Project Brief

Dr Paul Yoo

Dept CSIS

07/11/19

Overview

We covered:
- Linear Regression
- Logistic Regression
- Neural Network (MLP)
- Activation Functions
- Loss and Cost Functions

We will cover:
- Group Practical
- Application: IoT Intrusion Detection
- AWID (Kolias *et al* 2015)
- DEMISe (Parker *et al* 2019)
- Evaluation

Assessment (Group Practical)

- A report (inc. individual section) of a group project worth 30% of your total mark.

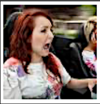
Code	Module title	Coursework element	Publication date	Deadline	Late cut-off deadline	Mark return date
BUCI077H7	Applied Machine Learning	Project	Monday, 11 November 2019	Sunday, 19 January 2020	Sunday, 2 February 2020	Friday, 14 February 2020

URL: https://www.dcs.bbk.ac.uk/intranet/index.php/Coursework_Deadlines_Autumn_2019

Aim of the Assessment

- The aim of this assessment is to provide a hands-on, practical, assessment of your machine learning skills for practical IoT intrusion detection application.

Five Components Of Autonomous Car Security
Forbes - 31 Oct 2019
Among all AI solutions, I believe the **cybersecurity** of **autonomous cars** is the most crucial aspect. One incident alone affecting human lives ...



Could a hacker hijack your connected **car**?
BBC News - 5 Oct 2017
As more carmakers adopt "over the air (OTA)" software updates for their increasingly connected and **autonomous cars**, is the risk of hacker ...

'Smart pods' blaze a trail for **autonomous** public transport
CNN - 1 Nov 2019
But perhaps the most crucial distinction is that this self-**driving vehicle** is ... The **autonomous** pods from NEXT and similar designs could play a key role in .... "We feel with these shuttle buses the technology is quite mature," Bahrozyan told **CNN**. How are we going to tackle **cybersecurity** with all these **cars** ...

Danger On The Road
Business Today - 17 Oct 2019
But connected **vehicles** are already under threat - not from retaliating truckers but sinister hackers as the industry looks towards an **automated** future. ... upon us, **cybersecurity** would be the most critical issue to protect this ubiquitous ... Their new **CNN** model was trained and tested on large, publicly available ...

Driverless **cars** to be subject to 'digital MOT' under ...
Telegraph.co.uk - 4 Sep 2019
Driverless cars to be subject to 'digital MOT' under Government ... Ministers have announced they are drawing new standards for **autonomous vehicles** that will set out ... cars will also have to be tested for **cyber security** and whether their .... **BBC** pay should be broken down into hours worked, Clare Balding ...

UK to develop 'world-leading' safety standard for autonomous cars

The new regime, called CAV PASS, is being developed by "world-leading" experts in vehicle safety and cyber security from the Government, industry and the academic world. It's intended to ensure that self-driving vehicles "are safe and secure by design and minimise any defects ahead of their testing, sale and wider deployment on UK roads."

16

Assessment Requirements

IoT Intrusion Detection Competition using Machine Learning

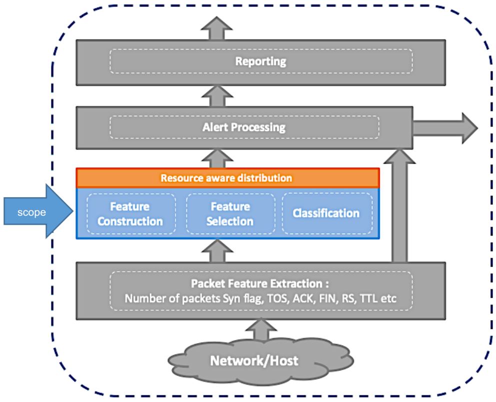
- Software to detect wireless network intrusions protects a computer system from various cyber attacks, including perhaps insiders.
- The task of intrusion detection learning is to build a predictive model (i.e. a machine-learning classifier) capable of distinguishing between "bad" traffic, called intrusions or attacks, and "good" normal traffic.

