

Machine Learning for Network Anomaly Detection

Winston K.G. Seah

Key contributions by: Janel Huang, Murugaraj Odiathevar,
Marcus Freaan & Alvin Valera

Engineering and Computer Science
Victoria University of Wellington
New Zealand

- 1 Introduction
- 2 Graph-based Network Model for BGP Hijacking Detection using Machine Learning
 - Graphical Network Representation
 - Centrality and Machine Learning
 - Observations and Future Work
- 3 Hybrid Online-Offline Framework for Network Anomaly Detection
 - Framework and Methodology
 - Results and Discussion
 - Future Work

Introduction

Network Anomaly Detection

Network anomaly detection refers to the problem of finding patterns in network data that do not conform to expected behaviour.

Caveat: in the literature, network anomaly detection is usually associated with intrusion detection; network anomalies encompass more than intrusions, including malware, faulty devices, surge of traffic, misconfigurations, etc.

An anomaly detection system S can be defined as:

$$S = (M, D)$$

where

M is the model of normal system behaviour,

D is a similarity measure that, given a history of activity, determines the degree of deviation of activities with regard to the model M .

Anomaly detection using:

- Statistics
- Classifier
- Finite State Machine
- Machine Learning

Statistics-based Anomaly Detection

System observes the activity of subjects and generates profiles based on measures, e.g., traffic rate, number of packets for each protocol, rate of connections, number of different IP addresses, etc.:

- current profile – constantly updated
- stored profile – based on past records

Process:

- 1 Anomaly score is periodically calculated by comparing the current profile with the stored profile based on the measures.
- 2 If anomaly score $>$ threshold, an alert is generated.

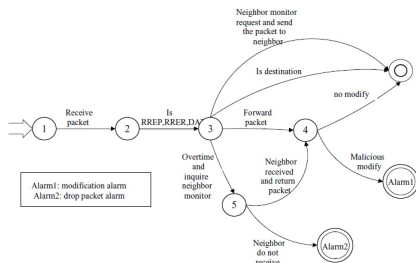
Classification typically involves the following steps:

- ① Identify classes and attributes from training data.
- ② Identify attributes (features) for classification.
- ③ Learn a model using the training data.
- ④ Use the learned model to classify the unknown data samples.

Finite State Machine-based Anomaly Detection

Finite state machine (FSM) – model of behaviour composed of states, transitions and actions.

- a state stores information about the past,
- a transition indicates a state change; described by a condition that would need to be fulfilled to enable the transition.
- an action is a description of an activity that is to be performed at a given moment.



FSM of a node monitoring the actions of another node, using the Dynamic Source Routing protocol, to detect anomalies. (Source: Yi *et al.*, "Distributed Intrusion Detection for Mobile Ad Hoc Networks," SAINT Workshops, 2005.)

Machine Learning-based Anomaly Detection

Machine learning (ML) is

- a subset of Artificial Intelligence (AI) and a powerful analytical tool based on statistics.
- used in complicated scenarios for identifying complex patterns that are not obvious to humans.

For known anomalies, ML learns from existing data to understand their characteristics.

For unknown anomalies, ML finds the outlier from the intrinsic patterns in the data.

Categories of ML:

- **Supervised Learning** – learns from existing labelled datasets, which is called *training set*, and by comparing with the known labels the predicted output can be evaluated.
- **Unsupervised Learning** – finds hidden patterns or intrinsic structures in data to group them; has input data but no expected output variables; mainly used for *clustering* and *dimensionality reduction*.
- **Semi-supervised Learning** – combines both labelled and unlabelled data to build the *classifier*, which is suitable for the scenario that has a paucity of labelled dataset.
- **Reinforcement Learning** – uses states, actions, and rewards to judge if the machine has made a good decision; RL algorithm is called an *agent*, and the agent is working in the object, called *environment*.

Machine Learning-based Anomaly Detection

Based on the available dataset, the network operator could choose:

- *supervised learning* to train a predictor when the size of labelled data is large, or
- *semi-supervised learning* when the number of labelled data is limited.

Running the same model to detect the same type of anomaly may (not unlikely at all) get different outcomes; outcomes vary depending on the features that you prefer the ML models to consider.

