National College of Ireland

Configuration Manual

MSc Research Project
Cloud Computing

Amit Kumar Chhabiraj Yadav
Student ID: 23294795

School of Computing
National College of Ireland

Supervisor: Prof. Aqeel Kazmi

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Amit Kumar Chhabiraj Yadav
Student ID:	23294795
Programme:	Cloud Computing
Year:	2025
Module:	MSc Research Project
Supervisor:	Prof. Aqeel Kazmi
Submission Due Date:	11/12/2025
Project Title:	Configuration Manual
Word Count:	XXX
Page Count:	11

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:	Amit Kumar Yadav
Date:	28th January 2026

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):	

Configuration Manual

Amit Kumar Chhabiraj Yadav
23294795

1 Introduction

This configuration manual covers the entire experimental setup of simulation to compare the performance of sheduling algorithms under the same workload scenarios and analyze energy efficiency, latency, makespan and throughput when the task is assigne to VM in the cloud datacenters. This process can be achieved with the help CloudSim Plus simulation modelPlus (2024) . The study draws a comparison between three types of scheduling. techniques:

- Heuristic RoundRobin, Min-Min, Max-Min.
- Metaheuristic: Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO)
- Machine Learning: Deep Q-Network (DQN).

The experiment is carried out in six realistic working load situations and verified, also applying a lot of statistical processing in Python.

2 System Requirements

2.1 Software Requirements

Table 1: Technologies and Tools Used

Component	**Technology**
Programming Language version	Java 17
Simulation Framework	CloudSim Plus 7.3.3 Plus (2024)
Machine Learning	DeepLearning4J (DL4J) DeepLearning4J (2024)
Numerical Backend	ND4J
Build Tool	Maven
Statistiical Analysis	Python 3 Foundation (2024)
Visualization lib	Matplotlib, Seaborn
Version Control	Git & GitHub

Since the hardware used in my machine is shown in figure 1

Figure 1: System Configuration

2.2 Hardware Requirements

2.2.1 System Requirements

- Processor: Apple M1/M2/M3/M4 or Intel i7+
- RAM: 16 GB or above
- Storages: 50 GB
- Operating System: macOS or Windows 11

2.2.2 Minimum System requirements

- Processor: Intel i5
- RAM: 8 GB
- Storage: 30 GB

3 GitHub Repository Configuration

All source code and experimental scripts are maintained and arranged in the github repository which is publicly available on below URL and just for the reference the screenshot of the home page of github is available in the figure 2

Repository URL: https://github.com/Amit-NCI/Energyconsumption

How to run this simulation experiments on your machine by using the below steps

Cloning the Repository

Git clone : `https://github.com/Amit-NCI/Energyconsumption.git` GitHub (2024) The Go to project directories by using below command: cd Energyconsumption

