (1.0 MB)
Requirement already satisfied: setuptools in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (from protob
uf==3.9.0) (63.4.1)
Requirement already satisfied: six>=1.9 in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (from protobuf
==3.9.0) (1.16.0)
Installing collected packages: protobuf
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviou
r is the source of the following dependency conflicts.
tensorflow 1.15.0 requires tensorboard<1.16.0,>=1.15.0, but you have tensorboard 2.8.0 which is incompatible.
tensorflow 1.15.0 requires tensorflow-estimator==1.15.1, but you have tensorflow-estimator 2.6.0 which is incompatible.
Successfully installed protobuf-3.9.0

(timegan) C:\Users\Diego Tamayo>python -m pip show tensorflow
Name: tensorflow
Version: 1.15.0
Summary: TensorFlow is an open source machine learning framework for everyone.
Home-page: https://www.tensorflow.org/
Author: Google Inc.
Author-email: packages@tensorflow.org
License: Apache 2.0
Location: c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages
Requires: absl-py, astor, gast, google-pasta, grpcio, keras-applications, keras-preprocessing, numpy, opt-einsum, protob
uf, six, tensorboard, tensorflow-estimator, termcolor, wheel, wrapt
Required-by: ydata-synthetic

(timegan) C:\Users\Diego Tamayo>
```

Figure 12: Tensorflow version information

It is also possible to find out the installed version of Tensorflow from the console of a Python interpreter. To do this we first run the following code:

```
import tensorflow as tf
tf.version
```

In order to be able to use the specific Python kernel for the environment created in Anaconda, which is associated with Tensorflow, the *ipykernel* package is required. Specifically, this package is required when working with *Jupyter Notebook*, as it associates the kernel with the corresponding environment. The *ipykernel* package must be installed with the following command:

```
conda install -c anaconda ipykernel (17)
```

Figure 13 shows the respective output after running the command (17).

```
C:\Windows\system32\cmd.exe - conda update -n base -c defaults conda - conda install spyder - conda install jupyter - conda inst...
The following packages will be downloaded:

package | build | size | channel
-----|-----|-----|-----
certifi-2022.6.15 | py37h... | 157 KB | anaconda
ipykernel-6.9.1 | py37h... | 196 KB | anaconda
-----|-----|-----|-----
Total: | 353 KB |

The following packages will be SUPERSEDED by a higher-priority channel:

ca-certificates pkgs/main::ca-certificates-2022.07.19~ --> anaconda::ca-certificates-2022.4.26-haa95532_0
certifi pkgs/main --> anaconda
ipykernel pkgs/main --> anaconda

Proceed ([y]/n)? y

Downloading and Extracting Packages
ipykernel-6.9.1 | 196 KB | ##### | 100%
certifi-2022.6.15 | 157 KB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Retrieving notices: ...working... done
(timegan) C:\Users\Diego Tamayo>
```

Figure 13: *ipykernel* package installation

In case it is necessary to uninstall a particular kernel, it can be done with the following command:

```
jupyter kernelspec uninstall name-unwanted-kernel (18)
```

Figure 14 shows the respective output after running the command (18).

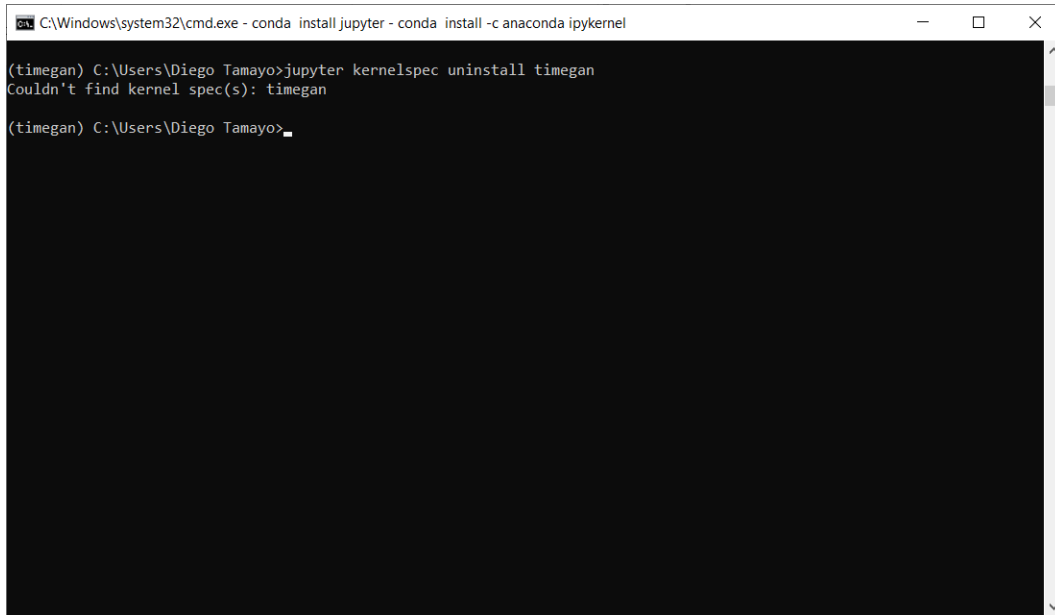
Once the *ipykernel* package is installed, it is possible to create a kernel to associate with the environment created for TensorFlow, which will be called *timegan*. The command used for is:

```
python -m ipykernel install --user --name=timegan (19)
```

Figure 15 shows the respective output after running the command (19).

The *pandas* package is very useful because of its methods for manipulating datasets. This package is installed with the following command:

```
pip install pandas (20)
```

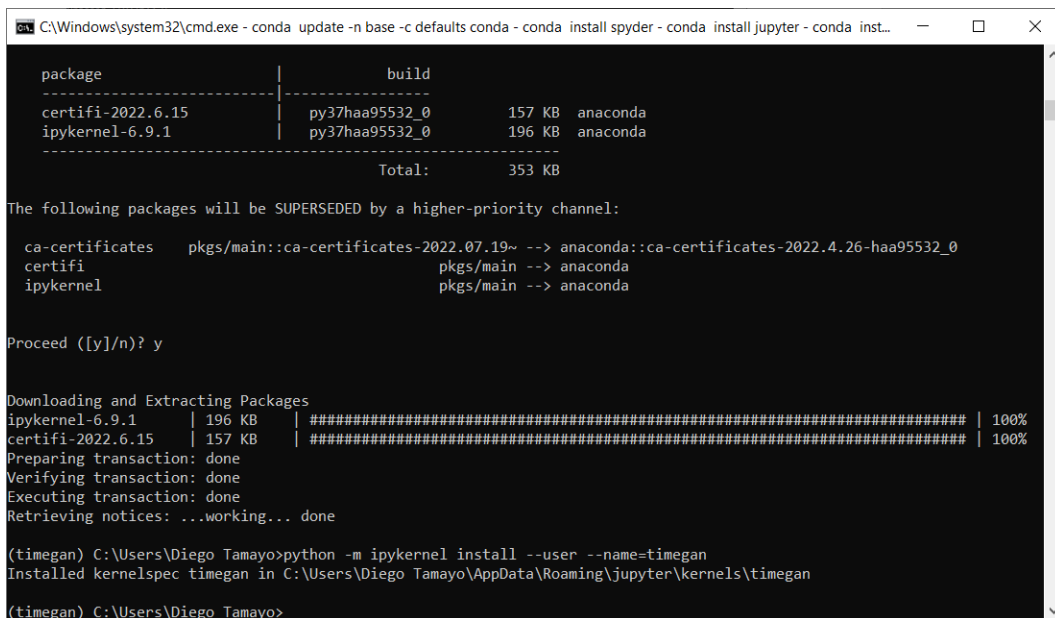
A terminal window titled "C:\Windows\system32\cmd.exe - conda install jupyter - conda install -c anaconda ipykernel". The prompt is "(timegan) C:\Users\Diego Tamayo>". The user enters "jupyter kernelspec uninstall timegan". The output is "Couldn't find kernel spec(s): timegan". The prompt returns to "(timegan) C:\Users\Diego Tamayo>".