Fact: most difficult step in ML → *data preparation*, from data collection to annotation (labelling); a high quality dataset is vital to the prediction, as the ML algorithm relies heavily on the data to learn how to distinguish anomalous from normal behaviours.

Graph-based Network Model for BGP Hijacking Detection using Machine Learning

Border Gateway Protocol (BGP) is the backbone of the Internet that determines how traffic is routed through networks or Autonomous Systems (AS) in the Internet. An AS defines a set of IP prefixes (Internet Protocol network addresses) that belong to a network or a group of networks.

BGP update packets are regularly exchanged by routers to determine the routes for sending datagrams to their intended destinations.

- BGP *bviews* – infrequent periodic exchange (usually once every hour) of the routing table of a BGP router;
- BGP *updates* – propagated (usually in 15-minute periods) to advertise routable paths, as routes may change more frequently than hourly timeframes.

BGP Updates – Key Attributes

- *Announcement Routes* – List of IP address prefixes for routes that should be added to the advertising node.
- *Withdrawn Routes* – List of IP address prefixes for routes that should be withdrawn.
- *AS_PATH* – List of ASes along the path.
- *NEXT_HOP* – IP address of the next router from the advertising BGP router to forward the message to the destination.
- *Network Layer Reachability Information* – List of IP address prefixes that specify how to reach prefixes.

BGP Anomalous Events

BGP Anomalies are caused by:

- Hijacking – attackers impersonate ASes by advertising false BGP routes to **maliciously** reroute Internet traffic, e.g. panix.com incident.
- Misconfiguration (non-malicious) events
 - *Intentional* – Pakistan Telecom incident, where invalid BGP routes were advertised with the intention being to ban youtube.com; multiple ASes' youtube traffic were redirected to the Pakistan AS → Denial of Service (DoS) for Pakistan AS and loss of connectivity to YouTube for affected ASes.
 - *Unintentional* – Global BGP CenturyLink outage caused by the misconfiguration of BGP routes.

BGP Anomaly Detection Techniques

Method	Effectiveness	Limitations
Time Series	Able to detect anomalies using data within a fixed period.	Incapable of real-time detection
Statistical Pattern Recognition	Able to correlate events to detect anomalies	Incapable of real-time detection. Incapable of determining the anomaly source
Historical BGP	Able to detect prefix hijacks.	Unable to detect sub-prefix hijacks. Unable to detect link failures and indirect anomalies.
Reachability Check	Able to detect prefix hijacks.	Unable to detect sub-prefix hijacks, link failures and indirect anomalies.
Machine Learning BGP	Capable of detecting occurred BGP anomalies.	Incapable of detecting new BGP anomalies. Unable to determine the anomaly source.

Problem

Current BGP anomaly detection methods (e.g. historical BGP, time series, and reachability check)

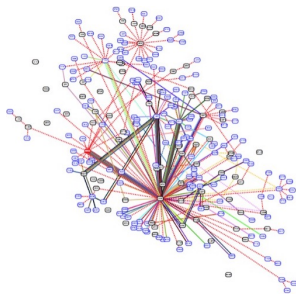
- cannot automatically learn from experience;
- use *node level features* to detect anomalies:
 - Average Autonomous System (AS) path length;
 - Number of withdrawals or announcements;
- do not consider the entire network graph;
- are incapable of real-time detection and determining the source of the anomaly.

Need to select network-level features to detect anomalies.

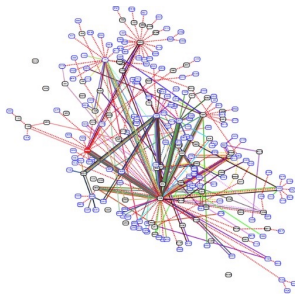
Approach

Select appropriate BGP update attributes.

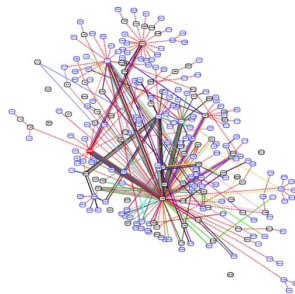
Construct network graph.



Before anomaly event



During anomaly event



After anomaly event

Extract features, such as:

- Node-level features (easy to compute)
 - average AS-path length
 - number of changed AS-paths
- Network-level features
 - connectivity
 - node centrality

Connectivity

Measures how interconnected nodes are within the network. Network topology changes significantly during BGP anomaly incidents.

