Deep Learning at the Shallow End: Malware Classification for Non-Domain Experts

Dr. Quan Le
Dr. Oisín Boydell
Dr. Brian Mac Namee
Dr. Mark Scanlon

DFRWS USA 2018

Malware analysis

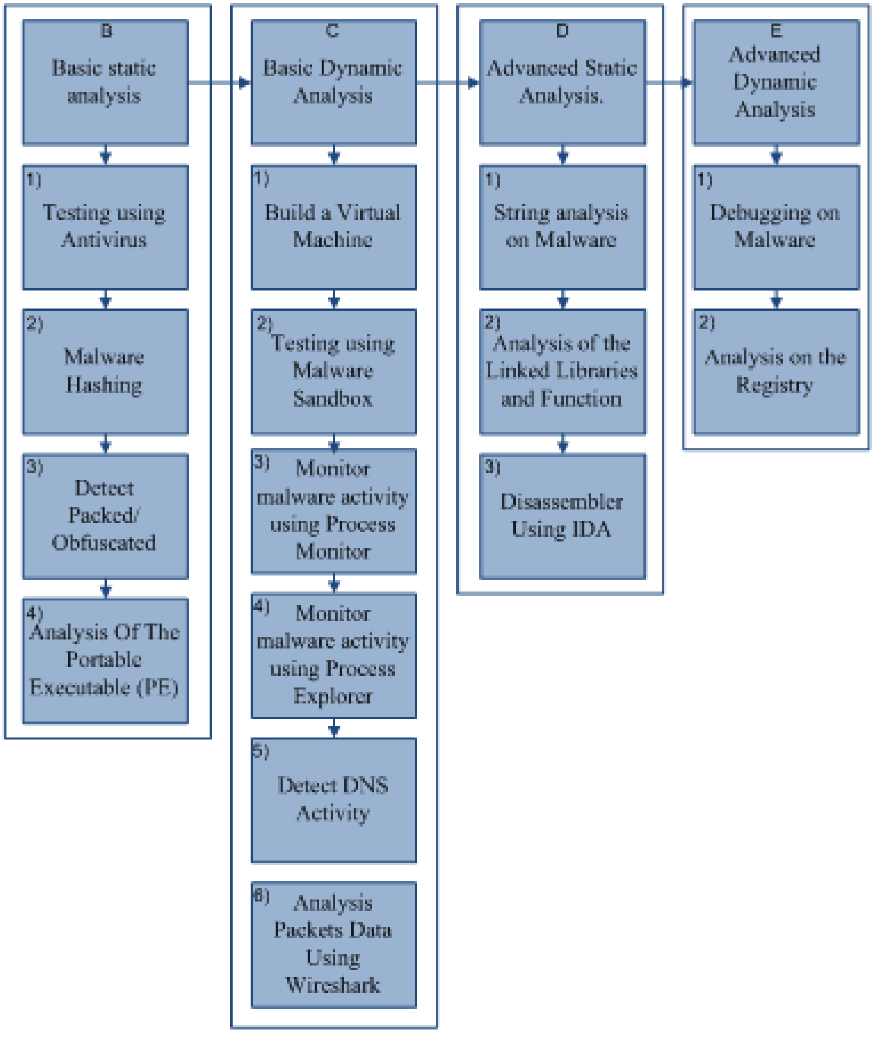
Malware analysis/detection/classification challenges…

- Huge volume and variation
- Dynamic - malware constantly changing
- Requires deep domain expertise
- Time consuming
- Hard to scale

Malware analysis

Traditional approaches require

- Specialist tools
- Computational resources – virtual machines, sandbox environments, isolated networks
- Time – malware often needs to be executed in real-time for analysis
- Expertise



