National College of Ireland

Configuration Manual

MSc Research Project
Artificial Intelligence

Vamshi Krishna Angala
Student ID: X22177213

School of Computing
National College of Ireland

Supervisor: Dr. Anh Duong Trinh

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Vamshi Krishna Angala
Student ID:	X22177213
Programme:	MSc in Artificial Intelligence **Year:** 2023
Module:	MSc Research Project
Lecturer:	Dr. Anh Duong Trinh
Submission Due Date:	05/01/2024
Project Title:	Transformers for Malware Detection
Word Count:	708 **Page Count:** 8

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: Vamshi Krishna Angala

Date: 05/01/2024

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: Transformers for Malware Detection

Forename Surname
Student ID:

1. Introduction

This manual provides detailed instructions for setting up and executing the code for implementing transformers in malware detection. The focus is on utilizing transformer models, specifically BERT, for effective identification of malicious software. The implementation is carried out in Python, leveraging the Hugging Face library.

2. System Specification

The malware detection system using transformers is developed on the following hardware specifications:

- Process: Intel i7 generation,
- Operating System: Windows 10,
- Ram: 16 GB (DDR4),
- Stroage Hard Drive: 512GB (SSD)

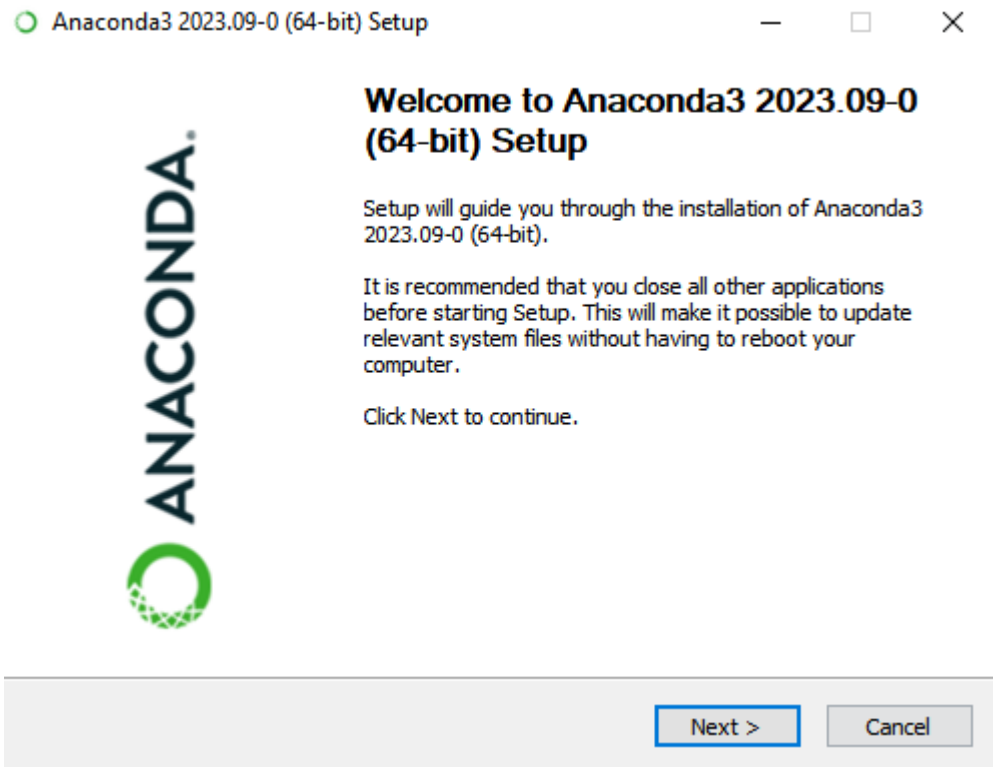
3. Softwares Used:

The following software tools are necessary for the development and execution of the malware detection system:

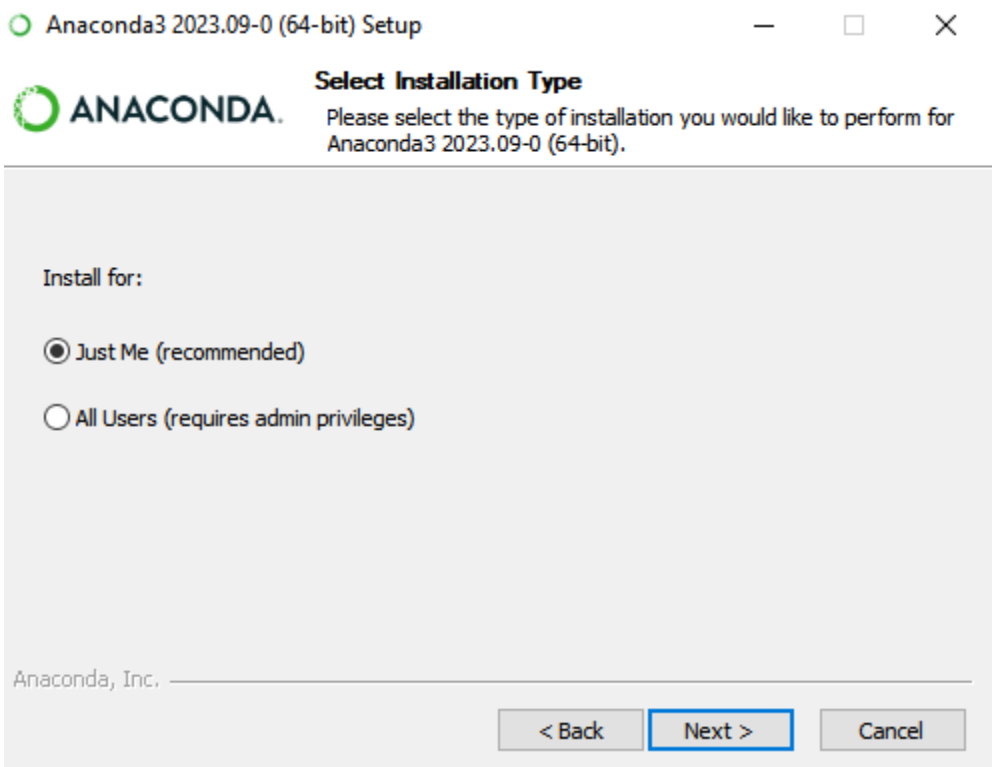
- Pycharm
- Anaconda
- TensorFlow and Keras
- Hugging Face Transformers Library
- Pandas
- NumPy
- Matplotlib

4. Installation of the Software:

- Download Anaconda from the official website: Anaconda.
- Follow the installation instructions.