Measures of connectivity:

- Clustering coefficients – measure the level at which nodes are clustered together;
- Level of overlapping neighbourhoods.

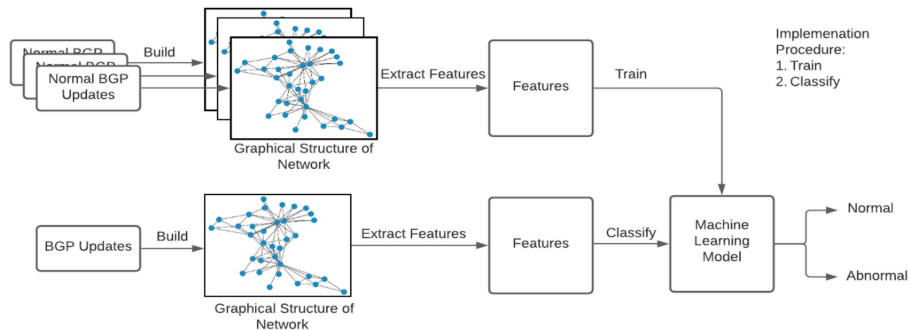
Drawbacks:

- Storage and computation intensive.
- Requires definition of “neighbourhood” which is inaccurate using simple methods, e.g. k -means clustering.
- Aggregation of neighbours by prefixes, articulation points, or spectral clustering can be used to summarise the connections of nodes, but methods lose individual network information.

Key centrality metrics include:

- Betweenness Centrality – Number of paths that pass through a node.
- Eigenvector Centrality – Combines the importance and number of immediate neighbours of a node.
- Degree Centrality (DC) – Number of immediate neighbours of a node.
- Closeness Centrality (CC) – Inverse distance to all the reachable neighbours of a node.

Workflow of Anomaly Detection Method

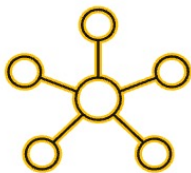


To achieve realtime anomaly detection, classification workflow (lower half) needs to be done before the next BGP update arrives. So, the methods for *network graph construction* and *feature extraction* must be simple, fast, yet accurate.



Closeness Centrality

Interconnectivity or clustering coefficients of the entire network lose *individual network* information.

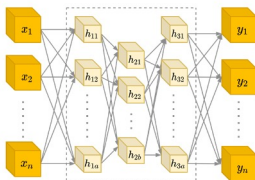


Degree Centrality

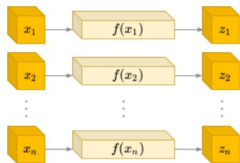
Eigenvector and betweenness centralities are not feasible in *memory* and *time*.

Machine Learning Models

Two models are selected for our approach:



Autoencoders



Univariate Gaussian

Autoencoders used to capture complex relationships amongst the datapoints in the dataset.

Individual network anomaly detection using Univariate Gaussian (UG).

UG is unstable in detecting anomalies from routers that have a limited visibility

Autoencoders

A type of Neural Network (NN) that aims to learn a reconstruction of data.

Anomalies can be detected through the reconstruction error generated in the learnt normal model.

If the reconstruction error is greater than a threshold, it will indicate anomalies.

Drawback: detection of anomalies for a specific AS is not possible.

Need: Identify problematic ASes to avoid routing to such ASes in an abnormal event.

UG is capable of modelling each AS as a Gaussian distribution to detect anomalies.

UG can also capture second-order statistics with a much lower computational overhead than Autoencoders.

Evaluation – CenturyLink outage

On 30th August 2020, around 10:04 (UTC), BGP misconfiguration by US ISP CenturyLink led to global Internet outage.

Root cause: Misconfigured flow specification (flowspec) rule;

flowspec “allows you to rapidly deploy and propagate filtering and policing functionality among a large number of BGP peer routers to mitigate the effects of a distributed denial-of-service (DDoS) attack over your network.” - *Cisco*

Incorrect rule → routing loop created in CenturyLink’s internal network
→ incorrect routes announced to other peer ISPs, propagating the error.