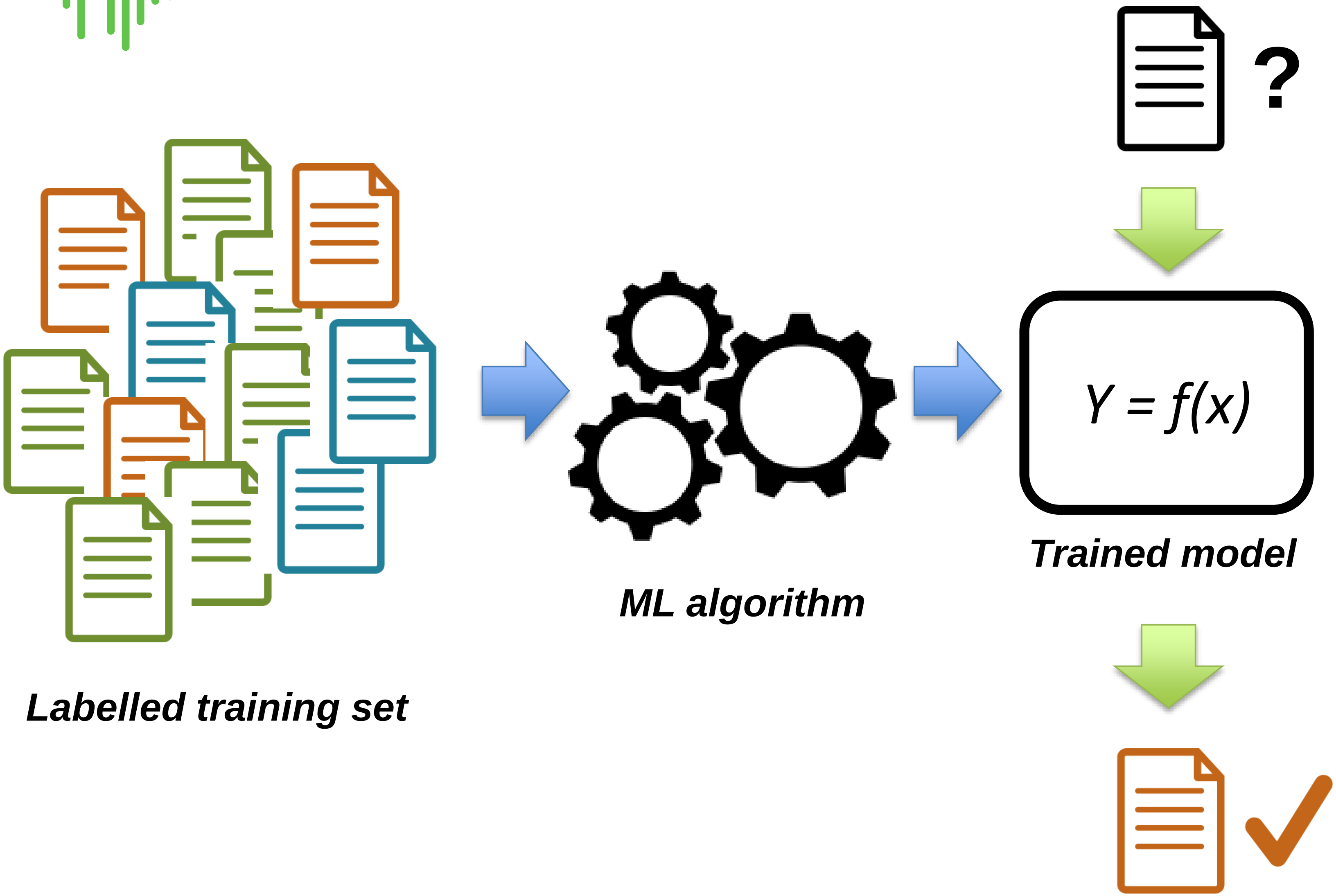
Source: "Implementation of Malware Analysis using Static and Dynamic Analysis Method", Yusirwan et al., International Journal of Computer Applications, volume 117

