**Pre-interview Assessment**

**Scenario: Anomaly Detection Challenge – Detecting Attacks on a robotic vehicle**

Develop a prototype anomaly detection machine learning classifier to identify attacks in a robotic vehicle, using primarily unsupervised methods. Candidates are free to use any algorithms they wish; however, the anomaly detection **MUST** initially originate from unsupervised methods.

This challenge is designed to assess both a candidate’s ability to tackle a machine learning task that is commonly related to intrusion detection, and their software development skills for fast prototyping. **NOTE**: Failing to complete fully any task **DOES NOT** automatically exclude the candidate from interview or from due consideration for the Machine Learning engineer position.

**Candidates are required to share a Github link to their repo containing the code they have developed for the tasks and any relevant documentation (e.g., overview, results, test harnesses) for review.**

**Task 1**

Find and detect the anomalies in the robotic vehicle dataset

* Employ any feature engineering and normalisation methods to optimise learning
* Identify and highlight the key feature(s) contributing to anomaly detection
* Present model train and test performance using validation techniques and criteria

**Task 2:**

Deploy the trained model as a REST API using Docker for exposing it as a service

* Demonstrate the service being queried with new data in JSON format
* Show REST API outputting result data in JSON format

**Task 3 (Desirable):**

Develop a mechanism for detecting the “type” of attack vector from the following categories and present these results by reporting the row numbers in the csv that correspond to each attack vector:

* Denial of Service
* Command Injection
* Malware Command and Control

**Dataset information**

The following features are included in the dataset“vulnerable\_robot\_challenge.csv”**:**

t Time step

CPU Total CPU utilisation

RxKBTot Received network traffic rate

TxKBTo Transmitted network traffic rate

WriteKBTot The rate of data being written to disk

Watts Power consumption reading in watts

Amps Power consumption reading in amps

RMS Root mean square reading of accelerometer for vibration of chassis

diff\_encoder\_l Quadratic rotary encoder reading for motor

flag Ground truth **ONLY** to be used for model validation (1 = attack, 0 = normal)

Candidates are encouraged to use feature engineering methods for improving the expressiveness and information gain of features in the dataset.