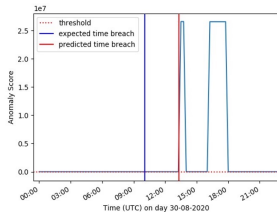
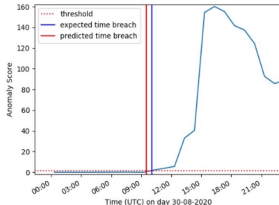
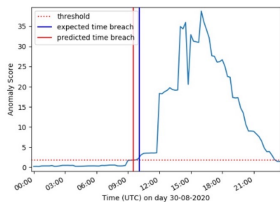
Solution: CenturyLink had to reset all equipment and start with clean BGP routing tables, a process that took almost 7 hours to complete.

Process BGP update data at various ASes, viz. NZ, WIDE (JP) and SOXRS (Serbia), with a network view up to 2 hops away; more hops give better global view but also increases computation load significantly.

Network-wide analysis show anomalies for entire network, but source or infected ASes also need to be identified to prevent routing to such networks.

Data are unlabelled and it is possible that network was unstable before anomaly event; hence, rise in anomaly score before anomaly event is possible.

Network-wide Closeness Centrality



ASes

Network of WIDE CC



ASes

Network of NZ CC

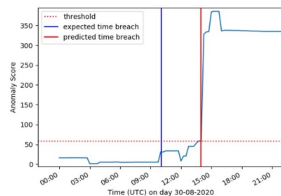
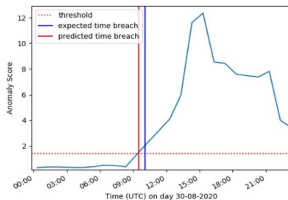
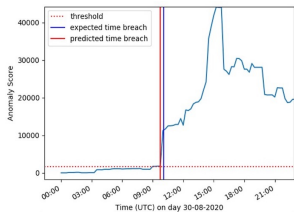


ASes

Network of SOXRS CC

Reachability distances for many nodes change significantly due to anomaly. NZ and WIDE ASes show rise in anomaly score before the event. SOXRS detected the anomaly event *after* it happened, as it is further away.

Network-wide Degree Centrality



ASes

Network WIDE DC



ASes

Network of NZ DC



ASes

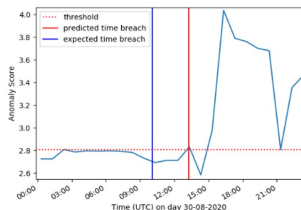
Network of SOXRS DC



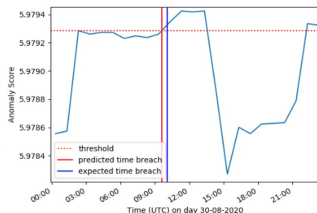
Severity images show which ASes are affected.

From NZ core router's viewpoint, it shows a large number of affected ASes.

AS38022 Anomaly Detection from NZ and WIDE



NZ AS38022 CC

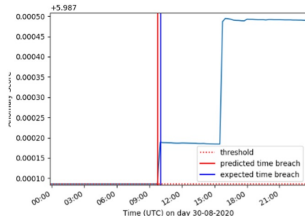


NZ AS38022 DC

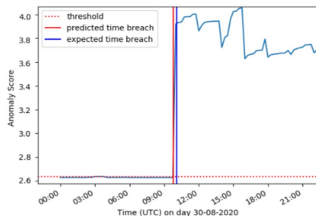
CC anomaly prediction (left figure) is later than actual event as the event was external.

Earlier detection for DC (right figure) due to number of immediate neighbours of AS38022 changing significantly during the anomaly event.

WIDE predicted anomaly earlier as it can view the anomalous activity between AS38022 and AS3561 from an outsider's point of view. AS3561 (anomaly source) is a trusted network peer of AS38022, the error that is propagated by AS3561 is deemed normal when transferred to AS38022.