Malware analysis – machine learning

- Currently, there is a lot of interest and ongoing research around using Machine Learning (ML) for malware analysis
  - *Ucci, D., Aniello, L., Baldoni, R., 2017.* ***Survey on the Usage of Machine Learning Techniques for Malware Analysis****. CoRR abs/1710.08189.* [*http://arxiv.org/abs/1710.08189*](http://arxiv.org/abs/1710.08189)
  - *Gandotra, E., Bansal, D., Sofat, S., 2014.* ***Malware Analysis and Classification: A Survey****, Journal of Information Security, 2014, 5, 56-64.* [*http://file.scirp.org/Html/4-7800194_44440.htm*](http://file.scirp.org/Html/4-7800194_44440.htm)

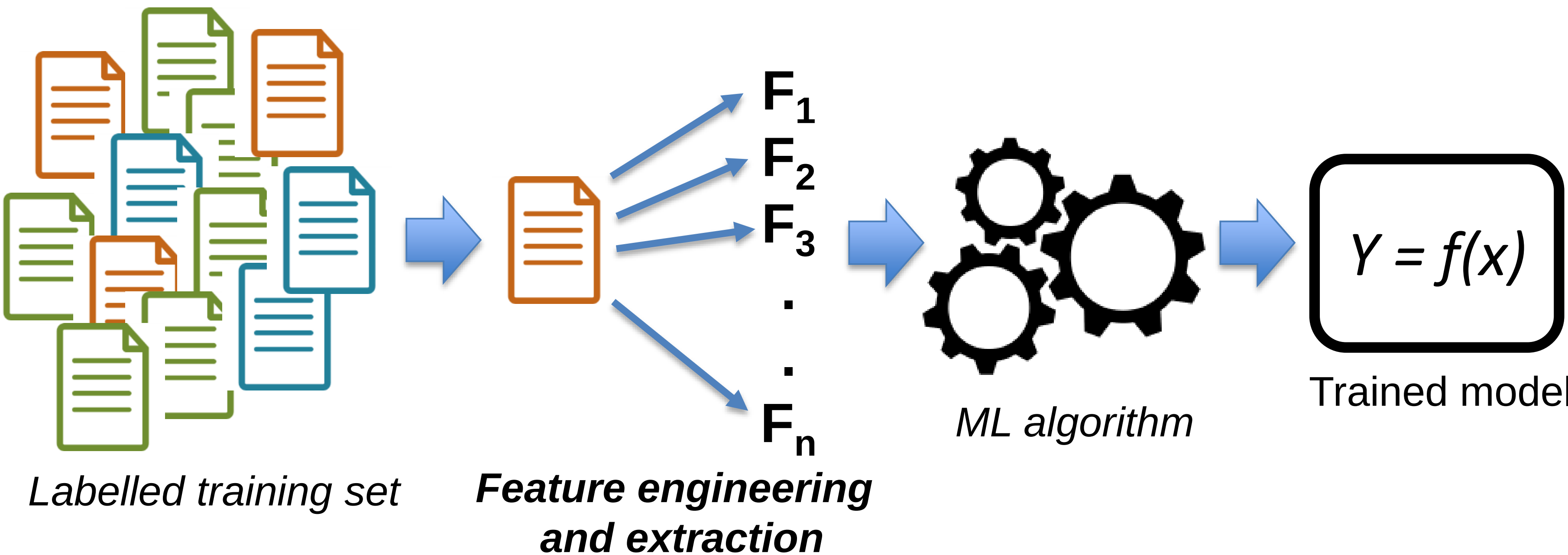
- ML has been used to automate and improve many malware analysis tasks, particularly malware classification

Machine Learning



Malware analysis – machine learning

- However, the majority of 'traditional' ML algorithms require input data in terms of higher level features derived from the data