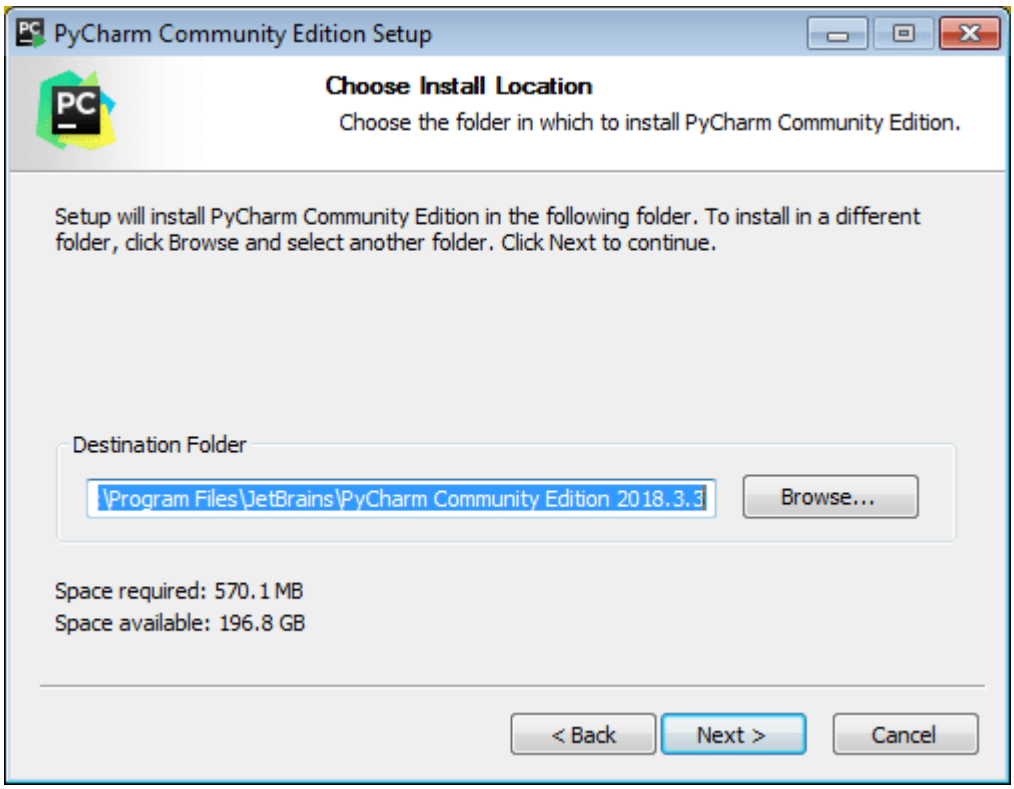


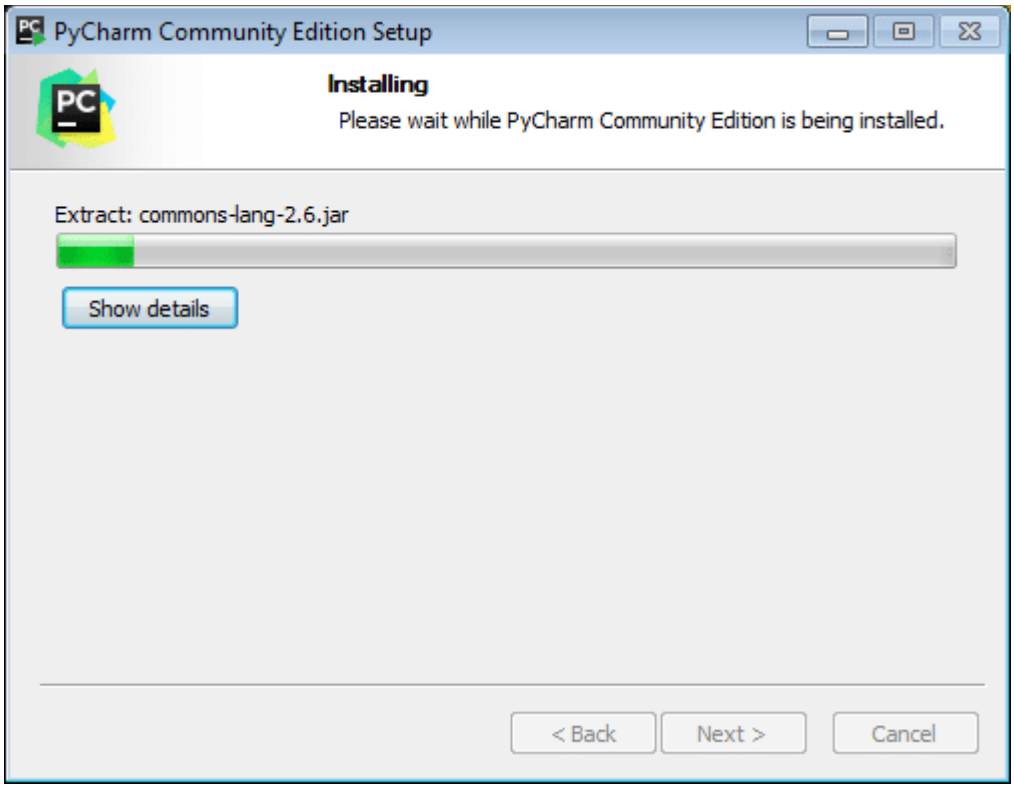
- Chosen it for (Just Me) and then clicked on Next until the installation get started.