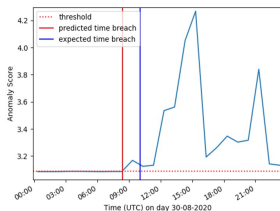


WIDE AS38022 CC

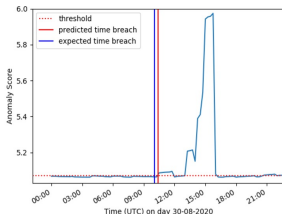


WIDE AS38022 DC

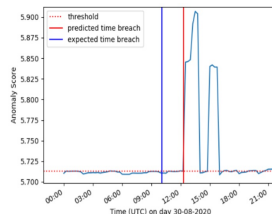
AS3561 Anomaly Detection from NZ, WIDE and SOXRS



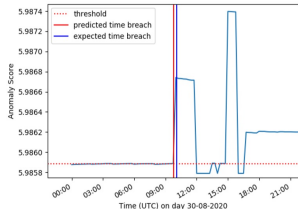
NZ AS3561 CC



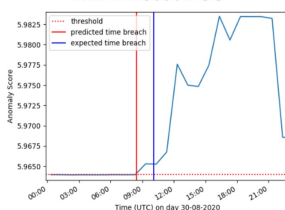
WIDE AS3561 CC



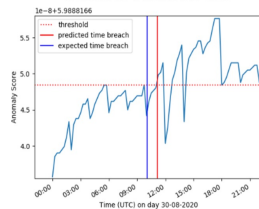
SOXRS AS3561 CC



NZ AS3561 DC



WIDE AS3561 DC



SOXRS AS3561 DC

NZ AS38022 is closest to AS3561, hence able to detect anomaly earlier than WIDE and SOXRS.

Summary of Approach

Use of graph-level features to represent and detect network anomalies before they occur.

Ability to detect network-wide as well as AS-specific anomalies.

Corroboration of multiple networks, e.g. NZ, Japan and Serbia, to provide better network anomaly detection capability.

Challenges

Resource constraints when dealing with major core routers, such as, London and Singapore, that contain gigabytes of data for each 15-minute BGP update in comparison to kilo/megabytes of data in NZ, WIDE and SOXRS.

Current work is limited to the NZ core router and its neighbours (up to 2 hops away); anomalies in other parts of the Internet are not evaluated. Limit on the size definition of the entire network is required as the memory and computation requirements are infeasible when considering more hops.

- Other application areas – which can be represented in a graph, detection of anomalies using centrality information is also possible; e.g. anomalies present in power grids and transport traffic can be detected using centralities where a change in the topology can indicate a power outage or a traffic accident, respectively, in both cases.
- Distributed processing – computational workloads are balanced amongst multiple processors to handle major core routers.
- Network Neighbourhood Aggregation - to scale to the whole Internet, networks can be aggregated into neighbourhoods to reflect a more condensed network.

Future Work

- Traffic Link Analysis – additional data such as the amount of traffic traversing in the network links can be used to determine anomalies, e.g. Distributed DoS (DDoS) attacks, as the centrality of nodes cannot reflect the amount of traffic that is travelling between ASes.
- Variational Autoencoders – Autoencoders can fail to represent data in latent space that is not within the observed data.
- Selection of Trusted Core Routers – to monitor Internet-wide anomalies; investigate the generation a trustworthiness scheme for each BGP router.
- Online or Batch Learning – select the an appropriate amount of data to continuously train the proposed machine learning models, so that they can adapt to changing normal network conditions.

Hybrid Online-Offline Framework for Network Anomaly Detection

What is normal?

Concept-drift → change in the underlying distribution of the data.

Cause

- new applications are being created almost daily
- Internet of Things (IoT) traffic profiles display large variations and heterogeneity

Models and Approaches

- **Offline models** – Models which are trained with certain data and used to detect anomalies. Signature based models, deep learning models, traditional models, etc., which can be optimised.
- **Online models** – Outlier detection, incremental learning with new data by updating thresholds. Continuous learners, unsupervised and learn from data grouped in time windows.
- **Combination** – An offline step to select features in an unsupervised way followed by online learning. Or, online learning by clustering then using offline model to train on cluster centers.

