VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui

School of Engineering and Computer Science
Te Kura Mātai Pūkaha, Pūrorohiko

PO Box 600
Wellington
New Zealand

Tel: +64 4 463 5341
Fax: +64 4 463 5045
Internet: office@ecs.vuw.ac.nz

Modelling BGP Updates for Anomaly Detection using Machine Learning

Janel Yi Lin Huang

Supervisor(s): Winston Seah, Marcus Frean and Murugaraj Odiathevar
Submitted in partial fulfilment of the requirements for
ENGR489

Abstract

Border Gateway Protocol (BGP) anomalies, such as hijacking, is currently growing in trend due to limited detection capabilities. BGP is the backbone of the Internet that determines how traffic is routed through networks, also known as Autonomous Systems (ASes). BGP hijacking maliciously reroutes Internet traffic, causing Denial of Service (DoS) to major Internet Service Providers (ISP) or redirection attacks to Internet users. Current literature has proven to detect BGP anomalies using machine learning methods. However, the features used to train these machine learning models are node-level features that do not consider the network structure or relationships. Therefore, in this project, an approach to extract BGP updates to build a network graph is proposed. Then, centrality information is used as a feature to build an anomaly detection tool using machine learning. The proposed method has been validated on a BGP incident, BGP CenturyLink outage to show capability in detecting individual networks and defined group of network anomalies. Furthermore, determination of the anomaly source has shown to be capable in the proposed method.

Commented [WS1]: Abstract should not have citations.

Contents

Chapter 1

Introduction

With more applications delivered via the Internet, the greater the assumption is for the secure transmission of information. More specifically, the routing protocol of the Internet, Border Gateway Protocol (BGP) must be secure to transfer information between different networks. Operating as the backbone of the Internet, BGP is the routing protocol used for transferring information across different Autonomous Systems (ASes) on the Internet. AS defines a set of IP prefixes (Internet Protocol network addresses) that belong to a network or a group of networks. The presence of BGP routers is known through BGP updates transmitted amongst routers across the Internet.

Over the past years, there have been many incidents caused by anomalous BGP updates. BGP anomalies are caused by events such as hijacking. This is where attackers impersonate ASes by advertising false BGP routes to maliciously reroute Internet traffic. In worst cases, BGP hijacking led to a loss of connectivity for domains such as panix.com [1], an Internet service provider (ISP) in the United States of America (USA). Unintentional BGP hijacking, such as misconfiguration events, is more commonly seen. In a well-known example, the Pakistan Telecom incident, invalid BGP routes were advertised with the intention being to ban *youtube.com* [1]. However, a misconfiguration led to redirecting multiple ASes' *youtube.com* traffic to the Pakistan AS, causing a Denial of Service (DoS) for the Pakistan AS and a loss of connectivity of *youtube.com* for the affected ASes. Another BGP anomaly example is the global BGP CenturyLink outage induced by the misconfiguration of BGP routes [1]. BGP anomalies have caused severe outages for Internet users and revenue loss for many businesses across the globe. Therefore, the detection of BGP anomalies is crucial when routing Internet traffic.

1.1. Problem Statement

From literature reviews, the survey by Al-Musawi *et al.* [2] had identified to provide evidence in detecting past BGP anomalies using the methods of time series, statistical pattern recognition, historical BGP, reachability check and machine learning BGP. This research uses machine learning BGP as it allows for automation and optimization of model training to detect anomalies compared to other techniques. In existing BGP anomaly detection methods, features such as the number of BGP announcements, average AS path length and average edit distance are used. However, such features cannot be comprehensive because there is no consideration of the entire network structure and relationships, given that new types of BGP anomalies are always being introduced. As a result, these features are unable to detect a wide range of BGP anomalies, including anomalies that have not been seen before.

All BGP anomalies historically have shown changes in the network structure and thus, this project captures the change in structure to build a BGP anomaly detection method. This will allow the ability to capture a wide range of BGP anomalies. Instead of using ad-hoc features, a network structure is built using BGP updates. Then, centrality features are extracted to model the network structure. This is passed into two machine learning algorithms to detect anomalies. A machine learning algorithm such as Autoencoders is used to detect anomalies in the entire network. Autoencoders is a model that rebuilds inputs and aims to minimize the reconstruction error. The anomaly score is measured based on the reconstruction error where it has shown to be effective in the literature [2]. Another machine learning algorithm such as Univariate Gaussian (UG) is used to detect anomalies in individual networks. UG computes the anomaly score as the probability of each centrality in respect to the trained normal Gaussian distributions. Based on current literature, the use of centrality features to detect BGP anomalies has never been explored.

1.2. Project Goals

The goals of this project include:

1. Map a core router's BGP updates into a network graph.
2. Extract graphical features from the network graph to pass into a machine learning model to detect anomalies.
3. Determine the source of anomaly upon detection.

1.3. Contributions

This project offers the following contributions:

- Enhanced efficiency of searching network prefixes using trie data structures.
- Use of graphical features from the network to detect anomalies.
- Reporting the severity of the incident using an image which shows the number of ASes affected.
- Corroborating multiple core routers to detect anomalies.

1.4. Organisation

The remainder of the report is structured as follows:

- Chapter 2 will discuss the existing solutions to the problem to identify the differences to the proposed solution.
- Chapter 3 will discuss the design of the solution. This chapter involves a discussion of the various design decisions.
- Chapter 4 will provide details on the solution implementation.
- Chapter 5 will evaluate the solution to verify its correctness.
- Chapter 6 will conclude and identify future work to be conducted on the project.

Chapter 2

Background

This chapter provides background information regarding the importance and the key features of BGP. Moreover, the indicators of BGP anomalies and various detection methods will be discussed to determine the improvements that should be applied to detect BGP anomalies.

2.1 BGP

Commented [LH2]:

BGP is the essential backbone protocol that connects ASes within the Internet to enable network traffic to be routed through ASes. BGP routers are positioned on the border of each AS to deliver traffic. BGP operates on the links between such border routers to deliver traffic.

BGP routers deliver traffic by searching for the next-hop router (next router in the path to route the traffic) in its routing table to forward the traffic to. To allow the delivery of traffic, BGP routers must advertise their presence to other BGP routers. All BGP routers are initially unaware of the presence of other routers, thus, routing of traffic is incapable as the next-hop router cannot be determined to forward the traffic to its specified destination. The presence of BGP routers is advertised by BGP *bviews* and updates. BGP *bivews* are an infrequent periodic exchange (usually once every hour) of the routing table of a BGP router. Routes may change more frequently than hourly timeframes, hence, BGP updates are propagated (usually in 15-minute periods) to advertise routable paths. BGP *bviews* and updates are propagated router to router across the Internet. This is because it is infeasible to have every BGP router connected to every other BGP router in the Internet. The key attributes of BGP *bviews* and updates as noted in RFC4271 [3] include:

1. Withdrawn Routes – List of IP address prefixes for routes that should be withdrawn.
2. Announcement Routes – List of IP address prefixes for routes that should be added to the advertising node.
3. Source and Destination – Defines the origin of the path location.
4. AS_PATH – ASes of the path
5. NEXT_HOP – The IP address of the next router from the advertising BGP router to forward the message to the destination.
6. AGGREGATE – Optional and used along with ATOMIC_AGGREGATE. Includes the ASN and IP address of the router that originated the aggregated route.
7. LOCAL_PREF - Used by a BGP router to determine the exit path for the AS.
8. ATOMIC_AGGREGATE – Optional. Informs BGP routers that the advertising BGP router is using a less specific or aggregated route to a destination.

9. Network Layer Reachability Information – List of IP address prefixes that specify how to reach prefixes.

Routing policies that are used to route traffic can differ for ASes. This is due to political and/or efficiency reasons where the neighbours of each BGP router (BGP peers) must be formally established using contract agreements. However, advertisements of misconfigured or redirected routes are possible as observed in events such as DoS of YouTube by Pakistan and the BGP CenturyLink outage [1].

2.2 BGP Anomalies

BGP updates are classified as anomalous when the path contains an invalid AS or reserved or invalid IP prefix(s). The BGP update is also considered anomalous if the prefix advertised is from an invalid AS. When the AS-PATH is not geographically existent or the routing policy is uncommon [4], the BGP update is also considered anomalous [5]. A network may be anomalous through multiple BGP updates that reroute traffic in abnormal routes. Events such as outages and hijacks are examples where the traffic in a network is anomalous.

