# Configuration Manual

MSc Research Project
Programme Name

Ravi Ranjan Singh
Student ID: x22203052

School of Computing
National College of Ireland

Supervisor: Michael Prior

## National College of Ireland

## Project Submission Sheet

**Student Name:** **Ravi Ranjan Singh**

**Student ID:** **X22203052**

**Programme:** **MSc Cybersecurity** **Year:** **Jan 2024**

**Module:** **MSc Research Project**

**Lecturer:** **Michael Prior**
**Submission Due Date:** **12/12/24**

**Project Title:** **Exploring the use of Explainable AI for improving intrusion detection systems**

**Word Count:** **565**

**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:** **Ravi Ranjan Singh**

**Date:** **11/12/2024**

**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.**

| **Office Use Only** | |
|---|---|
| Signature: | |

| | |
|---|---|
| Date: | |
| Penalty Applied (if applicable): | |

# AI Acknowledgement Supplement

## 1 MSc Research Project

## 2 Exploring the use of Explainable AI for improving intrusion detection systems

| Your Name/Student Number | Course | Date |
|---|---|---|
| **Ravi Ranjan Singh** | MSc Cybersecurity | 11/12/2024 |

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.

## 3 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** | | |
| | | |

## 4 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**.

| NA | |
|---|---|
| [NA | |
| [Insert Sample prompt] | [Insert Sample response] |

## 5 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.

## 6 Additional Evidence:

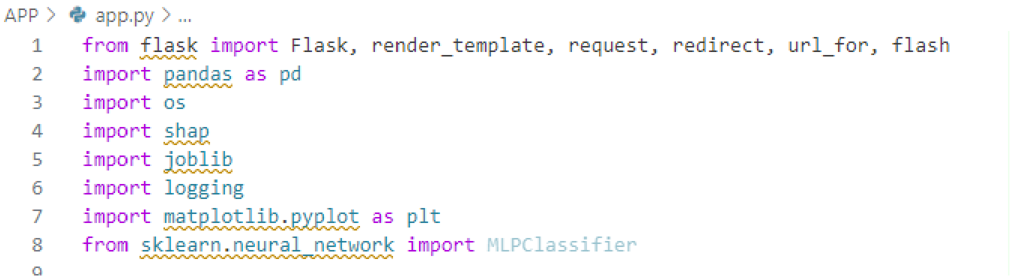
[Place evidence here]

# Configuration Manual

Ravi Ranjan Singh
Student ID: x22203052

## 1. Libraries Imported for Web Application

Such web applications would not have been developed were it not for the frameworks that are underlying web applications and machine learning model development. The web framework is constructed with Flask, which has methods for handling and redirecting users including render_template, request and redirect, and url_for methods (Ghimire, 2020). Pandas are employed in data processing and CSV file editing. The SHAP elements help improve explainable AI by incorporating SHAP values into the final output of models while the OS supports and manages the navigation within files and folders. Joblib is the package responsible for retrieving the ML elements that have previously been learned, while logging is the worker that notes events and failures that happen as the system runs, which helps with debugging and monitoring (Brownlee *et al.*, 2022).



```python
from flask import Flask, render_template, request, redirect, url_for, flash
import pandas as pd
import os
import shap
import joblib
import logging
import matplotlib.pyplot as plt
from sklearn.neural_network import MLPClassifier
```

**Figure 1: Libraries imported**

## 2. Methods used

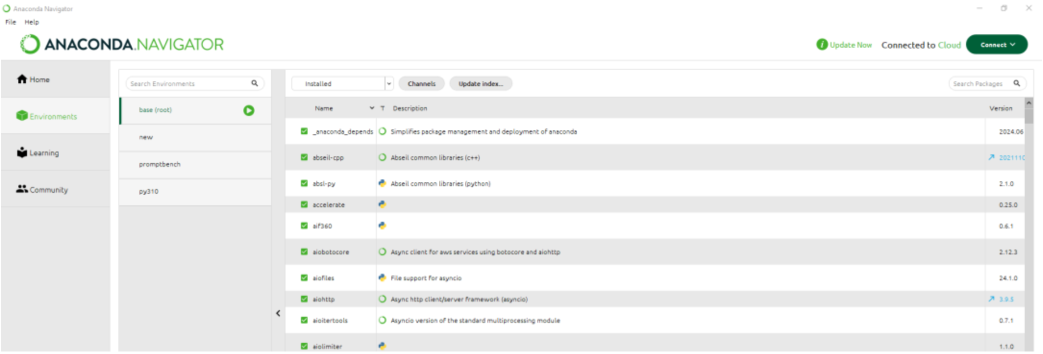
A web application that is developed on the Flask platform has a minimum of some critical endogenous processes. Index() is responsible for rendering the main page of the website which contains the index.html file. The upload_file() function is responsible for the process of uploading a file. It validates the file saves it in the uploads folder and afterwards transforms it into a Pandas DataFrame. It adds zero values in sparse columns of the feature set of the machine learning model during this process. model_rf.predict() is then executed in this case to apply the built model for making predictions. In addition, SHAP values can be computed using SHAP.TreeExplainer to enhance the interpretability of the SHAP values. The prediction and the SHAP features are along the result page in result.html. For debugging purposes, the error handling and logging code components are included.

