# Configuration Manual

MSc Research Project
MSc in Cybersecurity

Forename Surname
Student ID: x23251506

School of Computing
National College of Ireland

Supervisor: Michael Pantridge

**National College of Ireland**

**MSc Project Submission Sheet**

**School of Computing**

| | | | |
|---|---|---|---|
| **Student Name:** | Rohan Vivek Lahane | | |
| **Student ID:** | x23251506 | | |
| **Programme:** | MSc in Cybersecurity | **Year:** | 2024-25 |
| **Module:** | Practicum | | |
| **Lecturer:** | Michael Pantridge | | |
| **Submission Due Date:** | 11-08-2025 | | |
| **Project Title:** | AI-Driven Self-Learning Intrusion Detection & Response System for Cloud Security | | |
| **Word Count:** | 728 | **Page Count:** | 5 |

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:** Rohan Vivek Lahane

**Date:** 11-08-2025

**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): | |

# Configuration Manual

Rohan Vivek Lahane
Student ID: x23251506

## 1 System Requirements and Installation

Intrusion Detection System (IDS) was done in Python version 3.9 and tested on windows 11 using visual studio code. The data can be ensured proper functioning with the following hardware and software specifications:

**Hardware Requirements:**

- At least 8 GB RAM (16 GB or more is recommended to train faster)
- AMD Ryzen 7 7735 HS
- Min 2 GB of free disk space

**Software Requirements:**

- Python 3.9 and up
- Venv
- Python packages needed (see requirements.txt)

**Installation Steps:**

- A project repository can be cloned or downloaded to the local machine.
- In the terminal, go to project directory.
- Design a virtual surrounding: python -m venv .venv

**Turn on the virtual world:**

- Windows (PowerShell): activate .venv\Scripts
- Install the dependencies: pip install requirements.txt

# 2 Configuration and Data Setup

As well as training with historical data, the IDS can make real-time predictions trained on simulated AWS CloudTrail log data.

**Dataset Preparation:**

- Training Data: Put KDDTrain_cleaned.csv KDDTest_cleaned.csv in folder /data.
- Environmental Data to be predicted in Real Time:
- The mock AWS CloudTrail events that you can test against are in sample_cloudtrail_logs.csv or sample_cloudtrail_logs.json.
- The live detection module receives this file and mock ingests the AWS logs.

**Configuration Files:**

- paths and thresholds of predictions may be stored in config.json (not empty).
- In the code, change DATA_PATH in:
    - The live dashboard is in view_live_predictions.py.
    - The model evaluation dashboard is called dashboard.py.

# 3 Running the system

- **Model Training**

To preprocess, train the XGBoost model, evaluate it and save both, the trained model file (xgboost_model.pkl) and the scaler and encoder objects, run the following: python src/main

- **Live Prediction Generation**

The live detection script has to be run to analyze the AWS CloudTrail logs, and the predicted logs must be stored in predicted_live_logs.csv: python live_detection.py

- **Viewing Dashboards**
    - **Model Evaluation Dashboard:** streamlit run src/dashboard.py
      It shows performance measures, confusion matrix, feature influence and filtering.
    - **Live Prediction dashboard:** streamlit run src/view_live_predictions.py
      Displays prediction outcomes in real time, suspicious incidents, trends, possible downloads (CSV).

# References


1. Alharkan, I. & Martin, A., 2020. *An investigation of identity and access management (IAM) misuse in cloud computing*. *Journal of Cloud Computing*, 9(1), pp.1–14. [Accessed 10 July 2025].
2. Almseidin, M. et al., 2017. *Evaluation of machine learning algorithms for intrusion detection system*. *Procedia Computer Science*, 127, pp.503–509. [Accessed 12 July 2025].
3. Aljawarneh, S., Aldwairi, M. & Yassein, M.B., 2018. *Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model*. *Journal of Computational Science*, 25, pp.152–160. [Accessed 19 July 2025].
4. Dehghantanha, A. et al., 2016. *Identity-centric security model for IoT*. *IEEE Communications Magazine*, 54(12), pp.52–57. [Accessed 9 July 2025].
5. Jeyanthi, A. & Tiwari, S., 2023. *Generative Adversarial Network-based Intrusion Detection for Cloud Environments*. *Journal of Information Security and Applications*, 70, p.103249. [Accessed 7 July 2025].