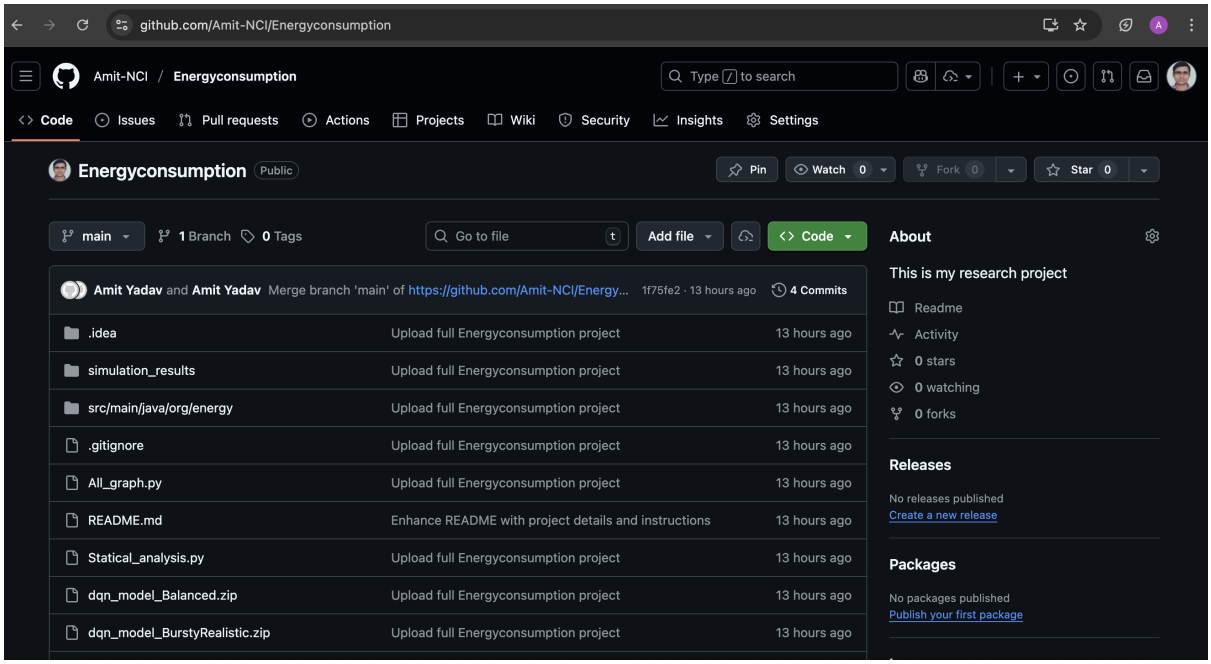

Figure 2: GitHub Home Page

4 Installation of Development Environment

4.1 Java 17 Installation

- brew install openjdk@17
- java -version

4.2 Maven Installation

- brew install maven
- mvn -version

The list of maven dependency is available in the figure 6 which I have used while implementing this project

4.3 Python Libraries Installation

- pip install pandas numpy scipy matplotlib seaborn

Since I have attached the terminal screenshot where the software related version is shown in the figure 3

4.4 Project Compilation

- The project is compiled using Maven here.
- mvn clean install
- After successful execution the executable JAR file is created at: target/Energyconsumption-1.0-SNAPSHOT-jar-with-dependencies.jar

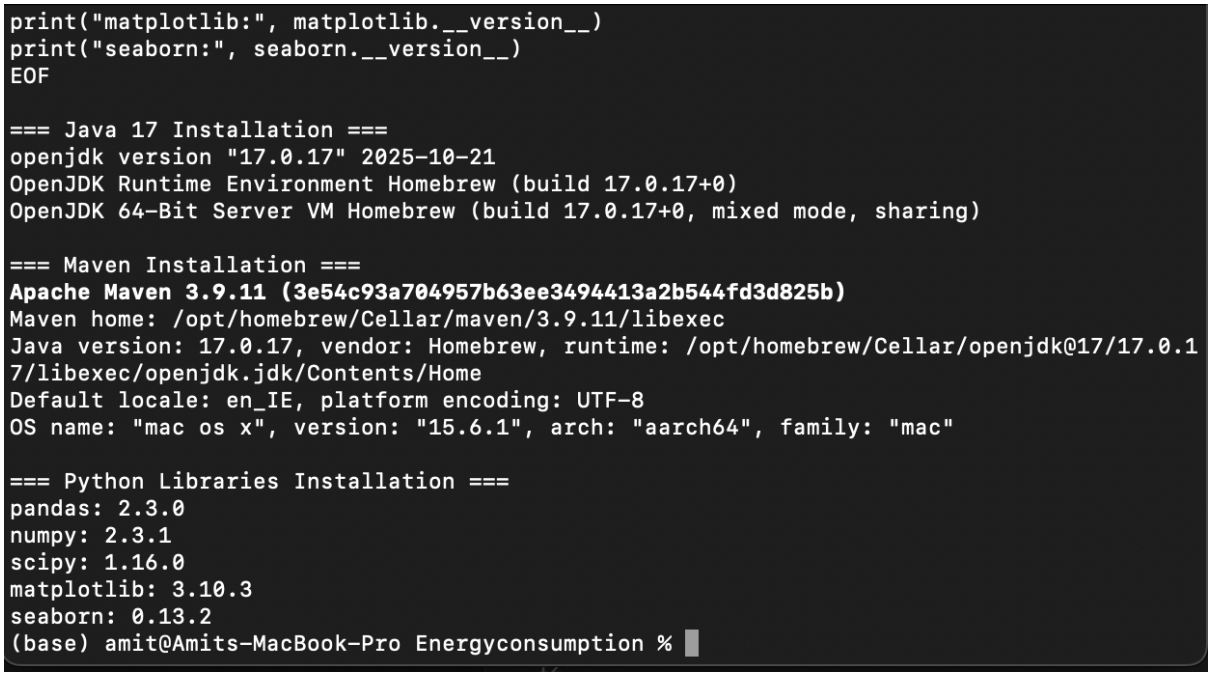

Figure 3: Software version

The figure 4 show the build success message in my terminal screenshot.also I have mention the folder strucuture and home page screenshot in the figure 5

5 Simulation Using IntelliJ IDEA (IDE)

The terminal based command execution is limited(because i have not mention the code in pom.xml to run from terminal only use for building the maven project) thus the entire CloudSim Plus simulation is run through IntelliJ IDEA Integrated Development Environment (IDE). The method will allow a graphical user interface to construct, execute and debug the project in a safe mannerJetBrains (2024). The figure show the home page of intellij where the project folder directories is available in the figure 5 .