Here are the general steps to install PyCharm:

- Visit the JetBrains website and go to the PyCharm download page.
- Download the appropriate version (Community or Professional) for your operating system (Windows, macOS, or Linux).
- Once the download is complete, run the installer.
- Follow the on-screen instructions to install PyCharm.





- After installation, launch PyCharm.

PyCharm will prompt you to create or open a project. Follow the prompts to set up your Python environment.

That's it! You should now have PyCharm installed and ready to use for Python development.

- Create a new virtual environment for the malware detection application.
- Activate the virtual environment.
- After activating the new virtual environment and install the required packages to make the our research would get done by necessary packages.

```
Collecting pandas
  Downloading pandas-2.1.4-cp311-cp311-win_amd64.whl.metadata (18 kB)
Collecting numpy<2,>=1.23.2 (from pandas)
  Downloading numpy-1.26.2-cp311-cp311-win_amd64.whl.metadata (61 kB)
     ---------------------------------------- 61.2/61.2 kB 3.2 MB/s eta 0:00:00
Collecting python-dateutil>=2.8.2 (from pandas)
  Downloading python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
     ---------------------------------------- 247.7/247.7 kB 7.7 MB/s eta 0:00:00
Collecting pytz>=2020.1 (from pandas)
  Downloading pytz-2023.3.post1-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.1 (from pandas)
  Downloading tzdata-2023.3-py2.py3-none-any.whl (341 kB)
     ---------------------------------------- 341.8/341.8 kB 7.1 MB/s eta 0:00:00
Collecting six>=1.5 (from python-dateutil>=2.8.2->pandas)
  Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Downloading pandas-2.1.4-cp311-cp311-win_amd64.whl (10.6 MB)
     ---------------------------------------- 10.6/10.6 MB 16.0 MB/s eta 0:00:00
Downloading numpy-1.26.2-cp311-cp311-win_amd64.whl (15.8 MB)
     ---------------------------------------- 15.8/15.8 MB 14.2 MB/s eta 0:00:00
Downloading pytz-2023.3.post1-py2.py3-none-any.whl (502 kB)
     ---------------------------------------- 502.5/502.5 kB 7.9 MB/s eta 0:00:00
Installing collected packages: pytz, tzdata, six, numpy, python-dateutil, pandas
```

5. Dataset Preparation

Prepare the malware dataset, specifically the BODMAS dataset, for training and evaluation. Ensure that the dataset is organized and accessible.

6. Code Execution

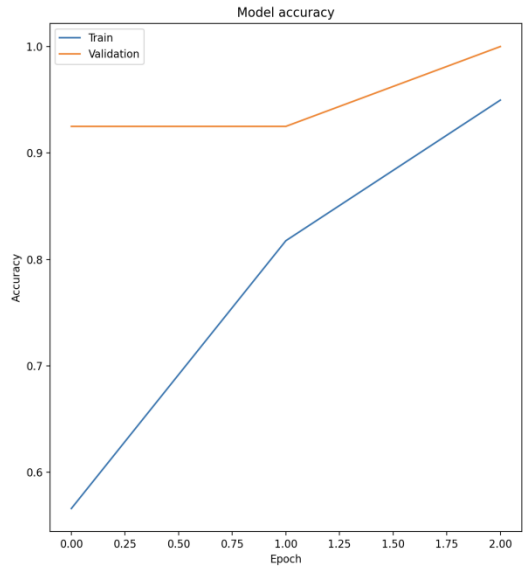
Open the pycharm ide to start developing or modifying the Scenario1&2.py and Scenario3.py scripts for the task for the malware detection.

