National College of Ireland

C-MRiC.ORG

Cloud Based Real-Time Network Intrusion Detection Using Deep Learning

Santhosh Parampottupadam & Arghir-Nicolae Moldovan

School of Computing, National College of Ireland, Mayor Street, IFSC, Dublin 1, Ireland
santhosh.parampottupadam@student.ncirl.ie; arghir.moldovan@student.ncirl.ie

The International Conference on Cyber Security and Protection of Digital Services (Cyber Security 2018)
Glasgow, Scotland, UK
12th June 2018

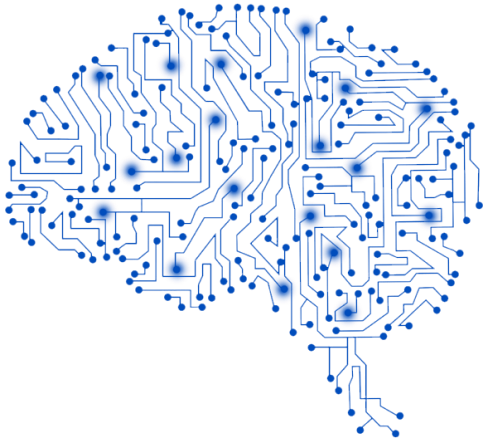
Outline

- Introduction
- Goals
- Methodology
- Prototype
- Results
- Conclusions
- Q&A

Introduction

- In 2017 the average cost of a data breach was $3.6 million, or $141 per data record (Ponemon Institute / IBM, 2018)
- NIDS systems
  - Signature or rule-based → look for specific patterns or signatures based on different factors such as IP addresses, ports, protocol, payload information, etc.
  - Anomaly-based → use machine learning to build an anomaly model based on different factors and then compare traffic to this model
- Deep learning
  - Neural network algorithms that transform data in multiple layers, where each layer uses output from the previous layer as input
  - Capable of automatic feature extraction, reducing the necessity to select features explicitly

Source: https://gbhackers.com/intrusion-detection-system-ids-2/



Source: http://earthsky.org/space/machine-deep-learning-2-astronomy-studies

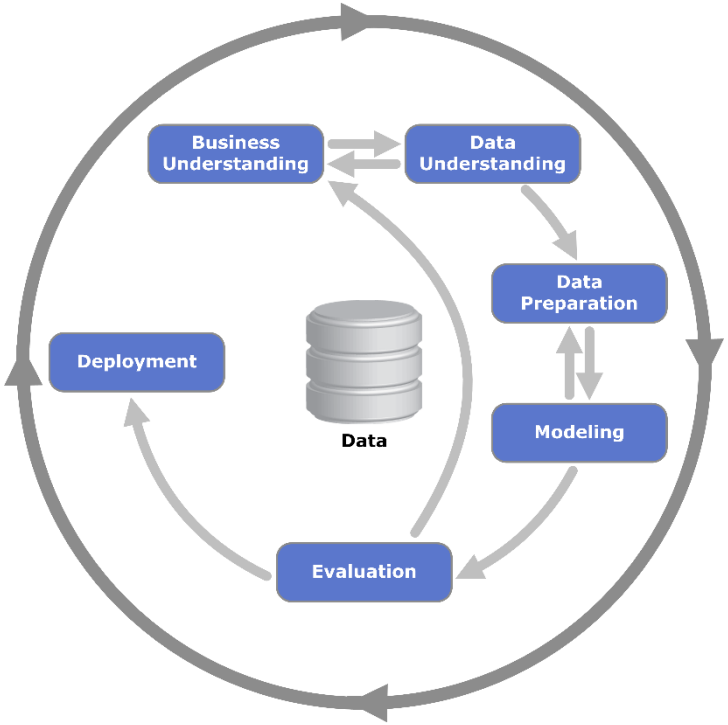
Research Goals

- Investigate the capabilities of Deep Learning for network intrusion detection
  - Compare DL models built using H2O and DeepLearning4J, with other commonly used ML models such as SVM, Random Forest, Logistic Regression and Naïve Bayes

- Propose a cloud-based prototype system for real-time network intrusion detection using Deep Learning

Methodology (1)

- Followed the cross-industry standard process for data mining (CRISP-DM)



- Problem understanding
  - Review of previous research studies showed that DL models do not always outperform other ML models
  - More research needed to investigate the capabilities of DL for intrusion detection

Methodology (2)

- Data understanding
  - NSL-KDD dataset
    - Selected records of KDDcup99 dataset, but without its shortcomings (i.e., removed duplicated records, more difficult to achieve high accuracy)
    - 41 attributes
      - 3 nominal (i.e., protocol, service, flag)
      - 38 numerical (e.g., duration, source and destination bytes, number failed logins, etc.)
    - Contains normal traffic and 39 attack types categorised into 4 classes
- Data preparation
  - NSL-KDD required little pre-processing
  - No record elimination or imputation was necessary
  - Removed one attribute as it was 0 for all train and test records
  - Added 2 columns for normal/intrusion and attack class

TABLE I
NSL-KDD INTRUSION ATTACK TYPE CLASSIFICATION.