5.1 Step 1: Install IntelliJ IDEA IDE

- Download IntelliJ IDEA from the given link: `https://www.jetbrains.com/idea/`
- To Install the Intellij IDE by following the basic installation steps.

5.2 Step 2: Open the Project in IntelliJ

- Launch IntelliJ IDEA
- Select **Open**.
- Go to the cloned `Energyconsumption` project folder.
- Click **Open** to load the project which is cloned from git hub.
- IntelliJ IDE having capabilities which detect the Maven project and begins downloading dependencies if it is not available in the project

```
[INFO] ------------------------------------------------------------------------
(base) amit@Amits-MacBook-Pro Energyconsumption % mvn clean install
[INFO] Scanning for projects...
[INFO]
[INFO] -------------------< org.example:Energyconsumption >--------------------
[INFO] Building Energyconsumption 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ Energyconsumption ---
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ Energyconsumption ---
[INFO] Copying 0 resource from src/main/resources to target/classes
[INFO]
[INFO] --- compiler:3.10.1:compile (default-compile) @ Energyconsumption ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 15 source files to /Users/amit/IdeaProjects/Energyconsumption/target/classes
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ Energyconsumption ---
[INFO] skip non existing resourceDirectory /Users/amit/IdeaProjects/Energyconsumption/src/test/resources
[INFO]
[INFO] --- compiler:3.10.1:testCompile (default-testCompile) @ Energyconsumption ---
[INFO] Changes detected - recompiling the module!
[INFO]
[INFO] --- surefire:3.2.5:test (default-test) @ Energyconsumption ---
[INFO]
[INFO] --- jar:3.4.1:jar (default-jar) @ Energyconsumption ---
[INFO] Building jar: /Users/amit/IdeaProjects/Energyconsumption/target/Energyconsumption-1.0-SNAPSHOT.jar
[INFO]
[INFO] --- install:3.1.2:install (default-install) @ Energyconsumption ---
[INFO] Installing /Users/amit/IdeaProjects/Energyconsumption/pom.xml to /Users/amit/.m2/repository/org/example/Energyc
onsumption/1.0-SNAPSHOT/Energyconsumption-1.0-SNAPSHOT.pom
[INFO] Installing /Users/amit/IdeaProjects/Energyconsumption/target/Energyconsumption-1.0-SNAPSHOT.jar to /Users/amit/
.m2/repository/org/example/Energyconsumption/1.0-SNAPSHOT/Energyconsumption-1.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.728 s
[INFO] Finished at: 2025-12-09T13:47:07Z
[INFO] ------------------------------------------------------------------------
(base) amit@Amits-MacBook-Pro Energyconsumption %
```

