Machine Learning in Network Anomaly Detection

Winston Seah

School of Engineering and Computer Science

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.

Types of Network Anomalies

- **Volume-based** – abnormal traffic volume
- **Contextual** – normal / abnormal in different contexts
- **Point** – single data point/event deviating from norm
- **Collective** – group of data points deviating from norm
- **Protocol/Port** - unusual activity on non-standard ports or protocols (e.g., HTTP traffic on SSH ports)
- **Behavioural** - deviations in user/device behaviour
- **Performance** – degradation in performance metrics

Anomaly Detection System

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.

Typical Anomaly Detection Methods

- Time Series Analysis
- Statistical
- Classifier
- Signature-based Detection
- Behavioural Analysis

Learning-based Methods

- Machine Learning (ML)
  - Supervised, Unsupervised, Semi-supervised
- Deep Learning (DL)
  - uses artificial NNs (ANNs) with multiple layers to automatically extract patterns from raw data; no need for manual feature engineering

ML-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.

ML-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 features that you tell the ML models to consider.

ML-based Anomaly Detection

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.

DL-based Anomaly Detection

- Enables identification of complex, subtle and evolving threats in high-dimensional network data.
- Typical methods:
  - Autoencoders (AEs)
  - Convolutional Neural Networks (CNNs)
  - Recurrent Neural Networks (RNNs/LSTMs)

Autoencoders

Compresses input data (e.g., network traffic logs) into a latent space and reconstructs it. Anomalies exhibit ***high reconstruction errors*** due to deviations from learned "normal" patterns.

Use cases:

- detecting DDoS in cloud environments;
- detecting BGP route hijacks by modelling normal routing behaviour.

Convolutional Neural Networks (CNNs)

Extracts spatial features from network traffic matrices or spectrograms (e.g., packet headers converted to 2D images)

Use cases:

- classifying intrusion in imbalanced datasets;
- detecting jamming attacks in wireless networks using signal strength patterns, e.g. RSSI, SNR, etc.

Recurrent Neural Networks (RNNs)

Models temporal dependencies in sequential data (e.g., time-series network logs) to flag deviations.

Use cases:

- Identifying APTs (Advanced Persistent Threats) via long-term behavioural analysis;
- Real-time detection of lateral movement in internal traffic.

Benefits of DL approaches

- *Useful for Unstructured Data*: Excels with images, audio, and text where traditional ML struggles.
- *Accurate*: Outperforms humans in tasks like object detection.
- *Scales well*: Improves with more data and compute power (unlike ML, which plateaus).

Limitations of DL Approaches

Data Hungry: Requires massive labelled datasets.

Computational Complexity: Needs GPUs/TPUs for training → expensive and energy-intensive.

Black Box: Hard to interpret how decisions are made, critical in many applications, e.g. healthcare, finance, ***network management***, etc.

Realtime Detection and Visualization of BGP Anomalies using Machine Learning

Routing in the Internet

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.

Routing in the Internet

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 Anomalities 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.

Problem

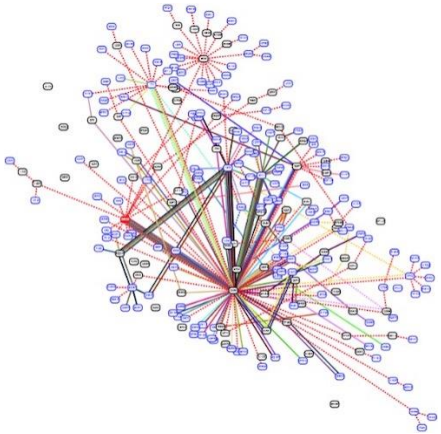
Existing BGP anomaly detection methods (e.g. historical BGP, time series, and reachability check)

- 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.

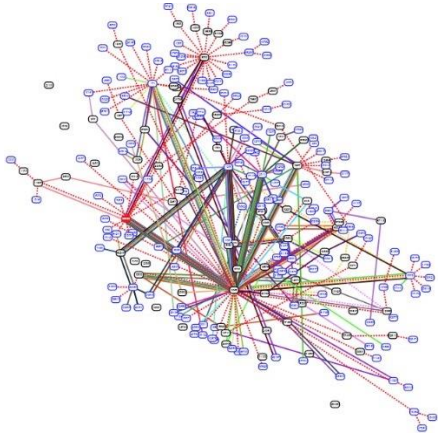
Need to select network-level features to detect anomalies.

Graph-based Approach

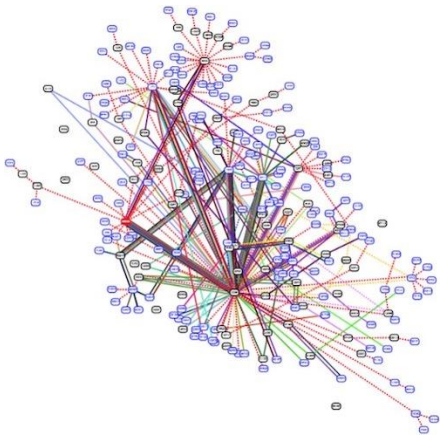
1. Select appropriate BGP update attributes.
2. Construct network graph.



