National College of Ireland

Resource Management in a Cloud Computing Environment using Generative Adversarial Networks (GANs) - Configuration Manual

MSc Research Project
Cloud Computing

Kelechukwu Chima
Student ID: 19202181

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Kelechukwu Chima
Student ID:	19202181
Programme:	Cloud Computing
Year:	2021
Module:	MSc Research Project
Supervisor:	Vikas Sahni
Submission Due Date:	16/08/2021
Project Title:	Resource Management in a Cloud Computing Environment using Generative Adversarial Networks (GANs) - Configuration Manual
Word Count:	537
Page Count:	7

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Kelechukwu Chima
Date:	16th August 2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Resource Management in a Cloud Computing Environment using Generative Adversarial Networks (GANs) - Configuration Manual

Kelechukwu Chima
19202181

1 Introduction

This configuration manual will help its readers to understand the system requirements, setup,software and hardware specifications that were used in this research. Also, this manual includes detailed explanation of the steps needed to follow when implementing this research project: Resource Allocation in Cloud Computing using GANs Algorithm.

2 System Configuration

2.1 Hardware Specification

- Model: Dell Latitude 7390
- Processor : Intel(R) Core(TM) i5- 8350U CPU@ 1.70GHz 1.90 GHz
- Operating System : Ubuntu 20.04 (Linux)
- RAM : 8.00 GB
- Hard Disk Drive : 70 gb SSD

3 Software Installation

3.1 Python

To implement and successfully perform the operations and get the results, python is used. The python software can be downloaded from https://www.python.org/downloads/. The required python version is python 3.8.

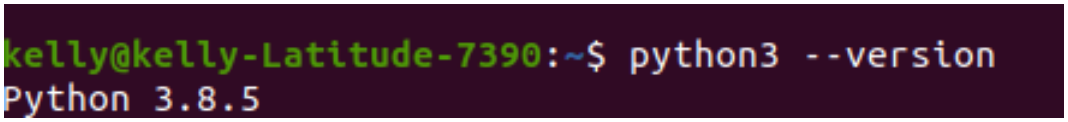

Figure 1: Python Version

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import tensorflow as tf
import math
import random
import time
import threading
import datetime
import pywt
from sklearn.linear_model import LinearRegression
```

Figure 2: List of Libraries used

3.2 The Required Python Libraries

The figure below shows the python libraries that were used in this research project. To install these libraries, the commands below are used for Linux OS;

- sudo apt install python3-pip;
- pip install –upgrade tensorflow ;
- pip install -U scikit-learn
- pip install -U matplotlib;
- pip install pandas; ;

3.3 Pip Installation

Pip Version -pip3.

Pip is a package installer for the python programming language. The "Pip" command allows you to install libraries and packages in python unto your system. The command for installation in Ubuntu is; **sudo apt install python3-pip**