Figure 4: mvn clean install

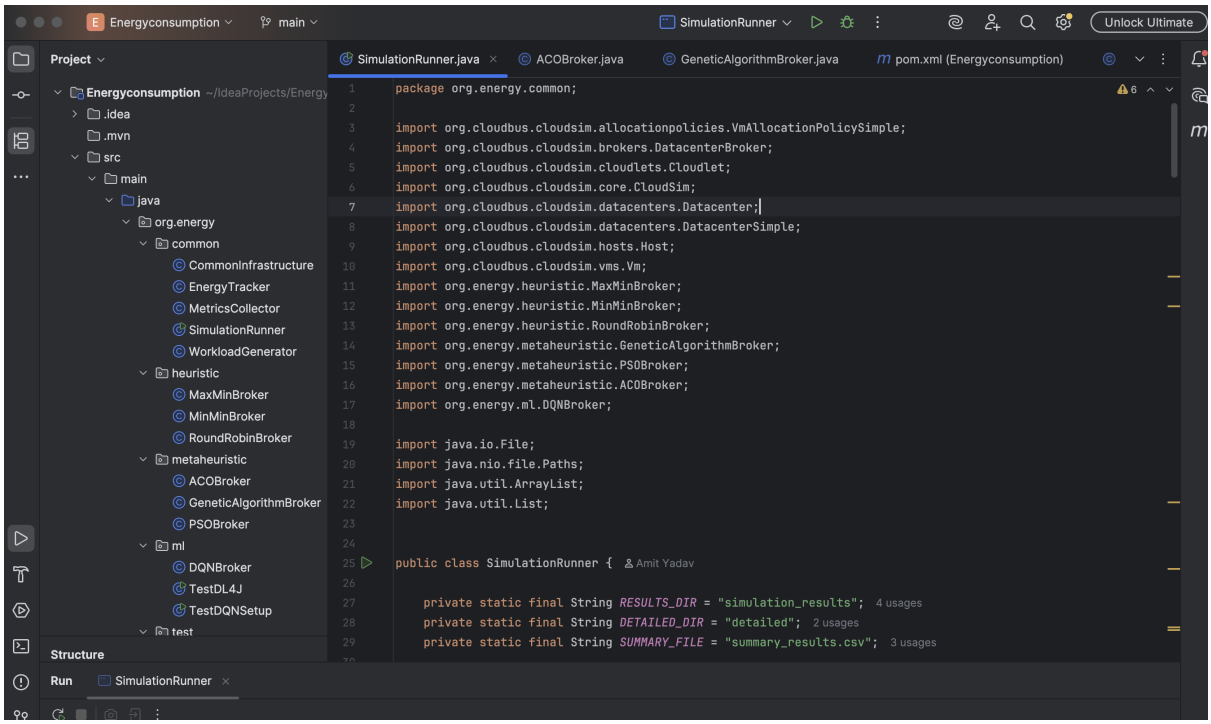

Figure 5: Open the Energyconsumption Project in IJ

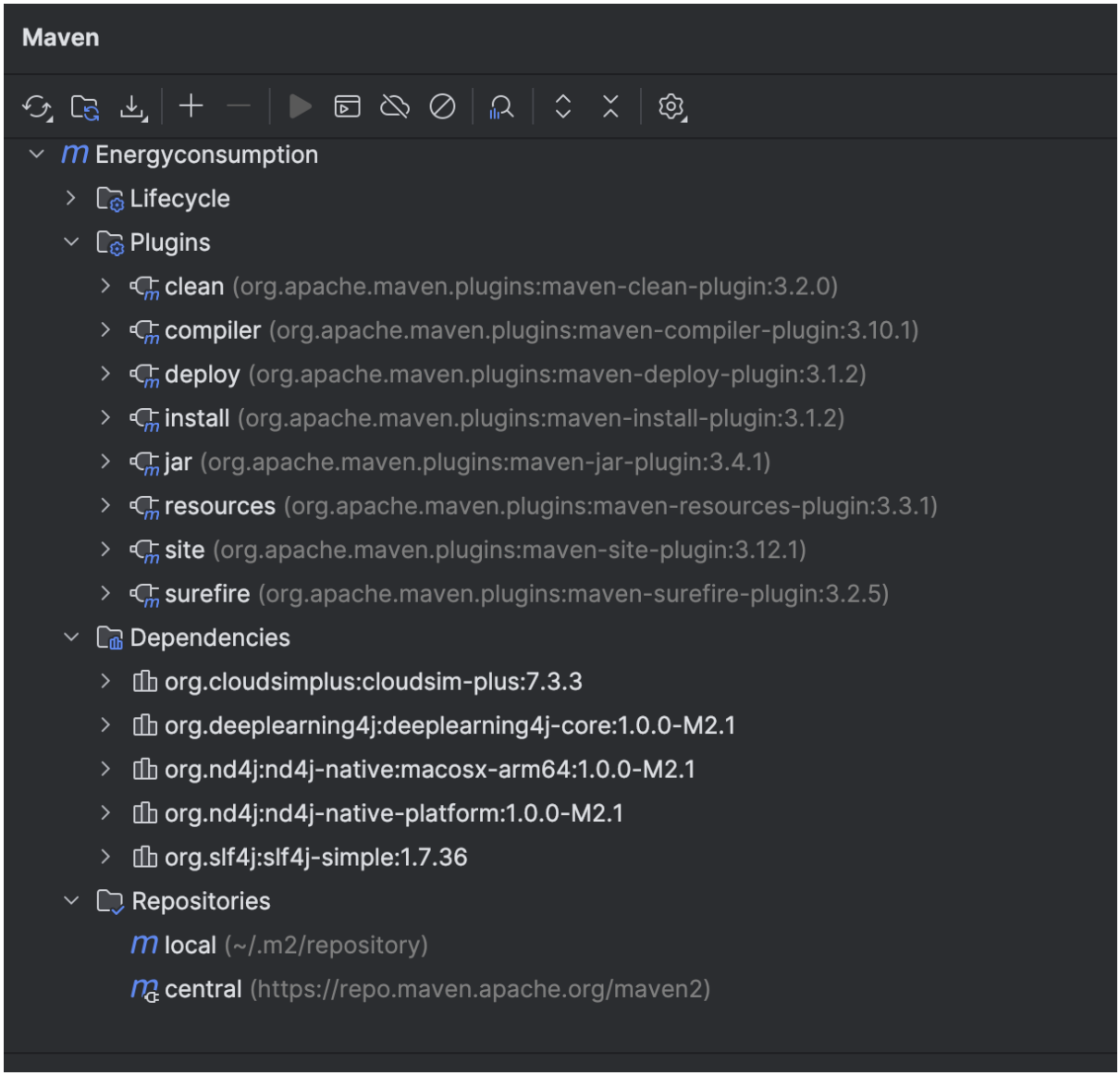

Figure 6: Maven Dependency

5.3 Step 3: Maven Dependency Syncronization

- After opening the project IntelliJ automatically loads the required packages and download te required de- pendency which are mention in pom.xml file.
- All required dependencies in the simulation project such as CloudSim Plus, DeepLearning4J, and ND4J are automatically downloaded in the project .
- When the build process completed then the status shows Build is Successful.