17

AI-based Intrusion Detection for IoT Enabled Devices

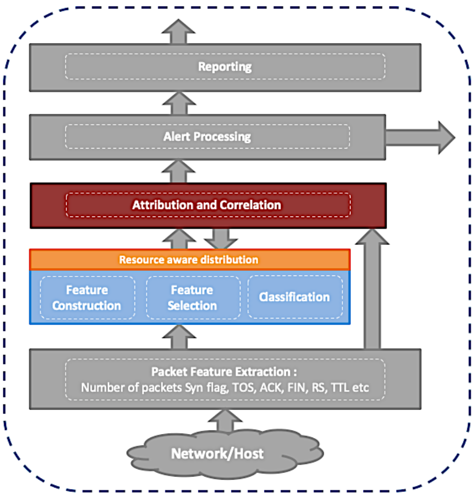
To develop replacements for traditional security technology using a sophisticated adaptive modelling approach that combines machine learning and behavioural analytics while minimising the use of computing power and energy.

- Sensors gather raw data from both the network and the host, filter incoming data, and extract interesting and potentially valuable information.
- DL includes classifiers trained with supervised machine learning techniques assesses each event and search for suspicious behaviour.
- The computational cost are distributed across the computing nodes.
- AP layer performs attack analysis and generates alerts.
- RL generates a report.



A Data-driven Framework for Attribution and Correlation in Intrusion Detection

- Attribution identifies the virtual actors responsible for cyberattacks.
- Provide proof of involvement by specific groups.
- Can be identified by their methods of attack, consistent errors and other unique characteristics (patterns).
- Support potential sanctions and policy decisions.
- Discourage attacks by providing transparency for activities that are normally hidden.
- Attributing attacks to specific groups or individuals could be partially achieved today.
- Is a manual process that requires highly skilled investigators and weeks or months to complete.
- Machine learning and behaviour analytics to scale up the attribution process to help companies and the government protect against bad actors.

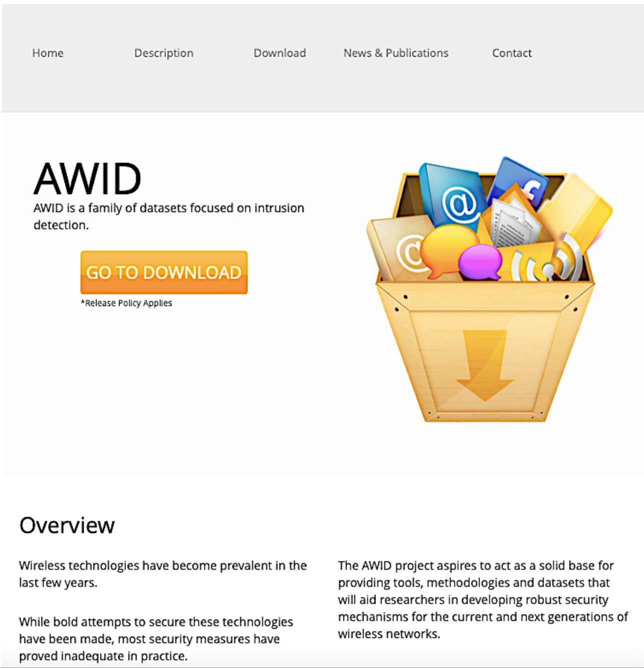


The Aegean WiFi Intrusion Dataset (AWID)

- Prepared and managed by George Mason University and University of the Aegean.
- The objective was to survey and evaluate ML research in intrusion detection.
- Real traces of both normal and intrusive 802.11 traffic
- A wide variety of intrusions simulated in a physical lab which realistically emulates a typical SOHO infrastructure.

20

URL: http://icsdweb.aegean.gr/awid/



21

Kolias, C., Kambourakis, G., Stavrou, A. and Gritzalis, S., 2015. Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset. IEEE Communications Surveys & Tutorials, 18(1), pp.184-208.