Before anomaly event



During anomaly event



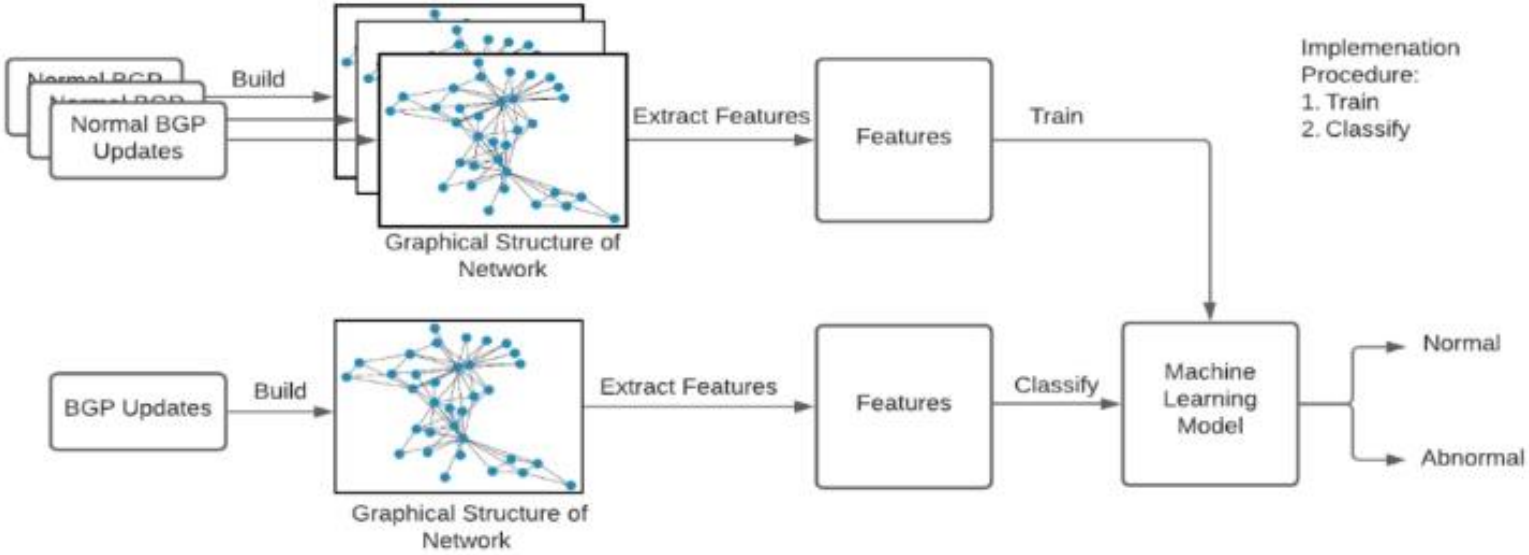
After anomaly event

Centrality

Ranks nodes within a network graph based on their network position; key centrality metrics include:

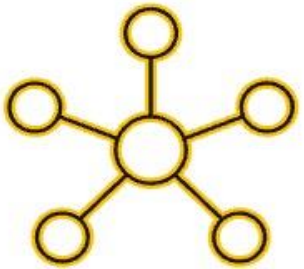
- Betweenness Centrality – Number of paths that pass through a node.
- Eigenvector Centrality – Combines the importance and number of immediate neighbours of a node.
- Degree Centrality (DC) – Number of immediate neighbours of a node.
- Closeness Centrality (CC) – Inverse distance to all the reachable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

Workflow of Anomaly Detection



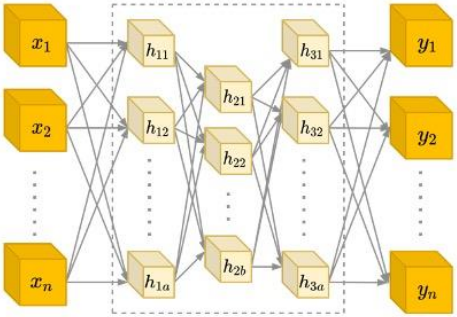
To achieve realtime anomaly detection, classification workflow (lower half) needs to be done before the next BGP update arrives. So, the methods for *network graph construction* and *feature extraction* must be simple, fast, yet accurate.

Features and Models

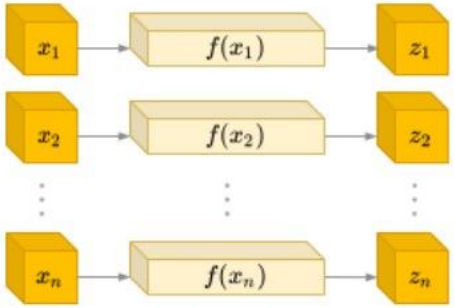


Degree Centrality

Closeness Centrality



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



Individual network anomaly detection using Univariate Gaussian

Test Case – CenturLink Outage

On 30th August 2020, around 10:04 (UTC), BGP

by US ISP CenturyLink led to global

Root cause: Misconfi **owspec) rule;**

flowspec "all filtering and policing

BGP peer routers to mitigate the effects

(DDoS) attack over your network." - *Cisco*

ting loop created in CenturyLink's internal network

rect 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.

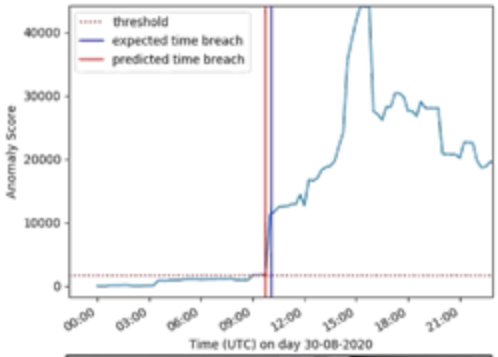
Evaluation Process

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.

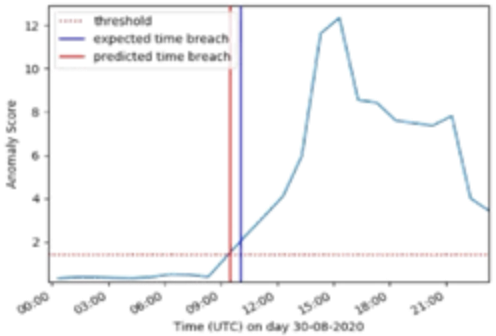
Network-wide analysis show anomalies for entire network, but source or infected ASes also need to be identified to prevent routing to such networks.

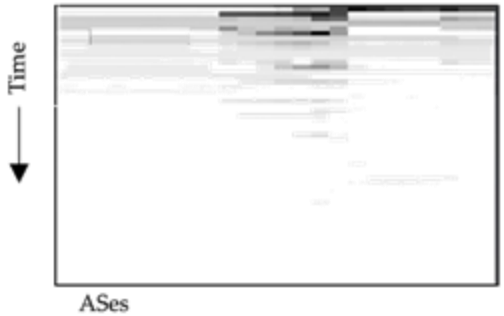
Data are unlabelled and it is possible that network was unstable before anomaly event; hence, rise in anomaly score before anomaly event is possible.

