National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Roopak Mallik
Student ID: 23433639

School of Computing
National College of Ireland

Supervisor: Dr. David Hamill

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Roopak Mallik

Student ID: 23433639

Programme: MSc Data Analytics **Year:** 2025-2026

Module: Research Practicum/ Master Thesis

Lecturer: Dr. David Hamill

Submission Due Date: 11th December 2025

Project Title: A Comparative Study of Reinforcement Learning Algorithms and Custom Technical Indicators for Short-Term Stock Trading

Word Count: 1284 **Page Count:** 9

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: Roopak Mallik

Date: 8th December 2025

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):	

National College of Ireland

Project Submission Sheet

Student Name:	Roopak Mallik		
Student ID:	23433639		
Programme:	MSc Data Analytics	**Year:**	2025-2026
Module:	Research Practicum/ Master Thesis		
Lecturer:	Dr. David Hamill		
Submission Due Date:	11th December 2025		
Project Title:	Configuration Manual: A Comparative Study of Reinforcement Learning Algorithms and Custom Technical Indicators for Short-Term Stock Trading		
Word Count:	1284		

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 references section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature:	Roopak Mallik
Date:	8th December 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**
6. **Please check that you read AI and Academic Integrity Acknowledgement Supplments in this document**

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

AI Acknowledgement Supplement

Research Practicum/ Master Thesis

Configuration Manual: A Comparative Study of Reinforcement Learning Algorithms and Custom Technical Indicators for Short-Term Stock Trading

Roopak Mallik/23433639	MSc Data Analytics	8th December 2025
NA	NA	NA

This section is a supplement to the main assignment, to be used if AI was used in any capacity in the creation of your assignment; if you have queries about how to do this, please contact your lecturer. For an example of how to fill these sections out, please click here.

AI Acknowledgment

This section acknowledges the AI tools that were utilized in the process of completing this assignment.

Tool Name	Brief Description	Link to tool
NA	NA	NA

Description of AI Usage

This section provides a more detailed description of how the AI tools were used in the assignment. It includes information about the prompts given to the AI tool, the responses received, and how these responses were utilized or modified in the assignment. **One table should be used for each tool used**.

[Insert Tool Name]	
[Insert Description of use]	
[Insert Sample prompt]	[Insert Sample response]

Evidence of AI Usage

This section includes evidence of significant prompts and responses used or generated through the AI tool. It should provide a clear understanding of the extent to which the AI tool was used in the assignment. Evidence may be attached via screenshots or text.

Additional Evidence:

[Place evidence here]

Additional Evidence:

[Place evidence here]

Configuration Manual: A Comparative Study of Reinforcement Learning Algorithms and Custom Technical Indicators for Short-Term Stock Trading

Roopak Mallik
x23433639
MSc Data Analytics
National College of Ireland
Supervisor: Dr. David Hamill

This configuration manual provides detailed information to ensure reproducibility and consistency across environments. It has all the specific required software tools, settings, and configurations required to replicate the experimental setup for training and evaluating the proposed framework.

1. System Requirements

This project was developed and implemented in multiple Jupyter Notebooks using Visual Studio Code on a MacBook Air M3, equipped with 16 GB of unified memory and a 256 GB SSD.

1.1 Hardware Requirements

CPU:

- Minimum: 4-core CPU.
- Recommended: 6-core or higher.

GPU:

- Minimum: None required (CPU-only training will work).

RAM:

- Minimum: 8 GB.
- Recommended: 16 GB or more.

Storage:

- Minimum: 20 GB free space.
- Recommended: 50 GB or more (SSD preferred).

1.2 Software Requirements

To implement, train and evaluate the reinforcement learning models developed in this project for trading, several software components, libraries and development tools are required. These tools ensure smooth data acquisition, environment setup, model execution and result visualization. The following list outlines the essential software requirements:

Operating System:

- macOS (Ventura or later).
- Windows 10/11 (64-bit).
- Linux (Ubuntu 20.04 or later).

All operating systems must support Python environments and the associated machine learning libraries.

Python Environment:

- Python Version:
  - Python 3.10 or 3.11 (recommended for compatibility with Stable-Baselines3 and Gymnasium).
- Environment Managers (any one):
  - venv (built-in).
  - Anaconda / Miniconda (recommended for dependency isolation).