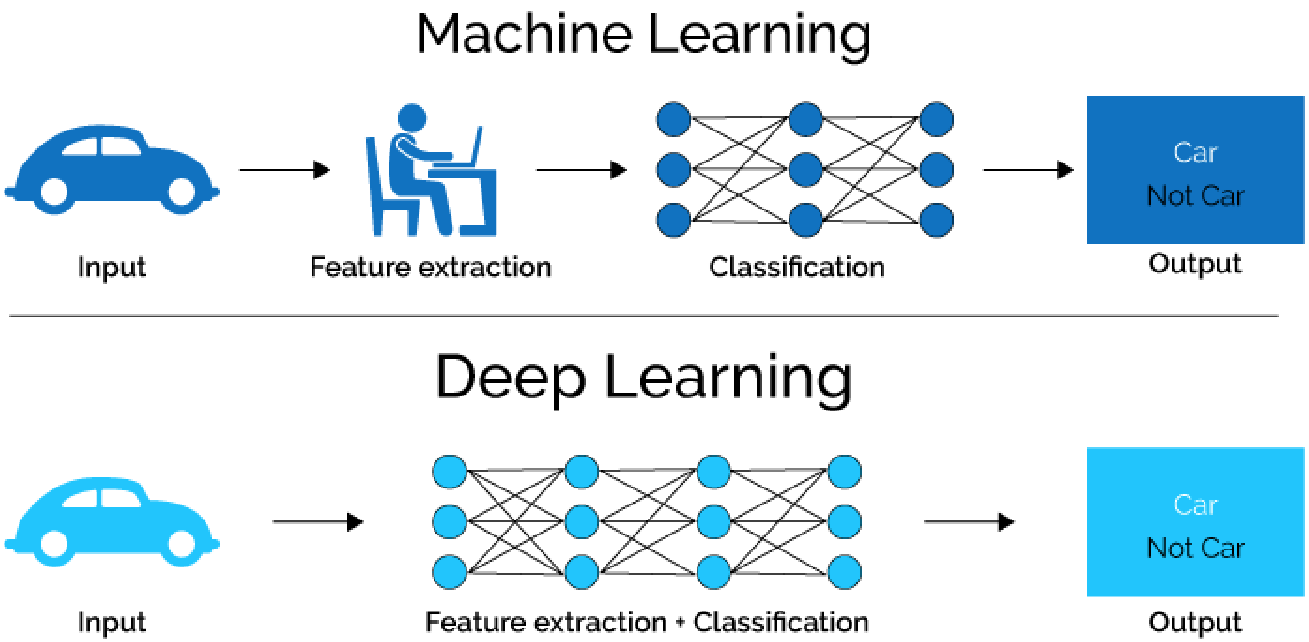


Malware features for ML classification

- The generation of these features is still a very manual process that relies on both domain expertise, as well as ML expertise
- Static features
  - *Processor instructions*
  - *Null terminated strings and other static resources contained in the code*
  - *Static system library imports*
  - *System API calls*
  - *Etc.*
- Dynamic features
  - *Dynamic system API calls*
  - *Interactions with other system resources such as memory and storage*
  - *Network communications*
  - *Etc.*