Network Wide Degree Centrality

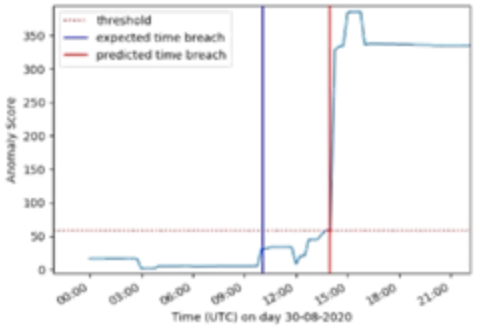


Network WIDE DC



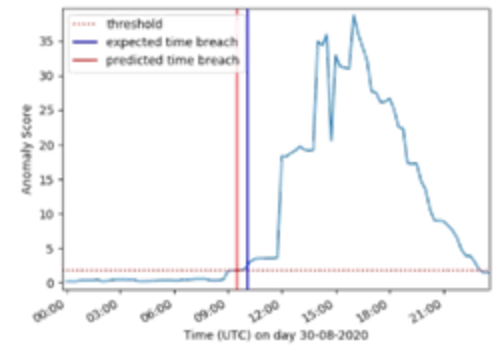


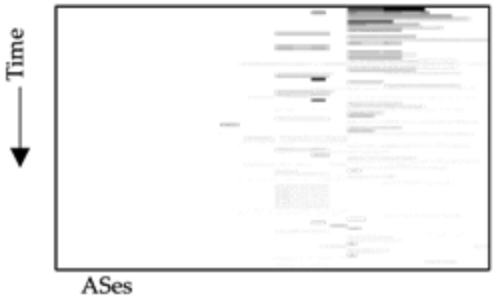
Network of NZ DC



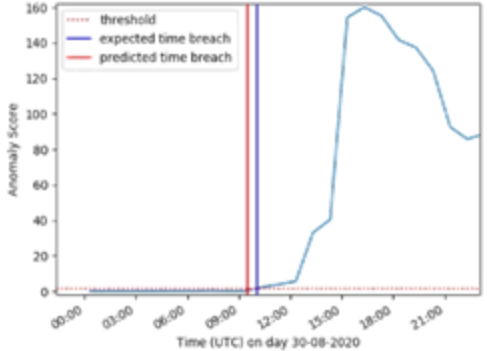
Network of SOXRS DC

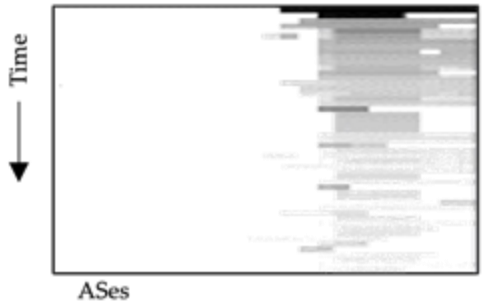
Network Wide Closeness Centrality



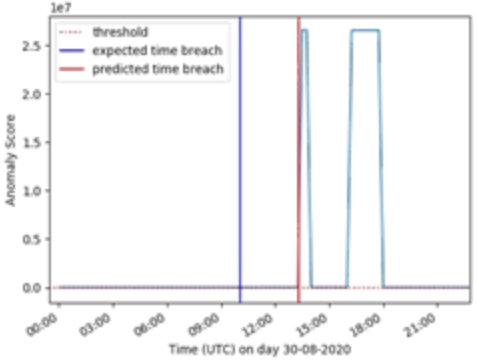


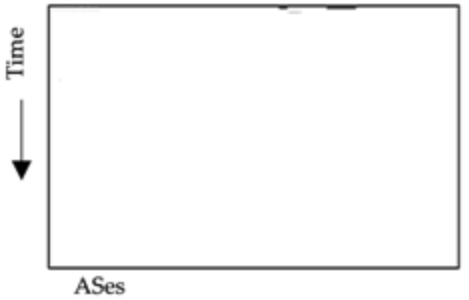
Network of WIDE CC





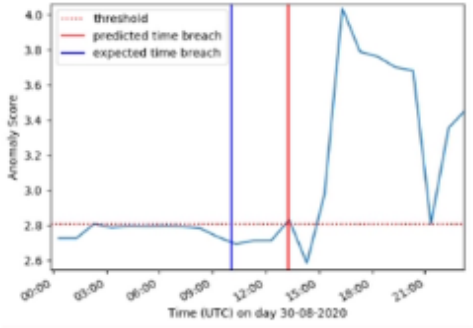
Network of NZ CC



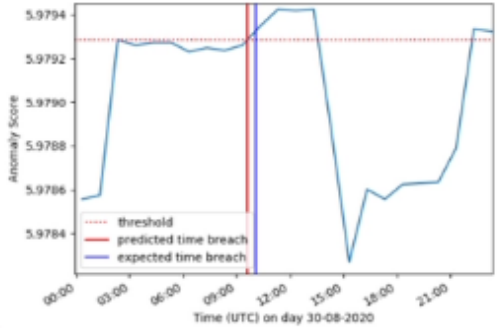


Network of SOXRS CC

AS38022 Anomaly Detection from NZ & WIDE



NZ AS38022 CC

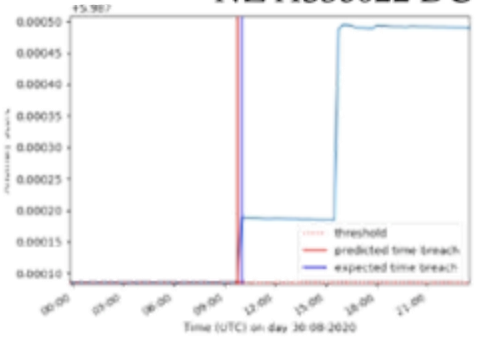


NZ AS38022 DC

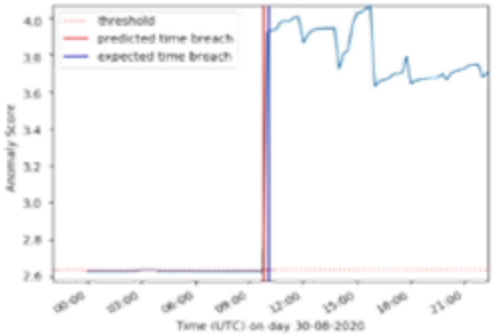
CC anomaly prediction (left figure) is later than actual event as the event was external.