Intrusion Detection in 802.11 Networks: Empirical Evaluation of Threats and a Public Dataset

Constantinos Kolias, Georgios Kambourakis, Angelos Stavrou, and Stefanos Gritzalis

***Abstract*—WiFi has become the de facto wireless technology for achieving short to medium-range device connectivity. While early attempts to secure this technology have been proved inadequate in several respects, the current, more robust, security amendments will inevitably get outperformed in the future too. In any case, several security vulnerabilities have been spotted in virtually any version of the protocol rendering the integration of external protection mechanisms a necessity. In this context, the contribution of this paper is multi-fold. First, it gathers, categorizes, thoroughly evaluates the most popular attacks on 802.11, and analyzes their signatures. Second, it offers a publicly available dataset containing a rich blend of normal and attack traffic against 802.11 networks. A quite extensive first-hand evaluation of this dataset using several machine learning algorithms and data features is also provided. Given that to the best of our knowledge the literature lacks such a rich and well-tailored dataset, it is anticipated that the results of the work at hand will offer a solid basis for intrusion detection in the current as**

of availability attacks but more importantly to attacks that threat the secrecy of its key, jeopardising the confidentiality of the entire communication. Posterior efforts such as WiFi Protected Access (WPA) and WPA2 proved to be more robust as far as confidentiality is concerned. However, with the increasing computational power and the instalment of low-cost cluster computing this will be soon inaccurate. Naturally, these mechanisms are anticipated to render themselves vulnerable even to brute force attacks [3]. On the other hand, cloud-based systems like CloudCracker [4] can test 300 million possible WPA passwords in just 20 minutes.

In any case, WPA/WPA2 share almost the same vulnerabilities as the early WEP versions as far as availability is concerned. Even the newest amendment, 802.11w [5], which concentrates in patching availability related shortcomings (leading

22

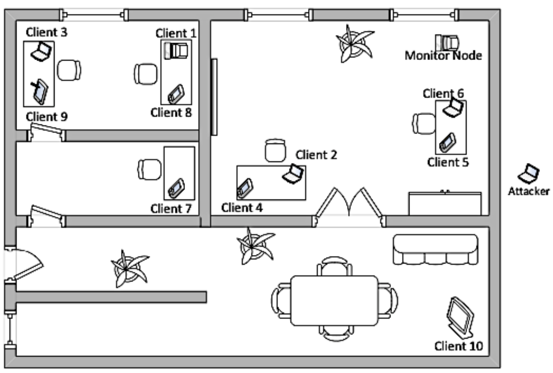


Fig. 11: Lab Blueprints

TABLE V: Specifications of the Equipment Used in the Experiments

Node	Type	Brand	OS	Network Card	CPU
Client1	Desktop	Custom	Ubuntu Linux 12.04 LTS	Netgear WNA3100 N300	Intel Core i7 3.2GHz
Client2	Laptop	Fujitsu-Siemens	Ubuntu Linux 12.04 LTS	Intel 3945ABG	Intel Core Duo T2050 1.6GHz
Client3	Laptop	Acer	Ubuntu Linux 12.04 LTS	Qualcomm Atheros AR9462	Intel Core i5 1.7GHz
Client4	Smartphone	iPhone 3G	iOS 4.2	NA	Samsung 32-bit RISC ARM 620MHz
Client5	Other	iPod Touch	iOS 3.1	NA	Samsung 32-bit RISC ARM 533MHz
Client6	Laptop	Acer Aspire 5750G	Windows 7	Broadcom BCM943227HM4L	Intel Core i5 2.8GHz
Client7	Smartphone	HTC Diamond	Windows Phone 6.1	NA	528 MHz ARM 11
Client8	Smartphone	Samsung Nexus	Android 4.2	NA	dual-core ARM Cortex-A9 1.2 GHz
Client9	Tablet	Samsung Galaxy Tab	Android 2.2	NA	Cortex-A8 1 GHz
Client10	Smart TV	LG 42LM7600S	Linux	NA	NA
Attacker	Laptop	Acer Aspire 5750G	Kali Linux 1.0.6	D-Link DWA-125/Linksys WUSB54GC	Intel Core i5 2.8GHz
Monitor Node	Desktop	Custom	Linux Debian 7.3	Alpha AWUS036H	Core i7 2.4Ghz

23

TABLE VIII: Confusion Matrices of Various Classification Algorithms on the 156 Feature Set. Best performer in red.