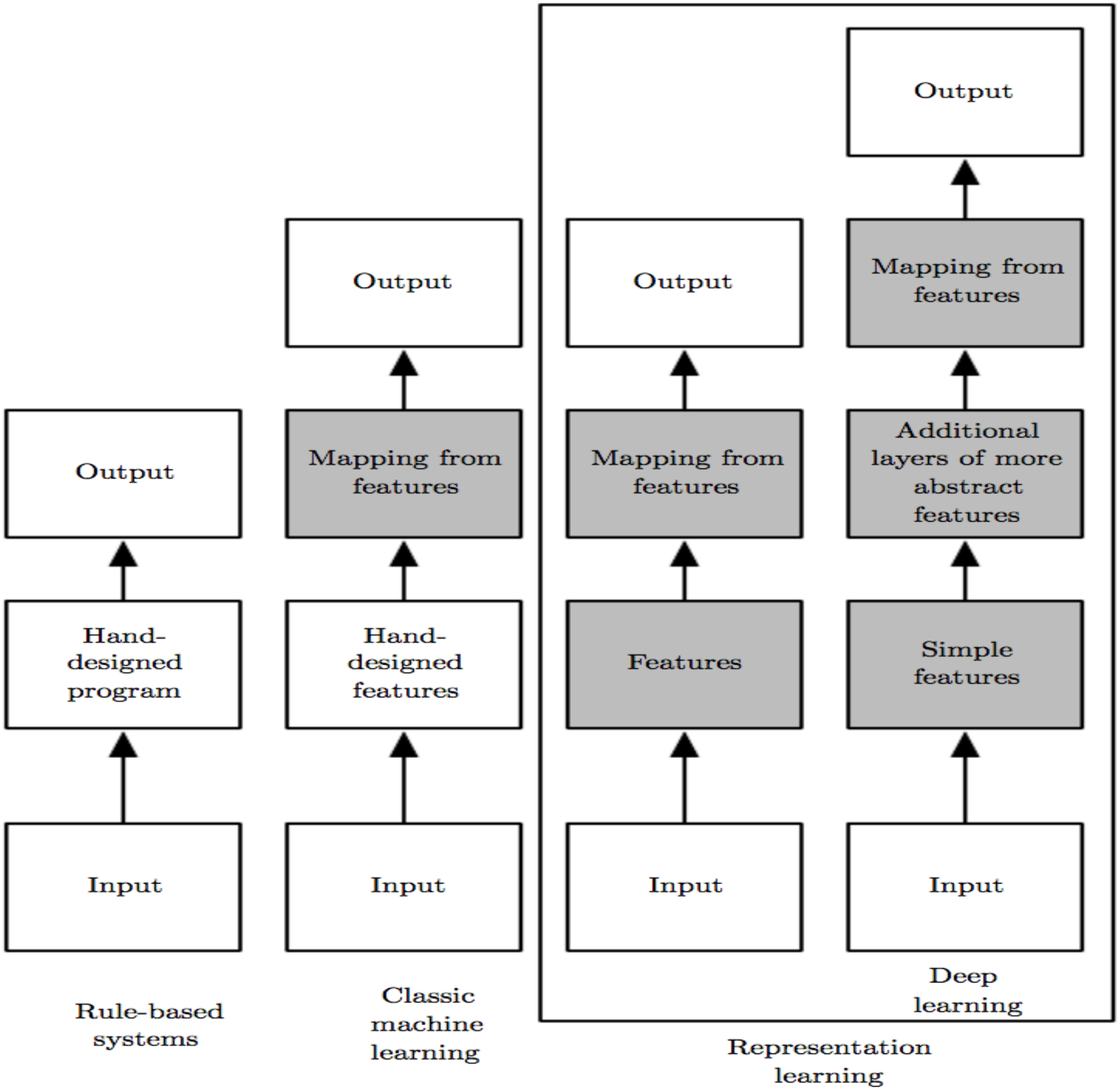
Deep learning

- Deep Learning is a type of ML based on Artificial Neural Networks (ANNs)
- A key feature is it's ability to operate on low level, raw data representations



Source: https://www.xenonstack.com/blog/data-science/log-analytics-deep-machine-learning-ai/