```
kelly@kelly-Latitude-7390:~$ sudo apt install python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential dpkg-dev
  fakeroot g++ g++-9 gcc gcc-10-base gcc-9 libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatomic1
  libbinutils libc-dev-bin libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0
  libctf0 libexpat1-dev libfakeroot libgcc-9-dev libgcc-s1 libgomp1 libitm1
  liblsan0 libpython3-dev libpython3.8 libpython3.8-dev libpython3.8-minimal
  libpython3.8-stdlib libquadmath0 libstdc++-9-dev libstdc++6 libtsan0
  libubsan1 linux-libc-dev make manpages-dev python-pip-whl python3-dev
  python3-distutils python3-lib2to3 python3-setuptools python3-wheel
  python3.8 python3.8-dev python3.8-minimal zlib1g-dev
Suggested packages:
  binutils-doc debian-keyring g++-multilib g++-9-multilib gcc-9-doc
  gcc-multilib autoconf automake libtool flex bison gcc-doc gcc-9-multilib
  gcc-9-locales glibc-doc libstdc++-9-doc make-doc python-setuptools-doc
  python3.8-venv python3.8-doc binfmt-support
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential dpkg-dev
  fakeroot g++ g++-9 gcc gcc-9 libalgorithm-diff-perl
```

Figure 3: Pip Installation

3.4 TensorFlow

Tensor Flow version - v2.5.

The TensorFlow Library helps to create large scale neural networks, it is used mostly for machine and deep learning purposes. The command for installation in Ubuntu is; **pip install –upgrade tensorflow** This command installs or performs an update of the TensorFlow library.

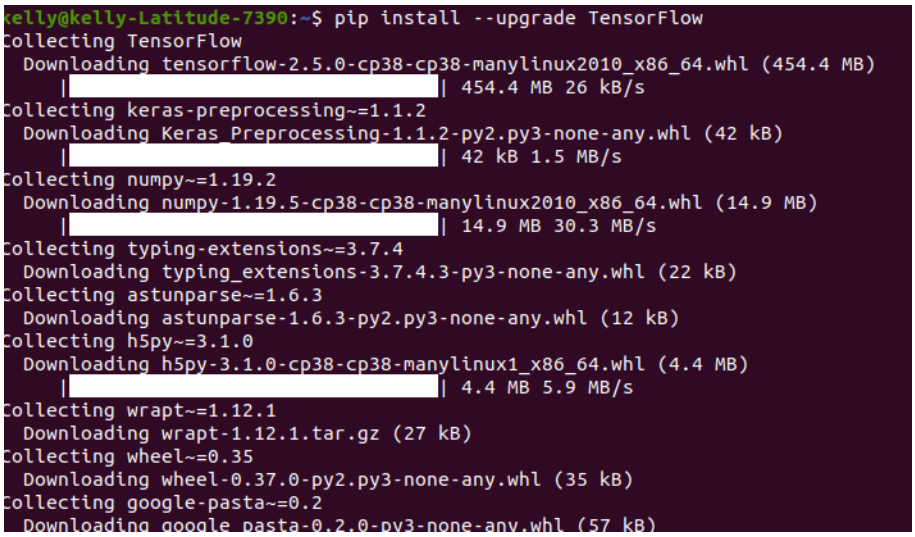

Figure 4: TensorFlow Installation

3.5 Scikit-Learn/SkLearn

Sckit-Learn Version - v0.24

The scikit library is used to build machine learning models. The command for installation in Ubuntu is; **pip install -U scikit-learn**

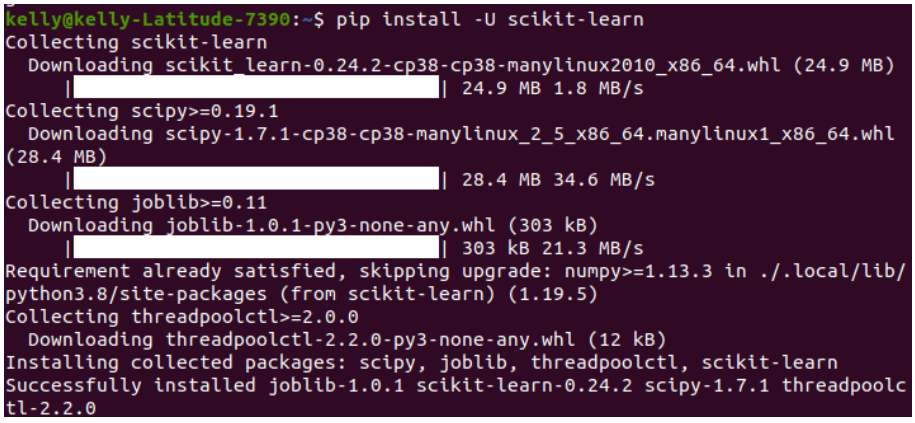

Figure 5: Sklearn Installation

3.6 MatplotLib

Mathplotlib version - v3.4

The matplotlib library is used to plot all the graphs in this project.

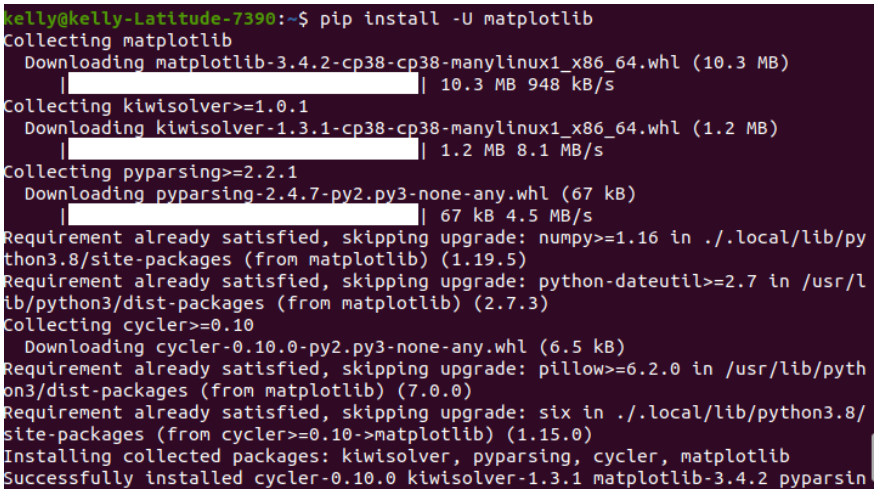

Figure 6: Mathplotlib Installation

3.7 Pandas

Pandas version - v0.24

The Pandas library allows data to be imported from different file formats such as csv, JSON,SQL and Ms Excel. It also allows data merging, cleaning and other manipulation operations.

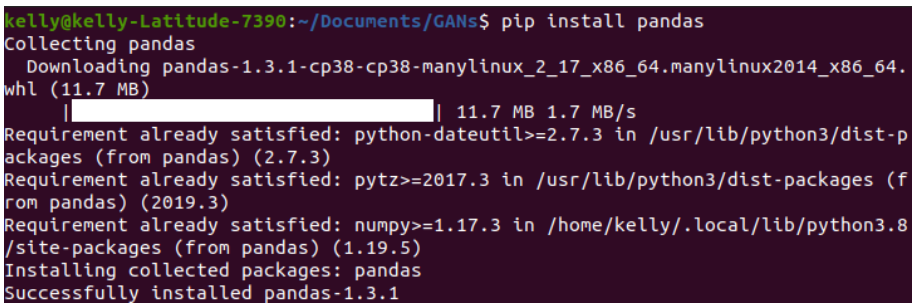

Figure 7: Pandas Installation

4 Implementation and Steps

4.1 Data Generation

In this research project, synthetic data is created by using pseudo randomness and the machine learning models are trained with this data. The generated data is stored in a csv file; dataset.csv.