Normal	Flooding	Injection	Impersonation	Classified As
530785	0	0	0	Normal
8097	0	0	0	Flooding
16682	0	0	0	Injection
20079	0	0	0	Impersonation

(a) Adaboost

Normal	Flooding	Injection	Impersonation	Classified As
530785	0	0	0	Normal
8097	0	0	0	Flooding
16682	0	0	0	Injection
20079	0	0	0	Impersonation

(b) Hyperpipes

Normal	Flooding	Injection	Impersonation	Classified As
530771	8	0	6	Normal
2641	4857	0	599	Flooding
2	0	16680	0	Injection
18629	0	0	1450	Impersonation

(c) J48

Normal	Flooding	Injection	Impersonation	Classified As
508621	22164	0	0	Normal
2189	5908	0	0	Flooding
16400	0	282	0	Injection
18750	1329	0	0	Impersonation

(d) Naive Bayes

Normal	Flooding	Injection	Impersonation	Classified As
530775	0	7	3	Normal
8097	0	0	0	Flooding
3038	0	13644	0	Injection
20079	0	0	0	Impersonation

(e) OneR

Normal	Flooding	Injection	Impersonation	Classified As
530729	1	54	1	Normal
4077	4020	0	0	Flooding
2470	0	14212	0	Injection
18760	0	28	1291	Impersonation

(f) Random Forest

Normal	Flooding	Injection	Impersonation	Classified As
518657	906	716	10506	Normal
3854	4243	0	0	Flooding
338	0	1930	14414	Injection
17550	0	1003	1526	Impersonation

(g) Random Tree

Normal	Flooding	Injection	Impersonation	Classified As
530785	0	0	0	Normal
8097	0	0	0	Flooding
16682	0	0	0	Injection
20079	0	0	0	Impersonation

(h) ZeroR

Hirte, Honeypot and EvilTwin impersonation attacks have previously been identified as the most severe threats to a wireless network.

24

TABLE X: Confusion Matrices of Various Classification Algorithms on the 20 Feature Set. Best performer in red.

Normal	Flooding	Injection	Impersonation	Classified As
530785	0	0	0	Normal
8097	0	0	0	Flooding
16682	0	0	0	Injection
20079	0	0	0	Impersonation

(a) Adaboost

Normal	Flooding	Injection	Impersonation	Classified As
530785	0	0	0	Normal
8097	0	0	0	Flooding
16515	0	167	0	Injection
20079	0	0	0	Impersonation

(b) Hyperpipes

Normal	Flooding	Injection	Impersonation	Classified As
530588	116	6	75	Normal
2553	5544	0	0	Flooding
2	0	16680	0	Injection
18644	148	0	1287	Impersonation

(c) J48

Normal	Flooding	Injection	Impersonation	Classified As
497199	8971	11899	12716	Normal
2123	5974	0	0	Flooding
3027	0	13655	0	Injection
14187	1473	0	4419	Impersonation

(d) Naive Bayes

Normal	Flooding	Injection	Impersonation	Classified As
530765	0	14	6	Normal
8097	0	0	0	Flooding
3038	0	13644	0	Injection
20079	0	0	0	Impersonation

(e) OneR

Normal	Flooding	Injection	Impersonation	Classified As
530746	1	1	37	Normal
2600	5497	0	0	Flooding
2763	0	13893	0	Injection
18607	0	28	1472	Impersonation

(f) Random Forest

Normal	Flooding	Injection	Impersonation	Classified As
530700	3	0	82	Normal
2442	5494	161	0	Flooding
273	0	16253	156	Injection
18609	0	0	1470	Impersonation

(g) Random Tree

Normal	Flooding	Injection	Impersonation	Classified As
530785	0	0	0	Normal
8097	0	0	0	Flooding
16682	0	0	0	Injection
20079	0	0	0	Impersonation

(h) ZeroR

25

The 14th ACM International Conference on Availability, Reliability and Security (ARES), 26-29 Aug. 2019, U.K.

DEMISe: Interpretable Deep Extraction and Mutual Information Selection Techniques for IoT Intrusion Detection

Luke R Parker
Defence Equipment and Support
Ministry of Defence
Bristol, UK
luke.parker890@mod.gov.uk

Paul D Yoo*
CSIS, Birkbeck College
University of London
London, UK
paul.d.yoo@ieee.org

Taufiq A Asyhari
School of Computing, Electronics and Mathematics
Coventry University
Coventry, UK
taufiq-a@ieee.org

Lounis Chermak
Centre for Electronic Warfare, Information and Cyber
Cranfield University
Shrivenham, UK
l.chermak@cranfield.ac.uk