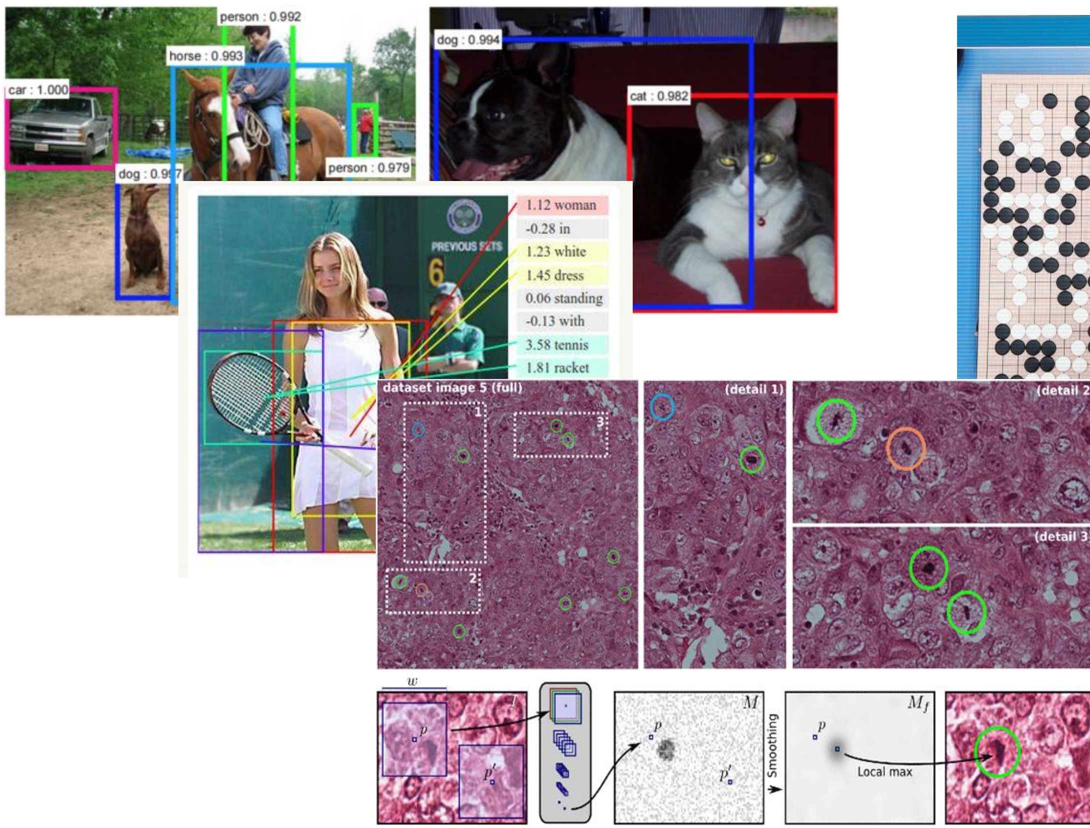
Deep learning



Source: 'Deep Learning' by Goodfellow, Bengio and Courville, MIT Press 2016

Deep learning

- Deep Learning has rapidly achieved state of the art performance across a broad range of application areas

