Configuration Manual

MSc Research Project
MSc in Cloud Computing

Zaid Ali Khan
Student ID: x22204016

School of Computing
National College of Ireland

Supervisor: Shreyas Setlur Arun

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Zaid Ali Khan …………………………………………………………………………………………………
Student ID:	22204016 …………………………………………………………………………………………………
Programme:	MSc in Cloud Computing ……………………………………………………… **Year:** 2023-2024 …………………………
Module:	Research in Computing …………………………………………………………………………………………………
Lecturer:	Shreyas Setlur Arun …………………………………………………………………………………………………
Submission Due Date:	16, September 2024 …………………………………………………………………………………………………
Project Title:	Assessing Machine learning Algorithms for Cloud Computing Threat Detection …………………………………………………………………………………………………
Word Count:	413 ……………………………………… **Page Count:** 4 …………………………………

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: Zaid Ali Khan ………………………………………………………………………………………………

Date: 16, September 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

Zaid Ali Khan
Student ID: 22204016

1. Introduction

The study demonstrates the utilization of machine learning for threat detection. The study considers the usage of CICIDS 2017 Dataset for training CNN, RNN and Reinforcement learning model for prediction of threats.

2. Hardware Requirements

2.1 For macOS (High-Performance Mac Configuration)

- Model: Mac Pro or MacBook Pro
- Processor: Apple M1 Max or Intel Core i7/i9 processor
- Memory: 32 GB or 64 GB of RAM (depending on workload intensity)
- Storage: 2 TB SSD
- Graphics: Integrated Apple GPU or AMD Radeon Pro (for Mac Pro)
- Operating System: macOS Ventura or later
- Network: High-speed Ethernet or Wi-Fi 6 for data-intensive tasks

2.2 For Windows(High-Performance Windows Configuration)

- Model: Workstation PC (e.g., Dell Precision, HP Z Workstation) or high-end laptop (e.g., Dell)
- Processor: Intel Core i7/i9 or AMD Ryzen 9
- Memory: 32 GB or 64 GB of RAM
- Storage: 2 TB SSD
- Graphics: NVIDIA RTX 3060 or higher for workstation: Integrated Intel Iris Xe for laptops
- Operating System: Windows 11 Pro
- Network: Gigabit Ethernet or Wi-Fi 6

3. Software Requirements

3.1 Operating System:

- Windows 11 Pro or macOS Ventura or later

3.2 Programming Environment:

- **Python**: Version 3.8 or later
- **Anaconda**: For managing Python packages and environments
- **Jupyter Notebook**: For interactive development and documentation

3.3 Machine Learning Libraries:

- **TensorFlow**: For deep learning model development
- **Keras**: High-level neural networks API for TensorFlow
- **Scikit-learn**: For traditional machine learning algorithms
- **Pandas**: For data manipulation and analysis
- **NumPy**: For numerical computations
- **Matplotlib**: For plotting and visualization