5.4 Step 4: Go to the Main Simulation File

- Navigate to: `src/main/java`
- The main java class which control all the schedling classes : SimulationRunner.java

For reference the figure 7 shows the simulation runner java files in the picture

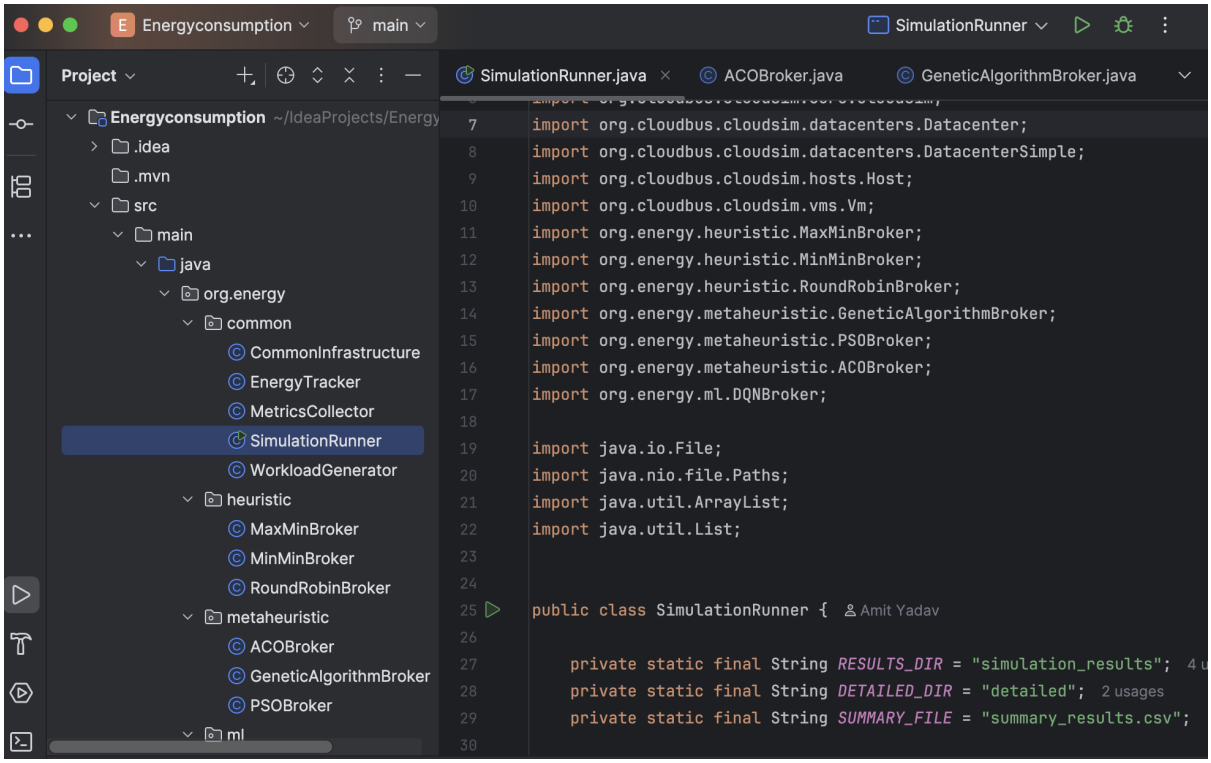

Figure 7: Simulation Runner main java file

5.5 Step 5: Configure Run Parameters

- Right-click on SimulationRunner.java.
- Select **Run 'SimulationRunner.main()'**.
- In the run configuration window:

The running code showing the console output in the console window at the bottom of the main page which has shown in figure 8

5.6 Step 6: Execute the Simulation

- Click on the run button which is present on the top of the window.
- The simulation starts and the code is executing and the current log are availabe on the console
- Real-time logs are displayed showing algorithm execution for all task scheduling strategies.