```
import math
'''
create wave function with different periodicities.
we are considering 2 cycles
1.) daily 24 hour cycle
2.) weekly 7 day cycle
'''

#24 hour cycle
def w1(h):
    h%=24
    return math.sin((h/24)*math.pi*2) + 1.1

#weekly cycle
def w2(h):
    h%=168
    return math.sin((h/(7*24))*math.pi*2) + 1.1

print(w1(3.6*24))
print(w2(3.6*24))

'''
plot the test data by combining 2 wave functions + some randomness
'''
import matplotlib.pyplot as plt
import random

NOISE_LEVEL = 0.2 #10% noise level
SCALE_FACTOR = 100 #the max amout of load on the system
def data_gen(h):
    return w1(h) + w2(h) + (random.choice([-1,1])*NOISE_LEVEL)

x = [i for i in range(24*2)]
y = [data_gen(i)*SCALE_FACTOR for i in x]

plt.plot(x,y)
plt.show()
'''
generate data for load on cluster.
our main challenge is to get good accuracy on very small amount of data.

data-structure:-
    timestamp = hourly timestamp.
    load_level = the amoun of incoming load on the system.
'''
import pandas as pd

#small dataset
x1 = [i for i in range(1200)]
y1 = [data_gen(i)*SCALE_FACTOR for i in x1]