Yoonchan Jhi
Security Research Team
Samsung SDS
Seoul, South Korea
yoonchan.jhi@samsung.com

Kamal Taha
ECE Dept
Khalifa University
Abu Dhabi, UAE
kamal.taha@kustar.ac.ae

ABSTRACT

Recent studies have proposed that traditional security technology – involving pattern-matching algorithms that check predefined pattern sets of intrusion signatures – should be replaced with sophisticated adaptive approaches that combine machine learning and behavioural analytics. However, machine learning is performance driven, and the high computational cost is incompatible with the limited computing power, memory capacity and energy resources of portable IoT-enabled devices. The convoluted nature of deep-structured machine learning means that such models also lack transparency and interpretability. The knowledge obtained by interpretable learners is critical in security

KEYWORDS

Security mobility applications, security of resource constrained devices, IoT, lightweight intrusion detection, feature engineering, mutual information, white-box modelling, deep learning.

1 Introduction

The Internet of Things (IoT) is an expanding network of devices that are predicted to become more mainstream as a result of their proliferation in the healthcare, retail, manufacturing and transportation markets [1,2]. The IoT comprises everyday devices with a degree of networked capability such that they provide an

26

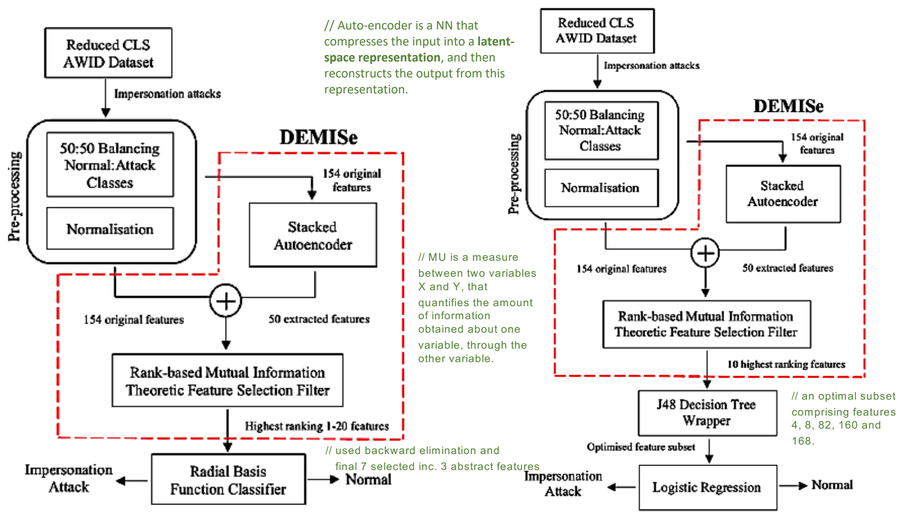


Figure. 1: High-level architecture of the DEMISe-RBFC model showing the core DEMISe element for feature extraction and selection, the Radial Basis Function Classifier, and the pre-processing stages.

Figure. 2: High-level architecture of the DETEReD model showing the core DEMISe element for feature extraction and selection, the C4.8 tree wrapper, the logistic regression, and the pre-processing stages.

27

Table VI: DETERED and DEMISE-RBFC versus Models previously tested against the CLS portion of the AWID dataset [11,12]

Classifier	Acc (%)	DR (%)	FAR (%)	F_1 (%)	Mcc (%)
DEMISe-RBFC	98.00	99.04	3.00	97.98	96.02
DETEReD	**98.04**	**99.07**	2.96	**98.01**	**96.09**
Kolias *et al* [12]	94.91	97.23	74.21	97.37	22.12
Aminanto *et al* [11]	97.60	85.00	**2.36**	NRA	NRA

NRA = No results available.

Table VII: Estimated resource requirements for DETEReD and DEMISe-RBFC

Model	Number of Parameters	Estimated Memory Requirement
DEMISe-RBFC	21 (4 output weights (from 2 layers), 14 unit centres (from 2 layers), 2 bias weights (one for each class) and a scale weight)	84 bytes
DETEReD	5 weights (+1 intercept)	24 bytes

28

IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 13, NO. 3, MARCH 2018 621

Deep Abstraction and Weighted Feature Selection for Wi-Fi Impersonation Detection

// the weight values between the first two layers – the weight represents the contribution from the input features to the first hidden layer.