5.7 Step 7: Output File Generation

- After successful simulation execution done the output CSV files are automatically generated.
- These files are stored in: `simulation_results/` available in the figure 9
- The folder contains:

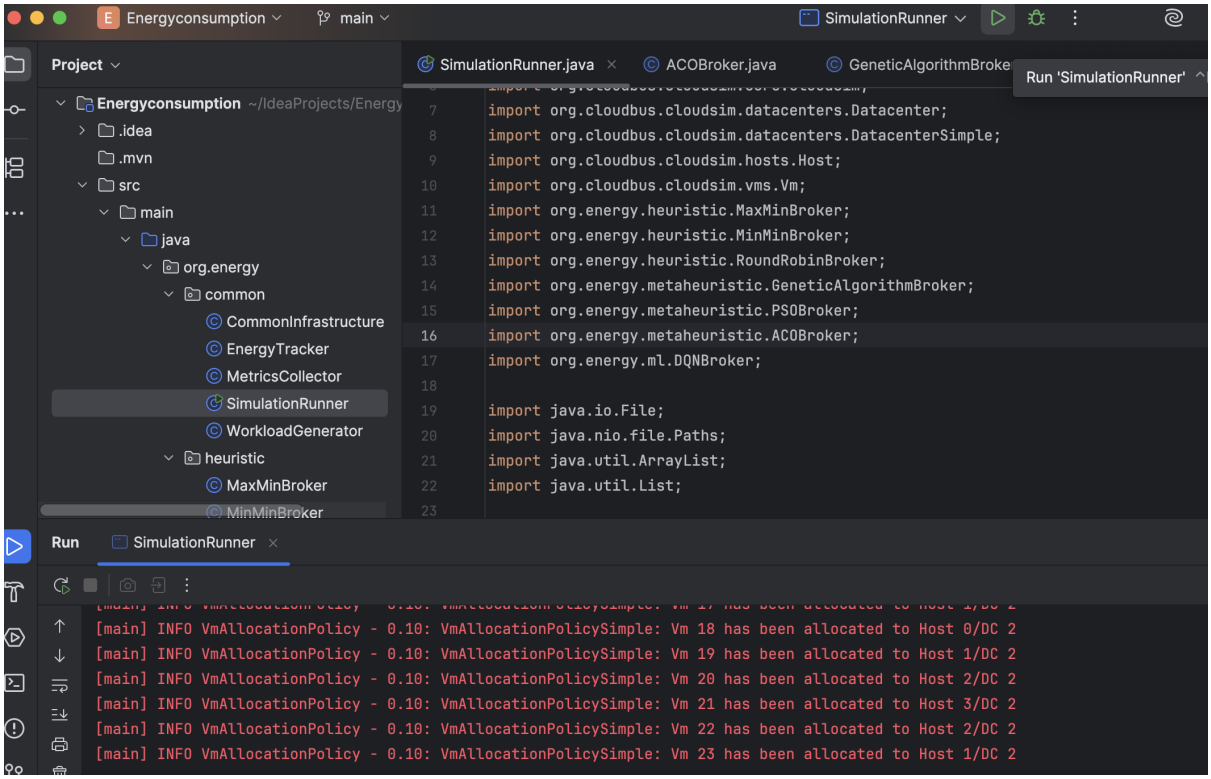

Figure 8: after running the simulation file with console output

- Inside the folder simulation results there is another folder name details which contain all csv files for all algorithms with workload scenario and seeds wise.
- Scenario wise performance result files will get in another folder after running then both python code manually. It generates two folder name plot(all matrics graph) and statical_analysis folder contain box plot and winner and comparision csv files.

6 Statistical Analysis and Graph Generation by using the python codes

After completinng the simulation execution the Python based statistical analysis is performed using two python files:Foundation (2024); Matplotlib (2024); Seaborn (2024)

- cd Energyconsumption
- python3 All_graphs.py
- The list of output files shows in the console and the results are store in the folder called plot 10
- cd Energyconsumption
- python3 Statical_analysis.py
- The list of output files show in the console and the results are stored in the folder called statistical analysis 11

