COM6012 Assignment Part 1 - Deadline: 03:00 PM, May 125, 2022

New Deadline: 03:00 PM, May 12, 2022

Please, carefully read the assignment brief before starting to complete the assignment

Assignment Brief

How and what to submit

A. Create a .zip file containing the following:

1) **ASPart1_report.pdf**: A report in PDF containing answers to ALL questions. The report should be concise. You may include appendices/references for additional information but marking will focus on the main body of the report.
2) **Code, script, and output files:** All files used to generate the answers for individual questions above, **except the data**. These files should be named properly starting with the question number: e.g., your python code as **QP1**_xxx**.py**, your script for HPC as **QP1**_HPC.sh, and your output files on HPC such as QP1_output.txt or QP1_figB.jpg. The results should be generated from the HPC, not your local machine.

B. Upload a .zip file to Blackboard (BB) before the deadline that contains the files in A (you can group them in a folder called Part1) and any other files requested for the solution of Part 2 of the Assignment (Dr Lu will be in charge of releasing this part).

C. **NO DATA UPLOAD**: Please do not upload the data files used. We have a copy already. Instead, please use a **relative file path in your code (data files under folder 'Data')**, as in the lab sheets so that we can run your code smoothly.

D. **Code and output**. 1) Use **PySpark** as covered in the lecture and lab sessions to complete the tasks; 2) **Submit your PySpark job to HPC** with **qsub** to obtain the output.

Assessment Criteria (Scope: Sessions 3-6; Total marks: 20)

1. Being able to use pipelines, cross-validators and a different range of supervised learning methods for large datasets
2. Being able to analyse and put in place a suitable course of action to address a large scale data analytics challenge

Late submissions: We follow the Department's guidelines about late submissions, i.e., "If you submit work to be marked after the deadline you will incur a deduction of 5% of the mark each working day the work is late after the deadline, up to a maximum of 5 working days" but **NO late submission will be marked after the maximum of 5 working days** because we will release a solution by then. Please see this link.

Use of unfair means: "*Any form of unfair means is treated as a serious academic offence and action may be taken under the Discipline Regulations.*" (from the MSc Handbook). Please carefully read this link on what constitutes Unfair Means if not sure.

Please, only use interactive HPC when you work with small data to test that your algorithms are working fine. If you use rse-com6012 in interactive HPC, the performance for the whole group of students will be better if you only use up to four cores and up to 10G per core. When you want to produce your results for the assignment and/or want to request access to more cores and more memory, **PLEASE USE BATCH HPC**. This will be mandatory. We will monitor the time your jobs are taking to run and will automatically "qdel" the job if it is taken much more than expected. **We want to promote good code practices (e.g. memory usage) so, please, once more, make sure that what you run on HPC has already been tested enough for a smaller dataset.** It is OK to attempt to produce results several times in HPC, but please, be mindful that extensive running jobs will affect the access of other users to the pool of resources. **Review Lab 4 where examples of good practices were described.**

Scalable supervised learning to study the Vulnerability of Resource-Constraint Internet of Things

In part 1 of the assignment, you will explore the use of scalable supervised classification algorithms to discover vulnerabilities in certain type of Internet of Things (IoT) devices. The dataset was generated as part of the paper "A Machine Learning-Based Security Vulnerability Study on XOR PUFs for Resource-Constraint Internet of Things" by Ahmad O. Aseeri; Yu Zhuang; Mohammed Saeed Alkatheiri published in the 2018 IEEE International Congress on Internet of Things (July 2018).

Quoting from the paper,

"Physical unclonable functions (PUFs) utilize variations in integrated circuits to produce responses unique for individual PUFs, and hence are not reproducible by manufacturers as well as by attackers even if they have obtained the exact PUF circuit design, possessing great potential for implementing secure mechanisms. Implementable with very simplistic circuits with extremely low energy and other resource requirements, PUFs are particularly promising for delivering high security for resource-constraint IoT devices"

"In a nutshell, a PUF is an input-output mapping in the form Υ: $\{0,1\}^n \rightarrow \{0,1\}^m$ where the input is n-bit binary vector (called *challenge*) and the output is m-bit binary vector (called *response*). A PUF circuit can receive 2^n different possible n-bit challenge vectors, each of which produces an m-bit output. Silicon PUFs can generate unique challenge-response pairs (CRPs) for different integrated circuit instances"

"While physically unreproducible, PUFs are reported to be "mathematically clonable" by machine learning-based modeling methods which can accurately predict the responses of PUFs. Mathematical clonability allows attackers to develop malicious software to impersonate the identity of trusted PUF-embedded devices by producing the same responses PUFs would give."

The dataset that you will use is from the UCI repository and can be downloaded from this link. The downloaded file is a .zip file. The uncompressed file includes two folders, namely,

5xor_128bit and 6xor_64bit. You will work with the files inside the folder **5xor_128bit.** There are two .csv files inside 5xor_128bit: train_5xor_128dim.csv and test_5xor_128dim.csv. Each file contains a table with 129 columns, where the last column is the label (1 or -1) and the first 128 columns are the features. Each row corresponds to a CRP, with the features representing the challenge and the label representing the (binary) response. When it comes to testing your algorithms, you will use the exact same train and test sets.

You will apply Random Forests, Logistic Regression (including regularisation) and (shallow) Neural networks over a subset of the dataset and over the full dataset. As performance measures use classification accuracy.

1. Working with a subset of the larger dataset. Use pipelines and cross-validation to find the best configuration of parameters for each model **[8 marks]**
    a. For finding the best configuration of parameters, use 1% of the data chosen randomly from the training set **[2 marks]**
    b. Use a sensible grid for the parameters (for example, at least the three most relevant parameters with at least three options for each parameter) for each predictive model:
        i. Pipeline and cross-validation for random forests **[2 marks]**
        ii. Pipeline and cross-validation for logistic regression **[2 marks]**
        iii. Pipeline and cross-validation for neural networks **[2 marks]**

    Please, use the batch mode to work on this. Although the dataset is not as large, the batch mode allows queueing jobs and for the cluster to better allocate resources.

2. Working with the larger dataset. Once you have found the best parameter configurations for each algorithm in the smaller subset of the data, use the full dataset to train on the whole training data and compare the performance of the three algorithms on the test data **[8 marks]**
    a. Use the best parameters found for each model in the smaller dataset of the previous step, for the models used in this step. You need to pass these parameters programmatically. Do not hard-code them. **[2 marks]**
    b. Provide accuracy, area under the curve, training time and testing time when using **five CORES** and **ten CORES** for each of the predictive models using train_5xor_128dim.csv for training and test_5xor_128dim.csv for testing.
        i. Five and ten cores results for random forests **[2 marks]**
        ii. Five and ten cores results for logistic regression **[2 marks]**
        iii. Five and ten cores results for neural networks **[2 marks]**

    Remember to use the batch mode to work on this.

3. Discuss at least four observations (e.g., anything interesting), with two to three sentences for each observation. If you need to, you can run additional experiments that help you to provide these observations **[4 marks] (one mark per observation)**

Do not try to upload the dataset to BB when returning your work. **It is around 200 Mb (compressed) and around 2.34Gb (uncompressed).**

COMMENTS: 1) An old, but very powerful engineering principle says: *divide and conquer*. If you are unable to analyse your datasets out of the box, you can always start with a smaller one, and build your way from it. **2)** Use **wget** to download the data file directly in HPC. You can navigate to /data/your_username/ScalableML/Data/ in HPC and once there type

```
wget https://archive.ics.uci.edu/ml/machine-learning-databases/00463/XOR_Arbiter_PUFs.zip
```

You can copy the link to the zip file after wget. To uncompress the .zip file, use **unzip**

```
unzip XOR_Arbiter_PUFs.zip
```

FAQ

- **How long does it take to run the whole assignment?**

  Using a minimum of three hyperparameters and a minimum of three options for each hyperparameter in Step 1, running the whole assignment in Batch mode should take less than 2 hours when using 10 cores.