Muhamad Erza Aminanto, Rakyong Choi, Harry Chandra Tanuwidjaja, Paul D. Yoo, *Senior Member, IEEE*, and Kwangjo Kim, *Member, IEEE*

***Abstract*—The recent advances in mobile technologies have resulted in Internet of Things (IoT)-enabled devices becoming more pervasive and integrated into our daily lives. The security challenges that need to be overcome mainly stem from the open nature of a wireless medium, such as a Wi-Fi network. An impersonation attack is an attack in which an adversary is disguised as a legitimate party in a system or communications protocol. The connected devices are pervasive, generating high-dimensional data on a large scale, which complicates simultaneous detections. Feature learning, however, can circumvent the potential problems that could be caused by the large-volume nature of network data. This paper thus proposes a novel deep-feature extraction and selection (D-FES), which combines stacked feature extraction and weighted feature selection. The stacked autoencoding is capable of providing representations that are more meaningful by reconstructing the relevant information from its raw inputs. We then combine this with modified weighted feature selection inspired by an existing shallow-structured machine learner. We finally demonstrate the ability of the condensed set of features to reduce the bias of a machine learner model as well as the computational complexity. Our experimental results on a well-referenced Wi-Fi network benchmark data set, namely, the Aegean Wi-Fi Intrusion data set, prove the usefulness and the utility of the proposed D-FES by achieving a detection accuracy of 99.918% and a false alarm rate of 0.012%, which is the most accurate detection of impersonation attacks reported in the literature.**

***Index Terms*—Intrusion detection system, impersonation attack, deep learning, feature extraction, stacked autoencoder, large-scale Wi-Fi networks.**

I. Introduction

THE rapid growth of the Internet has led to a significant increase in wireless network traffic in recent years. According to a worldwide telecommunication consortium [1], proliferation of 5G and Wi-Fi networks is expected to occur in the next decades. By 2020[1] wireless network traffic is anticipated to account for two thirds of total Internet traffic — with 66% of IP traffic expected to be generated by Wi-Fi and cellular devices only. Although wireless networks such as IEEE 802.11 have been widely deployed to provide users with mobility and flexibility in the form of high-speed local area connectivity, other issues such as privacy and security have raised. The rapid spread of Internet of Things (IoT)-enabled devices has resulted in wireless networks becoming to both passive and active attacks, the number of which has grown dramatically [2]. Examples of these attacks are impersonation, flooding, and injection attacks.

An Intrusion Detection System (IDS) is one of the most common components of every network security infrastructure [3] including wireless networks [4]. Machine-learning techniques have been well adopted as the main detection algorithm in IDS owing to their model-free properties and learnability [5]. Leveraging the recent development of machine-learning techniques such as deep learning [6] can be expected to bring significant benefits in terms of improving

29

Module Eval. – Confusion Matrix

```
=== Confusion Matrix ===
 a b   <-- classified as
 4 5 |  a = 1  Positive (Attack)
 3 2 |  b = 0  Negative (Normal)
```

TP (detection rate, sensitivity)

FN (type II error, beta)

FP (type I error, false alarm, alpha)

TN

Kolias *et al* 2015

Normal	Flooding	Injection	Impersonation	Classified As
530771	8	0	6	Normal
2641	4857	0	599	Flooding
2	0	16680	0	Injection
18629	0	0	1450	Impersonation

(c) J48

- True positive (TP): correct positive prediction
- False positive (FP): incorrect positive prediction
- True negative (TN): correct negative prediction
- False negative (FN): incorrect negative prediction

Detection Rate Vs False Alarm

- Detection Rate is a measure of positive cases that test positive
  - the ratio of TPs to the sum of TPs and FNs == TP/(TP+FN)
    (1 – Type II Error) // sensitivity = TP rate (# of TPs divided by the total # of positives)
- False Positive/Alarm (Type I Error) is a measure of negative cases that test positive
  - the ratio of FPs to the sum of FPs and TNs
    (1 – Specificity) == FP/(FP+TN)
    // specificity = TN rate (# of TNs divided by the total # of negatives)
- If a test has a high sensitivity, IDS accurately recognises attack traffics as being attack (TPs).
- If a test has a high specificity, IDS accurately recognises normal traffics as being normal (TNs).
- What if a test has a high Type II Error (FN/(FN+TP))?