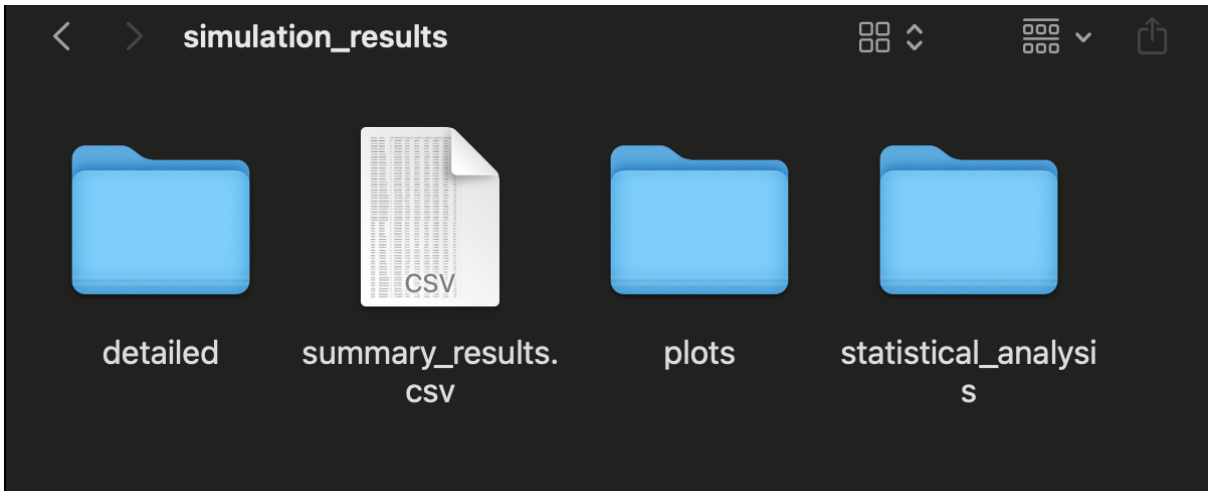

Figure 9: OutPut folder after running both python files as well

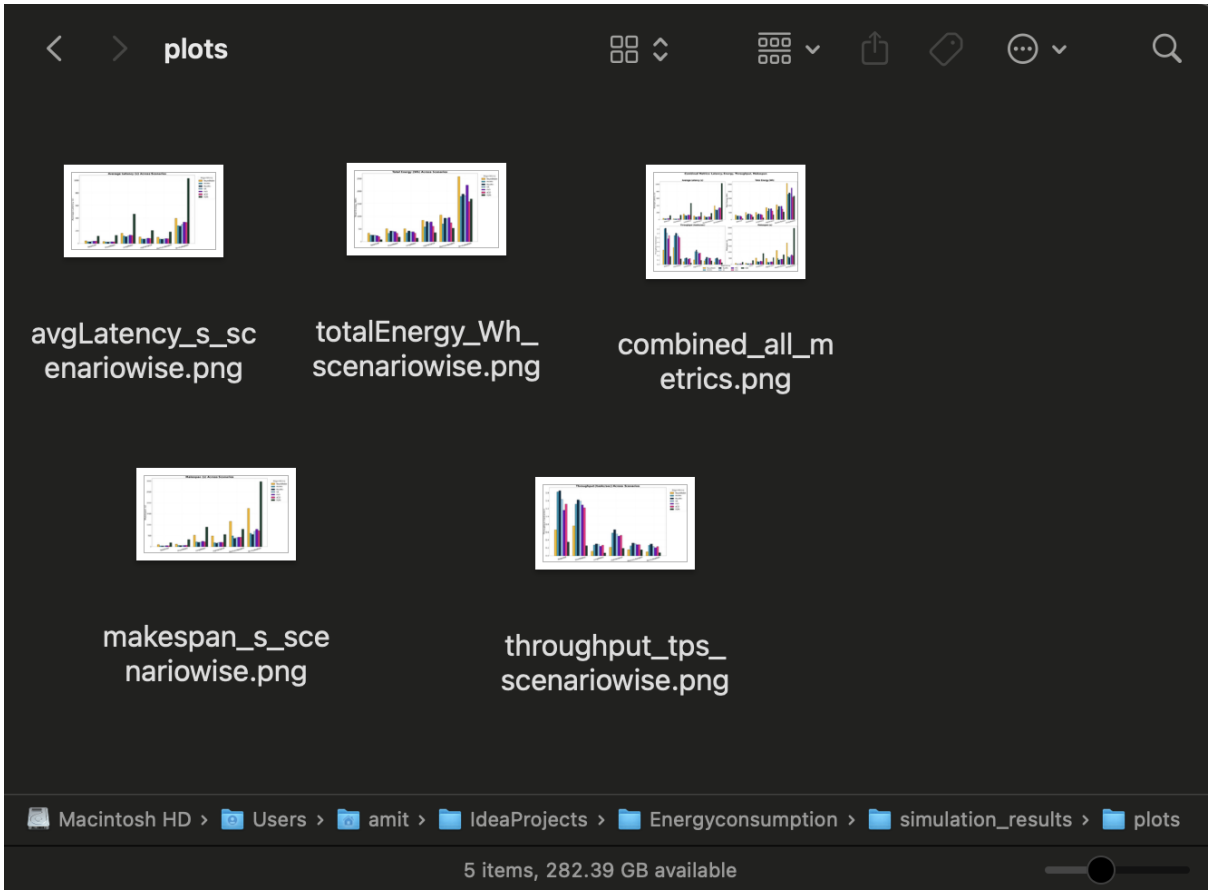

Figure 10: graph folder where all matrics are present

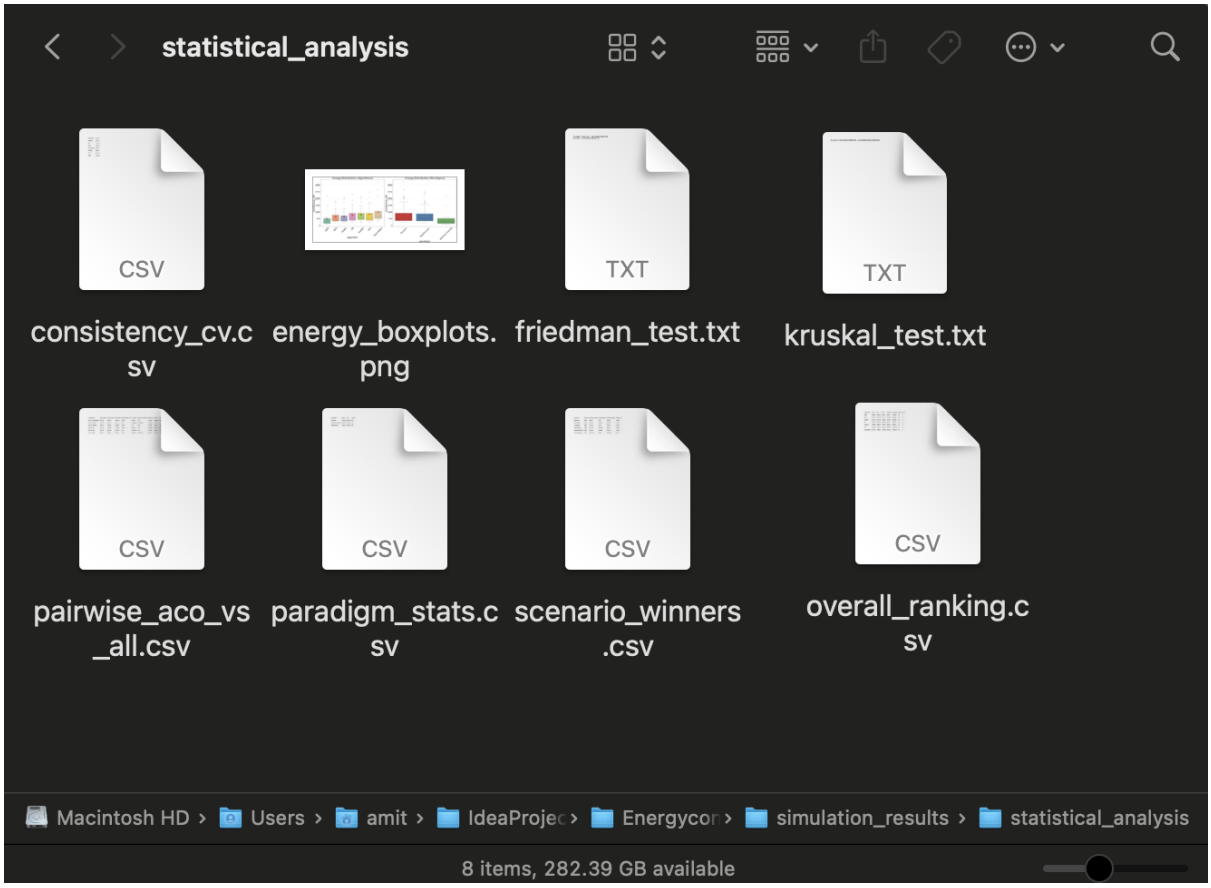

Figure 11: Box plot and winner csv folder

7 Conclusion

This experiment have explain an execution workflow describe an entire simulation, reproducibility for performing energy-saving task to vm scheduling experiments with CloudSim Plus. To guarantee scientific repeatability and integrity of research all software dependencies, simulation steps, workload configurations, statistical validation procedures and visualization tools are documented. The data sets and graphical results obtained are used to analyze the scheduler efficiency in heterogeneous cloud environments.

References

DeepLearning4J (2024). Deeplearning4j documentation. Accessed: 2025.
URL: *https://deeplearning4j.org*

Foundation, P. S. (2024). Python 3 documentation.
URL: *https://docs.python.org/3/*

GitHub (2024). Github documentation.
URL: *https://docs.github.com*

JetBrains (2024). Intellij idea documentation.
URL: *https://www.jetbrains.com/idea/*

Matplotlib (2024). Matplotlib documentation.
URL: *https://matplotlib.org*

Plus, C. (2024). Cloudsim plus documentation.
URL: *https://cloudsimplus.org*

Seaborn (2024). Seaborn statistical data visualization.
URL: *https://seaborn.pydata.org*