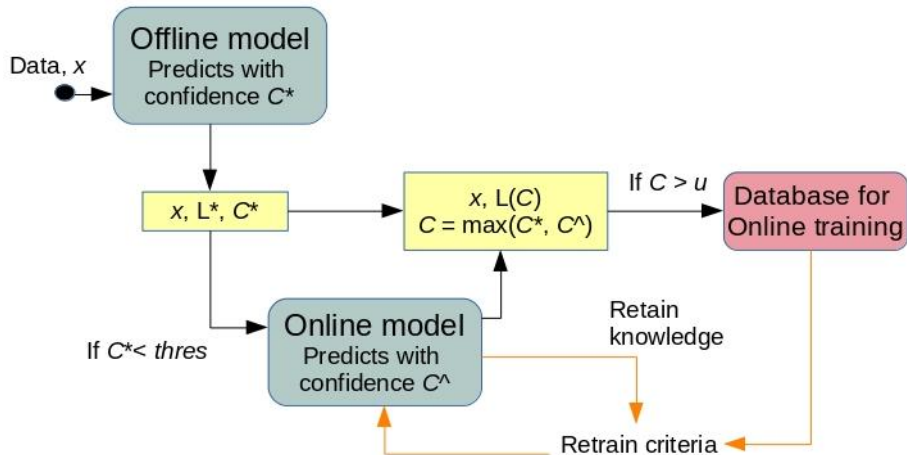
Issues faced by different methods:

- **Offline models** - not efficient in detecting new anomalies or variants of existing anomalies; unable to adapt to changing normal traffic or network conditions.
- **Online models** - Affected by noise or irrelevant points during training.
- **Combination methods** - offline step to select necessary features in a supervised/unsupervised manner followed by online learning.

Our proposed approach:

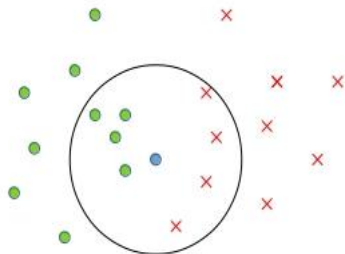
- A method to enable an Offline and an Online model to work together
- A technique to train an Online model effectively
- An approach to handle bias-variance trade-off separately

Framework



Radius Nearest Neighbour (Rad-NN)

$$C^*(x, k) = 1 - \frac{\sum_{i=1}^k D(x, NLN_i(x))}{\sum_{i=1}^k D(x, NUN_i(x))}$$



Support Vector Machine (SVM)

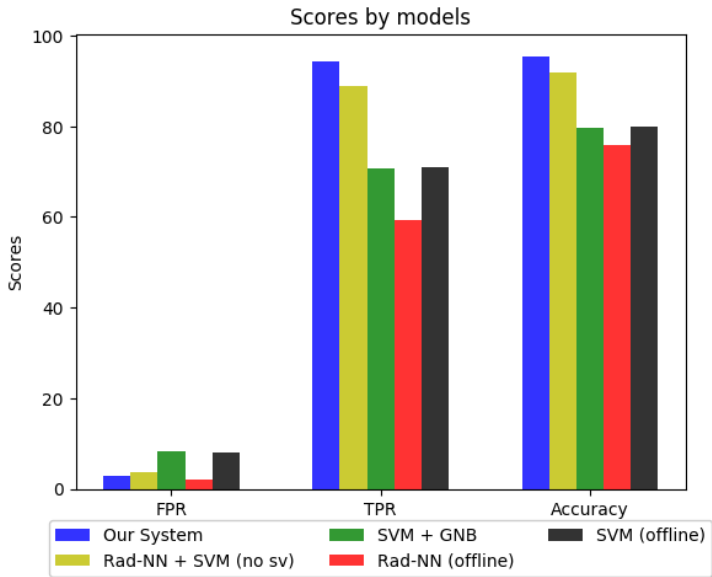
Platt-scaling

$$C^{\wedge}(x) = \begin{cases} \frac{1}{1 + \exp(Af(x) + B)} & \text{label} = 1 \\ 1 - \frac{1}{1 + \exp(Af(x) + B)} & \text{label} = 0 \end{cases}$$