Matthews Correlation Coefficient

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

- The MCC is used in ML as a measure of the quality of binary (two-class) classifications.
- It is generally regarded as a balanced measure which can be used even if the classes are of very different sizes.
- The MCC is in essence a correlation coefficient between the observed and predicted binary classifications;
- It returns a value between −1 and +1.
  - A coefficient of +1 represents a perfect prediction,
  - 0 means no better than random prediction and
  - −1 indicates total disagreement between prediction and observation.

Task to be completed

- Your task is to build a predictive model (i.e. a machine learning classifier) capable of distinguishing between “bad” traffic, called intrusions or attacks, and “good” normal traffic.

Tasks to be completed

This is a group task with individual element, and you will work in a group of 5 students.

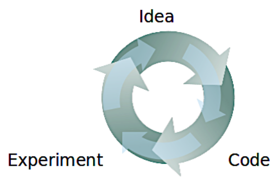
- Team forming
- Planning
- Searching literature
- Pre-processing
- Selecting features
- Exploring and selecting ML algorithms
- Refining ML algorithms
- Evaluating model and analysing the results
- Future work

Team forming

This is a group task with individual element, and you will work in a group of 5 students.

- Find your partners as soon as possible, and when a group is formed email the module leader (paul@dcs.bbk.ac.uk).
- The module leader will update info on VLE so we know who has partners and who does not.
- Teaching assistants also has support for soliciting partners.
- If you are having trouble finding partners, ask the teaching staff, and we will try to find you a group in a fair way.

Planning



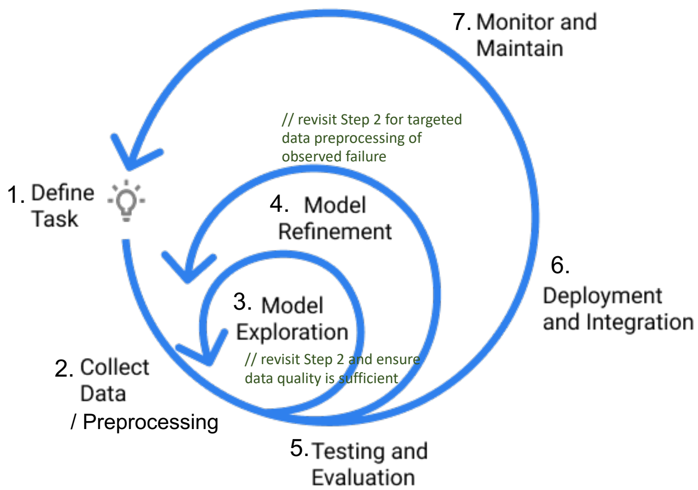
AML is an iterative process.

- ML projects are highly iterative; as you progress through the ML lifecycle, you'll find yourself iterating on a section until reaching a satisfactory level of performance, then proceeding forward to the next task (which may be circling back to an even earlier step).
- You need to plan carefully.
- You need to determine scope, resources, major tasks (and who is responsible for what) and schedule (e.g. Gantt chart).
- You may also need to discuss general model tradeoffs (accuracy vs speed).
- Read "How to plan and execute your ML and DL projects"

URL: https://blog.floydhub.com/structuring-and-planning-your-machine-learning-project/

37

Machine Learning Development Lifecycle



39

Searching Literature

You are suggested to undertake a literature search.

- This is a search designed to identify existing research and information on IDS using AWID dataset.
- Their findings will be very helpful for your task (particularly in feature and algorithm selection tasks).
- There should be no separate literature review section in your report.

Dataset

- The dataset for this project is available in VLE.
  - Use *train_imperson_without4n7_balanced_data.csv* for training
  - Use *test_imperson_without4n7_balanced_data.csv* for testing.
- The first row of each dataset gives variable numbers (this may need to be removed).
- Each dataset has 152 input variables and 1 target variable.
- Please be noted that two features numbered 4 and 7 (*frame.time_epoch* and *frame.time_relative*) have been removed from both datasets as they provide temporal information which may cause unfair prediction.
- The training set has 97,044 observations while testing set has 40,158 observations.