Class	Attack Type
DoS	apache2*, back, land, mailbomb*, neptune, pod, processtable*, smurf, teardrop, udpstorm*, worm*
Probe	ipsweep, mscan*, nmap, portsweep, saint*, satan
R2L	ftp_write, guess_passwd, httptunnel*, imap, multihop, named*, phf, sendmail*, snmpgetattack*, snmpguess*, spy†, warezclient†, warezmaster, xlock*, xsnoop*
U2R	buffer_overflow, loadmodule, perl, ps*, rootkit, sqlattack*, xterm*

Note: † indicates attack types present only in training data
* indicates attack types present only in test data

TABLE II
DISTRIBUTION OF RECORDS FOR NORMAL TRAFFIC AND DIFFERENT ATTACK CLASSES.

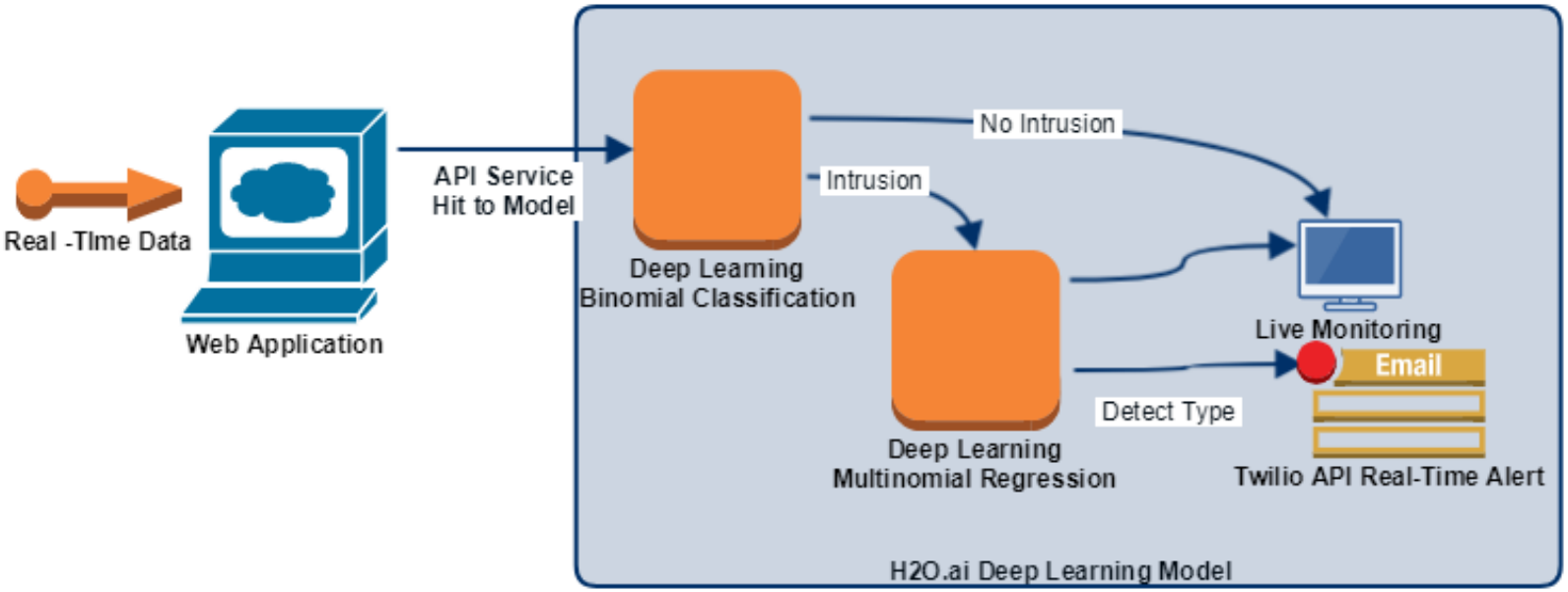
	Train Data		Test Data	
Class	Records[#]	Records [%]	Records[#]	Records [%]
Normal	67343	53.46	9710	43.07
Intrusion	58630	46.54	12833	56.93
DoS	45927	36.46	7460	33.09
Probe	11656	9.25	2421	10.74
R2L	995	0.79	2885	12.80
U2R	52	0.04	67	0.30
Total	125973	100.00	22543	100.00

Methodology (3)