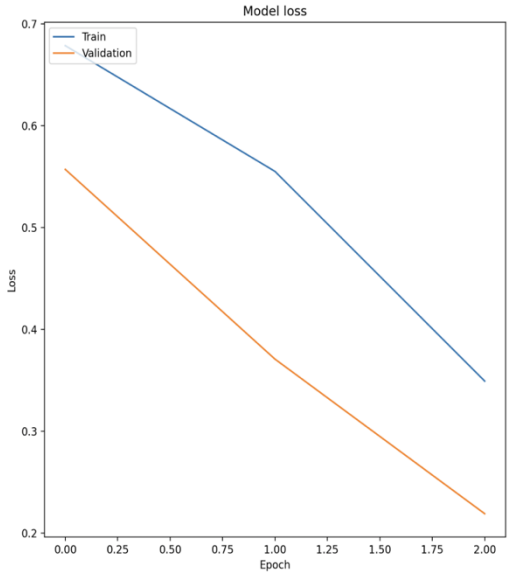
Execution Steps:

a) Selection of Transformer Architecture: Choose a suitable transformer architecture (e.g., BERT) based on specific requirements.

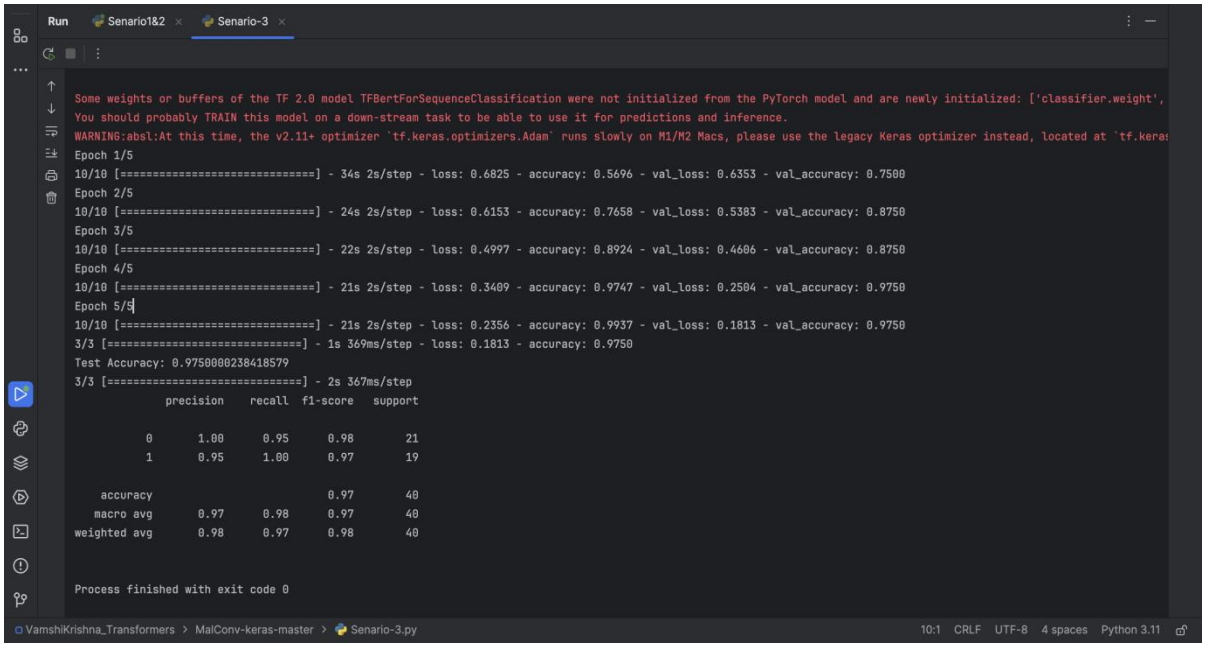
b) Pretraining of the Transformer: Implement pretraining tasks to allow the transformer to learn general-purpose representations from a diverse dataset.

```
WARNING:absl:At this time, the v2.11+ optimizer `tf.keras.optimizers.Adam` runs slowly on M1/M2 Macs, please use the legacy Keras
Epoch 1/3
10/10 [==============================] - 30s 2s/step - loss: 0.7014 - accuracy: 0.5157 - val_loss: 0.6272 - val_accuracy: 0.5250
Epoch 2/3
10/10 [==============================] - 20s 2s/step - loss: 0.5835 - accuracy: 0.8113 - val_loss: 0.4757 - val_accuracy: 0.9000
Epoch 3/3
10/10 [==============================] - 20s 2s/step - loss: 0.4228 - accuracy: 0.8994 - val_loss: 0.3732 - val_accuracy: 0.9000
```