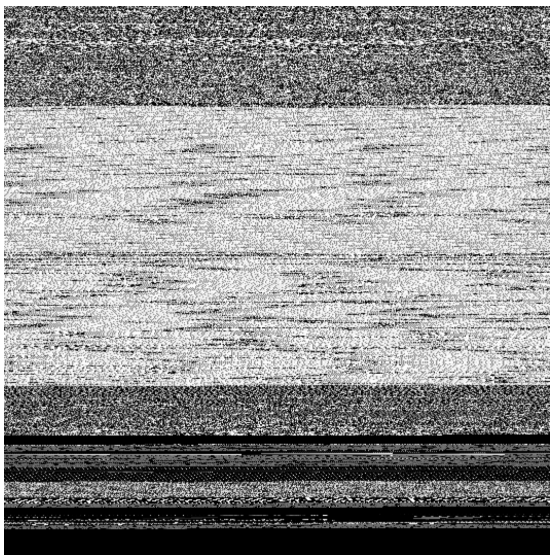


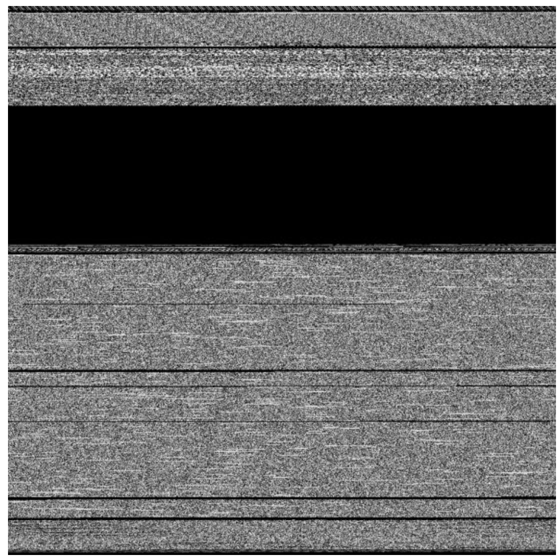
AlphaGo

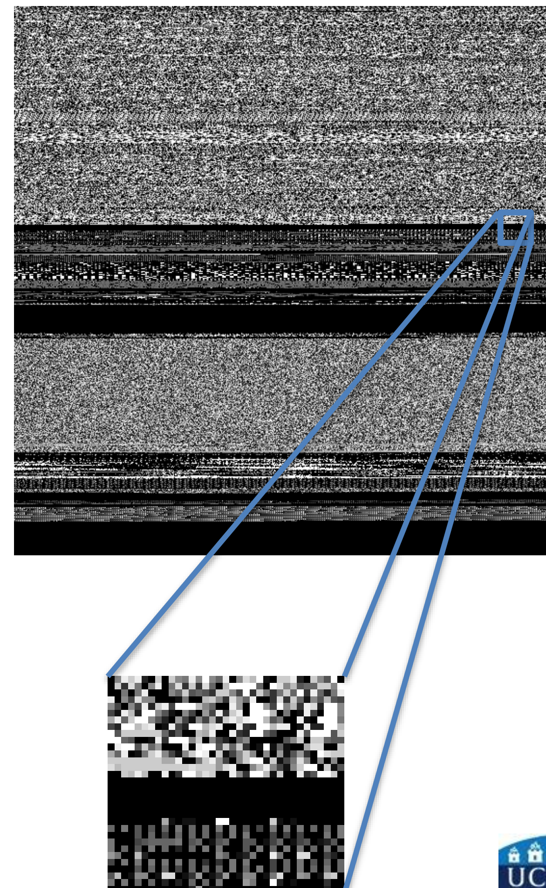
Google Translate

Our approach

- Malware classification using Deep Learning based on static, raw malware executable data







- Data driven approach
  - Allow the model to learn the features from the raw data (byte sequence) itself

Our approach

- Motivation - Why do this?
  - No deep malware domain expertise required
    - No sandbox environments
    - No code disassembly
    - No need to manually identify and extract features (static or dynamic)
  - Easily adaptable to new malware classes/types
    - Just requires labelled examples for training the model
  - Classification speed
    - No need to actually run or disassemble the code
    - Classification based on the static, raw byte code
- But how can an approach based purely on the static byte code which ignores human malware domain knowledge be any good?

Deep Learning model architectures

Three different model architectures evaluated

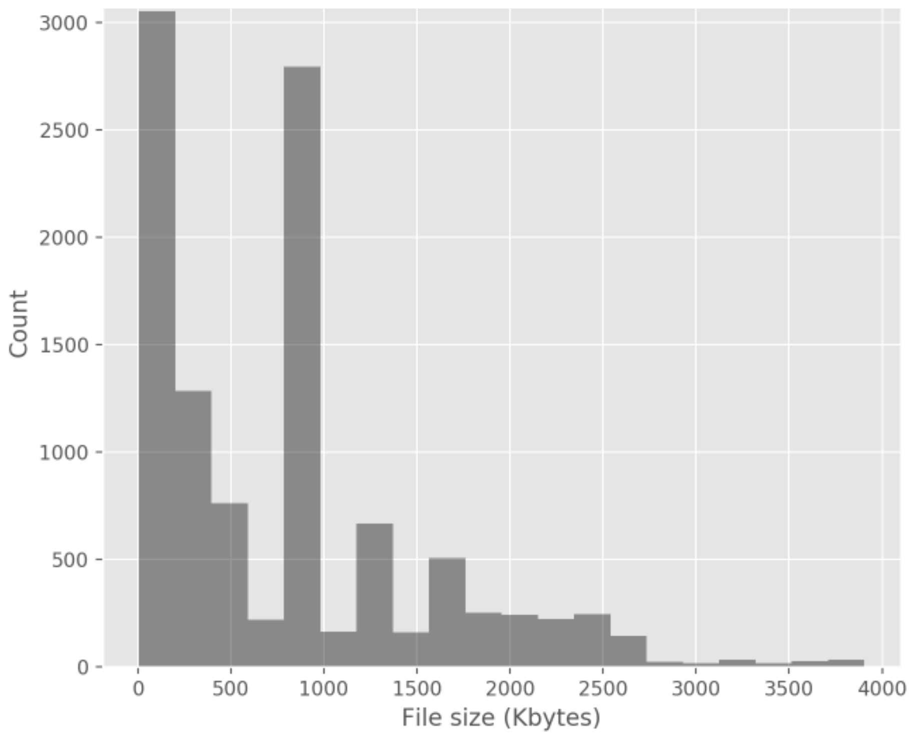
1. Convolutional Neural Network (CNN)
2. CNN + Unidirectional Long Short Term Memory (CNN UniLSTM)
3. CNN + Bi-directional Long Short Term Memory (CNN BiLSTM)