Preprocessing

Use suitable descriptive statistics and visualisation

- Existing studies focused on algorithms and features only.
- You need to consider various data pre-processing techniques such as data transformation, discretisation, cleaning, normalisation, standardisation, smoothing, feature construction, etc and use them if necessary.

42

Selecting features

Go beyond lectures and lab manuals!

- Consider various techniques within each of filter, wrapper and embedded methods.
- Consider some dimensionality techniques linear and non-linear
  - PCA, Factor analysis and LDA
  - MDS, Isomap, LLE, HLLE, Spectral Embedding, t-SNE
  - Autoencoder, GAN
- Findings from literature review may also be helpful.
- Two features (4. *frame.time_epoch* and 7. *frame.time_relative*) have been removed from the dataset - temporal information

43

Exploring and selecting ML algorithms

- Select candidate algorithms.
- Establish baselines for model performance and start with a simple model using initial data pipeline.
- Discuss your selection strategies.

Refining algorithms

Finding the best configuration for ML hyperparameters in such a high dimensional space is not a trivial challenge.

- Consider
  - model design components (e.g. # of layers, # of units per layer, loss function, activations, optimisers, dropout layer etc)
  - hyperparameters (e.g. learning rate, dropout rate, batch size etc).
- Perform model-specific optimisations and Iteratively debug model as complexity is added.
- Reproducibility – consistency in results
  - ML algorithms are stochastic in nature.
  - Accuracy of 90% today but you may get $\pm$ 1% change in accuracy with the same architecture.

Evaluating model and analysing the results

- Evaluate the classification performance
  - e.g. accuracy, detection rate, false alarm, type II error, MCC and TBM and TTM – go beyond these measures if necessary
  - Interpret the results
  - Compare the chosen model’s performance with the benchmarks

Future work

This is a place for you to explain where you think the results can lead you.

- What are the strengths and weaknesses of your work?
- What do you think are the next steps to take?
- What other questions do your results raise?
- Do you think certain paths seem to be more promising than others?
- This lets people know what you’re thinking of doing next and they may ask to collaborate if your future research area crosses over theirs.

Deliverables Required and Submission Information

- A report of 4,000 words (±10%) which includes
  - 2 groups components (i. Planning and ii. Future work) and
  - 5 individual components (i. Pre-processing, ii. Selecting features, iii. Exploring and selecting ML algorithms iv. Refining algorithms and v. Evaluating model and analysing the results).
- A group project – you need to work together
  - Each person needs to be responsible for one individual task and drafting the section.
- The cover page – show who is responsible for each individual component and wordcount.
- Substantial tables and figures (make a good use of appendix) help to cram all your information into the word count.
- Arial 10 point or Times New Roman 11 point font
- 1.5 line spacing.
- A minimum of 2.54 (1 inch) margins
- IEEE referencing must be used, for guidance see: https://ieeeauthorcenter.ieee.org/wp-content/uploads/IEEE-Reference-Guide.pdf
- Your code must also be submitted along the report. It should be either *.ipynb* or *.py*.

Estimated Time to Complete

- There will be time that is allocated for working on your group project in Weeks 7 and 8.
- However, it is your responsibility to allocate an appropriate amount of time to this piece of work and to form a group.

▶ OPEN ALL ▼ CLOSE ALL

Instructions: Clicking on the section name will show / hide the section.

1 ▶ **PRE-MODULE ACTIVITIES** Topic 1

2 ▶ **LECTURES/LABS** Topic 2

3 ▶ **FURTHER READING** Topic 3

4 ▼ **MODULE ASSESSMENT DETAILS AND SUBMISSION** Topic 4

Here is where you will find your assessment guidelines, brief and submission points. Remember that you can submit your work as many times as you like prior to the due date. Each time you resubmit your work it will take 24hrs to generate a report and the previous submission is erased from the database.

- Group Practical Brief
- Related Papers
- Datasets
- Group Practical Submission Point
- Project Discussions

You can use this discussion forum to find your partners and when a group is formed email the module leader (paul@dcs.bbk.ac.uk) with the details of your group members. The module leader will update info on VLE so we know who has partners and who does not. Teaching assistants also has support for soliciting partners. If you are having trouble finding partners, ask the teaching staff, and we will try to find you a group in a fair way.

50

Questions?

paul@dcs.bbk.ac.uk

51