2.3 BGP Anomaly Detection Methods

There are currently various methods to detect BGP anomalies. Methods such as time series, statistical pattern recognition, historical BGP, reachability check and machine learning BGP can be used. However, such methods are not widely used in the industry. Hence, this section investigates the underlying problems of the current BGP detection methods to determine potential enhancements to detect BGP anomalies.

Commented [WS3]: This section can be the intro for 2.2 before 2.2.1. In its place at the start of the chapter, give a short intro on how BGP is used for routing traffic through the internet, its key features, updates, etc. Then, zoom in on the updates especially the messages that you use. As discussed, you need to also mention *bview*.

2.3.1. Time Series

The earliest method used by Prakash *et al.* [6] and Mai *et al.* [7] to detect BGP anomalies is time series which gained popularity as it can find characteristics of abnormal behaviour within a set of BGP updates collected within a period. This method can detect anomalies in high-intensity short bursts (hours) or sustained low intensity (months) of BGP updates from the affected ASes involved in the Slammer Worm attack [6, 7]. However, even with over two years of data used, only a limited number of incidents can be detected. This is because statistical features such as the number of announcements and message volume are used. Such features have a distinct behaviour for a specific type of anomaly, thus, a wide range BGP anomalies are unable to be detected, including anomalies that have not occurred before. This proves that statistical features are unable to detect all and new types of anomalies. Furthermore, an analysis for a copious number of BGP updates is required, thus, this method is unable to detect anomalies in real-time.

2.3.2. Statistical Pattern Recognition

Building upon the time series method, Huang *et al.* [8] and Deshpande *et al.* [9] used a statistical pattern recognition method that is successful in determining existing BGP anomalies as it can find relationships amongst each BGP update. By correlating events, this method can detect, identify and differentiate BGP node and link failures [8, 9]. The use of features such as AS-

path and edit distance were found to be useful to determine the behaviour of the network topology. On the contrary, new types of BGP anomalies are unable to be detected because such features do not consider the entire network topology as features are looked at independently in an instance. Furthermore, without the construction of a topology, relationships amongst AS-path and edit distance cannot be accurately described. Thus, this supports that the network topology must be built using BGP updates to accurately describe the relationships present in the graph. The anomaly source cannot be feasibly found as the static router configuration cannot be obtained due to political and security reasons [8]. Furthermore, this method is incapable of real-time detection due to the limited range of historical data used to correlate events [9].

2.3.3. Historical BGP

To counteract limitation of identifying new types of BGP anomalies, Haeberlen *et al.* [1] and Shi *et al.* [10] presents a whitelisting approach of using historical BGP data to determine the abnormality of new BGP updates. This approach validates new BGP updates using a history of Routing Information Base (RIB) and BGP updates. In contrast to the time series and statistical pattern recognition methods, this method was able to detect anomalies in real-time and identify the root cause [1, 10]. BGP prefix hijacks can also be detected, where an attacker impersonates a prefix belonging to a victim AS [10]. However, this method is unable to detect sub-prefix hijacks, link failures and indirect anomalies as the feature of prefix origin change is not comprehensive to reflect all changes in the network topology [1]. Furthermore, the usage of RIB in this method is not feasible as many ASes do not reveal BGP updates and their routing policies for security and political reasons.

2.3.4. Reachability Check

In contrast to methods of time series, statistical pattern recognition and historical BGP, the reachability check method utilised by Zheng *et al.* [11] and Hu *et al.* [12], gained attraction as it is less computationally expensive. This method uses the data plane (that forwards or processes packets in a router) to check the reachability of a prefix. This method can detect prefix hijacks in real-time as a single hop count calculation is only required for each BGP update [11]. However, due to the large number of attacks that do not change the reachability of prefixes, this method is incapable of detecting sub-prefix hijacks, link failures and indirect anomalies [11].

2.3.5. Machine Learning BGP

All prior methods mentioned have not proven to allow the capability for a method to find complex patterns in data that humans cannot discover. To counteract this limitation, machine learning BGP anomaly detection methods can be used. This is where a machine learning model is trained using existing BGP updates to detect anomalies within a network. This method is sought-after as the objective function for modelling abnormal and normal behaviour can be found and optimized automatically. Currently, this method uses statistical or node-level features such as the number of announcements, withdrawals or the average AS path length to determine anomalies [13, 14, 15, 16]. The ability to use historical data to automatically train the machine learning model enables the method to detect direct and indirect anomalies in past

events [13, 16]. However, the anomaly source cannot be determined using this method [14, 15, 16]. This suggests that the source and new anomalies are unlikely to be detected as the relationships within the network are not considered due to the usage of node-level features. This reinforces that graph-level features must be extracted to capture a wide range of BGP anomalies, including anomalies that have not been observed before.

2.4 Analysis and Findings

Methods of time series, statistical pattern recognition, historical BGP, reachability check and machine learning BGP present potential methods for BGP anomaly detection. The capability of anomaly detection using patterns recognised in historical data is evident as shown in Table 1. However, an inverse relationship between real-time detection and feature extraction in historical data is also observed in Table 1.

The current literature shows that BGP anomaly detection methods are incapable of detecting anomalies as unsuitable features are used to determine anomalies in the network. No network topology is constructed to accurately derive the relationships and model the structure present in the graph. Features are not comprehensive as there may be new anomalies that escape these features. However, such features still impact the network topology. Thus, capturing the network topology is vital. By capturing the features that represent the topology behaviour, a simple model can be used to learn the correlations of normal activity for determining anomalies. Hence, this research proposes a method to use BGP updates to build the network structure to extract graph-level features for determining anomalies in real-time. As the network topology is complex and high dimensional, a method such as machine learning BGP that can automatically detect correlations in such data is suitable in time and accuracy. Hence, machine learning BGP will be used as a detection tool to enable automatic identification of normal and abnormal features in real-time.

Table 1 Summary of BGP Anomaly 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.

Chapter 3

Design

The design of the anomaly detection method concentrates on graphical feature extraction from BGP updates. This includes the construction of the network graph and the selection of graph-level features. This chapter highlights the design considerations of time, budget and space concerning the potential use cases of the system to formulate the overall process of graphical feature extraction from BGP updates.

3.1. BGP Updates

To formulate the graphical view of the network, appropriate BGP update attributes must be selected. A BGP router will advertise BGP updates to its peers. A BGP update message as shown in Appendix A is used to advertise the feasible routes or withdraw unfeasible routes from the advertising BGP router.

The graph structure is formulated using the determination of each AS or node's direct connections. Each node's direct connections can be determined through BGP update attributes. Attributes must include the AS Number (ASN) to allow identification of the corresponding node within the graph.

As shown in the RFC4271 [3] BGP update attributes discussed in Section 2.1, the AS_PATH attribute can be used where each node present in the path represents a direct connection to the next node in the path. The source and destination nodes within each update can also be included as a node in the graph. Each node can have connections added or removed within each BGP update. Therefore, the announcement and withdrawal attributes should be used respectively to ensure that connections are added or removed appropriately from the corresponding nodes. The AS_PATH, source, destination, announcement, and withdrawal attributes are used to construct the network graph.

3.2. Graph Construction

The selected BGP attributes in Section 3.1 are used to construct the network graph which allows features to be determined within the graph.

3.2.1. Data Structures

After constructing the network, the graph must then be represented in a form to train a machine learning model. The representation must model the graph structure and the node relationships to capture a graph-like form of the network. To efficiently store the nodes, Hamiliton *et al*. [17] suggested neighbourhood aggregation, where nodes' neighbours can be aggregated into clusters or neighbourhoods, where each node would only have links to the

neighbourhoods instead of all neighbours. However, aggregation of neighbours requires clustering and node classification where initial seeds to train the classification may not be sufficient within the BGP updates.

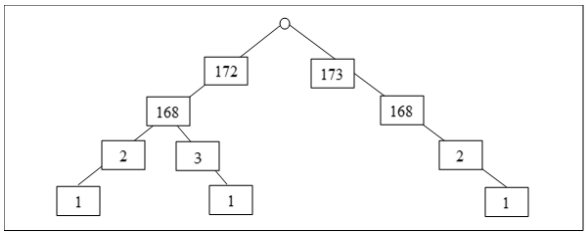

Figure 1 Example of a Computed Trie Structure for 172.168.2.1, 173.168.2.1 and 172.168.3.1

Another option to store the neighbours of a node that does not require clustering and node classification is to use adjacency lists, in which each node has a dictionary of its neighbours. Adjacency lists yield an efficiency of *O(max degree of the graph)* when finding all edges of a node. Furthermore, the use of adjacency lists requires less storage space in comparison to adjacency matrices (all nodes have an array of all connection points to other nodes). Therefore, adjacency lists can help increase efficiency and reduce the storage space of the network graph.

During announcements and withdrawals of BGP routes, a BGP router will add or remove routes respectfully. Therefore, the IP addresses of an AS must be present in each node to find withdrawal or announcement routes. Each node stores a dictionary of IP addresses as an AS may compose of multiple IP addresses. A naïve approach is to use a brute force search method where all nodes' IP addresses are searched when an announcement or withdrawal message is encountered. However, this yields an efficiency of $O(n^2)$, where n represents the number of nodes in the graph. A better approach is to use a trie structure to store all IP addresses with their associated nodes. For example, when storing the IP addresses of 172.168.2.1, 173.168.2.1 and 172.168.3.1, the trie structure will compute the tree as shown in Figure 1. By using a trie structure, this yields a better efficiency, *O(1)*. The pseudo code for building the graph using BGP updates is shown in Algorithm 1.

Algorithm 1 BGP Update Graph Creation

```
f = Read BGP Update
l = Read line in f
a                                   ▷ Announcing Node
while l is not null do
    if l == "ASPATH" then
        Add AS Path using l
    else if l starts with "TO" or l starts with "FROM" then
        n = add ip to node using l
        if l starts with "TO" then
            a = n
        end if
    else if l starts with "ANNOUNCE" then
        Add path, l to a
    else if l starts with "WITHDRAW" then
        Remove path, l from a
    end if
    l = Read next line in f
end while
```

3.3. Features

Graphical features are extracted from the constructed network graph and used in the machine learning algorithms to determine whether the BGP updates are abnormal. In this study, data

from the CenturyLink outage are used as it was a major BGP misconfiguration event in 2020 that brought down many Tier 1 ASes.

Graphical features such as the connectivity of nodes and node centrality can be used to determine anomalies. The connectivity of nodes can be used as the topology of the network changes significantly in BGP anomaly incidents. For example, over 1000 connections were changed during the BGP CenturyLink outage. However, using the connectivity of nodes as a feature is expensive in storage and computation. Clustering coefficients within the network or the level of overlapping neighbourhoods can also be used. Clustering coefficients measure the level at which nodes in the graph are clustered together. For example, in the BGP CenturyLink outage, nodes were more compactly clustered as redirection of traffic to other ISPs were required to compensate for the loss of the CenturyLink AS. However, clustering coefficients requires the definition of neighbourhoods which is inaccurate using simple methods such as k-means clustering. Although aggregation of neighbours by prefixes, articulation points, or spectral clustering [18, 19] can be used to summarise the connections of nodes, such methods lose individual network information and do not have sufficient information or are computationally infeasible due to the presence of over 60,000 nodes present in the network.

Commented [WS4]: Some elaboration/explanation of this statement is necessary to make it clearer.

Simple features can be extracted practically but lack in the ability to find a wide range of BGP anomalies. A simple feature such as the average AS-path length in the network can be used, where multiple AS paths can change in distance when a major AS is down. However, this feature is unable to detect anomalies in hijacking incidents where the average AS path length does not change significantly [6]. Another simple feature such as the number of changed AS paths in a period can also be used but the number of AS paths changed may be minimal in hijacking incidents where only a small number of victims are targeted [6]. This suggests that features that can be feasibly computed and combine the connectivity of nodes regarding the entire network must be used.

A computationally feasible graphical feature that can reflect a node's presence in the entire network is node centrality. Node centrality represents a node's position within a network based on a specified measure/metric. The BGPlay visualisation during the CenturyLink outage in Appendix B shows that the node centralities changed significantly. This is because a large portion of the traffic was rerouted, leading to several nodes having more or fewer paths routing through them, thereby changing their centrality values. Hence, node centrality can be used as a feature in the machine learning algorithm to detect anomalies because a large difference in the centrality of a node can indicate abnormal behaviour [19]. The key centrality metrics include:

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

It is impracticable to use the betweenness and eigenvector centralities as features as they are infeasible in computation. Betweenness centrality requires all possible paths to be

enumerated which is incomputable due to the presence of over 60,000 nodes in the network. The calculation of the eigenvector centrality requires the computation of the adjacency matrix which is unachievable due to the memory error generated from creating a large matrix size for over 60,000 nodes present in the BGP updates. Adjacency lists cannot be used as a matrix multiplication is required when computing the eigenvector centrality.

Algorithm 2 Centrality Feature Extraction

1: f = Read BGP Update
2: g = Get graph from f ▷ Contains each node with its immediate neighbours
3: N = Enumerate number of nodes in g
4: **for** $n = 1, \ldots, N$ **do**
5: $node$ = get n in $graph$
6: Calculate $paths$ as the shortest path length from each $node$ to its reachable neighbours
7: Calculate **CC** using number of $paths$, N and total length of $paths$
8: Calculate **DC** using number of immediate neighbours of $node$ in g and N
9: **end for**

In contrast, DC and CC were selected as graphical features as they are feasible to compute. DC is inexpensive to compute as it only requires enumerating the number of immediate neighbours of each node. CC is also computationally inexpensive as enumerating the distance to all reachable neighbours of a node is only required. The distance to all reachable neighbours is calculated using Dijkstra's algorithm to allow efficient computation of distances. Although the A* algorithm [18] can instead be used for greater efficiency, the generation of a heuristic is dependent on the geolocation of the nodes which is unobtainable due to insufficient information in BGP updates. The pseudo code for extracting centrality features is shown in Algorithm 2.

3.4. Summary

The design constraints presented in this chapter show that CC and DC are suitable in time and computation to detect anomalies within a network. Suitable BGP attributes are used to build the graphical structure of the network. The graphical structure of the network utilises adjacency lists and trie data structures to enhance the efficiency of the solution. A corresponding flow diagram of the implementation is also shown in Figure 2. The implementation of the proposed design will be discussed in the next chapter.

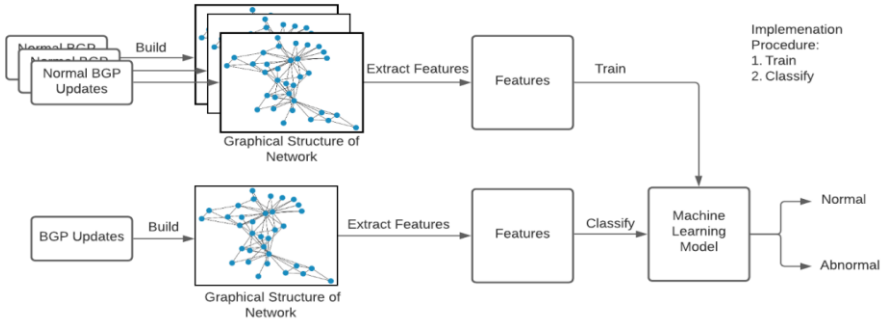

Figure 2 Workflow of Proposed Anomaly Detection Method

Chapter 4

Implementation

This chapter discusses the implementation of feature extraction using the designs from Chapter 3. Implementation of the machine learning models to process the extracted features for anomaly detection is also discussed. Details of the tools, parameters and implementation considerations are discussed in this chapter.

4.1. Processing Data

4.1.1. Programming language

There are multiple programming languages to process BGP updates into a graph-like structure. A programming language such as Java can be used but it is incapable of processing a copious amount of data. For example, when processing a single BGP update using Java, a Java Heap Space Memory exception was raised. This is because Java is an object-orientated language that stores additional attributes such as the object type and its features. A more suitable language is Python which is a general-purpose language that does not store the object type. This leads to less memory stored by the program, reducing the likelihood of Out-Of-Memory exceptions.

4.1.2. Input Data

To build the structure of the network graph, BGP *bviews* and updates were used. Firstly, *bviews* are exchanged amongst BGP routers. Then, BGP updates are exchanged after a BGP *bview*, that notify other routers of routable paths. Hence, a network graph is constructed after one *bview* is received. The next subsequent BGP update is built upon the graph generated from the *bview*. Each BGP update that occurs after is built on top of the graph from the previous update. Events occurring before and after the anomaly event are extracted to produce respective normal and abnormal features.

4.2. Generating Features

There are more than 60,000 centralities that can be passed into the machine learning algorithm. However, this is infeasible and not applicable in many cases where AS owners may only want to detect anomalies within their networks. Therefore, the New Zealand core router (AS38022) and its neighbours (up to two hops away) were used as input features which yielded 4000+ nodes. To further reduce the number of nodes and capture the necessary information, non-

articulation points were eliminated from the graph. Articulation points are crucial points in a graph that if removed would disconnect the network.

To compute the articulation points in the graph, there were over 60,000 nodes to be processed. Hence using the recursive method of the articulation points algorithm led to exceeding the limit for recursive call-backs. Therefore, the iterative version of the articulation points algorithm was used. The iterative version used a stack that had a larger memory space and did not involve any recursive call-backs.

The centrality values were calculated based on the entire network using the *Networkx* [20] library, instead of subnetworks. The disadvantage of calculating the centralities using a subnetwork is that information within the graph can be overlooked. Subnetwork features can be an incorrect representation of the entire network. Each centrality value is calculated using the following formulas:

i. $DC(u) = \frac{n}{N-1}$, where n represents the number of immediate neighbours of a node u and N represents the total number of nodes in the graph [19].

ii. $CC(u) = \frac{n-1}{\sum_{v=1}^{n-1} distance(u,v)} \frac{n-1}{N-1}$, where n represents the number of reachable neighbours of a node u and N represents the total number of nodes in the graph [21].

To ensure that the result is proportional for graphs of varied sizes, the centrality values are normalised by the number of nodes in the graph [19].

The CC formula suggested by Wasserman *et al.* [21] is used as it ensures that the CC for different sized network graphs can be comparable. For example, in the original formula of CC, as shown in Appendix C, the CC is only scaled by the number of reachable neighbours of a node. There is no consideration on the total amount of nodes in the graph. This means that the CC would not be weighted equivalently to the size of the network graph, where the same CC (computed using the original formula in Appendix C) would compute the same anomaly score in a large and a smaller sized network graph. This is incorrect as the anomaly score for that AS should be larger for a smaller sized network graph, as the impact of that AS is larger in a smaller sized graph with the same reachable neighbours [21]. Reachable nodes do not correspond to the total number of nodes in the graph as some nodes may be disconnected from the graph. For example, as shown in the network graph before, during and after the BGP CenturyLink outage in Appendix B, some nodes are not connected to the graph. Hence, the CC formula as suggested by Wasserman *et al.* [21] which incorporates the scaling ratio of a node's reachable neighbours and the total number of nodes in the graph is used.

Commented [WS5]: Your explanation is graph-theoretic, i.e. from the theory viewpoint although you try to relate to examples shown in the Appendix. You should try to explain from the perspective of how the two different ways of computing Closeness Centrality affect the way you determine/detect anomalies and hence your choice. (Page 18 still has space to accommodate more discussion.)

Commented [WS6]: Your explanation is graph-theoretic, i.e. from the theory viewpoint although you try to relate to examples shown in the Appendix. You should try to explain from the perspective of how the two different ways of computing Closeness Centrality affect the way you determine/detect anomalies and hence your choice. (Page 18 still has space to accommodate more discussion.)

4.3. Machine Learning Models

To detect anomalies in the extracted graphical features, multiple machine learning algorithms can be used. To detect anomalies, the machine learning model must identify the main patterns in normal data. The main patterns can reveal outliers or anomalies in the dataset. A commonly used anomaly detection method that captures complex relationships amongst the datapoints

in the dataset is Autoencoders [22]. To detect individual network anomalies, a less computationally expensive method than Autoencoders, such as UG, is used [23].

4.3.1. Autoencoders

Autoencoders are 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.

4.3.1.1 Training parameters

Multiple training parameters are used to define the Autoencoders model. A training parameter is epochs that determine the number of times all the training instances are used in the training process. The error between the inputs and outputs must minimise or converge during the training process. However, the training process must not cause overfitting where the error in unseen inputs increases as the number of epochs rises. Therefore, the number of epochs chosen was 100 as this allowed convergence as shown in Figure D.1 and did not overfit the model.

Another training parameter, batch size is used. The batch size must not be too small (inability to converge to a global optima for the reconstruction error) or be too large (poor generalization due to inefficient stochastic gradient descent). Therefore, a batch size of 80 was selected as it averaged the error for each batch and applied the corresponding adjustments to the NN weights. As shown in Figure D.1, the convergence of the error rate was achieved.

A threshold is used by the machine learning algorithm to determine whether an instance is abnormal. Hence, the threshold can be defined by the maximum error from the validation data which is defined as two days before the anomaly incident. For example, in reporting the entire network detection from the New Zealand core router, Figure D.2 shows that normal behaviour has a maximum error of 0.036. Therefore, any value that exceeds this error should be defined as anomalous.

Each feature's importance may be different according to the feature value. However, each feature should be considered as equal importance as a difference in any node in the network can indicate abnormal behaviour. Therefore, each feature, denoted by x, is normalised using the formula, $feature = \frac{feature-mean}{standard\ deviation}$. Figure D.3 and D.4 show that after normalizing the values, the anomaly score is more prominent in the anomaly period from 08-30-2020 10:00 UTC onwards.

4.3.2. Univariate Gaussian

The main disadvantage of Autoencoders is that the detection of anomalies for a specific AS is not addressed. Identification of problematic ASes is useful for network administrators 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. Hence, UG is used. UG is a probabilistic

Commented [7]: This is circular: "identification is useful to pinpoint". Identify and pinpoint are the same thing. "Problematic" is the same as "affected".

method that models a distribution directly as a PDF (probability density function), where a low PDF indicates an abnormal AS.

4.3.2.1 Parameters

Building a UG (one component) or a Gaussian Mixture Model (GMM) (more than one component) involves selecting the number of components. Too few components, the model will under-fit (actual clustering structure in the data will be missed); too many, the model will over-fit (inability to detect anomalies in a new testing set). An incorrect number of components can lead to incorrect classification of normal and abnormal centralities. Hence, the number of components is predicted by gathering the anomaly scores on the day of the anomaly event. As shown in Figure 3, the model with components one, two, and three show an increase in the anomaly score during the anomaly event and are similar in results, where neither under- nor over-fitting is observed. However, it takes significantly less time to compute one distribution than two or more distributions. This is because there are fewer Gaussian distributions to optimize, thus, the number of epochs required will be less. Hence, the number of components chosen is one, where the UG model is selected. This provides the fastest solution in comparison to GMM which has two or more components.

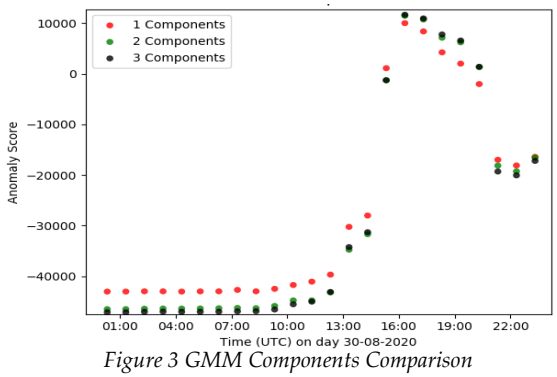

Figure 3 GMM Components Comparison

The PDF for a data point that consists of all the features at a period can be calculated using the product of the estimated probabilities for all test features. However, as there are many probabilities to be summed, this can result in an extremely small probability that may not be representable due to the limits in the decimal length floating-point numbers. The anomaly score is then the sum of log probabilities over all features. The equation, $Anomaly\ Score(v) = \sum_{i=0}^{N} \log\left(Probability(v_i)\right)$, where N represents the number of features and v represents a single feature is used to calculate the anomaly score for a data point. Individual anomaly scores for each AS can be calculated using the log-likelihood of each corresponding feature.

4.4. Summary

The implementation details such as parameters, algorithms and libraries discussed in this chapter reflect the time, computation and correctness considerations presented in Chapter 3. The evaluation of the implemented artifact will be discussed in the following chapter.

Chapter 5

Evaluation

Following the implementation of graphical feature extraction, an evaluation of the correctness of the proposed anomaly detection method using the extracted features is carried out. The correctness of the model is evaluated against the detection of abnormal behaviour during the BGP CenturyLink outage from 08-30-2020 10:04 (UTC). This helps to determine whether the learnt model can correctly classify anomaly incidents in the dataset. Anomaly detection for the entire network and specific ASes for the New Zealand (NZ), Japan (WIDE) and Serbia (SOXRS) core routers on the day of the anomaly event will be evaluated using Autoencoders and UG to evaluate the correctness of the proposed anomaly detection method. The importance of the evaluation allows validation of the detection method and identify the suitability to real-world applications.

5.1. Evaluation Method

To evaluate the accuracy of the detection method, experiments were run using data for the entire network (up to two hops away from the NZ core router). This enables the system to identify network-wide anomalies. This also enables earlier anomaly detection capabilities as BGP updates are propagated sequentially on the Internet, hence anomalies occurring in one part of the Internet can indicate a potential spread of BGP anomalies.

Although a network-wide analysis can indicate anomalies for the entire network, the source or infected ASes must be identified to ensure that traffic is not routed to such networks. This helps to prevent DoS for Internet users. Individual network monitoring can also be more suitable to ISPs who do not require or have sufficient computing power or resources to monitor an entire network. In consideration of the BGP CenturyLink outage which involves AS38022 (NZ core router) and AS3561 (BGP CenturyLink router), experiments will be run using BGP updates for each AS. This helps to evaluate the detection capabilities for the anomaly source and its neighbours.

To determine the correctness of the detection method, all experiments encompass a plot of the anomaly scores on the day of the anomaly event from 08-30-2020 00:00 UTC to 23:59 UTC. The results should expect a rise in the anomaly score on the day of the anomaly event. As the data are unlabelled, it is possible that the network was unstable before the anomaly event, hence a rise in the anomaly score before the expected anomaly breach is possible.

Commented [WS8]: The word "data" is plural.

5.2. Results

A series of experiments were run using data of the entire network and the individual networks of AS38022 and AS3561. Experiments included an evaluation of the DC and CC features used for determining the anomaly score.

5.3. Entire Network Anomaly Detection

Determination of whether an entire network is anomalous allows ISPs to have a generalised view of a network's stability. This allows faster determination of anomalous behaviour in comparison to monitoring multiple individual networks. As this project is New Zealand based, the anomaly for the entire network is defined by the NZ router and its neighbours (up to 2 hops away). Detection for other countries can also be achieved by changing the definition of the entire network.

5.3.1 Closeness Centrality

Graphs from Figures 4(a) and 5(a) show a rise in the anomaly score before the expected time breach. This is because the network is unstable during the anomaly period as shown in Appendix B, with a significant change in reachability distances for many nodes in the network.

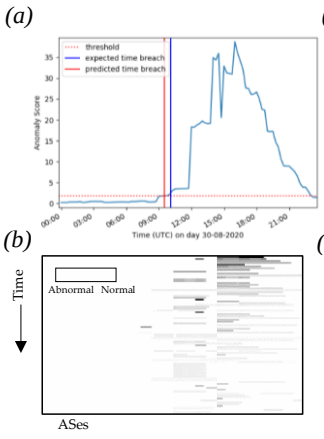

Figure 4 Entire Network of NZ CC

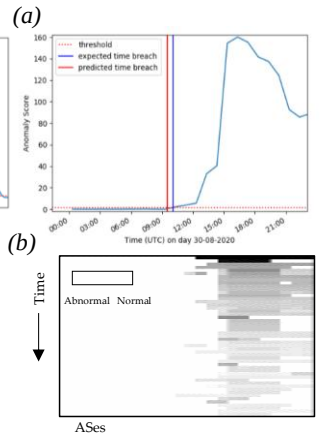

Figure 5 Entire Network of WIDE CC

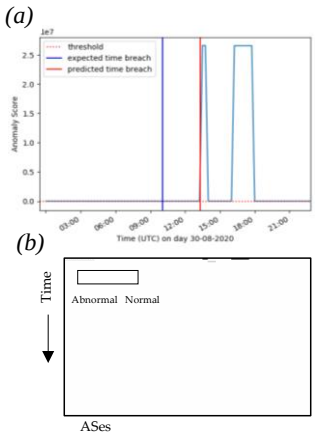

Figure 6 Entire Network of SOXRS CC

As shown in Figure 6(a), a later detection is observed for SOXRS as it is further away from the source of the incident in comparison to WIDE and NZ. BGP updates are transferred router to router, hence routers that are further away will experience a delay in receiving update messages. This suggests that to increase the detection accuracy of an anomaly event, multiple core routers' views are necessary to gain a wider picture of the activity on the Internet. The selection of core routers to monitor can be determined by the geolocation (where a sparse set of core routers can enable better anomaly detection capabilities around the world) and the trustworthiness of the country (using a contract agreement).

Network administrators need to determine the severity of the incident using the number of nodes affected. Severity level determines the privilege escalation procedure where the appropriate number of resources must be assigned to remediate the incident. However, the graphs as shown in Figures 4(a), 5(a), and 6(a) are unable to reflect the severity of the incident, thus corresponding severity images which show the number of ASes affected from the view of the router is shown in the images of Figures 4(b), 5(b), and 6(b). Severity images from Figures 4(b) and 5(b) show that NZ and WIDE routers can detect that a large number of nodes are affected. However, Figure 6(b) shows that only a small number of nodes are affected as the visibility level of the affected nodes is insufficient in comparison to NZ and WIDE. This helps to reinforce that multiple core routers should be used to detect anomalies within the Internet.

Commented [WS9]: May be better to give them sublabels, e.g. (a) for graphs and (b) for the images.

5.3.2 Degree Centrality

Figures 7(a) and 8(a) show an increase in the anomaly score before the expected time breach. Like CC for SOXRS, a detection later than the expected time breach is observed in Figure 9(a) as it is further away from the source of the incident. However, unlike the severity images as shown in Figures 7(b), 8(b), and 9(b), only the NZ core router shows that a large number of ASes are affected in Figure 8(b). Figures 7(b) and 9(b) do not indicate that a significant number of ASes are affected. This is due to the nature of DC which can only reflect anomalies for the immediate neighbours of a node. Both WIDE and SOXRS are not directly involved in the incident, hence the visibility of the number of ASes affected is not significant for both routers. Although this may suggest that DC should not be used to detect anomalies as it can yield false negatives, DC will unlikely yield false positives, as a significant change in immediate neighbours of a node can indicate instability in a network. Furthermore, for applications where there are constraints on computing power, an inexpensive computation will be ideal when using DC, as it has a complexity of $O(n)$, where n represents the number of nodes in the graph in comparison to the complexity of CC, $O(ne)$, where e represents the number of edges in the graph.

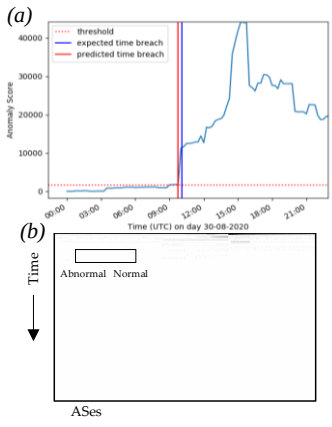

Figure 7 Entire Network WIDE DC

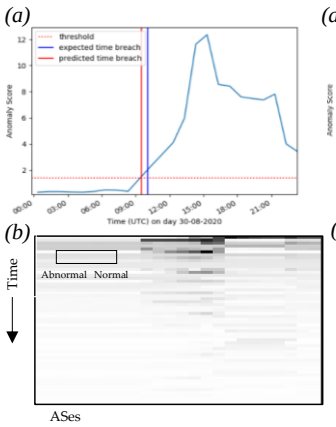

Figure 8 Entire Network of NZ DC

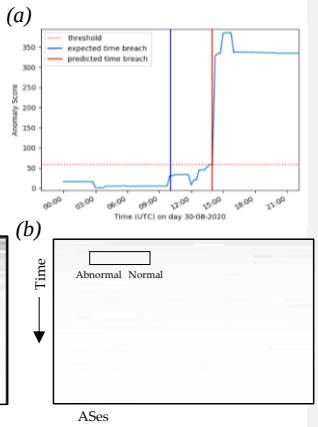

Figure 9 Entire Network of SOXRS DC

5.4. Individual Network Anomaly Detection

If the entire network is deemed anomalous using Autoencoders, specific ASes should be investigated to determine whether they are affected or is the source of the BGP incident. Individual anomaly scores for each AS are determined through DC and CC using UG. The gathered anomalous ASes can be used to update the routing table of ISPs such that traffic is not routed to/through them, thus minimising the chance of disruption for Internet users.

5.4.1 AS38022

Figures 11, 12, and 13 show a rise in the anomaly score before the expected time breach as the network was unstable. A later time breach is predicted for NZ as shown in Figure 10 as the anomaly event did not stem from AS38022. The distance from AS38022 to its reachable neighbours did not change until the error from the source of the anomaly, AS3561, propagated through the Internet. As AS3561 is a trusted network peer of AS38022, the error that is propagated by AS3561 is deemed normal when transferred to AS38022. However, an earlier time breach is predicted from WIDE as it can view the anomalous activity between AS38022 and AS3561 from an outsider's point of view.

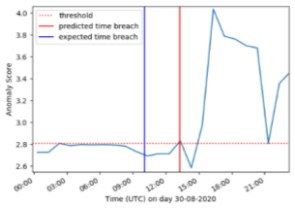

Figure 10 NZ AS38022 CC

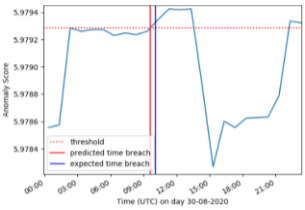

Figure 11 NZ 38022 DC

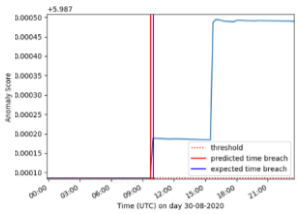

Figure 12 WIDE 38022 CC

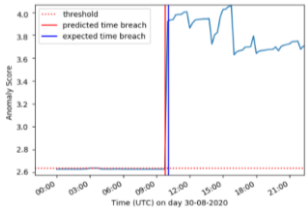

Figure 13 WIDE 38022 DC

An earlier detection for DC as shown in Figure 11 is due to the number of immediate neighbours of AS38022 changing significantly during the anomaly event as shown in Appendix B. This is because a significant change in the immediate neighbours of a node can instantly be reflected in a single BGP update. A significant change in the distance to all the reachable neighbours of a node can change in multiple BGP updates as such updates are propagated router to router. This suggests that DC can detect anomalies faster than CC. However, computation of CC is recommended as it can indicate whether all the reachable neighbours of a node are affected by the anomaly event. This is useful for network administrators to determine the severity of the anomaly incident and thus, allocate the appropriate amount of resources to remediate the incident.

No anomaly score is generated from SOXRS for AS38022 as it does not have AS38022 within its routing table during the detection period. This is because SOXRS is geographically further away from AS38022 and did not have any traffic that travelled to AS38022 within the detection period. This suggests that multiple core routers should be used for detection to allow anomalies to be discovered throughout the Internet.

5.4.1 AS3561

The anomaly source can be determined using the time at which the anomaly threshold is breached. For example, NZ detected abnormal behaviour at least 1 hour before the anomaly event as shown in Figures 14 and 17. This is because AS3561 is an immediate neighbour of NZ, hence it can detect abnormal behaviour before WIDE and SOXRS. In the experimental results, AS3561 is the earliest breached AS in the BGP CenturyLink outage which suggests that it is the source of the incident. This suspection is correct as the BGP CenturyLink outage stemmed from its own AS, which is AS3561.

Depending on the source of the anomaly, different core routers will detect anomalies at different times. This is because BGP works by transferring update messages router to router. Hence, routers that are further away from the announcing router will hear the BGP updates at later stages. For example, as SOXRS is geographically further away from AS3561, the detection of the anomalous activity is 2 hours after the anomaly event as shown in Figures 16 and 19. WIDE is closer to AS3561 than SOXRS, hence an earlier detection is observed as shown in Figures 15 and 18. However, a later detection is observed for the CC from WIDE as shown in Figure 15. This is because the distance to the reachable neighbours of AS3561 changed after the error had propagated within the network. This suggests that CC can also be used to detect whether the error has been propagated to the reachable neighbours of a node. As major ISPs of the Internet are interconnected, the reachable neighbours of a core router will consist of the Tier 1 ISPs, producing a generalised anomaly indication for the Internet.

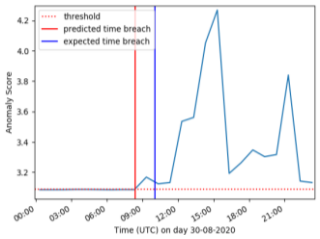

Figure 14 NZ AS3561 CC

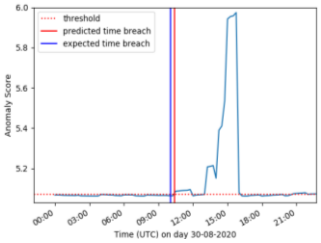

Figure 15 WIDE 3561 CC

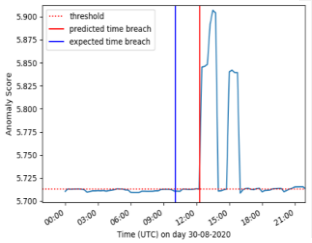

Figure 16 SOXRS AS3561 CC

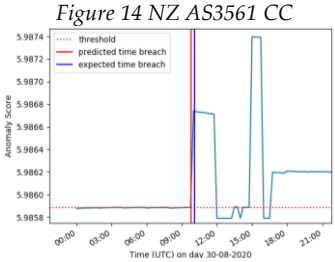

Figure 17 NZ AS3561 DC

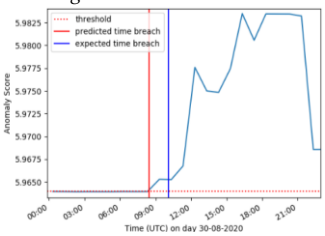

Figure 18 WIDE 3561 DC

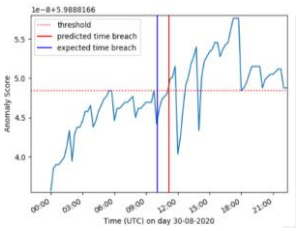

Figure 19 SOXRS AS3561 DC

5.5. Discussion

5.5.1 Evaluation Results

The anomaly detection for the entire network uses Autoencoders which shows to be successful in Section 5.3. In particular, Autoencoders can detect abnormal behaviour for routers that are located further away from the anomaly source. As shown in Figure 20, Autoencoders can

detect that there is an increase in the anomaly score from the point of view of SOXRS using DC during the anomaly incident. SOXRS can only detect that a small number of ASes are affected as it is located further away from the source of the anomaly. Detection is capable using Autoencoders as it can detect anomalies despite having a small number of nodes affected due to the usage of one distribution for all ASes.

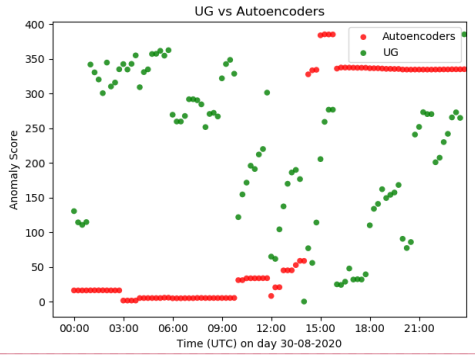

Figure 20 SOXRS DC using Autoencoders and UG

A full joint UG can also be used to detect anomalies in the entire network. For example, as shown in Appendix E, UG shows similar results in the prediction of the anomaly breach like Autoencoders, where the predicted time breach is before the expected time breach from the NZ and WIDE routers, and after the expected time breach from the SOXRS router. Similar results are shown as the covariance matrix resolves the limitation of UG, where it is unable to find correlations between features like Autoencoders as they are passed in separately. For example, UG does not show an expected increase in the anomaly score without computing the covariance matrix as shown in Figure 20. This is because Autoencoders considers all features as a singular distribution as shown in Figure F.1, whereas UG considers each feature as a singular Gaussian distribution as shown in Figure F.2.

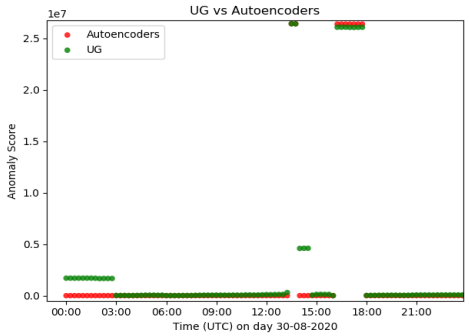

Figure 21 SOXRS CC using Autoencoders and UG

When there are resource constraints, UG would be preferred over Autoencoders as it is computationally faster. This is because features only need to be looked at once to train the model as opposed to multiple times when using Autoencoders. Additionally, UG would be more suitable to adapt for different ISPs as it does not require re-selection of multiple training parameters. Only the number of components in the dataset is required to be re-selected. Investigation of the number of components in the centralities also facilitates data analysis where the number of network structures present in the dataset can be predicted. This enables ISPs to investigate changes made to a specific network structure to determine anomalies and the reliability of the network.

Autoencoders, on the other hand, requires a re-selection process as it is a black box method. However, UG is not a stable model that optimizes the anomaly detection function.

Commented [WS10]: May be better to replace these figures (i.e. 22 and 23) with one of the those in the appendix (e.g. F.1, figures in Appendix G, etc.) that you reference to support your analysis. Figure F.1 is referred to a few times in the discussion and therefore would be worth putting it here in the main text.

For example, when comparing Autoencoders and UG's results of DC from SOXRS in Figure 9(a) and E.4 respectively, there is an instability of the anomaly score when using UG. This is because a Gaussian model captures (pair-wise) covariances, that is, second-order structure in the density function over features. In principle, Autoencoders embodies a richer representation of density, about to model higher-order structure rather than being limited to covariances. The correlations within DC are low in comparison to CC. For example, when using UG as an independent Gaussian per feature as shown in Figures 20 and 21, an expected rise is only observed for CC as correlations are initially present in the features before being passed to the machine learning model. Low correlations are observed for DC because only the immediate neighbours are used in computation as opposed to all reachable neighbours, as applied by CC. Furthermore, when insufficient visibility is encountered such as in the case of SOXRS, the covariances amongst features will be harder to find as there are fewer features to compute the covariances. Hence, this suggests that UG should be used for routers that are positioned geographically closer to other routers such as London and Singapore. In contrast, Autoencoders should be used for routers that are placed geographically further away from other routers, such as SOXRS, to enable anomaly detection.

Anomaly detection for the entire network shows the advantages and disadvantages of using DC and CC. Both DC and CC show similar results, with a similar time of anomaly detection. However, DC is cheaper to compute than CC as it requires the calculation of nodes' immediate neighbours instead of reachable neighbours. This suggests that DC should be used to alert anomalies in the entire network. CC, however, performs better than DC in determining the severity of the incident, as shown in Section 5.3.1. A large number of ASes were indicated as anomalous which supports the high number of affected ASes shown in the BGP CenturyLink outage in Appendix B. This is because CC takes into account the reachable neighbours of each node. This means that an outage of a major ISP will be reflected in more ASes when using CC in comparison to DC which only considers the immediate neighbours of nodes.

Another disadvantage of DC is that it can have a longer training period than CC. As shown in the entire and individual network monitoring of DC, the anomaly score is unstable in comparison to CC, where rapid peaks and dips can be seen. This is because DC can have a much more significant change in immediate neighbours in comparison to the reachable neighbours of a node used in CC. The result of rapid changes means that the machine learning model must train for a longer time to optimise the weights used in the computation. This also suggests that DC is more likely to encounter false positives than CC as it can experience rapid changes in the network.

To detect anomalies in real-time, such a capability from the proposed system is possible for NZ, WIDE and SOXRS. It takes on average 5 minutes and 26 seconds to generate features for one 15-minute period 40KB BGP update and one hourly 10MB *bview*. This means that the proposed system can compute the anomaly score for the defined entire network within 15 minutes when a BGP update arrives, thus the real-time detection capability of the system is possible.

Commented [WS11]: While I have not read the conclusions yet, such guidelines/recommendations would be very useful to emphasize in the conclusions chapter later. A similar recommendation would be the one on computational requirements.

Commented [WS12]: Too many "as" in a single statement; better to rephrase.

CC and DC reflect capabilities in the entire network and individual network anomaly detection, but further enhancements to allow a faster and more confident anomaly detection method could be achieved using traffic link analysis. The amount of traffic generated in DoS attacks such as Slammer Worm and WannaCry can generate a copious amount of traffic. By analysing the traffic alongside the generated network topology from BGP updates, the level of service disruption can be determined. As a result, ISPs can quickly identify current or potential bottlenecks and determine whether additional resources must be leveraged to combat the surge of Internet traffic.

During the evaluation, the selection of training datasets was important to determine the correctness of the model. For example, the network graph at certain periods, with a difference of years or months could be significantly different. This is because the network can change significantly within monthly periods due to the new establishment of peers between ISPs. However, the network does not change rapidly within weeks as BGP peers require contract agreements to be formally established. Hence, it is recommended that the model in practice should be trained using data at least two weeks from the current date. Using less than two weeks of data is inadequate as there are insufficient training data provided for the machine learning models to find correlations or define a correct distribution for detecting anomalies. If an anomaly incident occurs, the model must be retrained or should ignore the anomaly data. This suggests that further research on implementing online learning should be conducted to enhance the detection accuracy of the model.

5.5.2 Limitations

There are limitations presented in the evaluation method. Due to the time and processing power constraint, further evaluation on major core routers such as London and Singapore are unable to be conducted. Such routers contain gigabytes of data for each 15-minute BGP update in comparison to kilo or megabytes of data in SOXRS, NZ and WIDE. This is because routers such as London and Singapore are the heart of BGP exchanges of Europe and Asia respectively. This suggests that further improvements to increase the computation speed of the proposed detection method can be achieved using distributed processing where BGP updates are processed in multiple workloads.

Another limitation of the evaluation method is that the evaluation was limited to the NZ core router and its neighbours (up to 2 hops away). This means that anomalies in other parts of the Internet are not evaluated. A limit on the size definition of the entire network is required as the memory and computation requirements are infeasible when considering more hops. Hence, further investigation on using neighbourhood aggregation to detect anomalies on the Internet can be conducted to detect Internet-wide anomalies. Other anomaly incidents, such as Slammer worm, should also be evaluated to further determine the correctness of the proposed method.

Commented [WS13]: The limitations discussion can certainly be a good subsection within Section 5.5. See also my earlier comment about dividing the Discussion section into smaller subsections.

The entire network and individual network capabilities reflected in this chapter present the detection capability for BGP anomaly events. Further improvements to the evaluation method using traffic link analysis, network aggregation, and BGP updates from major core routers should be investigated to determine potential enhancements to the system. Future work recognised for the system will be discussed in the following chapter.

Chapter 6

Conclusions and Future Work

This chapter concludes and states the overall outcomes of the design, implementation and evaluation chapters presented above. Future work recommended to be examined will be discussed to enhance the proposed system of network anomaly detection.

6.1. Conclusions

In this project, a network anomaly detection method using graph-like features is proposed. The detection method can help detect BGP anomaly incidents such as misconfiguration events.

The first goal of the project where BGP updates are mapped into a network graph is achieved. BGP update attributes such as source, destination, withdrawals, announcements and AS path are used to construct the nodes and links of the network graph. Construction of the network graph enables graph-like features to be extracted and analysed for anomalies. The use of a trie data structure to store the IP address to node mapping is added to enhance the efficiency of the address search.

The project's second goal where graph-like features are extracted from the constructed network graph is achieved. Extracted features include DC and CC which reflect the connectivity of nodes that change significantly during a BGP incident. Such features are also feasible in time and complexity to allow practical uses for ISPs. The extracted features prove to be successful in detecting anomalies for the entire network and individual networks during a BGP incident as shown in Chapter 5. A further severity image showing the number of affected ASes is also produced to allow the determination of the incident severity. Three core routers (NZ, WIDE and SOXRS) are also used to evaluate the proposed detection method to determine a network-wide anomaly detection capability.

Commented [WS14]: Is there a more definitive name for this kind of image?

The third goal of the project where the determination of the anomaly source is achieved through the predicted time breach of the anomaly threshold. As shown in Chapter 5, breaches of the anomaly threshold can be at various times. Therefore, the earliest time breach for a specific network can indicate that the anomaly incident stemmed from such a network. By determining the source of the anomaly, prevention strategies such as re-routing traffic to avoid passing through the anomalous ASes can be conducted, thus, decreasing the chance of DoS caused for Internet users.

6.2. Future Work

In addition to the completion of the project goals listed above, future work recommended to be examined to improve the proposed system is listed below:

1. Distributed Processing – Distributed processing is where computational workloads are balanced amongst multiple processors. To enhance the computational speed of extracting graphical features from BGP updates, distributed processing with classical multilateration of BGP *bview* and update sets can be used to process data-intensive routers such as London and Singapore [24].
2. Network Neighbourhood Aggregation - To scale the entire network to the Internet, networks can be aggregated into neighbourhoods to reflect a more condensed network. This enables earlier anomaly detection within the Internet. Further investigation in using existing datasets to train a machine learning model to automatically aggregate networks within the Internet is recommended to be examined.
3. Traffic Link Analysis – Alongside BGP updates, additional data such as the amount of traffic that is travelling in the network links can be used to determine anomalies. This helps to better detect Distributed DoS (DDoS) attacks which are commonly seen in [25] as the centrality of nodes cannot reflect the amount of traffic that is travelling between ASes.
4. Variational Autoencoders – The proposed method for detecting anomalies in the entire network uses Autoencoders which can fail to represent data in latent space that is not within the observed data [26]. This is because there is no learning of the latent state or probability distributions of inputs. Although methods such as UG can represent inputs in probability distributions, it may describe all inputs to have similar characteristics as every observation is described in the Gaussian distribution. Instead, a method such as variational autoencoders should be used which can simultaneously describe data in latent space in a probabilistic manner [27]. This can help to better detect anomalies by enhancing the classification of data.
5. Selection of Core Routers – The selection of core routers to monitor Internet-wide anomalies should be investigated. The selection method can investigate on generating a trustworthiness scheme for each BGP router. Such a scheme can comprise the number of valid routes that a router has advertised and whether the router is positioned at a place that can cover a part of the Internet that other routers cannot observe. This helps to ensure that Internet-wide anomalies can be detected using trusted core routers.
6. Online or Batch Learning – A scheme to select the an appropriate amount of data to continuously train the proposed machine learning models is recommended to be conducted. Such schemes should investigate on using methods such as stochastic gradient descent to improve the proposed loss function.
7. Other Application Areas - In other application areas which can be represented in a graph, detection of anomalies using centrality information can also be possible. For example, 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. Thus, modifying the proposed system to extract centrality information to detect anomalies for a certain application area can be investigated to detect anomalies.

References

[1] A. Haeberlen, I. Avramopoulos, J. Rexford, and P. Druschel, “NetReview: Detecting when interdomain routing goes wrong,” in Proc. 6th USENIX Symp. Netw. Syst. Design Implement. (NSDI), Boston, MA, USA, 2009, pp. 437–452.

[2] B. Al-Musawi, P. Branch and G. Armitage, "BGP Anomaly Detection Techniques: A Survey," in IEEE Communications Surveys & Tutorials, vol. 19, no. 1, pp. 377-396, Firstquarter 2017, doi: 10.1109/COMST.2016.2622240.

[3] “A Border Gateway Protocol 4 (BGP-4),” *IETF Tools*.

[4] W. Goralski, “Border Gateway Protocol”, in *The Illustrated Network*, 2nd Edition, M. Kaufmann, 2017, pp. 409-458.

[5] M. Wubbeling, M. Meier, and T. Elsner, “Inter-AS routing anomalies: Improved detection and classification,” in Proc. 6th Int. Conf. Cyber Conflict (CyCon), Tallinn, Estonia, 2014, pp. 223–238.

[6] B. A. Prakash, N. Valler, D. Andersen, M. Faloutsos, and C. Faloutsos, “BGP-lens: Patterns and anomalies in Internet routing updates,” in Proc. 15th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min. (KDD), Paris, France, 2009, pp. 1315–1324.

[7] J. Mai, L. Yuan, and C.-N. Chuah, “Detecting BGP anomalies with wavelet,” in Proc. IEEE Netw. Oper. Manag. Symp. (NOMS), Salvador, Brazil, Apr. 2008, pp. 465–472.

[8] Y. Huang, N. Feamster, A. Lakhina, and J. J. Xu, “Diagnosing network disruptions with network-wide analysis,” SIGMETRICS Perform. Eval. Rev., vol. 35, no. 1, pp. 61–72, Jun. 2007.

[9] S. Deshpande, M. Thottan, T. K. Ho, and B. Sikdar, “An online mechanism for BGP instability detection and analysis,” IEEE Trans. Comput., vol. 58, no. 11, pp. 1470–1484, Nov. 2009.

[10] X. Shi, Y. Xiang, Z. Wang, X. Yin, and J. Wu, “Detecting prefix hijackings in the Internet with Argus,” in Proc. ACM Conf. Internet Meas. Conf. (IMC), Boston, MA, USA, 2012, pp. 15–28.

[11] C. Zheng, L. Ji, D. Pei, J. Wang, and P. Francis, “A lightweight distributed scheme for detecting IP prefix hijacks in real-time,” in Proc. Conf. Appl. Technol. Archit. Protocols Comput. Commun. (SIGCOMM), Kyoto, Japan, 2007, pp. 277–288.

[12] X. Hu and Z. M. Mao, "Accurate Real-time Identification of IP Prefix Hijacking," 2007 IEEE Symposium on Security and Privacy (SP '07), 2007, pp. 3-17, doi: 10.1109/SP.2007.7.

[13] J. Li, D. Dou, Z. Wu, S. Kim, and V. Agarwal, “An Internet routing forensics framework for discovering rules of abnormal BGP events,” SIGCOMM Comput. Commun. Rev., vol. 35, no. 5, pp. 55–66, Oct. 2005.

[14] N. M. Al-Rousan and L. Trajković, "Machine learning models for classification of BGP anomalies," *2012 IEEE 13th International Conference on High Performance Switching and Routing*, 2012, pp. 103-108, doi: 10.1109/HPSR.2012.6260835.

[15] A. Lutu, M. Bagnulo, J. Cid-Sueiro and O. Maennel, "Separating wheat from chaff: Winnowing unintended prefixes using machine learning," *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, 2014, pp. 943-951, doi: 10.1109/INFOCOM.2014.6848023.
[16] I. O. de Urbina Cazenave, E. Köşlük and M. C. Ganiz, "An anomaly detection framework for BGP," *2011 International Symposium on Innovations in Intelligent Systems and Applications*, 2011, pp. 107-111, doi: 10.1109/INISTA.2011.5946083.
[17] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation Learning on Graphs: Methods and Applications."
[18] F. Li, E. Yang, A. Ma, and R. Dong, "Optimal Representation of Large-Scale Graph Data Based on Grid Clustering and K^2-Tree", *Mathematical Problems in Engineering*, vol. 2020, Article ID 2354875, 8 pages, 2020.
[19] H. Chen, H. Yin, T. Chen, Q. V. H. Nguyen, W. Peng and X. Li, "Exploiting Centrality Information with Graph Convolutions for Network Representation Learning," *Proceedings of the IEEE 35th International Conference on Data Engineering (ICDE),* Macao, China, April 8-11, 2019, pp. 590-601, doi: 10.1109/ICDE.2019.00059.
[20] "Software for Complex Networks," *Software for Complex Networks - NetworkX 2.5 documentation*, 22-Aug-2020.
[21] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*. Cambridge: Cambridge University Press, 1994.
[22] V. L. Cao, M. Nicolau and J. McDermott, "A hybrid autoencoder and density estimation model for anomaly detection," *Proceedings of the International Conference on Parallel Problem Solving from Nature*, Edinburgh, UK, September 17-21, 2016, pp. 717–726.
[23] M. Odiathevar, W. K. G. Seah, M. Frean and A. Valera, "An Online Offline Framework for Anomaly Scoring and Detecting New Traffic in Network Streams," in *IEEE Transactions on Knowledge and Data Engineering*, 11 January 2021, doi: 10.1109/TKDE.2021.3050400.
[24] D. Munoz, F. Bouchereau, C. Vargas, and R. Enriquez, "Terrestrial-Based Location Systems" in *Position Location Techniques and Applications*, 1st Edition, Elsevier Science, 2009, pp. 153-206.
[25] M. Lad, X. Zhao, B. Zhang, D. Massey, and L. Zhang, "Analysis of BGP Update Surge during Slammer Worm Attack". In: Das S.R., Das S.K. (eds) Distributed Computing - IWDC 2003. IWDC 2003. Lecture Notes in Computer Science, vol 2918. Springer, Berlin, Heidelberg.
[26] J. Hajewski and S. Oliveira, "An Evolutionary Approach to Variational Autoencoders," 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), 2020, pp. 0071-0077, doi: 10.1109/CCWC47524.2020.9031239.
[27] X. Hou, L. Shen, K. Sun and G. Qiu, "Deep Feature Consistent Variational Autoencoder," 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), 2017, pp. 1133-1141, doi: 10.1109/WACV.2017.131.
[28] J. Cowie, A. Ogielski, B. Premore, E. Smith, and T. Underwood. Impact of the 2003 blackouts on Internet communications. Technical report, Renesys, November 2003.
[29] V. L. Cao, M. Nicolau and J. McDermott, "Learning Neural Representations for Network Anomaly Detection," in *IEEE Transactions on Cybernetics*, vol. 49, no. 8, pp. 3074-3087, Aug. 2019, doi: 10.1109/TCYB.2018.2838668.