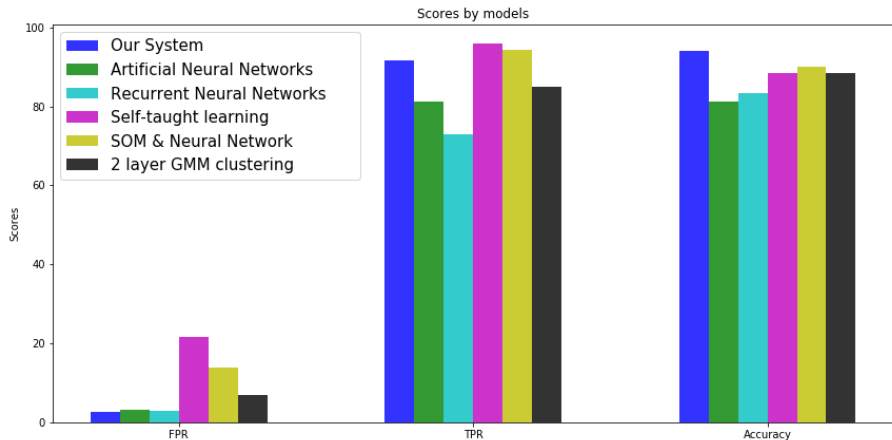
System parameters based on 20% of training set

- $k = 10$
- $thres = 92.2\%$
- $u = 96\%$
- Retrain Criteria = 1300 points using grid search
- SVM parameters $C=100$, $\gamma = 0.1$ using grid search

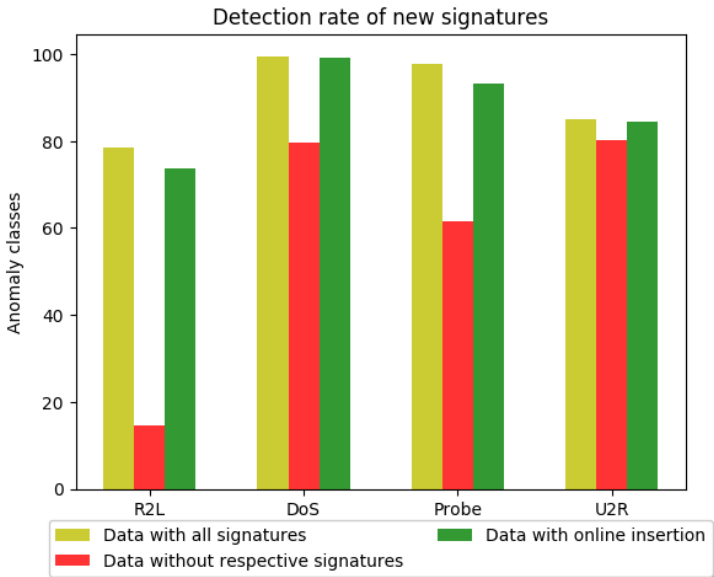
Results on NSL-KDD 2009 dataset



Results on NSL-KDD 2009 dataset



Results on NSL-KDD 2009 dataset



An Online and Offline model working together to overcome their weaknesses

- Offline Rad-NN
 - classifies by obtaining votes from data within a radius
 - provides a bias for Online SVM to select points to train on
- Online SVM
 - trains on new data and retains support vectors
 - classifies if the Rad-NN has low confidence
 - can be trained with high variance
- New signatures can be inserted in the Online model instead of retraining the Offline model

Benefits and Further Extensions

Online-Offline Framework helps to improve overall detection rate by identifying unseen normal data. It is trained on normal data and adapts to changing normal data.

To label an attack, a classification model can be trained and deployed on top of it as part of a pipeline. (Assumption: the attack behaviour does not change.)


There is potential to explore:

- 1 training the Online-Offline framework with attack data, then it adapts to changing attack data. This is still binary: normal vs attack
- 2 training a classifier with the Online-Offline framework.

- A behaviour based version using only the 'normal' class
- Multiple Online models trained on different features
- Further processing on low confidence points
- Use of other methods instead of Rad-NN and SVM

- Murugaraj Odiathevar, Winston K.G. Seah and Marcus Freat, “A Hybrid Online Offline System for Network Anomaly Detection,” *Proceedings of the 28th International Conference on Computer Communications and Networks (ICCCN 2019)*, July 29 - August 1, 2019, Valencia, Spain.
- Murugaraj Odiathevar, Winston K.G. Seah, Marcus Freat and Alvin Valera, “An Online Offline Framework for Anomaly Scoring and Detecting New Traffic in Network Streams,” *IEEE Transactions on Knowledge and Data Engineering*, 11 January 2021.

Thank you

 winston.seah@ecs.vuw.ac.nz

 WinstonSeah

 winstonkgseah

For other related publications:

