Final Report - Machine Learning for Detecting BGP Anomalies

Carlos Marcelo Martinez, {carlos@cagnazzo.uy, carlos@lacnic.net}

Workshop IA en Sistemas Ciber-Fisicos, Nov-Dec 2024

Table of Contents

Introduction

BGP, the Border Gateway Protocol, needs little introduction. BGP is crucial for internet connectivity and is prone to vulnerabilities like hijacking, misconfiguration, and distributed denial of service (DDoS) attacks. The protocol was designed without robust authentication mechanisms, leaving it susceptible to malicious activity.

Within the framework of the “Workshop IA en Sistemas Ciber-Fisicos” which took place in November 2024 in Montevideo, Uruguay, I would like to propose an alternative hands-on, open task where I will try to apply Machine Learning algorithms to the problem of BGP anomaly detection.

Report Objectives

I will work following the lines of work described in two papers, [1] and [2]. My goals are for this work are:

- Identify two or three *benchmark BGP incidents* to which to apply ML detection algorithms.
- Produce clean datasets of BGP data for the identified incidents, using the features mentioned in [1] and [2]
- Apply ML algorithms and check whether the BGP incident is detected appropriately (or not!)

A *benchmark incident* is a BGP routing anomaly that was publicly reported, can be approximately timestamped and its geographical location, “blast radius” can be pinpointed at least approximately. In this way we can validate our detection strategies.

Benchmark Incidents

Some incidents that are mentioned in [1] and [2] as benchmark incidents are listed in the following table. I have added references found in Internet news for each one.

Incident Name and Operator	Date	Time of Day	Duration Approx.	Type	General Geographical Region	References
YouTube Hijack (Pakistan Telecom)	2008-02-24	Afternoon	2 hours	Hijack		
Belarus Traffic Diversion (Belarus Telecom)	2013-02	Various	Few minutes to several hours	Hijack (MITM)		
China Telecom Incident	2010-04	Morning	18 minutes	Leak		
Level 3 Route Leak	2017-11-06	9:45am-11:25am Pacific	90 minutes	Leak	US East Coast	[TE2] [ME1]

Google Traffic Hijack (MainOne)	2018-11-12	1:00 PM - 2:23 PM PST	74 minutes	Hijack	Africa West Coast / Europe	[TE3]
Amazon Route 53 Hijack	2018-04-24	Afternoon	2 hours	Hijack		
Vodafone India Route Leak	2015-06	Unknown	Few hours	Leak		
MyEtherWallet BGP Hijack	2018-04-24	Morning	2 hours	Hijack		
RosTelecom Incident	2020-04-01	7:30PM UTC	Approximately 30 minutes	Leak		[TE1]
Moscow Power Outage	2005-05-25	Unknown	Several hours	Blackout	Central Europe	[WIKI1]

The incidents marked in yellow are those I selected as baseline incidents.

Data Sources

The two most used sources of routing and BGP data are RIPE RIS [RIS1] and the University of Oregon's RouteViews project [4]. Both projects provide detailed, timestamped dumps of BGP messages that can be processed using appropriate tools.

BGP data is usually stored in files following a format called MRT defined in RFC6396 [6]. In this work I will use RIS data primarily.

RIPE RIS Vantage Points

RIPE RIS publishes a list of route collectors in [RIS2]. See the full list in Annex 1.

Extracting BGP Data

BGPReader allows for getting BGP data from RIS from the command line using the following command:

```
bgpreader -w '2024-01-01 00:00:00','2024-01-01 23:59:59' -p ris -c rrcXX -o
cache=./cache-dir -t updates
```

The used flags mean:

- "-w" defines the *time window* from which we need to get data
- "-p" defines the *source* of the data (ris, routeviews, etc.)
- "-c" defines the route collector to use
- "-t" specifies the type of BGP message to get, in this case, UPDATES
- "-o cache" keeps some MRT files locally in order to speed up further queries for the same data

Tools

In addition to Jupyter Lab, Numpy, Pandas, Scikit-Learn and other Python libraries, I will use the toolkit provided by CAIDA's BGPStream [5] project in order to process the MRT files and streams provided by RIPE RIS and RouteViews.

BGPStream can be used as a library or with a standalone CLI tool called "bgpreader". It can either stream information directly from RIS or RouteViews servers and it also can be used to parse locally stored MRT files.

Features

From [1] and [2] we can identify a list of potentially relevant features for BGP data sets.

ID	NAME	DESCRIPTION	COMMENTS
0	update_count	The number of various BGP messages received over time	Analyzed within rolling windows of given length (5min)
1	announcement_count	The number of BGP announcements over time	Analyzed within rolling windows of given length (5min)
2	withdrawals_count	The number of BGP withdrawals over time	Analyzed within rolling windows of given length (5min)
3	as_path_len_avg	The average length of the AS Path contained in BGP updates	Analyzed within rolling windows of given length (5min)
4	as_path_len_median	The median / average length of the AS Path contained in BGP updates	Analyzed within rolling windows of given length (5min)
5	as_path_change	The number of prefixes that change the as_path in their announcements	
N/I*	origin_as_ch	The number of prefixes suddenly changing its origin AS**	Not implemented in this work
N/I*	origin_as_count	The number of origin ASes seen in in BGP NLRI	Not implemented in this work
N/I*	Number of sessions changing state	The number of BGP sessions that change state (active, idle, etc.)	Not implemented in this work

All feature files are computed aggregating or averaging the values over rolling windows of varying length. The best results were obtained using a window of 6 seconds.

The directory “scripts” in the repository has python scripts to extract features from 0 to 5 from bgpstream data. The code repository includes the extracted features but does not include the raw bgpstream data as these raw files are very large (several gigabytes).

Incident Analysis

For each of the chosen incidents I provide a short description of the incident, the collector(s) more likely to have been within the impact radius of the incident, the ML approach chosen and the results (if any).

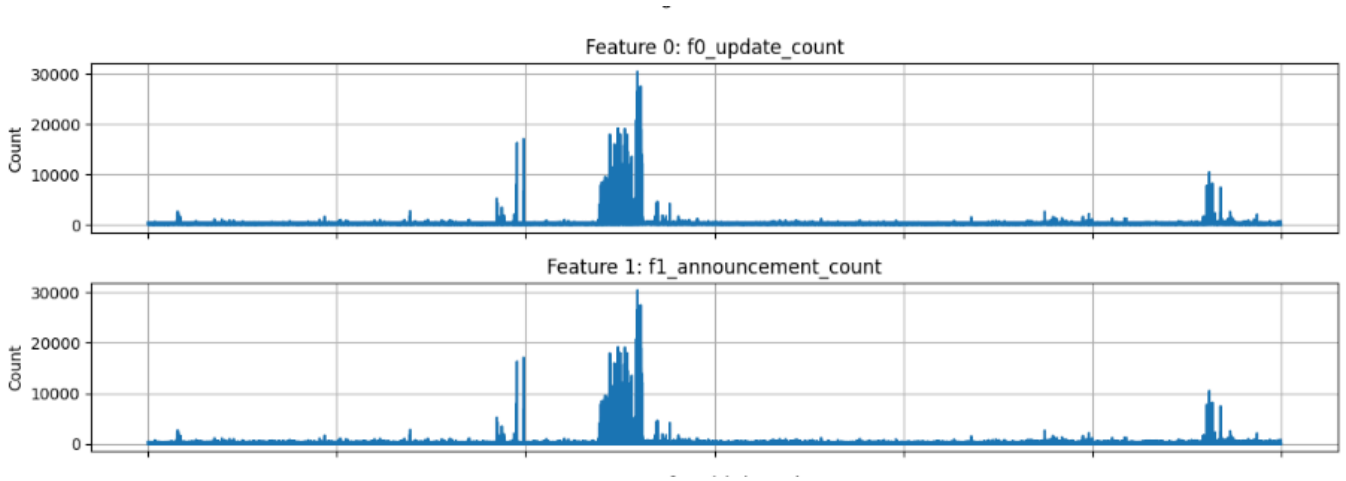
2005 Moscow Power Outage

On May 25, 2005 a massive power outage hit Moscow. Since 80% of Russia’s Internet traffic passed through Moscow at the time this incident had a significant impact on Internet routing [TE1].

The RIS collectors more likely to have seen impact from this incident are rrc03 (Amsterdam AMS-IX) and rrc05 (Vienna).

This incident has been studied before in the literature ({Updating})

Exploring data from rrc05 (Vienna) the anomaly is clearly visible:



rrc05 routing data for May 24-26, 2005

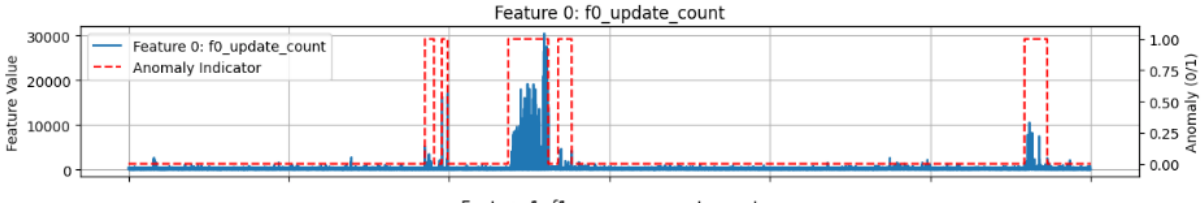
Algorithms and techniques applied

After data retrieval from RIS using bgpstream and performing feature extraction I tried the following approaches:

1. Supervised classification techniques: Logistic Regression

   Steps followed:

   - I labeled the data visually determining “anomaly intervals” from the above plots.



- I split the dataset into training and validation sets, with a 75% training-validation proportion.
- I trained a Logistic Regression model on this dataset

Issues found:

- Class 0 (normal intervals) is over represented in the dataset, creating a significant imbalance between classes. This creates a significant bias towards class 0.
- Timing for the incidents is not that clear cut, we usually look at press notes and blog posts and these times may have significant deviations

Results Obtained with the Logistic Regression Model:

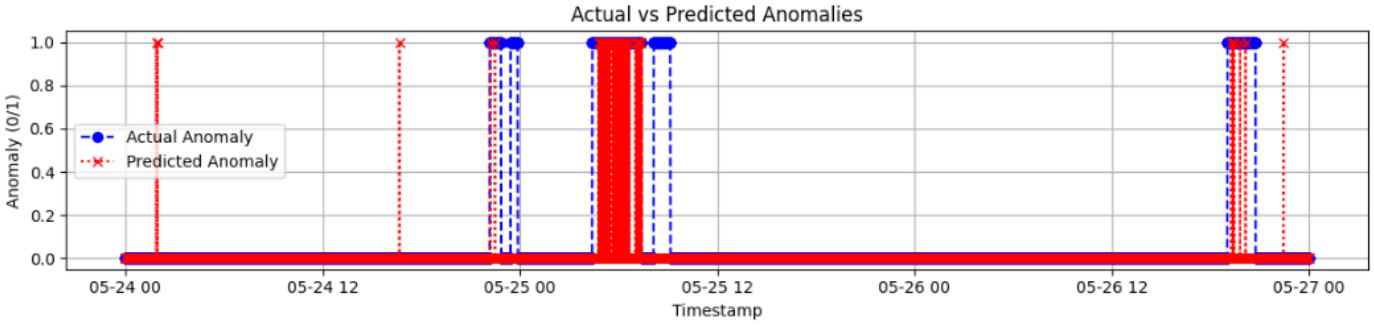
```
Selected Training Features Shape: (32399, 2)
Selected Validation Features Shape: (10800, 2)
Training Classification Report:
              precision    recall  f1-score   support

           0       0.91      1.00      0.95     29324
           1       0.95      0.08      0.15      3075

    accuracy                           0.91     32399
   macro avg       0.93      0.54      0.55     32399
weighted avg       0.92      0.91      0.88     32399
```

Comments on the results obtained with the logistic regression model:

- The model obtains an acceptable precision score, but recall for class 1 is very low
- The resulting F1 score reflects this and is also quite low



2. Supervised Classification Techniques : Downsampling and Logistic Regression with Custom Threshold

In an attempt to improve the results obtained applying the supervised learning approach, I tried the following:

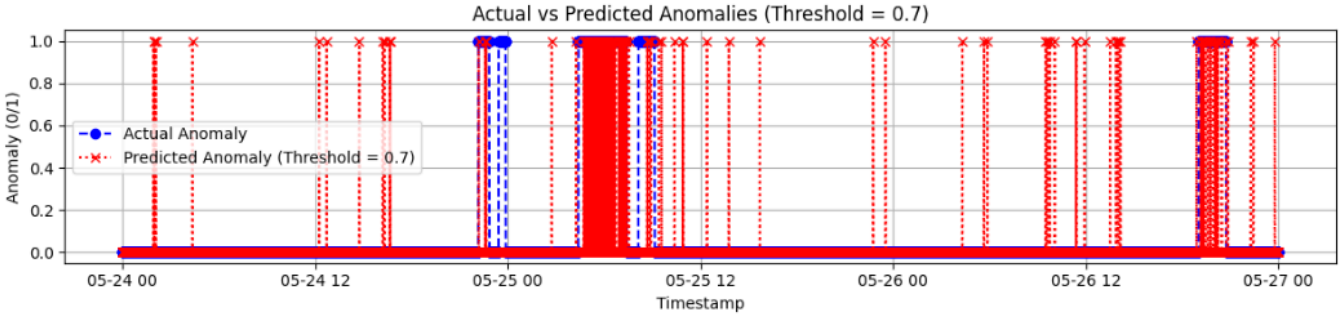
- Downsampling the majority class, to improve the imbalance between the two classes
- Setting a custom probability threshold in the logistic regressor

The results were only marginally better than in the previous case:

```
Validation Classification Report (Threshold = 0.7):
              precision    recall  f1-score   support

           0       0.92      1.00      0.96      9770
           1       0.78      0.15      0.26      1030

    accuracy                           0.92     10800
   macro avg       0.85      0.57      0.61     10800
weighted avg       0.90      0.92      0.89     10800
```



3. Decision Trees, Random Forests

Other supervised techniques I tried:

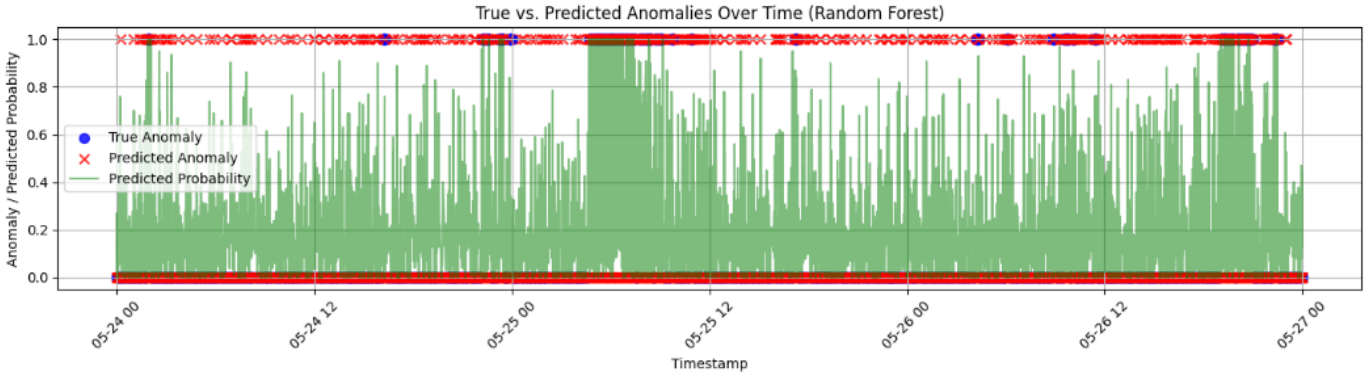
- I trained a Decision Tree classifier with the same labeled data obtained by visually identifying anomaly intervals.
- I trained a Random Forest model with the same labeled data obtained by visually identifying anomaly intervals.

Again here the results were not useful. See below for the results obtained with the Random Forest classifier.

```
Random Forest Model Evaluation:
Accuracy: 0.7974285714285714
Confusion Matrix:
 [[5335  265]
 [1153  247]]
Classification Report:
              precision    recall  f1-score   support

           0       0.82      0.95      0.88      5600
           1       0.48      0.18      0.26      1400

    accuracy                           0.80      7000
   macro avg       0.65      0.56      0.57      7000
weighted avg       0.75      0.80      0.76      7000
```



4. Unsupervised anomaly detection: One-Class SVM and Local Outlier Factor (LOF)

   I also explored the use of unsupervised anomaly detection techniques, in particular I tried:

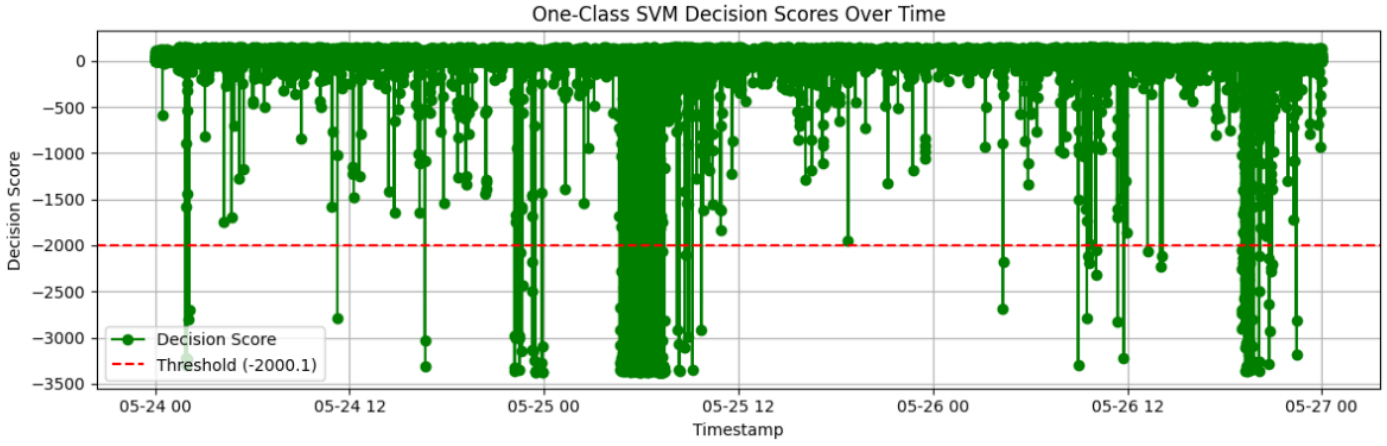
   - Isolation Forests
   - One-Class SVM
   - Local Outlier Factor (LOF)

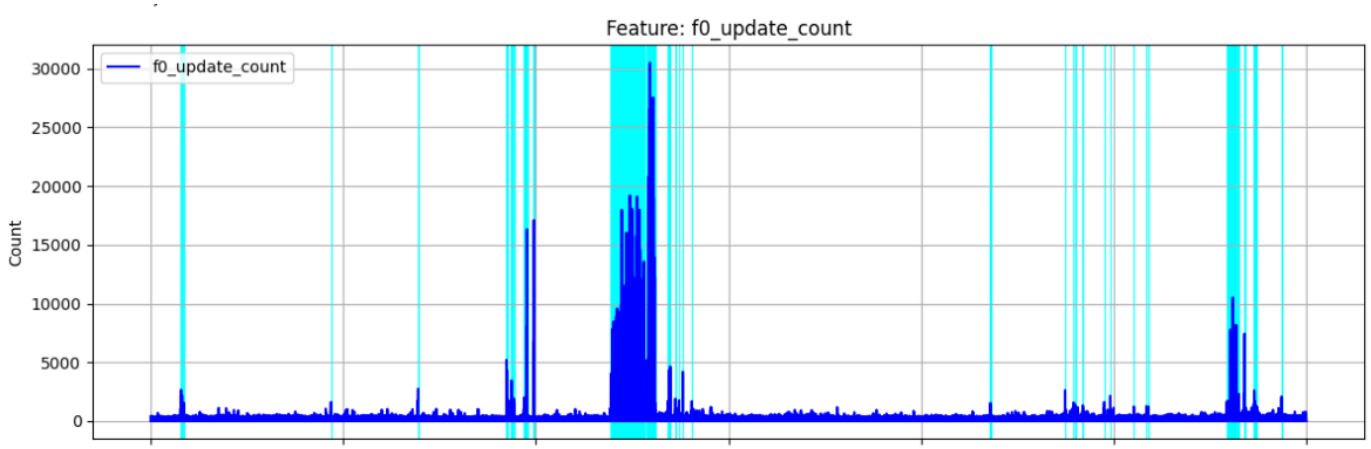
   While differing in their approach these algorithms all try to identify anomalies in data. Anomalies are data points that are very different from the feature data baseline. They are surprising / unexpected points that point to something out of the ordinary happening in the system under study.

   In this case the approach is different. I apply the unsupervised algorithm to my feature data and afterwards I compare the identified anomalous intervals to the intervals I visually identified when analyzing the incident during the exploratory data analysis phase.

   The best results were obtained applying One-Class SVM and LOF. All three algorithms provide a way of creating an “anomaly indicator”, and while some adjustment was indeed needed for determining the correct threshold for this anomaly indicator, I was successfully able to automatically identify anomalous intervals that correspond to those anomalies visually identified during the exploratory data analysis phase.

   ### Using One-Class SVM:

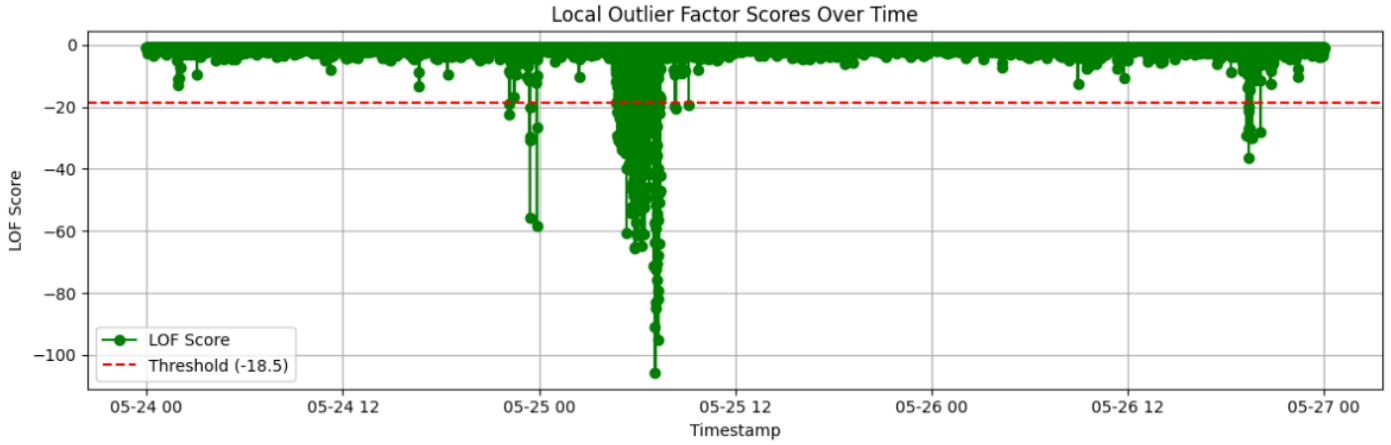




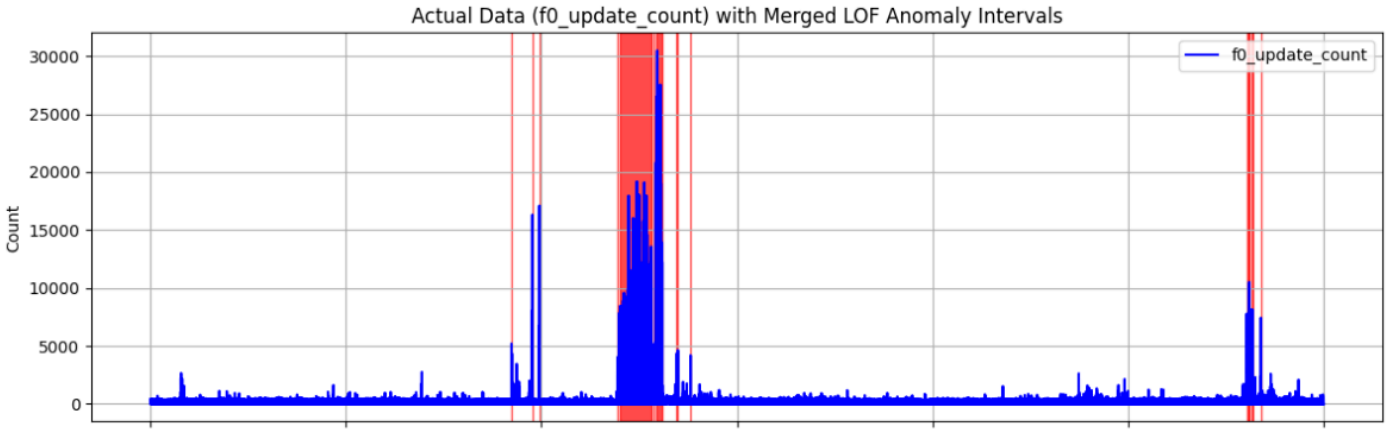
In this graph the light-blue intervals represent the intervals that the OneClass-SVM model identified as anomalous considering a threshold of -2000. They reflect very accurately those intervals identified visually.

Since the decision score is computed per-interval I created an additional post-processing step where anomaly intervals that are very close (separated by less than 60s) are joined into a single, longer, anomaly interval.

Local Outlier Factor:



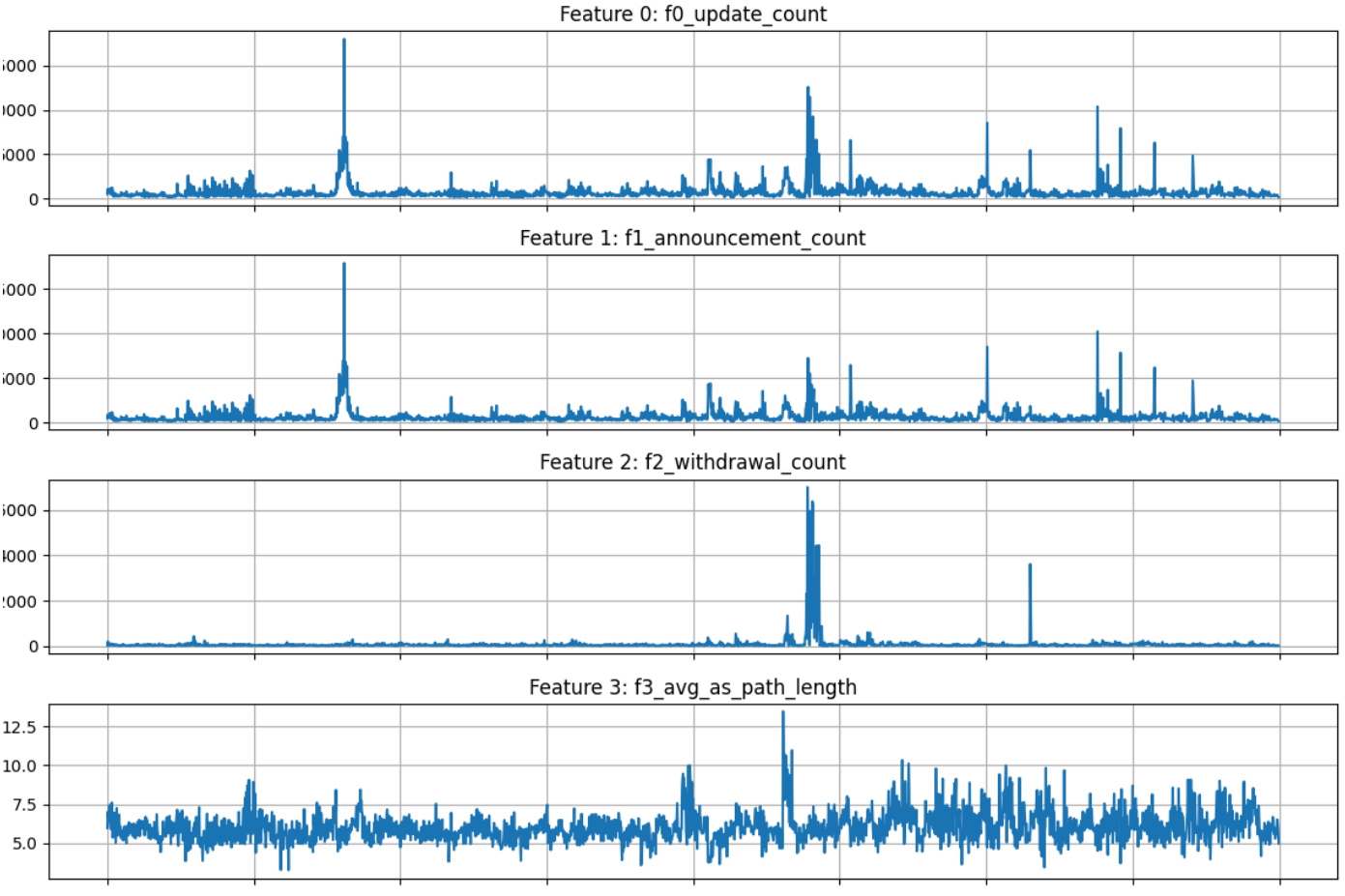
LOF Scores over time. The red dotted line represents the manually selected threshold.



Merged anomaly intervals.

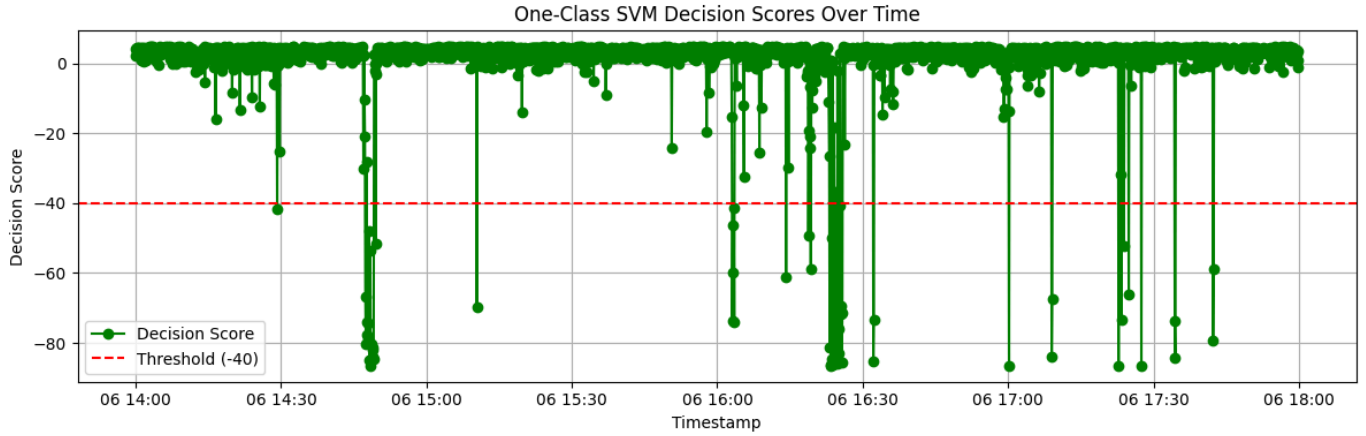
Level 3 Route Leak

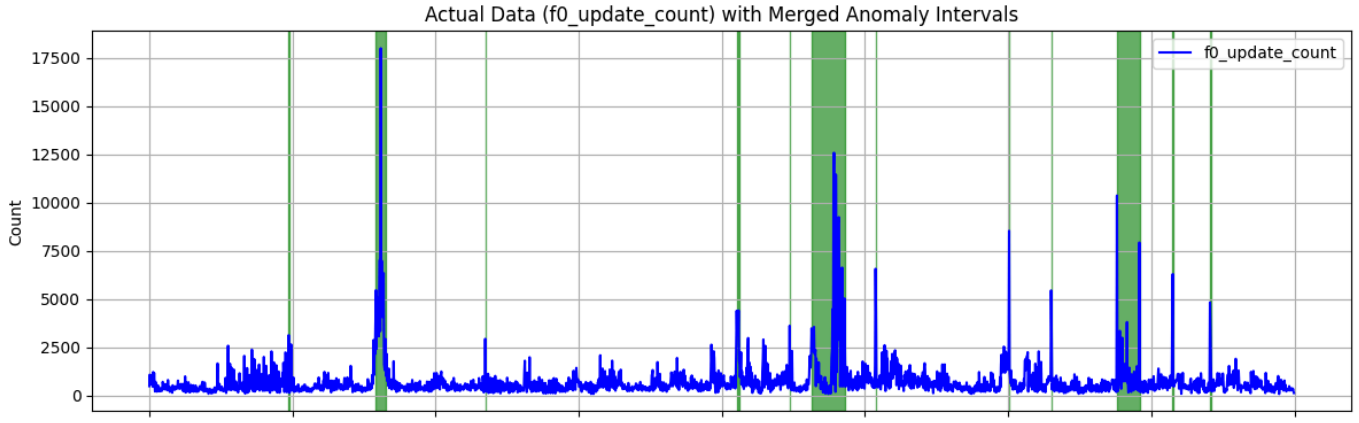
On November 6, 2017, Level 3 (AS3356) began announcing thousands of routes for which it was unable to properly carry traffic onwards. This type of incident is known as a “route leak”, and may produce widespread outages and service quality degradation.



2017 Level 3 Route Leak Exploratory Data Analysis

The unsupervised methods are the better performers here as well. One-Class SVM and LoF provide the best results.

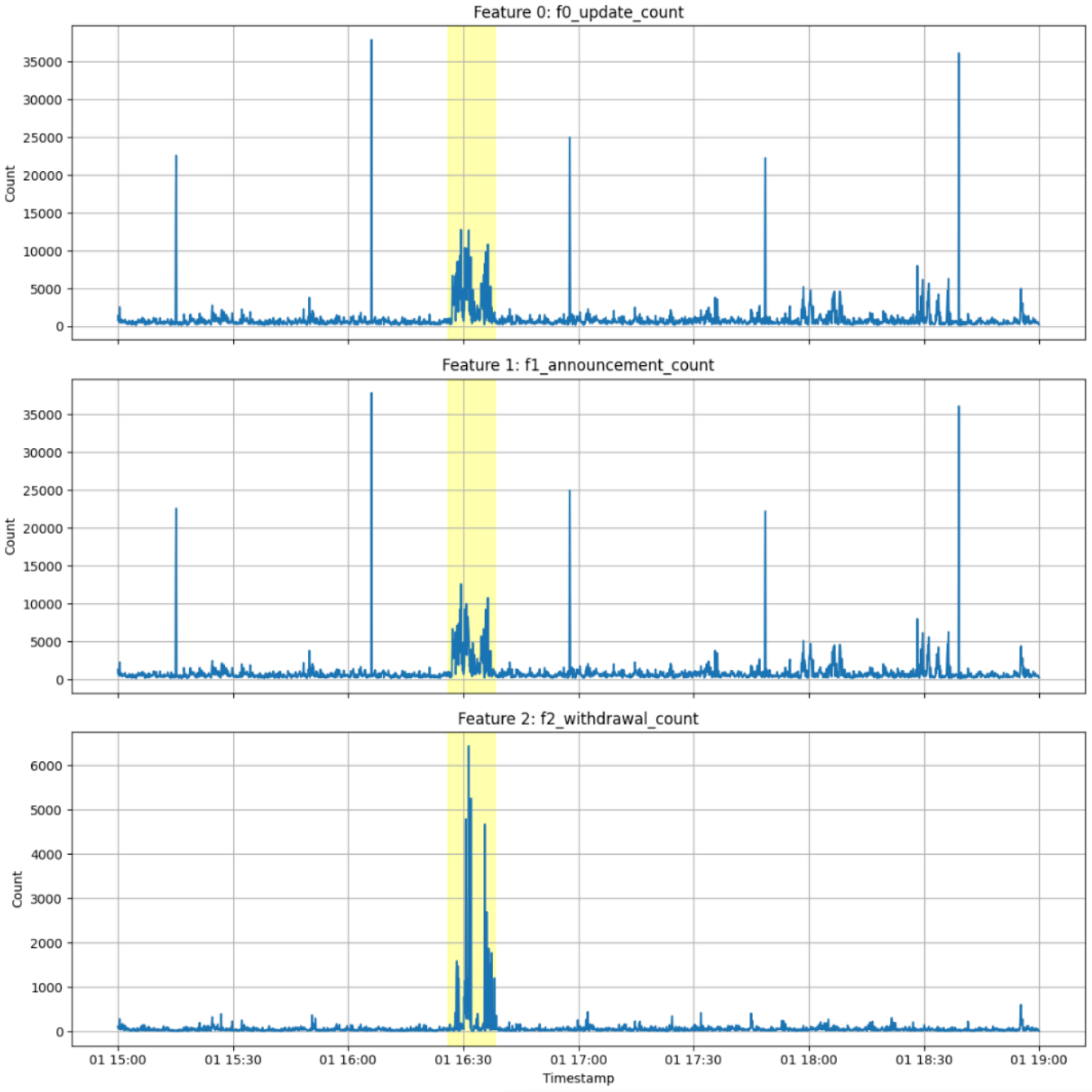




2020 Rostelecom Leak

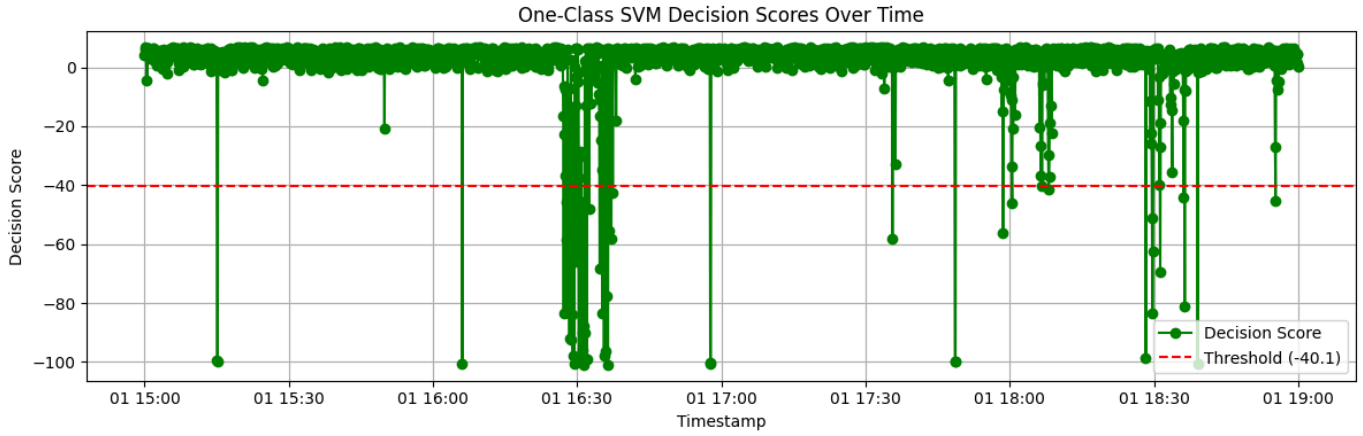
On April 1, 2020 the Russian telecom operator Rostelecom (AS12389) leaked several thousand prefixes, belonging to several European hosting and cloud providers among others. The incident lasted for approximately 30 minutes. This is an interesting incident in the sense that while several highly visible operators were affected, there is not much public information available about it. See [TE4]

The incident lasted for approximately 30 minutes. Our exploratory data analysis allows us to pinpoint an anomaly interval approximately between 19:26 UTC and 19:38 UTC where there is clearly anomalous behaviour.

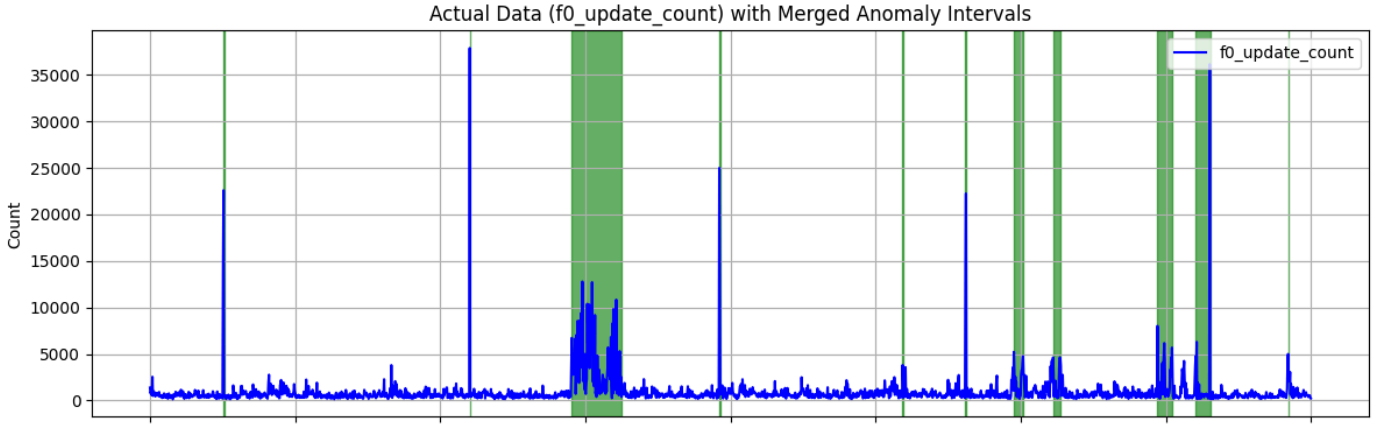


2020 Rostelecom Leak / Exploratory data analysis

Again the unsupervised anomaly detection algorithms perform very well.



One Class SVM Decision Scores



Anomaly intervals as detected by One-Class SVM

Conclusions

I explored the application of machine learning techniques for detecting anomalies in **Border Gateway Protocol (BGP) routing data**. Using real-world BGP data as collected by the RIS project, plus publicly available information about routing incidents I analyzed both supervised and unsupervised learning approaches to identify anomalies in BGP data.

Key lessons learned:

- **Supervised learning methods** such as **Logistic Regression, Decision Trees, and Random Forests** were effective in detecting anomalies but were highly sensitive to class imbalance. Despite downsampling efforts and threshold tuning, the **recall for anomaly detection remained low**, making these models prone to false negatives.
- **Unsupervised anomaly detection methods** such as **One-Class SVM and Local Outlier Factor (LOF)** provided better results. These methods were able to identify **previously known anomalous intervals** without requiring labeled data, making them suitable for real-world applications where labeling is difficult.

- **Clustering** time slots identified as anomalous was very effective in identifying incidents
- The **BGP feature engineering process** played a crucial role in model performance. Features such as **AS path changes, announcement/withdrawal rates, and session state changes** proved valuable in detecting anomalies. A key insight was that varying the rolling time windows (e.g., using 6-second intervals instead of 1 minute intervals) improved anomaly visibility.
- Incident timing information available in public news sources proved to be inaccurate and in a few cases, plain wrong. This made the use of supervised techniques difficult.

Ideas for Future Work

The study demonstrates the potential of ML-based anomaly detection in BGP routing but also highlights several areas for improvement:

- **Real-time processing**: Implementing **streaming-based detection** using **Kafka or other real-time data pipelines** would allow for the immediate identification of anomalies, reducing response time for network operators.
- **Additional work on identifying useful features is needed**. The methods I explored in this study were very useful when the number of affected prefixes was significantly large. Incidents affecting just a few tens of prefixes (although potentially affecting millions of users) would not be visible.
- **Interpretation of anomalies:** This study focused on identifying anomalies but the techniques applied do not help in interpreting whether an anomaly is an attack, a network fault or a benign network change.
- **Filtering by prefix or origin AS** and applying the techniques outlined in this study could perhaps allow for not only identifying anomalies but narrowing down the affected ASes or prefixes.

Annex 1 - Full List of RIS Collectors

Name	Physical Location	Type	Scope
RRC00	Amsterdam, NL	multihop	global
RRC01	London, GB	IXP	LINX, LONAP
RRC03	Amsterdam, NL	IXP	AMS-IX, NL-IX
RRC04	Geneva, CH	IXP	CIXP

RRC05	Vienna, AT	IXP	VIX
RRC06	Otemachi, JP	IXP	DIX-IE, JPIX
RRC07	Stockholm, SE	IXP	Netnod
RRC10	Milan, IT	IXP	MIX
RRC11	New York, NY, US	IXP	NYIIX
RRC12	Frankfurt, DE	IXP	DE-CIX
RRC13	Moscow, RU	IXP	MSK-IX
RRC14	Palo Alto, CA, US	IXP	PAIX
RRC15	Sao Paolo, BR	IXP	PTTMetro-SP
RRC16	Miami, FL, US	IXP	Equinix Miami
RRC18	Barcelona, ES	IXP	CATNIX
RRC19	Johannesburg, ZA	IXP	NAP Africa JB
RRC20	Zurich, CH	IXP	SwissIX
RRC21	Paris, FR	IXP	France-IX Paris and France-IX Marseille
RRC22	Bucharest, RO	IXP	Interlan
RRC23	Singapore, SG	IXP	Equinix Singapore
RRC24	Montevideo, UY	multihop	LACNIC region
RRC25	Amsterdam, NL	multihop	global
RRC26	Dubai, AE	IXP	UAE-IX

Annex 2 - References

[1] Edwards, P., Cheng, L., & Kadam, G. (2019). *Border Gateway Protocol Anomaly Detection Using Machine Learning Techniques*. SMU Data Science Review, 2(1), Article 5. Retrieved from https://scholar.smu.edu/datasciencereview/vol2/iss1/5

[2] Al-Musawi, B., Branch, P., & Armitage, G. (2017). *BGP Anomaly Detection Techniques: A Survey*. IEEE Communications Surveys & Tutorials, 19(1), 377–395. DOI: 10.1109/COMST.2016.2622240

[RIS1] RIPE NCC Routing Information System, https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/

[4] University of Oregon RouteViews Project, https://www.routeviews.org/routeviews/

[5] CAIDA BGPStream Project: https://bgpstream.caida.org/

[6] Multi-Threaded Routing Toolkit (MRT) Routing Information Export Format, https://datatracker.ietf.org/doc/html/rfc6396

[TE1] RosTelecom Incident: https://www.thousandeyes.com/blog/rostelecom-route-leak-targets-ecommerce-services#:~:text=The%20Rostelecom%20Route%20Leak,%2C%20Cogent%2C%20Telia%20and%20Tata.

[TE2] Level 3 Route Leak https://www.thousandeyes.com/blog/comcast-outage-level-3-route-leak

[RIS2] https://ris.ripe.net/docs/route-collectors/

[TE3] Google Main One https://www.thousandeyes.com/blog/internet-vulnerability-takes-down-google

[ME1] Widespread Outage Caused by Level 3 Route Leak, https://medium.com/oracledevs/widespread-impact-caused-by-level-3-bgp-route-leak-internet-intelligence-3dbd724d9ac5

[TE4] 2020 Rostelecom Route Leak https://www.thousandeyes.com/blog/rostelecom-route-hijack-highlights-bgp-security