```
C:\Windows\system32\cmd.exe - conda install jupyter - conda install -c anaconda ipykernel

(timegan) C:\Users\Diego Tamayo>jupyter kernelspec uninstall timegan
Couldn't find kernel spec(s): timegan

(timegan) C:\Users\Diego Tamayo>
```

Figure 14: Kernel uninstallation command

A terminal window titled "C:\Windows\system32\cmd.exe - conda update -n base -c defaults conda - conda install spyder - conda install jupyter - conda inst...". The prompt is "(timegan) C:\Users\Diego Tamayo>". The user enters "python -m ipykernel install --user --name=timegan". The output shows a table of packages to be installed, a warning about superseded packages, and the successful installation of the kernel. The prompt returns to "(timegan) C:\Users\Diego Tamayo>".

```
C:\Windows\system32\cmd.exe - conda update -n base -c defaults conda - conda install spyder - conda install jupyter - conda inst...

package | build | size | channel
-----|-----|-----|-----
certifi-2022.6.15 | py37h... | 157 KB | anaconda
ipykernel-6.9.1 | py37h... | 196 KB | anaconda
-----|-----|-----|-----
Total: | | 353 KB |

The following packages will be SUPERSEDED by a higher-priority channel:

ca-certificates pkgs/main::ca-certificates-2022.07.19~ --> anaconda::ca-certificates-2022.4.26-haa95532_0
certifi pkgs/main --> anaconda
ipykernel pkgs/main --> anaconda

Proceed ([y]/n)? y

Downloading and Extracting Packages
ipykernel-6.9.1 | 196 KB | ##### | 100%
certifi-2022.6.15 | 157 KB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Retrieving notices: ...working... done

(timegan) C:\Users\Diego Tamayo>python -m ipykernel install --user --name=timegan
Installed kernelspec timegan in C:\Users\Diego Tamayo\AppData\Roaming\jupyter\kernels\timegan

(timegan) C:\Users\Diego Tamayo>
```

Figure 15: Kernel creation using *ipykernel*

Figure 16 shows the respective output after running the command (20).

```
C:\Windows\system32\cmd.exe - conda update -n base -c defaults conda - conda install spyder - conda install jupyter - conda inst...
certifi                                pkgs/main --> anaconda
ipykernel                              pkgs/main --> anaconda

Proceed ([y]/n)? y

Downloading and Extracting Packages
ipykernel-6.9.1      | 196 KB | ##### | 100%
certifi-2022.6.15   | 157 KB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Retrieving notices: ...working... done

(timegan) C:\Users\Diego Tamayo>python -m ipykernel install --user --name=timegan
Installed kernelspec timegan in C:\Users\Diego Tamayo\AppData\Roaming\jupyter\kernels\timegan

(timegan) C:\Users\Diego Tamayo>pip install pandas
Requirement already satisfied: pandas in c:\users\diego tamayo\appdata\roaming\python\python37\site-packages (1.1.5)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\diego tamayo\appdata\roaming\python\python37\site-packag
ages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2017.2 in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (from pand
as) (2017.2)
Requirement already satisfied: numpy>=1.15.4 in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (from pan
das) (1.21.5)
Requirement already satisfied: six>=1.5 in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (from python-d
ateutil>=2.7.3->pandas) (1.16.0)

(timegan) C:\Users\Diego Tamayo>
```

Figure 16: *pandas* package installation

A very useful library for displaying graphs is *matplotlib*, which can be installed with the following command:

```
pip install matplotlib (21)
```

Figure 17 shows the respective output after running the command (21).

```
C:\Windows\system32\cmd.exe - conda update -n base -c defaults conda - conda install spyder - conda install jupyter - conda inst...
Requirement already satisfied: pytz>=2017.2 in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (from pand
as) (2017.2)
Requirement already satisfied: numpy>=1.15.4 in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (from pan
das) (1.21.5)
Requirement already satisfied: six>=1.5 in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (from python-d
ateutil>=2.7.3->pandas) (1.16.0)