3.4 Integrated Development Environment (IDE):

- **Visual Studio Code**: Lightweight editor with Python support

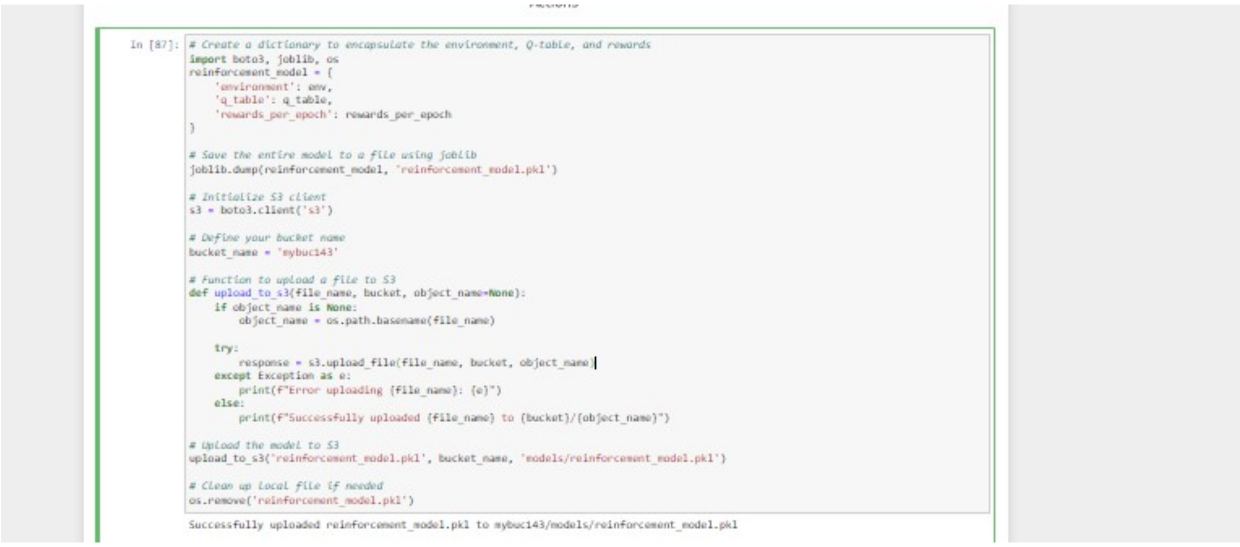
4. Dataset Used

The project makes use of the CICIDS2017 dataset provided by the Canadian Institute, for Cybersecurity which's a benchmark dataset specifically created for assessing network intrusion detection systems (NIDS). This dataset contains a range of network traffic data covering both activities and various types of cyber-attacks like DDoS and Brute Force making it well suited for evaluating threat detection models within cloud environments. The dataset, which is accessible in formats such as CSV and PCAP underwent pre-processing steps to prepare it for use in the project. These steps included handling missing data detecting outliers normalizing the data and extracting features to ensure that it was properly prepared for training machine learning algorithms aimed at enhancing security, in cloud environments.

5. Follow below steps for the code demonstration

Step 1: Open sage maker for the configured model. The graph has a –ve axis for better visualizations of the binary actions.

Step 2: Run sage maker notebook to save the model to s3 bucket.



```python
In [87]: # Create a dictionary to encapsulate the environment, Q-table, and rewards
import boto3, joblib, os
reinforcement_model = {
    'environment': env,
    'q_table': q_table,
    'rewards_per_epoch': rewards_per_epoch
}

# Save the entire model to a file using joblib
joblib.dump(reinforcement_model, 'reinforcement_model.pkl')

# Initialize S3 client
s3 = boto3.client('s3')

# Define your bucket name
bucket_name = 'mybuc143'

# Function to upload a file to S3
def upload_to_s3(file_name, bucket, object_name=None):
    if object_name is None:
        object_name = os.path.basename(file_name)

    try:
        response = s3.upload_file(file_name, bucket, object_name)
    except Exception as e:
        print(f"Error uploading {file_name}: {e}")
    else:
        print(f"Successfully uploaded {file_name} to {bucket}/{object_name}")

# Upload the model to S3
upload_to_s3('reinforcement_model.pkl', bucket_name, 'models/reinforcement_model.pkl')

# Clean up local file if needed
os.remove('reinforcement_model.pkl')

Successfully uploaded reinforcement_model.pkl to mybuc143/models/reinforcement_model.pkl
```

Figure 1: Save Model to S3

Step 3: Run python app.py and browse to local host http://localhost:5000 for the frontend

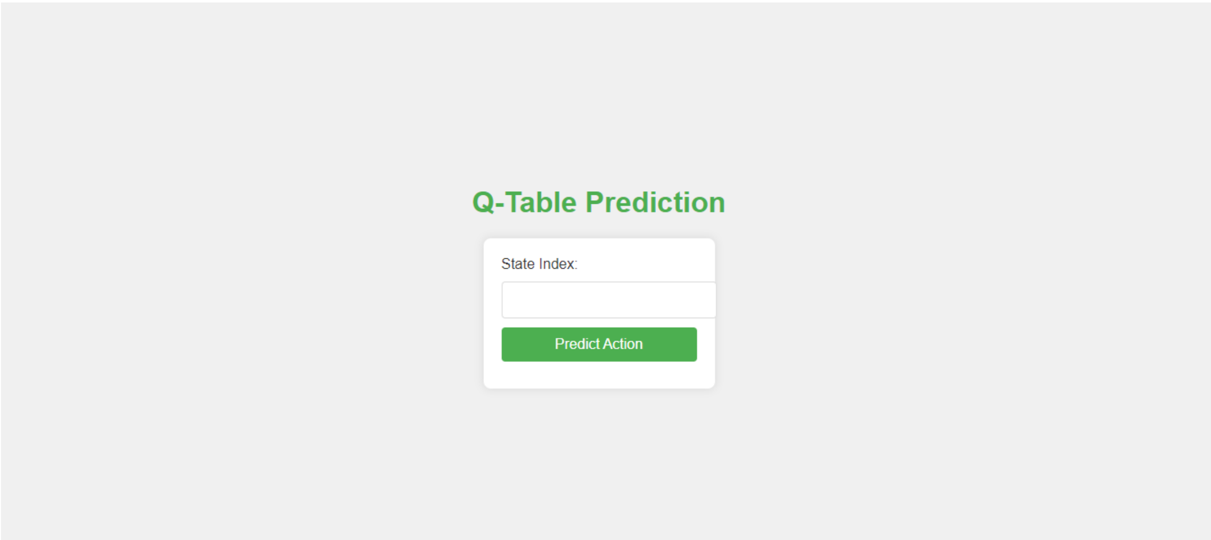


Figure 2: User-Interface of RL Model

Step 4: Enter values to predict Action.

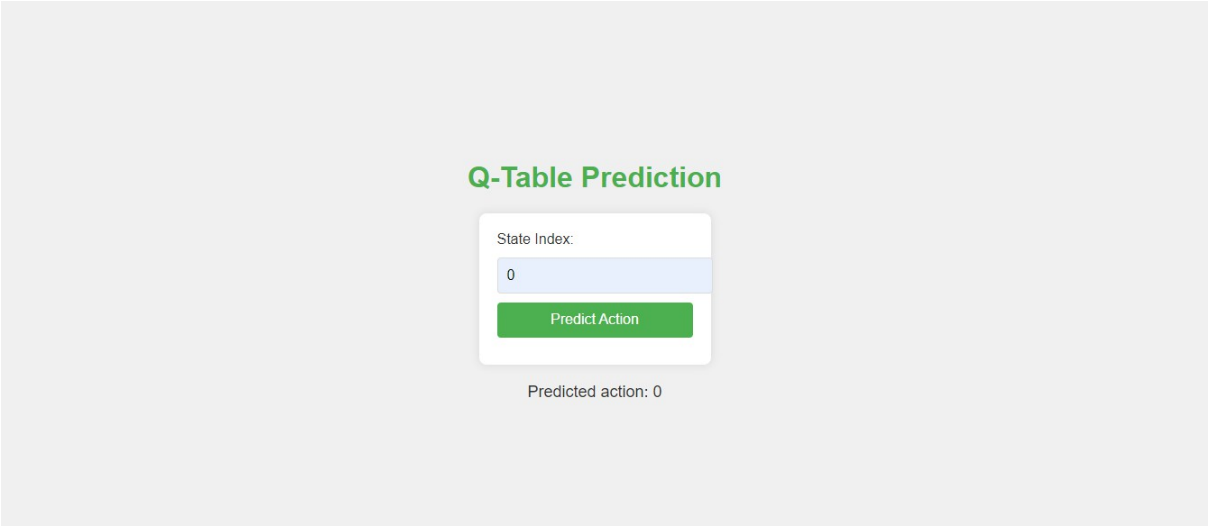


Figure 3: User-Interface of RL Model (Insert State)

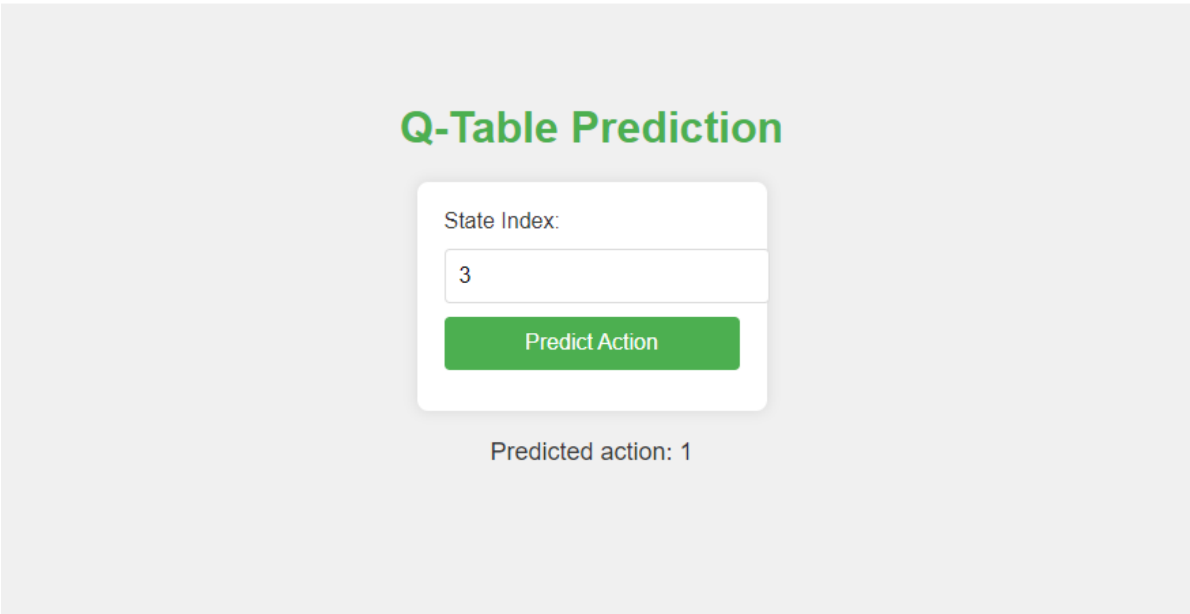


Figure 4: User-Interface of RL Model (Predict Action)