- Modelling
  - Binomial classification models to detect intrusions from normal traffic
  - Multinomial models to detect the intrusion class (i.e., DoS, Probe, R2L and U2R)
  - Models were built with the default settings of the libraries, and without performing class balancing
  - Java-based ML Libraries used
    - DL: H2O Deep Learning, DeepLearning4J
    - ML: Support Vector Machines (LibSVM), Random Forest, Logistic Regression, Naïve Bayes
- Evaluation
  - 5-fold cross validation on the NSL-KDD train data
  - Train models on NSL-KDD train data and test models on the test data
  - Metrics: Accuracy, Precision, Recall, F-Measure, AUC, Detection Rate

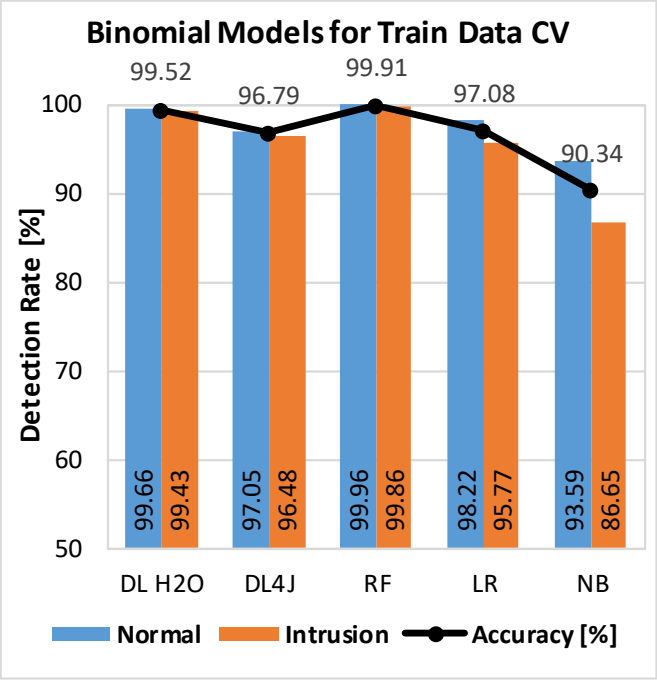
Prototype System

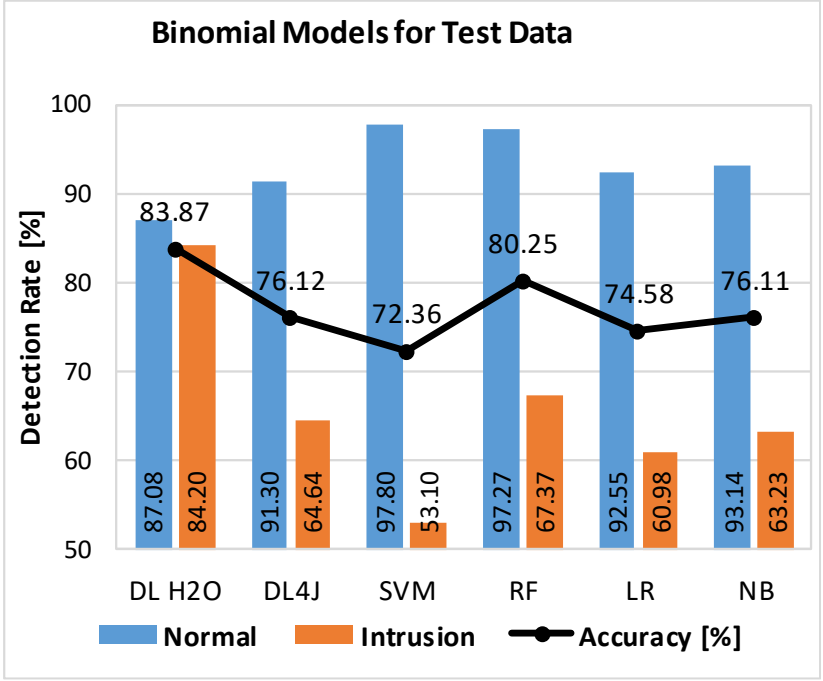
- Web app developed using: Java, jQuery, Bootstrap, Gradle
- Integrates the POJO binomial and multinomial DL models
- Twilio API integrated for real-time notifications to admin
- Cloud-based deployment to AWS EC2 instance



Evaluation Results (1)