(timegan) C:\Users\Diego Tamayo>pip install matplotlib
Requirement already satisfied: matplotlib in c:\users\diego tamayo\appdata\roaming\python\python37\site-packages (3.3.2)
Requirement already satisfied: certifi>=2020.06.20 in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (fr
om matplotlib) (2022.6.15)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in c:\users\diego tamayo\anaconda3\envs\timegan\
lib\site-packages (from matplotlib) (2.0.3)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\diego tamayo\appdata\roaming\python\python37\site-packag
es (from matplotlib) (2.8.2)
Requirement already satisfied: pillow>=6.2.0 in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (from mat
plotlib) (6.2.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\diego tamayo\appdata\roaming\python\python37\site-packages
(from matplotlib) (1.4.4)
Requirement already satisfied: cycler>=0.10 in c:\users\diego tamayo\appdata\roaming\python\python37\site-packages (from
matplotlib) (0.11.0)
Requirement already satisfied: numpy>=1.15 in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (from matpl
otlib) (1.21.5)
Requirement already satisfied: typing-extensions in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (from
kiwisolver>=1.0.1->matplotlib) (4.3.0)
Requirement already satisfied: six>=1.5 in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (from python-d
ateutil>=2.1->matplotlib) (1.16.0)

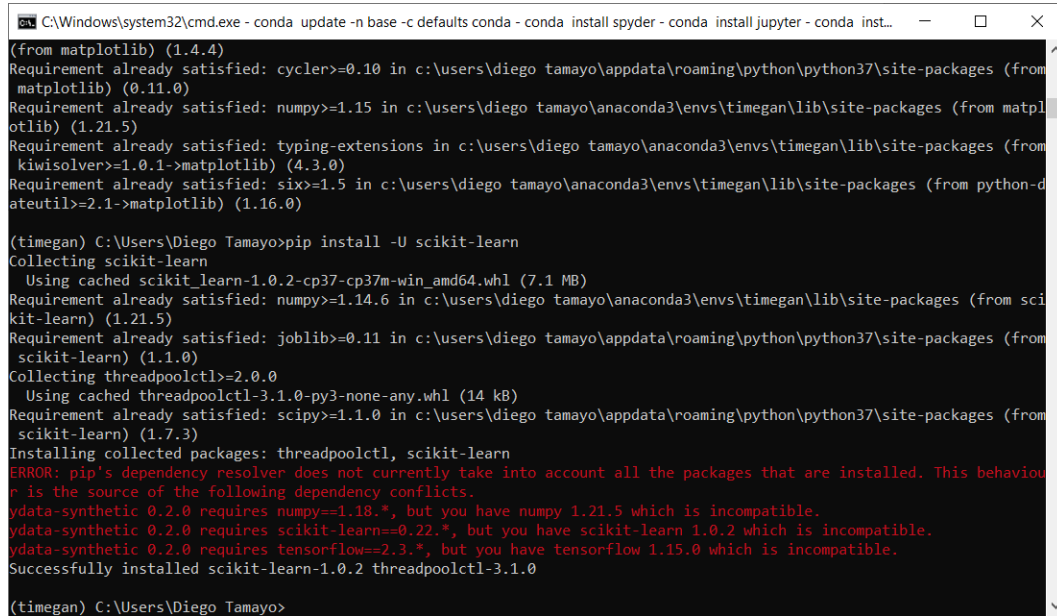
(timegan) C:\Users\Diego Tamayo>
```

Figure 17: *matplotlib* package installation

A very useful package for data analysis is *scikit-learn*, which can be installed with the command:

```
pip install -U scikit-learn (22)
```

Figure 18 shows the respective output after running the command (22).



```
(from matplotlib) (1.4.4)
Requirement already satisfied: cycler>=0.10 in c:\users\diego tamayo\appdata\roaming\python\python37\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: numpy>=1.15 in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (from matplotlib) (1.21.5)
Requirement already satisfied: typing-extensions in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (from kiwisolver>=1.0.1->matplotlib) (4.3.0)
Requirement already satisfied: six>=1.5 in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (from python-dateutil>=2.1->matplotlib) (1.16.0)