## 3 Style.css

In the style.css file for the web application, there is a modern layout with a combination of red and white color schemes in the app design. A CSS reset is performed at every beginning for the reason that the margin and padding are similar in all browsers. Most of the contents of the body are shaded in pale gray color, while the header section is also dominated by a bold red color along with white words to attract the attention of the relevant viewers to the title of the page. The user can see white bold links in the horizontal navigation bar, which turn into highlighting ones after the pointer gets onto one of them. The content section visually stands out on the page with the help of a normal white page with a little shadow on it for contrast and with sufficient padding. To avoid bulkiness and for comprehension, the forms are centrally located and the input boxes are placed vertically. An action that gives feedback to the user even as he submits an action which is normally associated with a submit button, in this case, the submit button has a red color and has a hover effect. The footer of the application still carries out the theme of professionalism and neatness of the application presenting black color and white text.

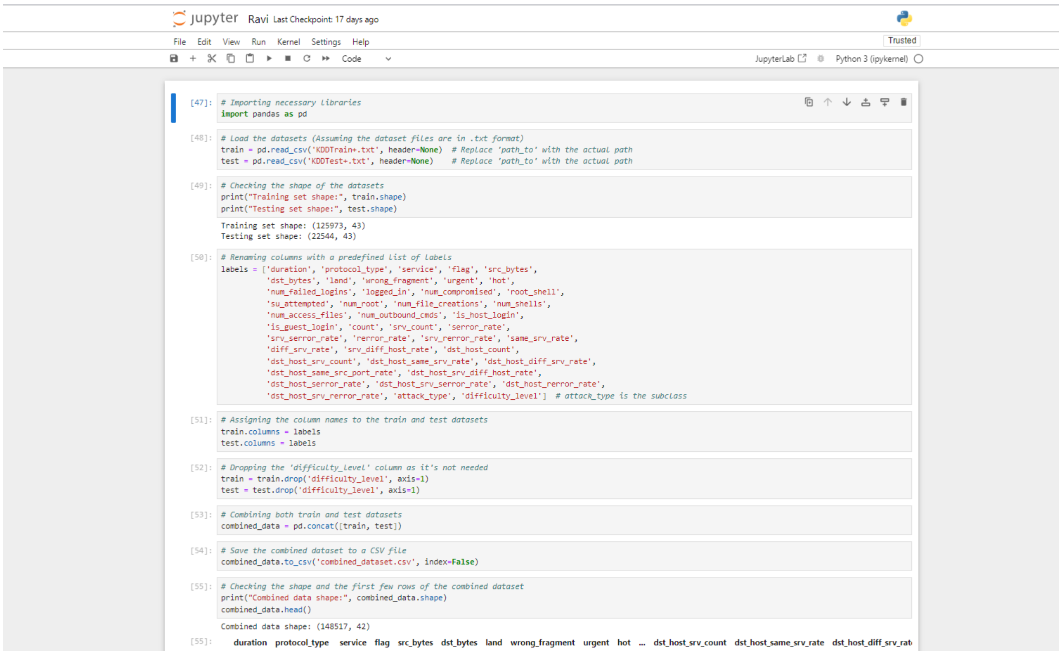
## 4 Software and Hardware Specifications

Software used for this paper are Anaconda 2.6.3, Jupyter Notebook for Model Evaluation and for web application development we used Visual Studio Code and Flask. The hardware specifications include Ryzen 5 processor, Ram 16GB, SSD 512GB.



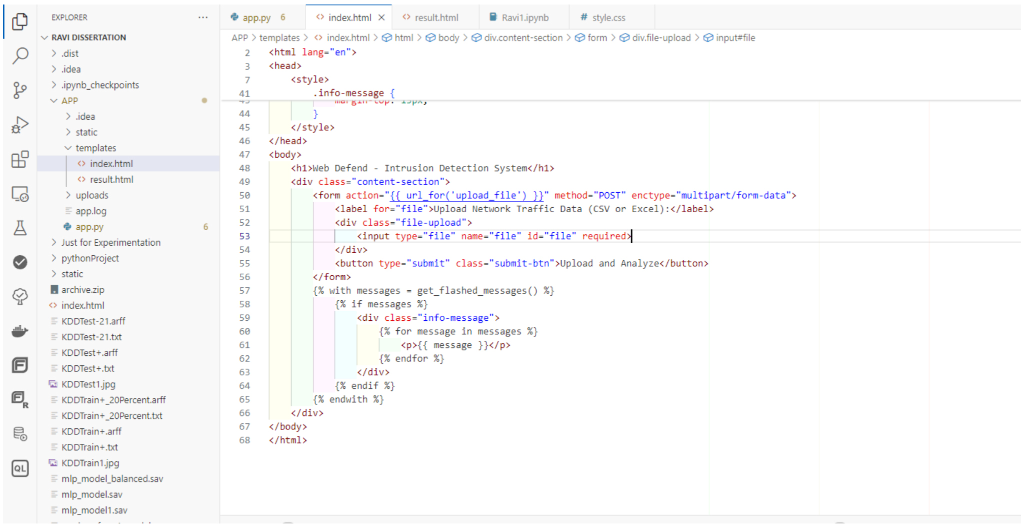
| | Name | T | Description | Version |
|---|---|---|---|---|
| ☑ | _anaconda_depends | ○ | Simplifies package management and deployment of anaconda | 2024.06 |
| ☑ | abseil-cpp | ○ | Abseil common libraries (c++) | ↗ 2021110 |
| ☑ | absl-py | | Abseil common libraries (python) | 2.1.0 |
| ☑ | accelerate | | | 0.25.0 |
| ☑ | aif360 | | | 0.6.1 |
| ☑ | aiobotocore | ○ | Async client for aws services using botocore and aiohttp | 2.12.3 |
| ☑ | aiofiles | | File support for asyncio | 24.1.0 |
| ☑ | aiohttp | ○ | Async http client/server framework (asyncio) | ↗ 3.9.5 |
| ☑ | aioitertools | ○ | Asyncio version of the standard multiprocessing module | 0.7.1 |
| ☑ | aiolimiter | | | 1.1.0 |

Figure 2: Anaconda Navigator



```
[47]: # Importing necessary libraries
      import pandas as pd

[48]: # Load the datasets (Assuming the dataset files are in .txt format)
      train = pd.read_csv('KDDTrain+.txt', header=None)  # Replace 'path_to' with the actual path
      test = pd.read_csv('KDDTest+.txt', header=None)    # Replace 'path_to' with the actual path

[49]: # Checking the shape of the datasets
      print("Training set shape:", train.shape)
      print("Testing set shape:", test.shape)

      Training set shape: (125973, 43)
      Testing set shape: (22544, 43)

[50]: # Renaming columns with a predefined list of labels
      labels = ['duration', 'protocol_type', 'service', 'flag', 'src_bytes',
                'dst_bytes', 'land', 'wrong_fragment', 'urgent', 'hot',
                'num_failed_logins', 'logged_in', 'num_compromised', 'root_shell',
                'su_attempted', 'num_root', 'num_file_creations', 'num_shells',
                'num_access_files', 'num_outbound_cmds', 'is_host_login',
                'is_guest_login', 'count', 'srv_count', 'serror_rate',
                'srv_serror_rate', 'rerror_rate', 'srv_rerror_rate', 'same_srv_rate',
                'diff_srv_rate', 'srv_diff_host_rate', 'dst_host_count',
                'dst_host_srv_count', 'dst_host_same_srv_rate', 'dst_host_diff_srv_rate',
                'dst_host_same_src_port_rate', 'dst_host_srv_diff_host_rate',
                'dst_host_serror_rate', 'dst_host_srv_serror_rate', 'dst_host_rerror_rate',
                'dst_host_srv_rerror_rate', 'attack_type', 'difficulty_level']  # attack_type is the subclass

[51]: # Assigning the column names to the train and test datasets
      train.columns = labels
      test.columns = labels

[52]: # Dropping the 'difficulty_level' column as it's not needed
      train = train.drop('difficulty_level', axis=1)
      test = test.drop('difficulty_level', axis=1)

[53]: # Combining both train and test datasets
      combined_data = pd.concat([train, test])

[54]: # Save the combined dataset to a CSV file
      combined_data.to_csv('combined_dataset.csv', index=False)

[55]: # Checking the shape and the first few rows of the combined dataset
      print("Combined data shape:", combined_data.shape)
      combined_data.head()

      Combined data shape: (148517, 42)

[55]:    duration  protocol_type  service  flag  src_bytes  dst_bytes  land  wrong_fragment  urgent  hot  ...  dst_host_srv_count  dst_host_same_srv_rate  dst_host_diff_srv_rat
```

## Figure 3: Jupyter Notebook



```
<html lang="en">
<head>
    <style>
        .info-message {
            margin-top: 15px;
        }
    </style>
</head>
<body>
    <h1>Web Defend - Intrusion Detection System</h1>
    <div class="content-section">
        <form action="{{ url_for('upload_file') }}" method="POST" enctype="multipart/form-data">
            <label for="file">Upload Network Traffic Data (CSV or Excel):</label>
            <div class="file-upload">
                <input type="file" name="file" id="file" required>
            </div>
            <button type="submit" class="submit-btn">Upload and Analyze</button>
        </form>
        {% with messages = get_flashed_messages() %}
            {% if messages %}
                <div class="info-message">
                    {% for message in messages %}
                        <p>{{ message }}</p>
                    {% endfor %}
                </div>
            {% endif %}
        {% endwith %}
    </div>
</body>
</html>
```

## Figure 4: VS CODE

## References

Brownlee, J., Chng, Z.M., Chung, D., Cristina, S., Saeed, M. and Tam, A., 2022. *Python for Machine Learning: Learn Python from Machine Learning Projects*. Machine Learning Mastery.

Ghimire, D., 2020. Comparative study on Python web frameworks: Flask and Django.