Required Python Libraries:

These libraries support data preprocessing, financial indicator computation, reinforcement learning algorithms and visualisation.

Purpose	Libraries
Data handling & numerical computations	`pandas, numpy`
Data acquisition	`yfinance`
Technical indicators	`Finta or ta`
Reinforcement learning	`stable-baselines3, gymnasium, gym-anytrading`
Vectorised environments	`stable_baselines3.common.vec_env`
Plotting & visualisation	`matplotlib, plotly`
Performance analysis (optional)	`quantstats`
Jupyter environment	`jupyterlab or notebook`

Table 1. Required Libraries

Development Tools:

- Visual Studio Code or any another code editor.

Reinforcement Learning Frameworks:

- Stable-Baselines3 (SB3):
  - Provides implementations for DQN, PPO, and A2C algorithms used in the project.
- OpenAI Gymnasium:
  - Used for creating custom trading environments.
- gym-anytrading:
  - Used as the base environment for financial time-series simulations.

2. Code Editor Setup

Use Visual Studio Code or any other code editor like Pycharm.
Link to download **Visual Studio Code.**

3. Data Gathering

This section guides you on how to download and prepare the dataset from starting to final stage.

3.1 Dataset Source

The historical stock market data used in this project was collected through the Yahoo Finance (YFinance) API, an open-source Python library that provides convenient access to publicly available financial datasets. The API was selected due to its reliability, ease of integration and suitability for academic research, allowing seamless retrieval of daily stock prices without licensing restrictions. For more information, refer to documentation available at **yfinance.**

3.2 Data Collection Procedure

To acquire the data locally on your workstation, begin by executing the following command to install the library on your system as shown in Figure 1. To download via pip on **Windows** simply run the command in command prompt or terminal, **'pip install yfinance'.** For **MacOS** run the command **'pip3 install yfinance'.**

Figure 1. Installing yfinance library

The datasets were acquired programmatically using Python through publicly accessible financial data APIs. A custom data-collection workflow was created in the accompanying '**fetch_data.ipynb**' notebook, which automated the retrieval of daily stock market information. Data points collected included:

- Daily opening and closing prices.
- High and low values for each trading day.
- Trading volume.

The data acquisition process followed a structured pipeline:

- **API Request Construction:** API calls were generated using stock ticker symbols and date ranges for AMD, Intel and Nvidia. An example on how to gather historical stock data of a company is shown in Figure 2.
- **Fetching:** The system retrieved data at the daily level and exported the results into CSV format for each company.
- **File Storage:** The resulting datasets were stored as:
  - intel_stock_data_jan_to_aug_2025.csv
  - amd_stock_data_jan_to_aug_2025.csv
  - nvda_stock_data_jan_to_aug_2025.csv
- **Verification:** Each dataset was checked for completeness, missing values, duplicate entries and consistency.

```
# Stock tickers
tickers = ["NVDA"]

# Fetching daily stock data from Jan 1, 2025 to Aug 31, 2025
nvda_data = yf.download(tickers, start="2025-01-01", end="2025-08-31", interval="1d")

# Reset index to make the timestamp a separate column
nvda_data = nvda_data.reset_index()

nvda_data.columns = ['Date', 'Open', 'High', 'Low', 'Close', 'Volume']

nvda_data.head()
```

```
/var/folders/gl/gtqpk9s52v7_d3g5hfr_v3940000gn/T/ipykernel_3697/1839666459.py:5: FutureWarning:
  nvda_data = yf.download(tickers, start="2025-01-01", end="2025-08-31", interval="1d")
[**********************100%**********************]  1 of 1 completed
```

	Date	Open	High	Low	Close	Volume
0	2025-01-02	138.279877	138.849760	134.600685	135.970382	198247200
1	2025-01-03	144.438538	144.868437	139.699564	139.979502	229322500
2	2025-01-06	149.397446	152.126862	147.787811	148.557632	265377400
3	2025-01-07	140.109467	153.096642	139.979490	152.996658	351782200
4	2025-01-08	140.079483	143.918643	137.530035	142.548946	227349900

Figure 2. API call to gather historical daily data for Nvidia

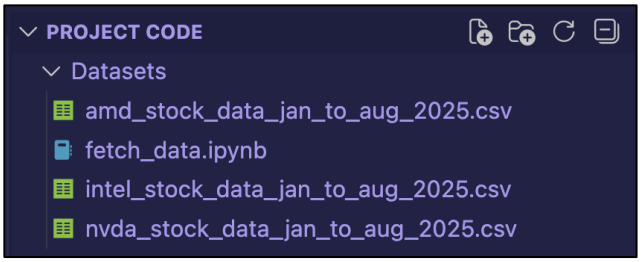


Figure 3. Data Collection as CSV file format

4. Execution

Prior to executing the main Reinforcement Learning for Trading scripts, a preliminary analysis was conducted to understand the characteristics of the collected datasets. The **'General_Analysis.ipynb'** file contains several visualizations and basic steps undertaken to explore and understand the data properties.