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. As *AS3561 (anomaly source) is a trusted network peer of AS38022, the error that is propagated by AS3561 is deemed normal when transferred to AS38022.*

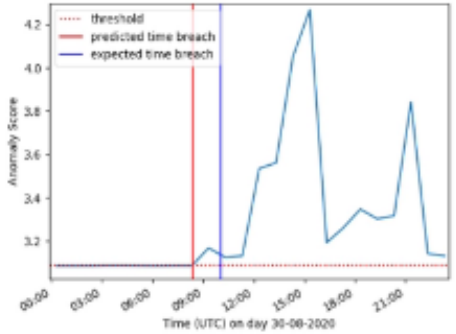


WIDE AS38022 CC

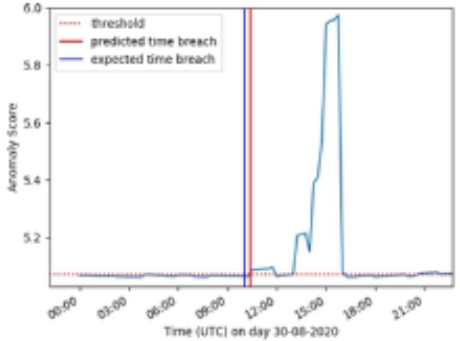


WIDE AS38022 DC

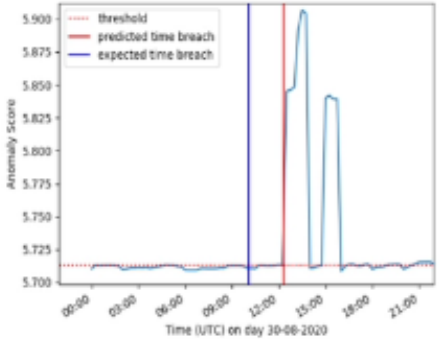
AS3561 Anomaly from NZ, WIDE & SOXRS



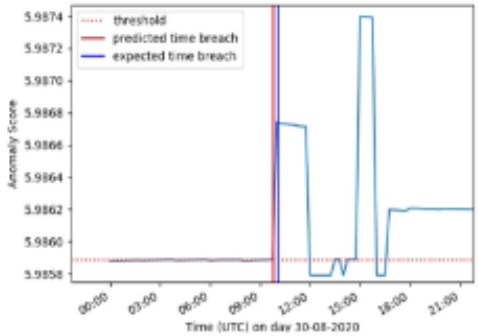
NZ AS3561 CC



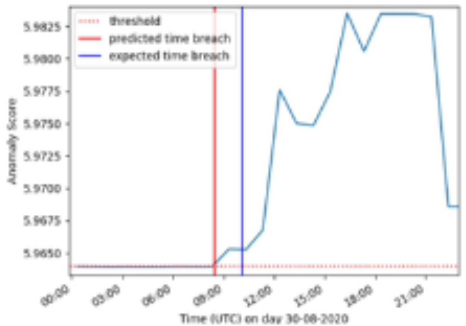
WIDE AS3561 CC



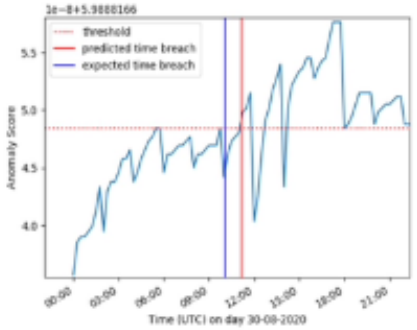
SOXRS AS3561 CC



NZ AS3561 DC



WIDE AS3561 DC



SOXRS AS3561 DC

Observations

- 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

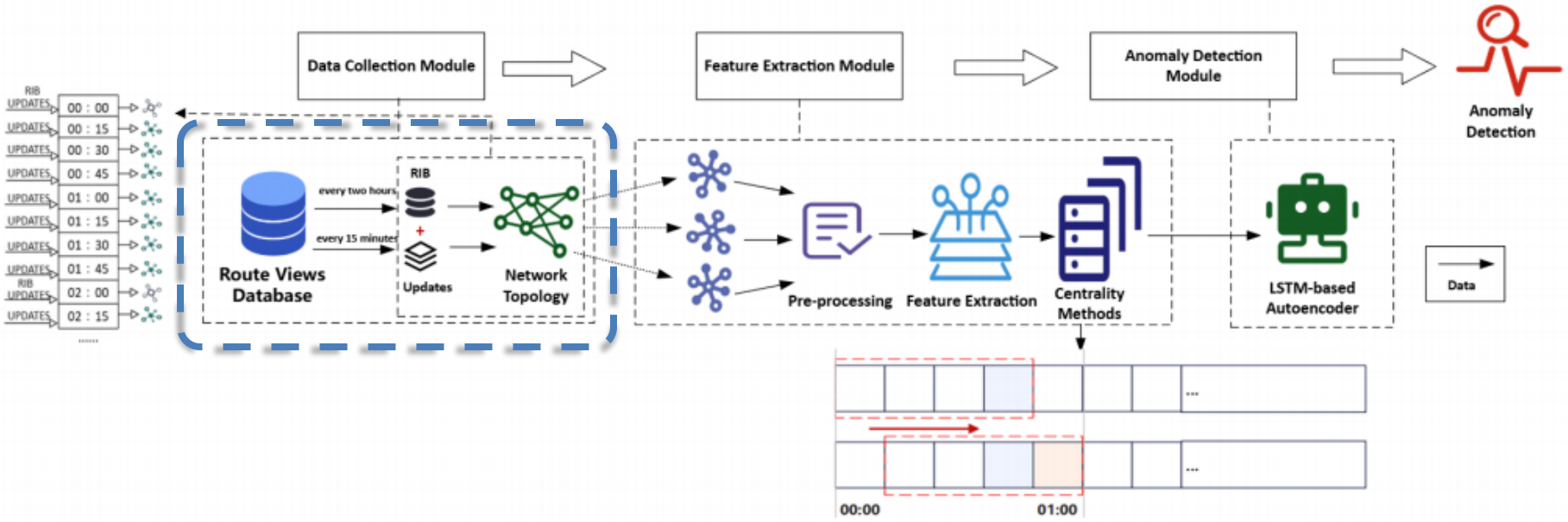
Resource constraints when dealing with major core routers, such as, London and Singapore, that contain GBs of data for each 15-minute BGP update in comparison to KBs/MBs of data in NZ, WIDE and SOXRS.