(timegan) C:\Users\Diego Tamayo>pip install -U scikit-learn
Collecting scikit-learn
  Using cached scikit_learn-1.0.2-cp37-cp37m-win_amd64.whl (7.1 MB)
Requirement already satisfied: numpy>=1.14.6 in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (from scikit-learn) (1.21.5)
Requirement already satisfied: joblib>=0.11 in c:\users\diego tamayo\appdata\roaming\python\python37\site-packages (from scikit-learn) (1.1.0)
Collecting threadpoolctl>=2.0.0
  Using cached threadpoolctl-3.1.0-py3-none-any.whl (14 kB)
Requirement already satisfied: scipy>=1.1.0 in c:\users\diego tamayo\appdata\roaming\python\python37\site-packages (from scikit-learn) (1.7.3)
Installing collected packages: threadpoolctl, scikit-learn
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.
ydata-synthetic 0.2.0 requires numpy==1.18.*, but you have numpy 1.21.5 which is incompatible.
ydata-synthetic 0.2.0 requires scikit-learn==0.22.*, but you have scikit-learn 1.0.2 which is incompatible.
ydata-synthetic 0.2.0 requires tensorflow==2.3.*, but you have tensorflow 1.15.0 which is incompatible.
Successfully installed scikit-learn-1.0.2 threadpoolctl-3.1.0

(timegan) C:\Users\Diego Tamayo>
```

Figure 18: *scikit-learn* package installation

Another library for data visualization is *seaborn*, which is based on matplotlib, but has higher quality than matplotlib. To install this package, the following command is executed:

```
pip install seaborn (23)
```

Figure 19 shows the respective output after running the command (23).

4. Code execution

An example of a command to execute the code is shown in Figure 20.

```
C:\Windows\system32\cmd.exe - conda update -n base -c defaults conda - conda install spyder - conda install jupyter - conda inst...
(timegan) C:\Users\Diego Tamayo>pip install seaborn
Requirement already satisfied: seaborn in c:\users\diego tamayo\appdata\roaming\python\python37\site-packages (0.11.2)
Requirement already satisfied: numpy>=1.15 in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (from seaborn) (1.21.5)
Requirement already satisfied: pandas>=0.23 in c:\users\diego tamayo\appdata\roaming\python\python37\site-packages (from seaborn) (1.1.5)
Requirement already satisfied: matplotlib>=2.2 in c:\users\diego tamayo\appdata\roaming\python\python37\site-packages (from seaborn) (3.3.2)
Requirement already satisfied: scipy>=1.0 in c:\users\diego tamayo\appdata\roaming\python\python37\site-packages (from seaborn) (1.7.3)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\diego tamayo\appdata\roaming\python\python37\site-packages (from matplotlib>=2.2->seaborn) (2.8.2)
Requirement already satisfied: pillow>=6.2.0 in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (from matplotlib>=2.2->seaborn) (6.2.0)
Requirement already satisfied: certifi>=2020.06.20 in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (from matplotlib>=2.2->seaborn) (2022.6.15)
Requirement already satisfied: cycler>=0.10 in c:\users\diego tamayo\appdata\roaming\python\python37\site-packages (from matplotlib>=2.2->seaborn) (0.11.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (from matplotlib>=2.2->seaborn) (2.0.3)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\diego tamayo\appdata\roaming\python\python37\site-packages (from matplotlib>=2.2->seaborn) (1.4.4)
Requirement already satisfied: pytz>=2017.2 in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (from pandas>=0.23->seaborn) (2017.2)
Requirement already satisfied: typing-extensions in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (from kiwisolver>=1.0.1->matplotlib>=2.2->seaborn) (4.3.0)
Requirement already satisfied: six>=1.5 in c:\users\diego tamayo\anaconda3\envs\timegan\lib\site-packages (from python-dateutil>=2.1->matplotlib>=2.2->seaborn) (1.16.0)
(timegan) C:\Users\Diego Tamayo>
```

Figure 19: *seaborn* package installation

```
python main_timegan.py --data_name serie_temporal
                        --seq_len 24
                        --module lstm
                        --hidden_dim 80
                        --num_layer 3
                        --iteration 100
                        --batch_size 128
                        --metric_iteration 10
```

Figure 20: Python code execution

where:

- `data_name` is the name of the data file.
- `seq_len` is the length of the sequence.
- `module` is the type of RNN network (LSTM or GRU).
- `hidden_dim` is the number of units in a LSTM or GRU cell.
- `num_layer` is the number of layers of the TimeGAN networks.
- `iteration` is the number of iterations for training.
- `batch_size` is the number of samples or sequences from each batch.
- `metric_iteration` is the number of iterations for the calculation of the metrics.