4.1 Library Installation and Execution Requirements

To ensure proper reproducibility of the reinforcement learning experiments conducted in this study, all software dependencies and library versions must be installed prior to running the notebooks. The project was developed in Python (version 3.10+), using a combination of scientific computing, machine learning, and reinforcement learning frameworks.

Refer to the official documentation available in Table 2 for the exact steps to install the libraries.

Library	Documentation Link
gymnasium **(as gym)**	https://gymnasium.farama.org/
gym_anytrading	Gym Trading Environment (similar functionality)
stable_baselines3	Stable Baselines3 Docs
finta	finta PyPI Page (includes installation and usage)
quantstats **(as qs)**	QuantStats GitHub/Documentation
pandas **(as pd)**	The Official Documentation of Pandas
numpy **(as np)**	NumPy Documentation
matplotlib.pyplot**(as plt)**	Matplotlib Documentation
os	Python os Module Documentation

Table 2. Library Documentation

4.2 Executing the files

The **RL Algorithms** folder is organized into two main directories: **Standard** and **Custom Indicators**. Each directory contains three algorithm-specific folders: **A2C, DQN, and PPO.**

The Standard folder houses reinforcement learning implementations applied to traditional **OHLCV datasets**, with files named intuitively (e.g., A2C_on_AMD.ipynb, PPO_on_Intel.ipynb). Similarly, the Custom Indicators folder follows the same structure but **incorporates technical indicators** into the trading algorithms, reflected in its naming convention (e.g., A2C_on_AMD_Custom_Indicators.ipynb, PPO_on_Intel_Custom_Indicators.ipynb).

Each notebook is fully **self-contained and can be executed independently**. To run any file, simply copy the dataset path from the Datasets folder (stored in CSV format) and paste it into your chosen algorithm notebook. The dataset loads via pandas, and you can then execute the cells sequentially. In total, there are 18 notebooks, 9 utilizing standard datasets and 9 incorporating custom technical indicators. Figures 4 and 5 provide visual examples of loading specific stock data into these implementations.

The environment operates with a 2-day window size and evaluates agent performance using Nvidia, AMD, and Intel stock data from January to August 2025. The model trains on days 50-130 and is tested on a separate period spanning days 131-165 as shown in Figures 9 and 10.

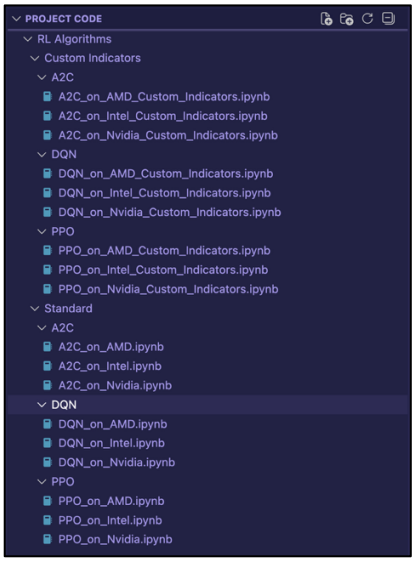


Figure 4. Layout of RL Algorithm Files

Figure 5. Loading a stock dataset for algorithmic trading

Steps for files using Standard Datasets:

1. Importing Libraries.
2. Loading Dataset and Pre-Processing.
3. Building Trading Environment.
4. Training the Reinforcement Learning Model (A2C, DQN or PPO).
5. Evaluating Model Performance using Time Series Graph.

Figure 6. Importing Libraries

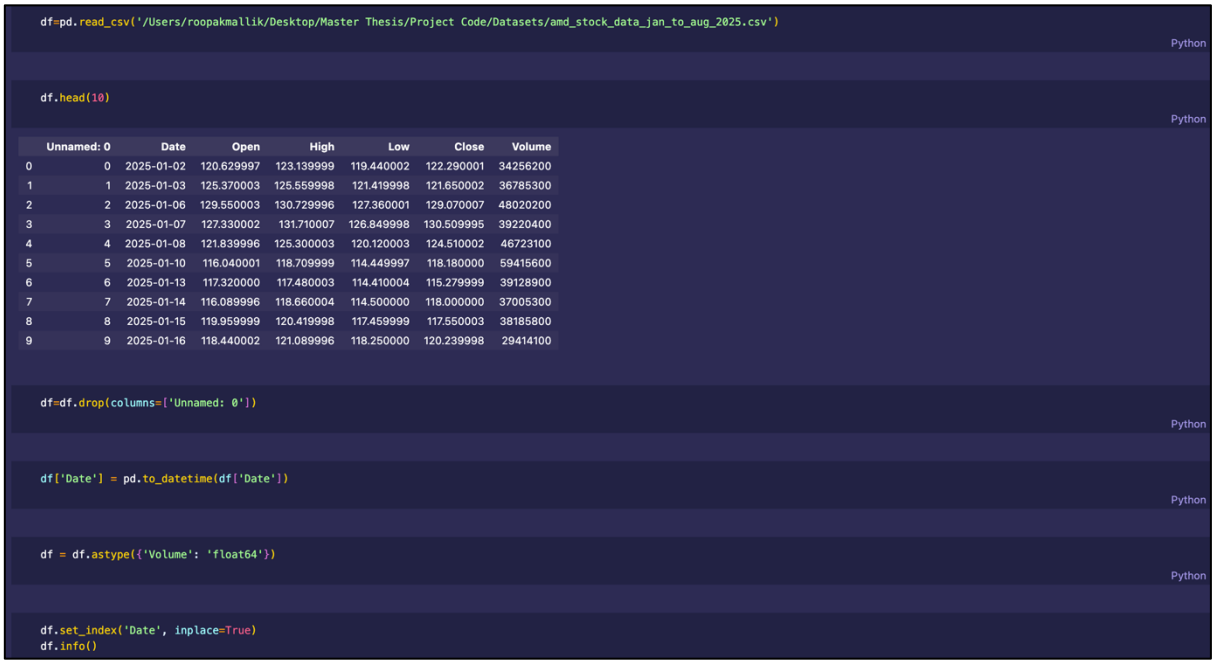


Figure 7. Loading Dataset and Pre-Processing

Figure 8. Building Trading Environment

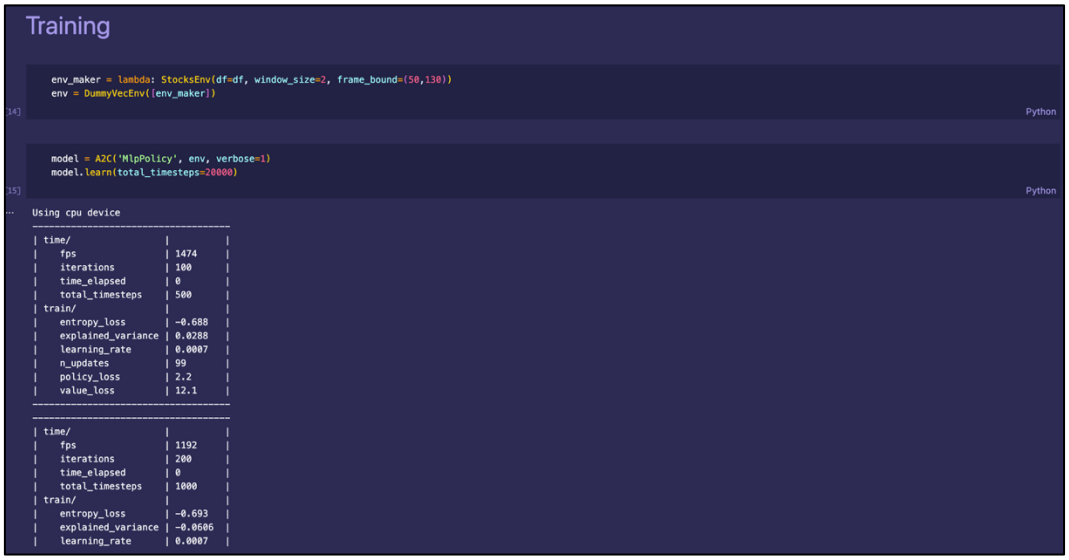


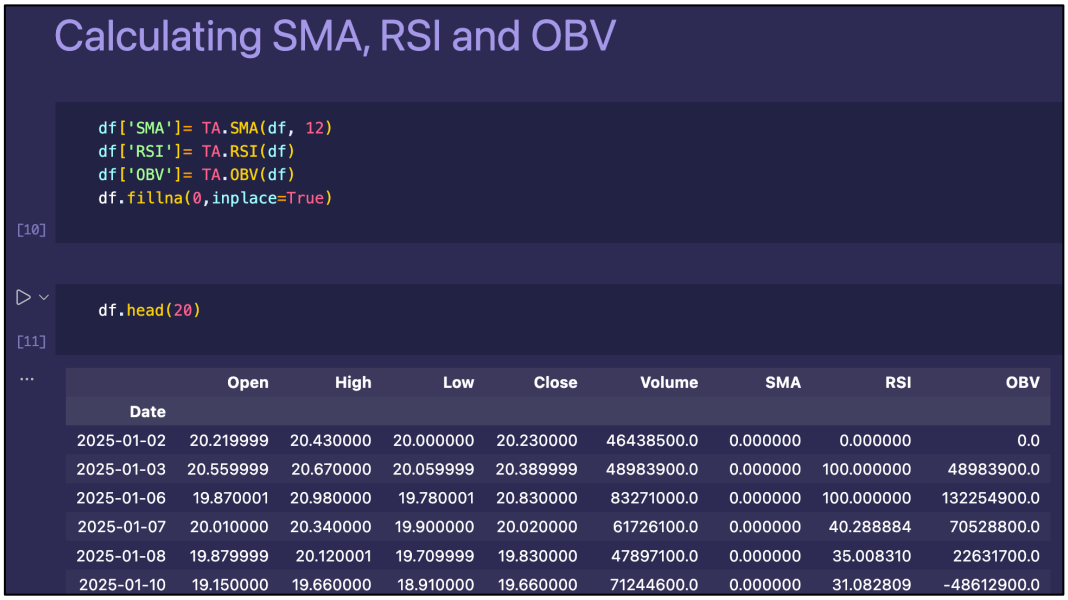
Figure 9. Training the Reinforcement Learning Model (A2C, DQN or PPO).

Figure 10. Evaluating Model Performance using Time Series Graph

Figures 6 through 10 provide a comprehensive, step-by-step guide to implementing reinforcement learning algorithms for trading using standard OHLCV (Open, High, Low, Close, Volume) datasets.

Steps for files with inclusion of Custom Indicators in Standard Datasets:

1. Importing Libraries.
2. Loading Dataset and Pre-Processing.
3. Calculating SMA, RSI and OBV.
4. Building Trading Environment.
5. Adding five features, 'Low','Volume','SMA','RSI','OBV'.
6. Training the Reinforcement Learning Model (A2C, DQN or PPO).
7. Evaluating Model Performance using Time Series Graph.



Date	Open	High	Low	Close	Volume	SMA	RSI	OBV
2025-01-02	20.219999	20.430000	20.000000	20.230000	46438500.0	0.000000	0.000000	0.0
2025-01-03	20.559999	20.670000	20.059999	20.389999	48983900.0	0.000000	100.000000	48983900.0
2025-01-06	19.870001	20.980000	19.780001	20.830000	83271000.0	0.000000	100.000000	132254900.0
2025-01-07	20.010000	20.340000	19.900000	20.020000	61726100.0	0.000000	40.288884	70528800.0
2025-01-08	19.879999	20.120001	19.709999	19.830000	47897100.0	0.000000	35.008310	22631700.0
2025-01-10	19.150000	19.660000	18.910000	19.660000	71244600.0	0.000000	31.082809	-48612900.0

Figure 11. Calculating SMA, RSI and OBV

Figure 12. Selecting Low, Volume, SMA, RSI and OBV as Custom Technical Indicators

As illustrated in Figures 11 and 12, implementing custom technical indicators requires just two additional steps: computing and adding the custom indicators, then selecting five key features (Low, Volume, SMA, RSI and OBV). Beyond these modifications, the execution process follows the same procedure as the standard implementation.