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.

Current Work

A Lightweight Framework for BGP Anomaly Detection with Centrality Trajectories and Subgraph Partitioning

Partition-based Anomaly Detection



Data Collection Module

- MRT format RIBs and UPDATEs from PhOpenIX (FRR bgpd, from route-views.phoix.routeviews.org)
- MRT format RIBs and UPDATEs from TELXATL (FRR bgpd, from route-views.telxatl.routeviews.org)
- MRT format RIBs and UPDATEs from DIXIE (WIDE) (quagga bgpd, from route-views.wide.routeviews.org)
- MRT format RIBs and UPDATEs from SYDNEY (FRR bgpd, from route-views.sydney.routeviews.org)
- MRT format RIBs and UPDATEs from SAOPAULO (quagga bgpd, from route-views.saopaulo.routeviews.org)
- MRT format RIBs and UPDATEs from 2nd SAOPAULO (FRR bgpd, from route-views2.saopaulo.routeviews.org)
- MRT format RIBs and UPDATEs from SINGAPORE (quagga bgpd, from route-views.sg.routeviews.org)
- MRT format RIBs and UPDATEs from PERTH (FRR bgpd, from route-views.perth.routeviews.org)
- MRT format RIBs and UPDATEs from PERU (FRR bgpd, from route-views.peru.routeviews.org)
- MRT format RIBs and UPDATEs from SFMIX (FRR bgpd, from route-views.sfmix.routeviews.org)
- MRT format RIBs and UPDATEs from SIEX (FRR bgpd, from route-views.siex.routeviews.org)
- MRT format RIBs and UPDATEs from SOXRS/Serbia (quagga bgpd, from route-views.soxrs.routeviews.org)
- MRT format RIBs and UPDATEs from RIO (FRR bgpd, from route-views.rio.routeviews.org)

(1) Data Source: **Route Views**

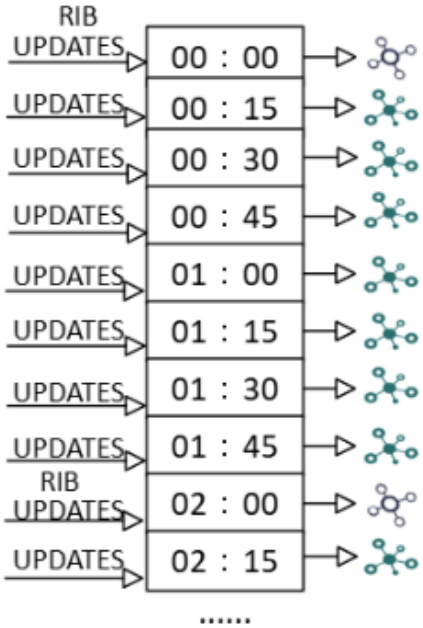
(2) Types of Data Obtained:

- **RIB** (Routing Information Base): obtained *every two hours* and provides a complete, static view.
- **Updates**: They are collected *every 15 minutes* to notify neighbors of changes in network topology or path attributes, ensuring dynamic updates to the routing table.

(3) Tools Used:

- Data Parsing: **BGP dump** parsing is performed using tools within the Linux system.

Data Collection Module



RIB (only major attributes listed):

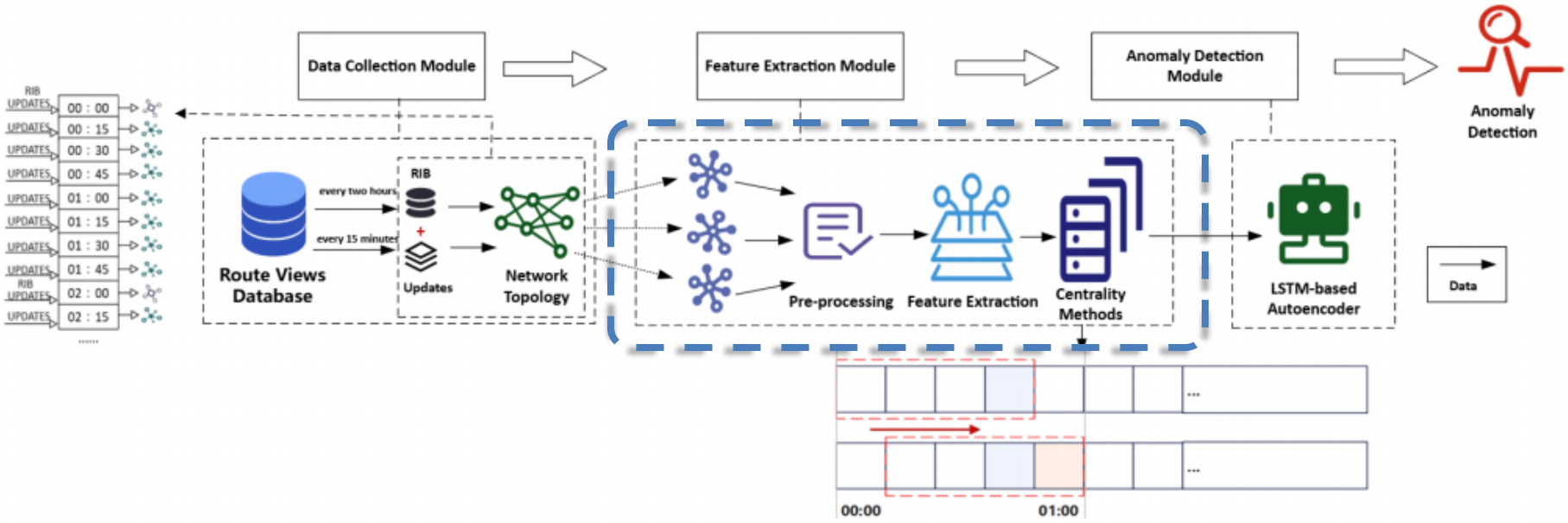
AS Path, Prefix, Next Hop, Origin AS

Updates:

Announcements, Withdrawals

```
Timestamp: 2019-04-09 00:00:04
Prefix: 2c0f:f690::/32
AS Path: 2500 2914 6939 37406
AS Number: 37406
Next Hop: 2001:200:0:fe00::9c4:11

Timestamp: 2019-04-09 00:00:04
Prefix: 2c0f:f690::/32

Timestamp: 2019-04-09 00:00:04
Prefix: 2620:144:a00::/40
AS Path: 2500 2914 209 53692
AS Number: 53692
Next Hop: 2001:200:0:fe00::9c4:11

Timestamp: 2019-04-09 00:00:04
Prefix: 2620:144:a00::/40
AS Path: 2500 2914 209 53692
AS Number: 53692
Next Hop: 2001:200:0:fe00::9c4:11

Timestamp: 2019-04-09 00:00:04
Prefix: 2c0f:f690::/32
AS Path: 2500 2914 6939 37406
AS Number: 37406
Next Hop: 2001:200:0:fe00::9c4:11

Timestamp: 2019-04-09 00:00:04
Prefix: 2c0f:f690::/32
AS Path: 2500 2914 6939 37406
AS Number: 37406
Next Hop: 2001:200:0:fe00::9c4:11

Timestamp: 2019-04-09 00:00:04
```

Partition-based Anomaly Detection



Feature Extraction Module

① Degree

② Closeness

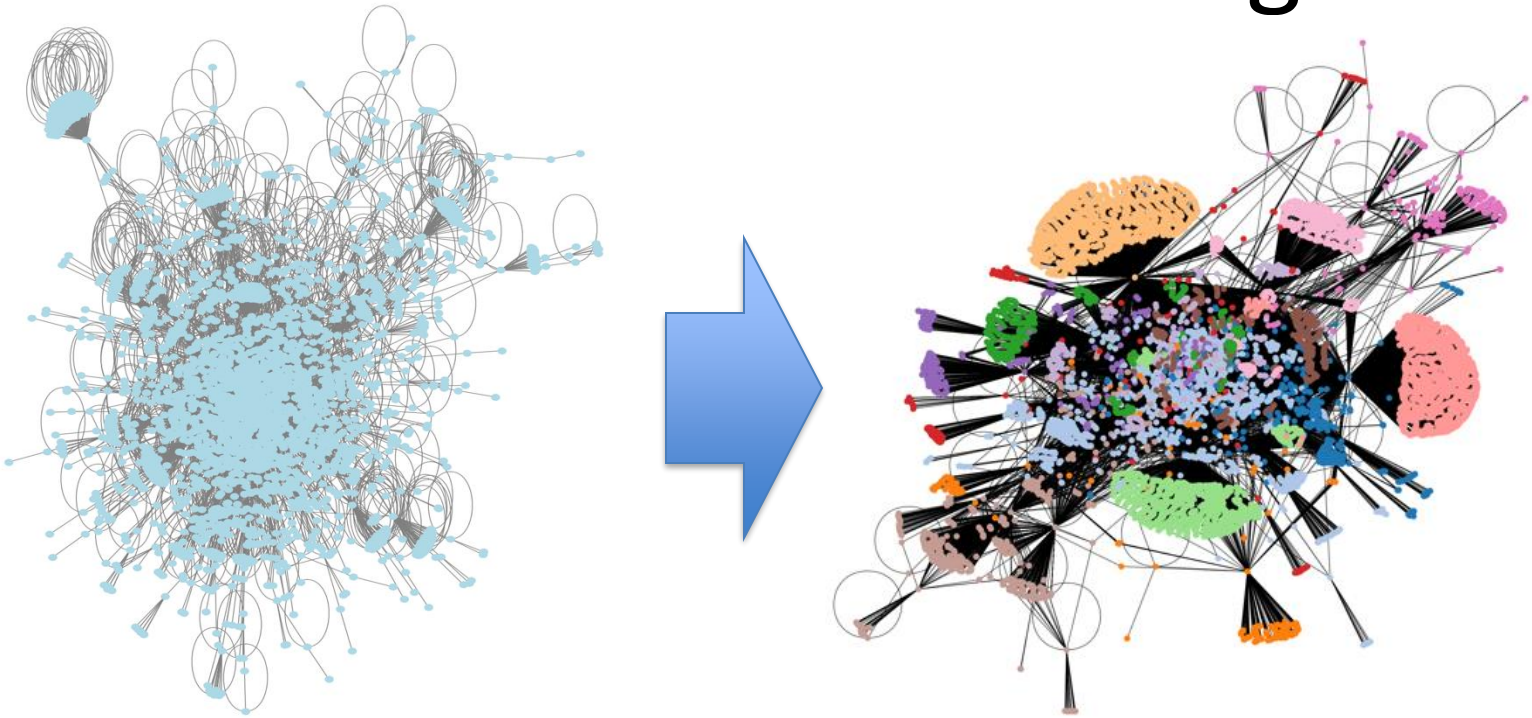
③ Eigenvector

④ Betweenness

Calculating Centrality Combinations:

1. ①, ②, ③, ④
2. ①②, ①③, ①④, ②③, ②④, ③④
3. ①②③, ①②④, ①③④, ②③④
4. ①②③④

Network Partitioning



Network Partitioning

Feature	Louvain Algorithm	Leiden Algorithm
Optimization Quality	Prone to local optima	Higher quality, stronger community connectivity
Efficiency	Moderate, faster for large-scale networks	Faster, suitable for large-scale networks
Connectivity	May produce disconnected communities	Ensures community connectivity
Robustness	Sensitive to initial conditions	More stable, higher robustness