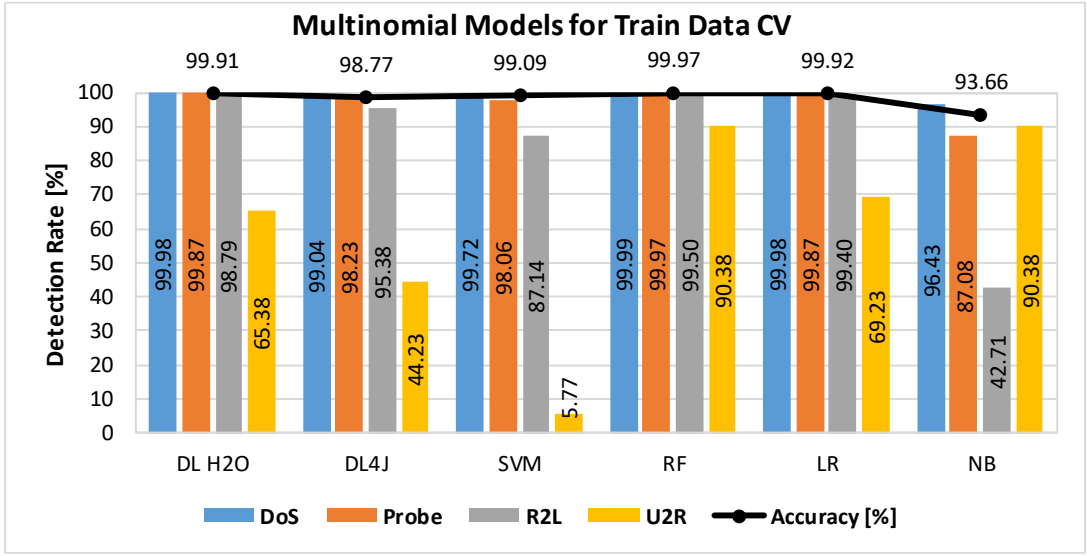
- Binomial Classification
  - All models achieved over 90% accuracy on training data
  - DL H2O achieved highest accuracy of 84% on test data
    - Higher than the 78.06% value of DL model based on soft-max regression (SMR), but less than the 88.39% value of the self-taught learning (STL) model from Niyaz et al. (2016)
  - DL H2O also has most balanced performance for detecting normal traffic and intrusions

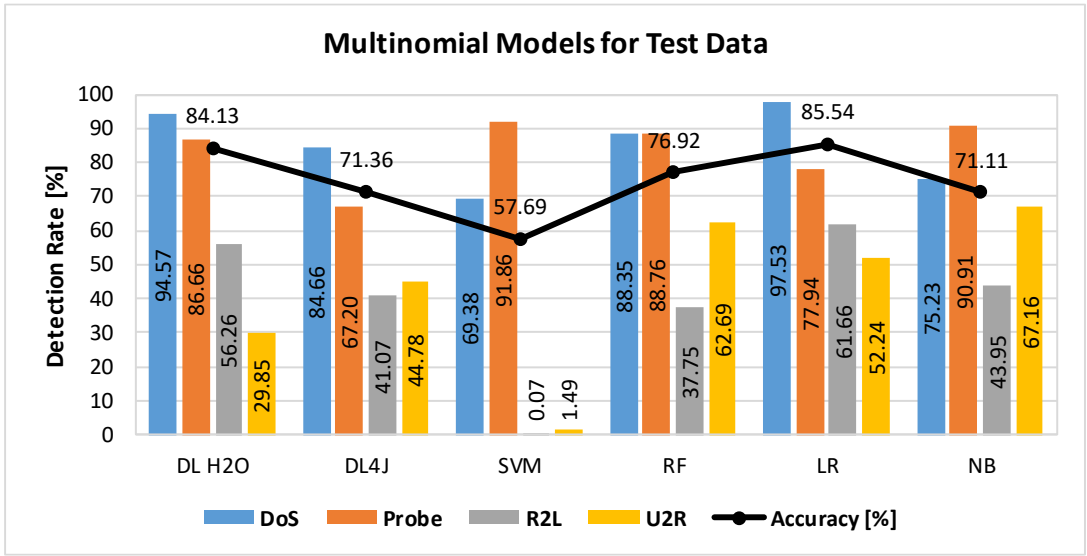




Evaluation Results (2)

- Multinomial Classification
  - All models except NB achieved over 99% accuracy
  - DL H2O achieved 84% accuracy on test data
    - Higher than the 5-class SMR (75.23%) and STL (79.10%) DL models from Niyaz et al. (2016)
  - DL H2O provides good detection rates for DoS and Probe, second best for R2L
  - NB provides highest detection rate for U2R





Conclusions

- Proposed a cloud based prototype system that integrates two DL models
  - binomial model to identify if there is an intrusion or not
  - multinomial model to detect the attack class in case of an intrusion
- DL H2O binomial and multinomial models outperformed DL4J and the other ML models, in terms of accuracy
- DL H2O binomial model also provides better intrusion detection rates than the other models
- No model provided consistent and best detection rate for all four intrusion classes
- Future work directions
  - Investigate ensemble approaches that combine multiple ML algorithms to improve detection rates
  - Use additional datasets and/or real-time network traffic data

Thank you for your attention!

Q&A