#create a dataframe
data_dict = {'timestamp':pd.Series(x1),'load_level':pd.Series(y1)}
df = pd.DataFrame(data_dict)
#store in csv file
df.head(10)
```

Figure 8: Data Generation source code

4.2 Steps

After installing the necessary libraries, the user can implement the project on their Ubuntu Machine. To Implement, run the three algorithms used; Threshold Based Method, Linear Regression and Generative Adversarial Networks (GANs). These are the following commands;

- python3 threshold.py
- python3 linear_regression.py
- python3 GAN.py

When the first command is executed, tasks are generated and the system tracks and invoke resources if the demand is met(higher or lower than the threshold set)

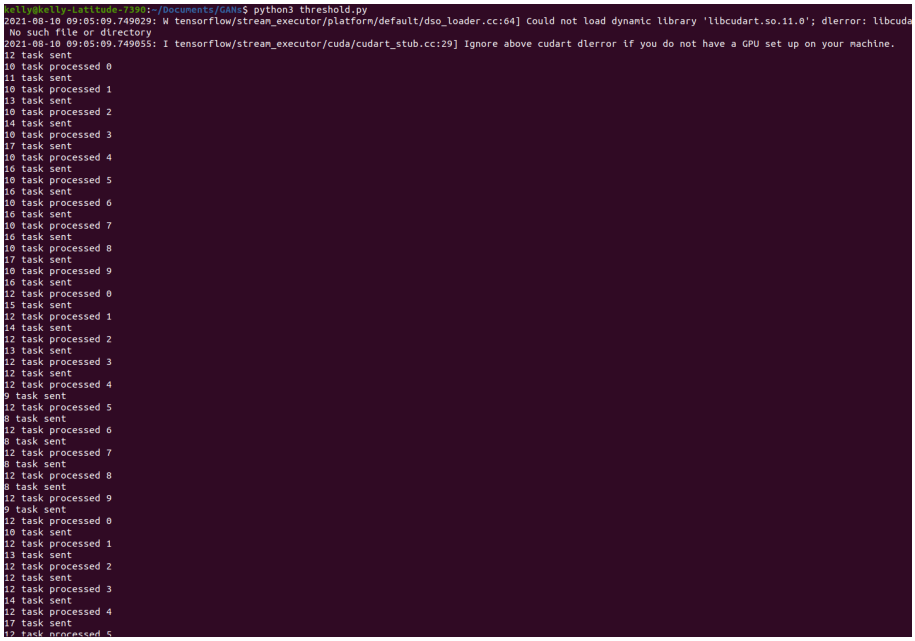