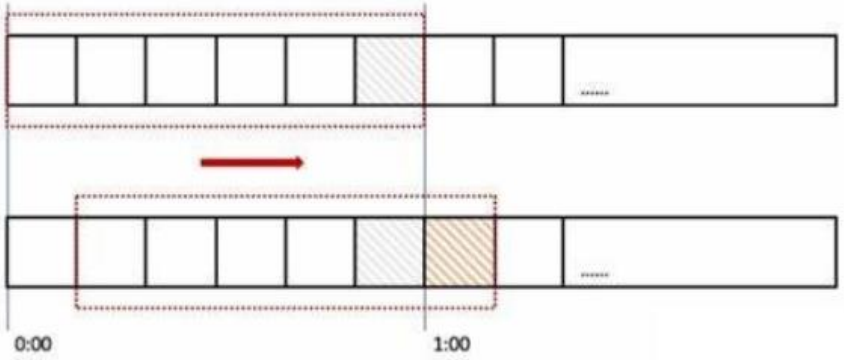
Feature Extraction Module

Centrality Vector per Subgraph

- Degree, Betweenness, Closeness, Eigenvector
- Concatenated as a structural snapshot vector

Sliding Window Trajectory

- Fixed-length window over time
- Each subgraph forms a centrality trajectory



Input for Detection

- Trajectories fed into LSTM-AutoEncoder
- Captures temporal structural evolution

Whole Network, Louvain, and Leiden

Feature	Whole Network(WIDE)				Louvain(WIDE)				Leiden(WIDE)				Whole Network(NWAX)				Louvain(NWAX)				Leiden(NWAX)			
	Acc	Pre	Recall	F1	Acc	Pre	Recall	F1	Acc	Pre	Recall	F1	Acc	Pre	Recall	F1	Acc	Pre	Recall	F1	Acc	Pre	Recall	F1
Betweenness (B)	0.90	1.00	0.76	0.86	0.82	1.00	0.59	0.74	0.90	1.00	0.93	0.96	0.88	1.00	0.69	0.81	0.70	1.00	0.29	0.45	0.60	1.00	0.07	0.14
Closeness (C)	1.00	1.00	0.76	0.86	0.93	1.00	0.83	0.91	0.75	1.00	0.41	0.59	0.82	1.00	0.53	0.69	0.98	1.00	0.95	0.97	0.72	1.00	0.34	0.51
Degree (D)	0.91	1.00	0.78	0.88	0.90	0.94	0.80	0.87	0.73	1.00	0.37	0.54	0.92	0.82	1.00	0.90	0.95	1.00	0.88	0.94	0.70	1.00	0.29	0.45
Eigenvector (E)	0.98	0.95	1.00	0.98	0.83	1.00	0.61	0.76	0.94	1.00	0.85	0.92	0.94	1.00	0.84	0.92	0.95	1.00	0.88	0.94	0.96	1.00	0.90	0.95
B and C	0.93	1.00	0.83	0.91	0.89	1.00	0.73	0.85	0.99	1.00	0.98	0.99	0.98	1.00	0.94	0.97	0.80	1.00	0.54	0.70	0.73	1.00	0.37	0.54
B and D	0.93	1.00	0.83	0.91	0.94	0.93	0.93	0.93	0.91	1.00	0.78	0.88	0.87	1.00	0.66	0.79	0.85	0.75	1.00	0.85	0.83	1.00	0.61	0.76
B and E	0.96	1.00	0.90	0.95	0.99	1.00	0.98	0.99	0.99	1.00	0.98	0.99	0.95	1.00	0.88	0.93	0.69	1.00	0.27	0.42	0.84	1.00	0.63	0.78
C and D	0.93	1.00	0.83	0.91	0.80	0.68	1.00	0.81	0.91	1.00	0.78	0.88	0.94	1.00	0.84	0.92	0.97	1.00	0.93	0.96	0.97	1.00	0.93	0.96
C and E	0.99	0.98	1.00	0.99	0.98	0.95	1.00	0.98	0.72	1.00	0.34	0.51	0.98	1.00	0.94	0.97	0.90	1.00	0.76	0.86	0.98	1.00	0.95	0.97
D and E	0.91	0.98	0.78	0.88	0.92	0.84	1.00	0.91	0.97	1.00	0.93	0.96	0.88	1.00	0.69	0.81	0.99	1.00	0.98	0.99	0.85	1.00	0.66	0.79
B, C and D	0.93	1.00	0.83	0.91	0.96	0.91	1.00	0.95	0.99	1.00	0.98	0.99	0.98	1.00	0.94	0.97	0.91	0.83	0.98	0.90	0.85	1.00	0.66	0.79
B, C and E	0.98	1.00	0.95	0.97	1.00	1.00	1.00	1.00	0.99	1.00	0.98	0.99	0.99	1.00	0.97	0.98	0.96	0.93	0.98	0.95	0.93	1.00	0.83	0.79
B, D and E	0.96	1.00	0.90	0.95	1.00	1.00	1.00	1.00	0.99	1.00	0.98	0.99	0.89	1.00	0.72	0.84	0.97	0.95	0.98	0.96	0.99	1.00	0.98	0.99
C, D and E	0.95	1.00	0.88	0.94	0.99	0.98	1.00	0.99	0.99	1.00	0.98	0.99	0.88	1.00	0.69	0.81	0.97	0.93	1.00	0.96	1.00	1.00	1.00	1.00
All	0.96	1.00	0.90	0.95	1.00	1.00	1.00	1.00	0.99	1.00	0.98	0.99	0.94	1.00	0.84	0.92	0.98	0.95	1.00	0.98	0.98	1.00	0.98	0.99