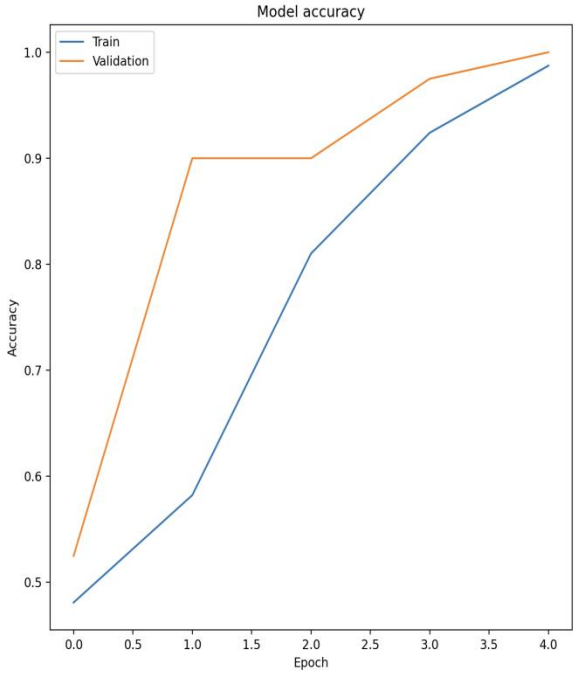


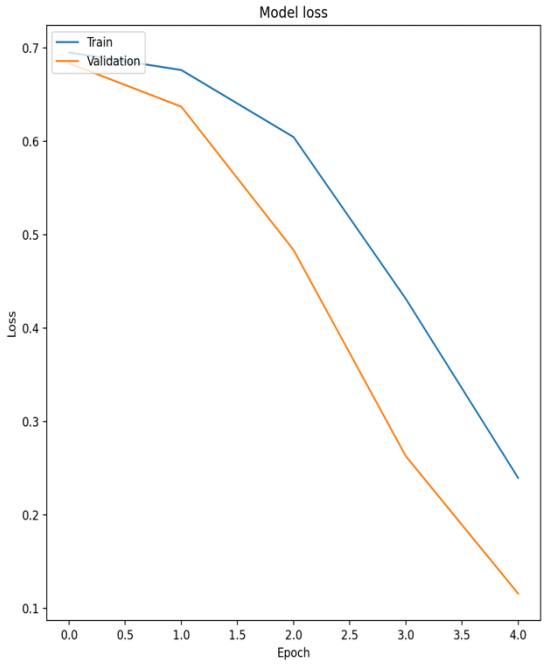
c) Fine-Tuning for Malware Detection: Fine-tune the transformer specifically for malware detection using labeled datasets.



d) Model Evaluation: Evaluate the model using the provided dataset and generate performance metrics.

Model Evaluation on Test set	
Metrics	**Results**
Test Accuracy	97.50%
Precision	95.00%
Recall	100.00%
F1 Score	97.00%





This manual guides the comprehensive configuration of the required software and tools for implementing transformers in malware detection. It covers setting up the environment, installing necessary libraries, preparing the dataset, and executing the code for effective malware identification.

References

- Anaconda: [Anaconda Installation Guide] (https://www.anaconda.com)
- Hugging Face Transformers: [Transformers Library] (https://huggingface.co/transformers)