Figure 9: Execution of Threshold based Method

Linear regression is the linear modelling approach for performing forecasting and predictions. When the second command is executed, the linear regression algorithm is applied to the load generated and prediction occurs.

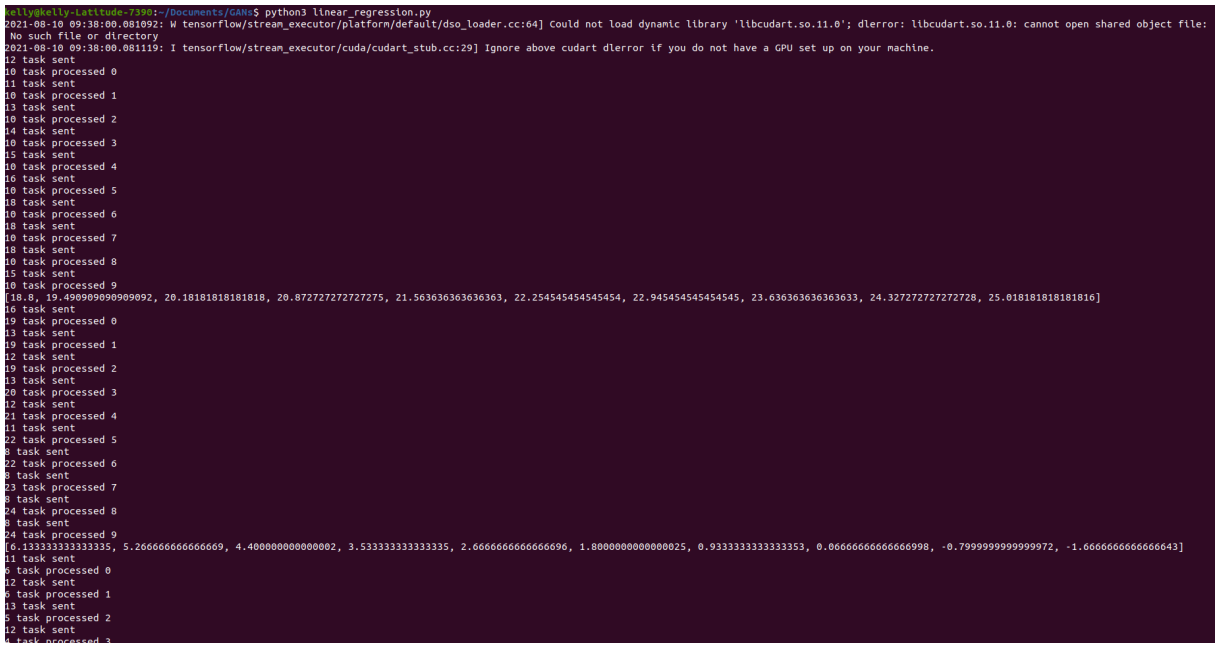

Figure 10: Execution of Linear Regression Technique

4.3 Proposed Model Implementation

GANs are usually unsupervised and learn using a cooperative zero-sum game framework. GANs generate their training data which makes it suitable for very long term purposed; As the feedback loop between the adversarial networks continues, the generator will produce higher-quality output.

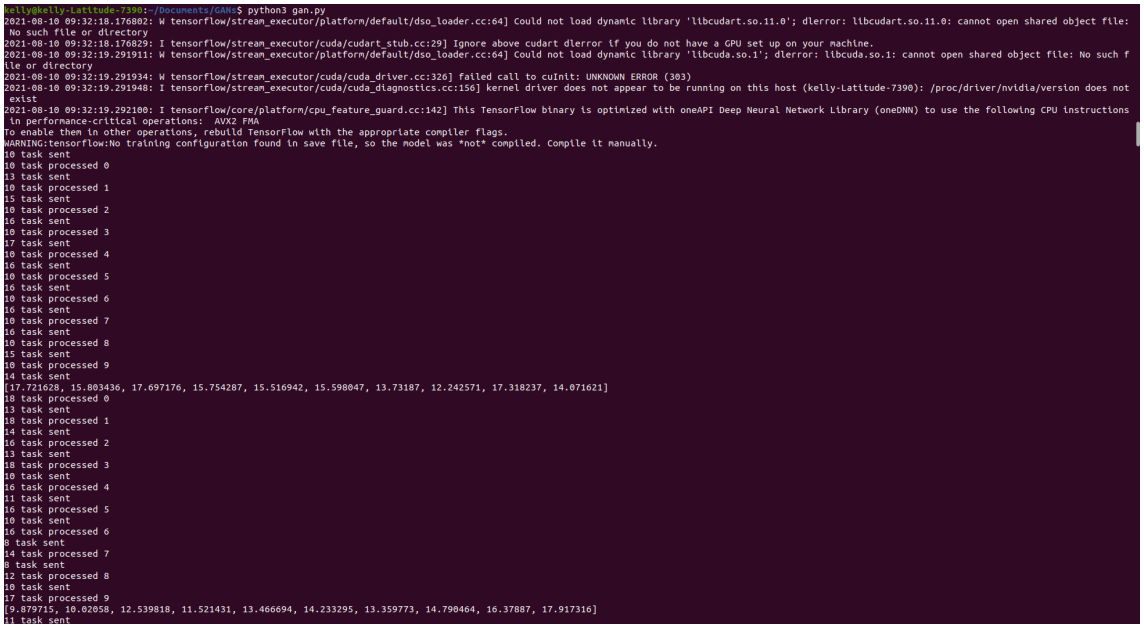

Figure 11: Execution using GANs