Dataset

- Kaggle Microsoft Malware Classification Challenge (BIG 2015)
  - https://www.kaggle.com/c/malware-classification
- Over 400 GB uncompressed.
- 9 labelled malware classes.
- 10,868 malware files as raw byte code with labels in the training set.
- 10,873 files in the test set without labels.
- Original challenge closed April 2015

Data Pre-processing

- Although our approach is designed to work on the raw, static malware byte code, some pre-processing is required (but this is easily automated).



- OpenCV to compress each sample to a length of 10,000 bytes

Class Imbalance

Malware Class	Number of Examples
Ramnit	1,541
Lollipop	2,478
Kelihos_ver3	2,942
Vundo	475
Simda	42
Tracur	751
Kelihos_ver1	398
Obfuscator.ACY	1,228
Gatak	1,013

- Two approaches
  - Preserve class imbalance in the training set
  - Re-sampling to balance class representation in the training set

5-fold cross validation results

Deep Learning Conf	Acc (%)	F1 (%)
CNN - Def Sampl	95.1	92.14
CNN - Reb Sampl	95.8	92.14
CNN UniLSTM - Def Sampl	97.64	94.15
CNN UniLSTM - Reb Sampl	98.12	95.92
CNN BiLSTM - Def Sampl	97.91	95.52
CNN BiLSTM - Reb Sampl	98.20	96.05

Results in context

Ahmadi et al. **Novel feature extraction, selection and fusion for effective malware family classification**. *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy. CODASPY '16*

- Feature engineering approach using features from disassembled binaries, combined with classic visual image analysis features from raw binaries, using XGBoost classifier
- **95.5%** accuracy using same 5-fold cross validation evaluation

Gibert Llauradó D., **Convolutional neural networks for malware classification**. *Master's thesis, Universitat Politècnica de Catalunya (2016)*

- Log-loss public score 0.1176, private score 0.1348
- Our results: public score 0.0655, private score 0.0774

Practical runtime considerations

- Training

Configurations	No Params	Train time (m)
CNN - Def Sampl	1,842,069	5.6
CNN - Reb Sampl	1,842,069	10.1
CNN UniLSTM - Def Sampl	155,669	32.1
CNN UniLSTM - Reb Sampl	155,669	55.1
CNN BiLSTM - Def Sampl	268,949	62.1
CNN BiLSTM - Reb Sampl	268,949	106.2

- Classifying a binary file: 20 ms

Summary

- Our deep learning approach for malware classification…
  - Does not require deep domain knowledge of malware
  - Does not require time, tools and resources for complex feature extraction
  - Classifying new instances is fast so is practical in online, live, near real-time applications
  - Scalable to newly identified malware types
  - Achieves high accuracy

Conclusions and Future Work

- Evaluate on other datasets
  - Particularly the binary malicious/benign classification task
- Explore the capability to identify and report similarity between malware classes and variants (analysis)
- Apply to the task of determining the type of binary packing used
  - Irish National Cyber Security Centre
- Other applications?

Questions?

Dr Oisín Boydell,

Principal Data Scientist,

CeADAR (Centre for Applied Data Analytics Research) at University College Dublin

Email: oisin.boydell@ucd.ie

Code from this paper at:
https://bitbucket.org/ceadarireland/deeplearningattheshallowend