Execution Time

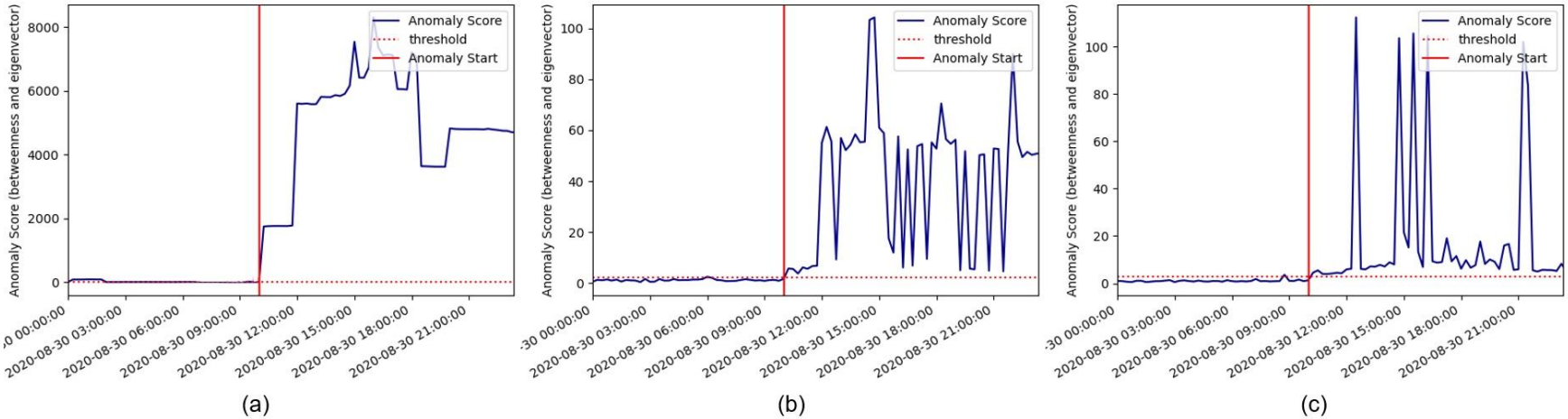
Centrality	WIDE			NWAX		
	Whole Network(s)	Louvain(s)	Leiden(s)	Whole Network(s)	Louvain(s)	Leiden(s)
Degree	0.01	0.17	0.01	0.05	0.22	0.01
Eigenvector	2.33	8.04	1.98	14.89	10.71	1.00
Closeness	1059.0	364.2	43.91	7587.66	562.81	22.18
Betweenness	8084.95	695.51	92.06	44394.09	1035.56	50.31

GNN Performance for Different Layers

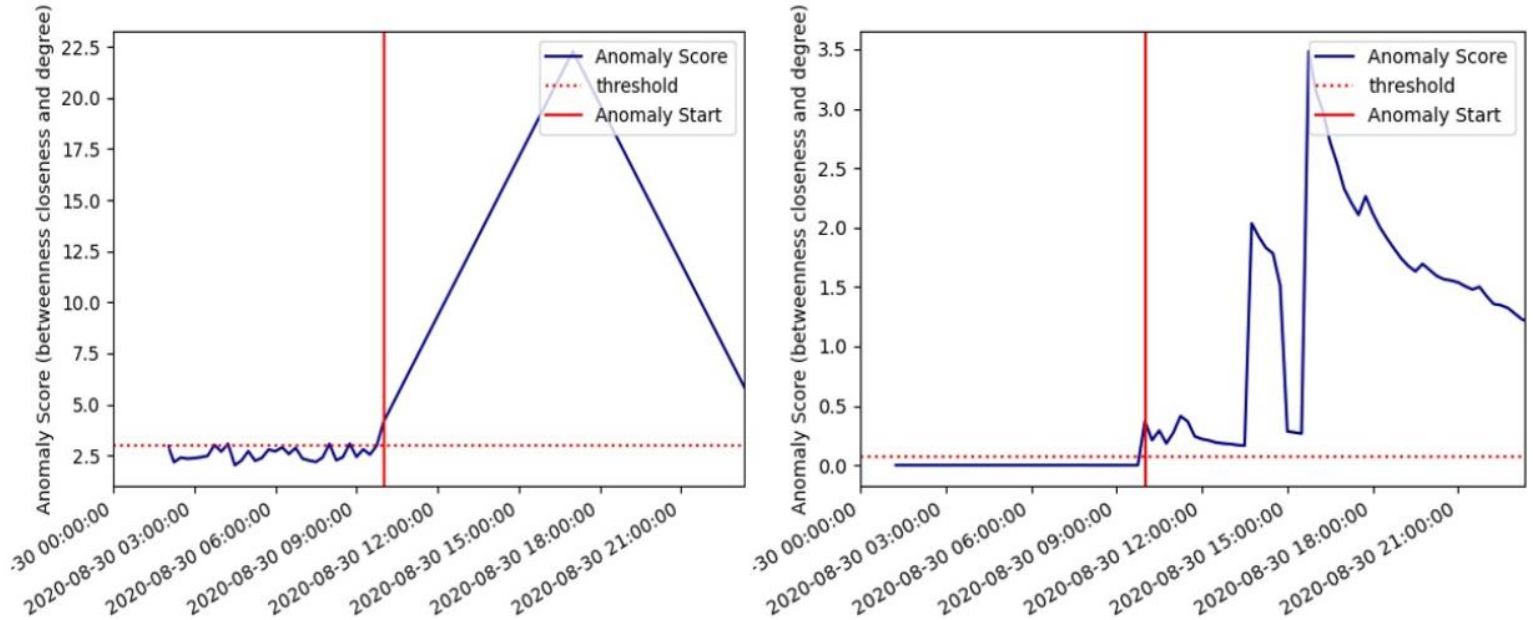
Layer	Accuracy	F1-score	Precision	Recall	Time/h
1	0.85	0.83	0.83	0.75	5.77
2	0.91	0.91	0.91	0.87	5.30
4	0.92	0.92	0.90	0.86	11.10
8	0.90	0.90	0.87	0.85	49.70
16	0.81	0.77	0.83	0.70	234.90

Whole Network, Louvain, and Leiden

Detection by WIDE using Betweenness & Eigenvector combination



SFMIX using whole network & Leiden



Conclusion

Effectively detect BGP anomaly by:

- Fusing multiple centrality measures to enhance detection performance.
- Partitioning (using Louvain and Leiden) significantly reduce computation time, with Leiden being more suitable for real-time analysis.

Future Work

- *Adaptive retraining pipeline* to significantly improve long-term robustness by aligning the model with the ever-changing routing environment.
- *Dynamically adjusting observation windows* based on streaming data could increase sensitivity to short-lived, high-impact anomalies and enable more timely mitigation responses.
- Detecting multiple types of anomalies through parallelism and federated learning.

Contact Details

winston.seah@ecs.vuw.ac.nz

WeChat

WhatsApp

LINE

For other related publications: