Université de Strasbourg

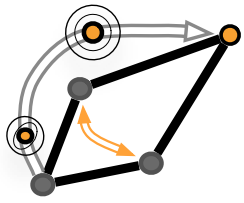
Internet Initiative Japan

Master de Science, mention Informatique,
spécialité Réseaux Informatiques et Systèmes Embarqués

Presented by:
Andréas Guillot
andreas.guillot@unistra.fr

Internet Background Radiation and Outage Detection

Supervized by:
Cristel Pelsser — pelsser@unistra.fr
Romain Fontugne — romain@iij.ad.jp
Pascal Mérindol — merindol@unistra.fr



NETWORK RESEARCH TEAM

January-July 2018

Acknowledgements

I would like to thank Pascal Mérindol, Romain Fontugne, and Cristel Pelsser for their supervision and help during this internship. I'd also like to thank François Michel and Brandon Foubert for their precious help during this internship. Lastly, I am very grateful to the IIJ II team for their warm welcome and good work environment.

Table of contents

Chapter 1 Introduction

Internet outages are known as frequent and destructive, their impacts on service performance is not negligible. But first, how to formally define and describe them? In [1], Giuseppe Aceto et al. define an Internet outage as 'the particular condition in which the network lies when one or multiple network elements located in a specific geographical area either do not work properly or are not reachable due to intentional or accidental events.'

In [17], the authors unveiled that persistent reachability problems affected more than 10 000 prefixes, with one in five of the events lasting over 10 hours. Moreover, Internet outages can happen for a wide variety of reasons, including attacks, misconfigurations, or even natural disasters. This creates a need for systems and approaches to *detect* and *mitigate* Internet outages.

Table 1: Main metrics that are used to measure network performances

Indicator	Metric	Analyzed objects	Tool
Delay	+	(One or two)-way latency	owamp, ping
Jitter	+	Delay variations	QosMet
Loss Ratio	*	Ratio of successful over total number of deliveries	ping, QosMet
Capacity	min	Amount of data that can be transferred over a link	iperf
Topology	composition	Determining the path from a source to a destination	traceroute

An outage will be detected if some of the network's components do not work properly. The need to specify performance indicators arises from this, as one needs to be able to evaluate a network based on its own characteristics. These performance indicators should be able to assert both a network's state and its level of performance. Table 1 gives an overview of standard performance indicators. For example, the analysis of the delay is based on an additive metric considering positive values. Its purpose is to estimate the time it takes to reach a destination (and generally the way back also accounts in this analysis as it is not straightforward to discriminate the two and time synchronization limitations can also interfere). *owamp* and *ping* are most used tools for measuring delay. This indicator, as the other presented in the Table, can be used to detect outages since a significant deviation from the normal trend may indicate that an outage is occurring. With delays, an outage might be detected when the time to reach a destination significantly increases (possibly to a pseudo infinite duration modeled with an arbitrary timer).

The goal of this chapter is to describe the context in which my internship took place. Section 1 is going to describe the different actors that were involved in this internship. Section 2 will define the objective and the different missions of this internship. Lastly, Section 3 will describe the structure of this document.

1. Hosting structures and actors

My internship is a collaboration between the ICube laboratory[1] and the Internet Initiative Japan Innovation Institute[2] (IIJ-II). The Center for Applied Internet Data Analysis[3] (CAIDA) is also involved as they provide us with the data that is at the basis of our study.

1.1 The ICube laboratory

ICube is a French research laboratory that focuses on engineering sciences, computer sciences, and imaging. There are four departments:

1. Computer science department
2. Imaging, robotics, remote sensing and biomedical department
3. Solid-state electronics, systems and photonics department
4. Department of mechanics

The Computer science department where I started my internship regroups six different teams, including the Network Research Group[4]. This group is then subdivided into two different research topics: the Internet of Things and Core Networks.

My local supervisors (Prof. Cristel Pelsser[5] and Pascal Mérindol[6]) are both part of the Core Networks theme.

1.2 Internet Initiative Japan Innovation Institute

IIJ is a Japanese Internet Service Provider (ISP) with more than 3000 employees. IIJ Innovation Institute is a division of IIJ that pursues core Internet technologies. Their goal is to create innovative technologies while following three missions:

1. Innovation: developing new fundamental technologies essential to the next generation Internet.
2. Incubation: supporting entrepreneurs with innovative technologies to build up their business.
3. Education: contributing to education for bringing up innovative engineers.

As a branch of IIJ, the members of the Innovation Institute get to interact with the different divisions to analyze their data and to provide feedback into their solutions. They work on several projects, including network Measurement and Analysis. Dr. Romain Fontugne[7], my third supervisor, works on this project. I was also a part of the innovation group.

1.3 CAIDA

CAIDA conducts network research and builds infrastructure to support large-scale data collection and redistribution. They provide valuable datasets and tools that are extensively used by the scientific research community working in the Internet field. In our study, we use one of their numerous dataset in particular, the Internet Outage Detection and Analysis[8]

[1] https://icube.unistra.fr/
[2] https://www.iij-ii.co.jp/
[3] https://www.caida.org/home/
[4] http://icube-reseaux.unistra.fr/fr/index.php/Accueil
[5] https://clarinet.u-strasbg.fr/~pelsser/
[6] https://dpt-info.u-strasbg.fr/~merindol/
[7] https://www.iij-ii.co.jp/en/members/romain.html
[8] https://www.caida.org/projects/ioda/

(IODA), whose main purpose is to identify macroscopic internet outages.

2. Problem statement

The inevitability and expensiveness of network outages is what drives the study of outage detection techniques forward. Combining the increase in size and in complexity both in terms of services and technologies created a need to systematize and to analyze different outage detection techniques. The study of IBR provides a unique opportunity to study network outages since it is a pervasive data source that receives traffic from most of the Internet [5].

The objective of this report is twofold. First, it aims at providing an overview of the potential of using *Internet Background Radiation* as a data source to detect outages. Second, it describes our implementation of an adaptive solution that monitors Internet-wide outages. Our solution tries to identify network outages thanks to a statistical analysis of the IBR data. It compares the ground collected data to artificial predictions in order to look for significant shifts that characterizes outages.

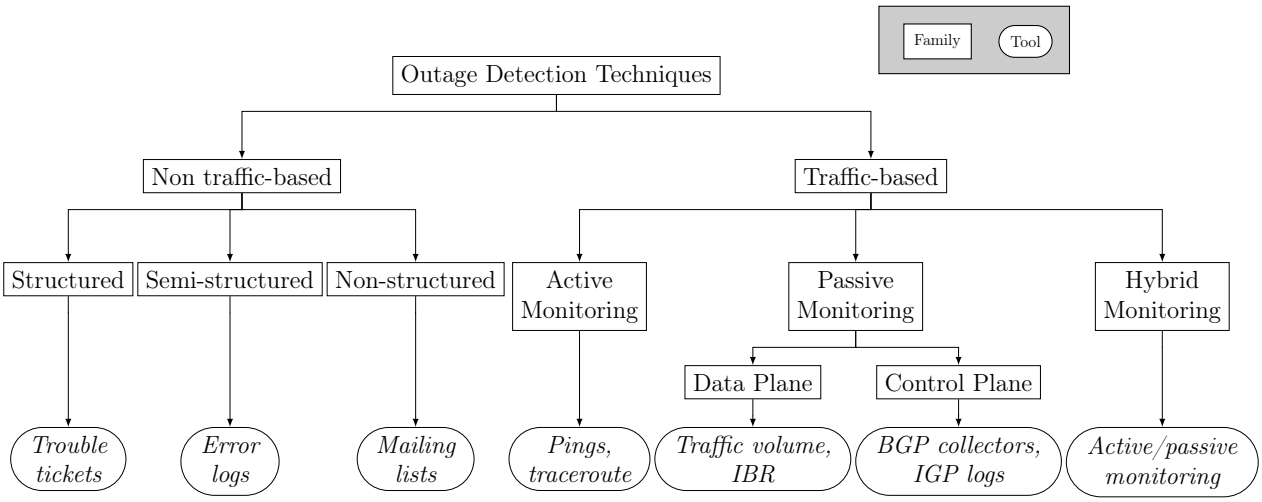
3. Outline

To evaluate and position the utility of the IBR dataset as an Internet-wide outage detection data source, Chapter 2 describes the scientific context behind its use, belonging to the field of IP monitoring and measurement. Chapter 3 describes our contributions, in particular the method and the outage detection tool that we propose.

Chapter 2 Scientific Context

Outage detection can be achieved with different measurement techniques analyzing various subset of performance indicators. These techniques can be divided into several families which are illustrated on figure 1. The following sections aim to introduce the main methods used in each family. In particular, as we opt for a traffic based method that passively monitors the data plane, we will provide a detailed background for this family of tools.

Figure 1: Tree of the different families of outage detection techniques



Section 1 is going to describe non-traffic based approaches, while Section 2 is going to describe traffic-based approaches. Lastly, Section 3 will describe how time series can be modeled and how predictions are evaluated.

1. Non traffic-based approaches

Non-traffic based approaches use data from other sources than passive or active traffic to study outages. There are three types of non-traffic based approaches: *structured*, *semi-structured*, and *non-structured*.

Approaches using structured objects take advantage of their clearly-defined structures to extract and analyze the information that is contained inside them. An example of such an object is trouble tickets, which have been used in papers such as [20] to correlate network events with human-made tickets. An example of a ticket can be found on figure 2, where information such as the beginning, the end, and the description of a network-related event can be found. In this example, one can observe the addition of a new link that is presented as a minor maintenance event with a limited impact on the service. Semi-structured data sources range from error and activity logs to customer emails. Non-structured data sources comprise technical blogs and mailing lists (e.g. NANOG). In [3], the authors used *Natural Language*

Processing (NLP) on the *outages* mailing list [23] to determine which network entities were likely to be part of a network outage.

Figure 2: Example of a network ticket issued by a Network Operating Center

```
----------------------------------------------------------------
N°Ticket                          : 59756
Type de ticket                    : MAINTENANCE
Etat du ticket                    : Ouvert
----------------------------------------------------------------
Emetteur                          : NOC-RENATER
Elément concerné                  : LIAISON
Localisation de la Maintenance    : STRASBOURG-LYON1
Impact sur le service             : MINEUR
Service(s) impacté(s)             :
----------------------------------------------------------------
Début de Maintenance              : 29/07/2016 06:00 CEST
Fin prévue de la Maintenance      : 29/07/2016 08:00 CEST
Durée de l'impact (minutes)       : 0
----------------------------------------------------------------
Date/Heure Ouverture (du ticket)  : 28/07/2016 11:07 CEST
----------------------------------------------------------------

Description de la maintenance :
Maintenance RENATER pour mise en service de la liaison STRASBOURG-LYON1 10
Go.

----------------------------------------------------------------
NOC-RENATER   Tél. : 0800 77 47 95 (+33 800 77 47 95)
Email : noc-renater@noc.renater.fr
----------------------------------------------------------------
```

Non-traffic based approaches are used as complementary data sources, but they are not sufficient on their own. Indeed, the fact that they often are human-made implies that they are potentially error-prone and might require manual processing to extract relevant data. However, the fact that they contain information about user-related metrics as opposed to network-related ones contributes to their usefulness. Indeed, each indicator can be subdivided into multiple indicators that might have a different meaning to different users. For example, the loss ratio can both be applied to network-related indicators with the loss ratio of a link, and to user-related indicators with the end-to-end loss ratio. Network-related indicators tend to study the Quality of Service of a network whereas user-related indicators study the Quality of Experience. A study of the difference and of the relationship between these two can be found in [12].

Besides, such a piece of information is useful for example to train automatic methods that we will describe later. As such, those source of data are also valuable as ground truth data necessary to validate statistical tools.

The next section describes traffic-based approaches, which draw from data that has been gathered on the networks themselves. This is the most popular approach used by the research community to automate outage detection.

2. Traffic-based approaches

Traffic-based approaches use data that originates from network measurements. These measurements are performed by network devices. This family can be divided into three categories as illustrated on figure 1:

- *Active Monitoring*, that injects purposely forged synthetic measurements into a network;
- *Passive Monitoring*, that takes advantage of the real traffic-related information that is already stored or captured on network devices;

- *Hybrid Monitoring*, which is a combination of both active and passive monitoring.

The following sections describe the differences between the three, and discuss how researchers have leveraged these mechanisms to detect outages.

2.1 Active Monitoring

Active monitoring techniques will insert artificial traffic into existing networks in order to study its characteristics and behavior in general.

These methods can be used in order to test the reachability of a destination or to compute end-to-end statistics such as latency or loss in real-time. The data is crucial to determine if a network meets certain criteria, which are also referred to as Service Level Agreements (SLA). Active monitoring can also detect *non-transient* faults, which cannot be recovered by routing protocols. These faults occur when physical links fail or when network devices have faulty configurations, and can only be resolved by the intervention of a network operator.

Most active monitoring approaches are based on variants of *ping* and *traceroute*, and often rely on a set of *vantage points* (*i.e.* the devices that perform the measurements) that can be distributed across different networks.

RIPE Atlas [27] is a network measurement network that is composed of over 10 000 probes. It uses active measurements to study the reachability, the delay and the path taken by packets originating from one of their vantage points. The measurements performed by these vantage points are mostly scheduled, but it is also possible to perform user-defined measurements. These measurements can then be analyzed to detect outages. For example, Disco [32] uses the long-running TCP connections of RIPE to identify bursts of disconnections.

Trinocular [26] is another outage detection system that can use a single machine to track the connectivity of 3.4 million /24 networks. The failure to reach a destination will increase the likelihood of the address block to be down, which means that more messages are going to be sent to determine if the block is truly down, and if an outage is occurring or not.

Thunderping [31] is a measurement tool that measures the connectivity of residential Internet hosts before, during, and after forecast periods of severe weather using data from the US National Weather Service. The authors found that failures are four times more likely during thunderstorms and two times more likely during rain by applying unrelated data sources to outage detection.

NetDiagnoser [10] constructs a topology from *traceroute*-like measurements to determine if an outage is occurring by determining the state of links using binary tomography. Binary tomography can be summarized as a technique which assumes either a 'good' or 'bad' state for links, and tries to identify the smallest set of links that corresponds to the unsuccessful end-to-end measurements.

In summary, active monitoring is useful because it is a valuable method to gather instantaneous knowledge on foreign networks (i.e. those for which one cannot have access to internal privileges such as packet capture or real time routing configurations) using tools that are widely available. However, it adds an additional burden on existing networks by adding control traffic that might decrease a network's users quality of experience. Additionally, some networks might prevent one from measuring their performances through filtering in order to protect their privacy. Thus the quality of one's measurements might be hindered by factors that are outside one's control.

2.2 Passive Monitoring

Passive monitoring methods only rely on existing traffic. Generally speaking, they use the traffic-related data that is already stored or can be captured on some internal networking devices. Analyzing real user-traces ensures that the inferred statistics correspond to real

traffic, thus granting a view of a network's current state. The generated data may then be aggregated in order to extract behavioral patterns.

There are two sub-families of passive monitoring: *control plane* and *data plane*. The following sections are going to describe how these two different abstractions can be used to detect outages.

2.2.1 Control plane

Control plane traffic refers to routing-related information including the signaling in particular.

Inter-domain routing data is gathered from the Border Gateway Protocol (BGP), the *de facto* inter-domain routing protocol. Global Internet dynamics can be studied with different BGP databases such as Routeviews [29], RIPE RIS [28], or the BGP looking glasses, which provide network routing information. Indeed, the disappearance of prefixes that have been announced in BGP signifies that these prefixes cannot be reached anymore, which is indicative of a network outage.

Intra-domain routing data is gathered from Interior Gateway Protocols (IGP) such as Intermediate-System to Intermediate-System (IS-IS) or Open Shortest Path First (OSPF). Intra-domain outages can be detected when a subset of links or network devices cease to operate properly and start to have a negative effect on a network's performances. There is less public data from IGP protocols available because the competitiveness of different ISPs desensitizes sharing such information.

Control plane measurement tools include BGP eye [33], a tool for visualization-aided root-cause analysis of BGP anomalies, or I-Seismograph [18], an Internet seismograph that quantifies the impact of BGP updates on the entire Internet.

Since the control plane can be consistent and outages still occur (as it is the case for persistent ones in particular), one also needs to look at the data-plane that is the closest plane to the user experience.

2.2.2 Data plane

Data plane traffic refers to the study of the traffic that transits through the considered network.

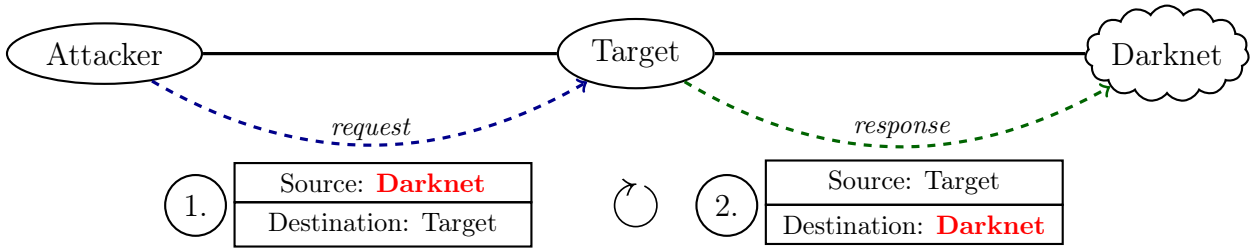
It offers a different point of view since a faulty data plane does not necessarily comes from a faulty control plane. This can take the form of non-transient faults that have been mentioned earlier. Another example would be what happened in Syria in 2011 [9], where the BGP (*i.e.* control plane) signalization did not change, even if the entire traffic had disappeared due to packet filtering mechanisms that were put in place. These matters are discussed in [6], where Randy Bush et al. highlight how data plane measurements can differ from control plane measurements.

Data plane measurement tools include FACT [30], a system that tracks connectivity problems by using *flow-level* data. A flow is defined by Augustin et al. in [25] as a set of packets that share the same *flow-identifier*, which is a 5-tuple containing the IP source address, IP destination address, protocol, source port, and destination port. Other tools such as Disco [32] or Tartiflette [13] rely on traces from RIPE Atlas [27] to measure data plane traffic without injecting additional traffic into existing networks.

An original data source that can be used to detect outages is the *Internet Background Radiation* (IBR). It consists in nonproductive traffic sent to a collection of allocated, routable, but unused IP addresses (i.e., the hosts using those IP do not trigger any communications by themselves). These IP blocks are commonly referred to as *Network Telescopes* or as *Darknets*. IBR has first been characterized in 2004 by Pang et al. [24]. More recently, it has been also studied in 2010 [36] by Benson et al. and in 2015 [5] by Wustrow et al.

The way IBR can be used to detect outages is by recording the traffic that reaches a network telescope and by looking at its properties. In practice, a network telescope can be

Figure 3: Illustration of how *backscatter* (*i.e.* spoofed) traffic can reach network telescopes.



easily setup by any ISP as a sub-block of its larger IP prefix. The sub-block thus simply consists in contiguous IP that are not — yet — used by the given ISP (neither for transit nor stub sub-networks). Then, the ISP just has to capture all the traffic destined to this sub-block. By definition, the traffic destined to such IP addresses has not been solicited by any internal hosts belonging to the ISP.

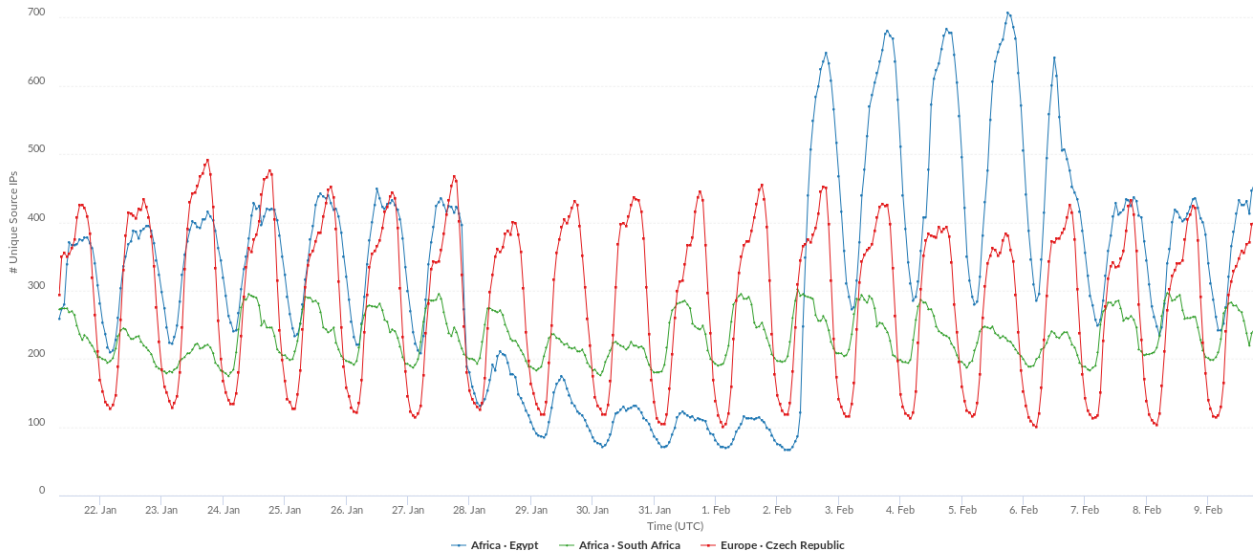
This data source can be divided into two main categories: benign and malicious traffic. Benign traffic is the result of software or hardware errors, such as bit-flipping or hard-coded IP addresses. Malicious traffic is composed of several kinds of attack attempts. It can take the form of scans that (more or less) randomly crawl through the Internet address space in the hope of finding vulnerable targets. It can also be the result of backscatter traffic (i.e. due to spoofing). Backscatter traffic is created when an attacker sends a request to a target with a fake source IP address (step 1.), which means that the response will be routed towards the fake source IP address instead (step 2.). In turn, it means that attackers that chose an IP address that belongs to a network telescope will indirectly generate traffic towards it. This process is illustrated on figure 3. Obviously, the malicious traffic can be used to study network attacks such as worm and virus propagation [35, 15, 16], or Distributed Denial of Service (DDoS) attacks [11].

A first strategy to perform an analysis at a coarser grain than packet-er packet would be to look at the flow-level information that is contained inside the inbound traffic. In [4], the authors use the retransmission behavior of TCP and the TTL[1] to determine if packets belonging to a TCP flow had been lost or if packets took a different path to reach the network telescope (*i.e.* arriving with a different TTL).

Another option is to consider where the incoming traffic is originating from. Geolocation services such as Maxmind Geolite [19] or Net Acuity [21] can be used to determine the spatial location of the inbound traffic by using its source IP address. There are multiple ways to aggregate traffic that is coming from a single location, namely the *number of bytes*, the *number of packets*, and the *number of unique source IP addresses*. The first two methods are self-explanatory. The number of unique source IP addresses [9] is defined as the number of IP addresses originating from the same location that contact the network telescope during a given time interval. This method is superior to the other two to study outages since the number of unique IP addresses will indicate *how many sources* contact the network telescope as opposed to *how much traffic* reaches it, which means that traffic bursts from the same location will be ignored while small traffic from a variety of sources will become more visible. The number of unique source IP addresses can then be used to see if the number of machines that try to reach the network telescope varies from the usual trend for this geographical area. For example, figure 4 showcases a severe network disruption that occurred during the Egyptian revolution in 2011. Three different signals of unique source IP addresses are plotted: Egypt, South Africa, and Czech Republic. These two other countries have been chosen since they generate a similar quantity of traffic and because they are on a similar timezone, which causes the day/night cycle to align itself with the other signals. The traffic

[1]TTL refers to the *time to leave* field in the IP header. It is used as a counter to control the maximal remaining number of hops of a packet before it is discarded.

Figure 4: Illustration of how the Egyptian traffic collapsed during the Egyptian revolution in 2011 while other traffic sources stayed consistent. (source: ioda.caida.org)



collapses around the 28th of January, which is when packet filtering mechanisms were put into place [9]. However, the other signals were not affected by the disruption resulting from such a brutal filtering. Indeed, these two other signals continue to generate a quantity of traffic that is aligned with the one of the previous days. Even the more general characteristics of the traffic (e.g. its variability in particular) remain almost constant. Hence, the analysis of such a traffic is both capable of identifying that an outage occurred and to localize where it happened.

Besides, the usage of IBR to detect outages is particularly pertinent since it is **pervasive**. Indeed, [36, 24] show that the amount of IBR that reaches network telescopes is considerable, incessant, and originates from a variety of applications. In [5], Benson *et al.* performed a spatial analysis where they determined that IBR provided an **Internet-wide view**. As a point of fact, all countries, except for 3 with a population lower than 4000 inhabitants, and more than half of all ASes have been observed in their dataset. Note that half of the ASes that do not show up in the dataset are small, as they only advertise a /24 prefix, while 86% of ASes advertise a /16 or more are visible in their dataset. A fifth of the invisible large ASes are unused blocks that belong to the US government.

The temporal analysis shown on figure 5 illustrates how the median time between obser-

Figure 5: Temporal analysis of how frequently different network entities contact the network telescope [5]

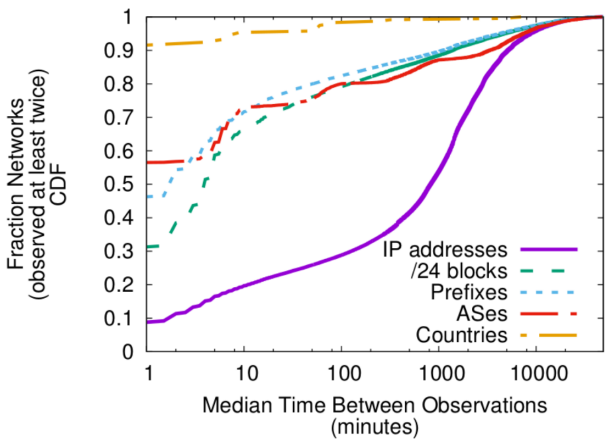
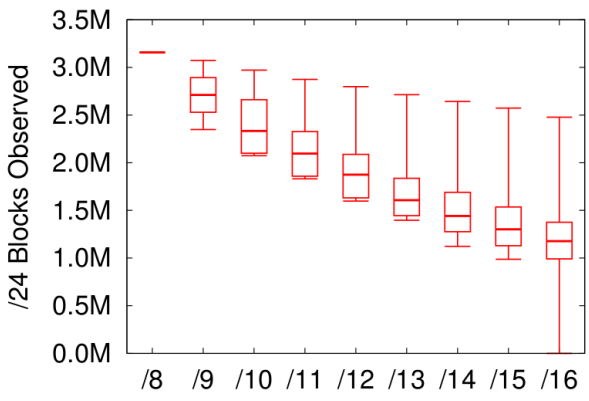


Figure 6: Study of the relationship between network telescope size and number of observed /24s [5]

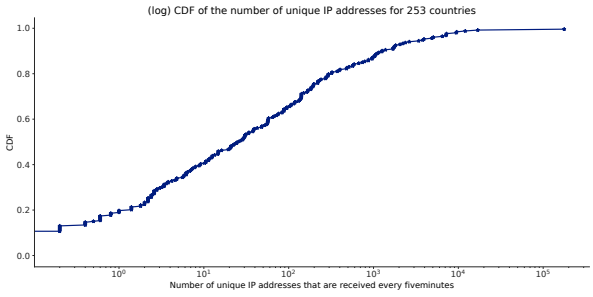


vations is lower than 1 minute for over 90% of countries, and is lower than 10 minutes for about 75% of the ASes. It means that **most networks frequently generate IBR traffic** in particular when considering a sufficient level of aggregation (i.e. a sufficiently coarse grain). It also illustrates the fact that single IP addresses do not contact the network telescope at the same frequency.

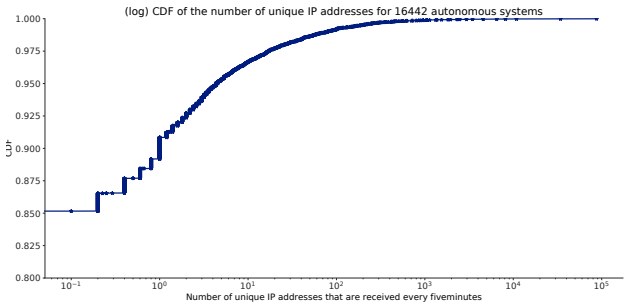
The quantity of observed source prefixes naturally depends on the size of the network telescope. Figure 6 illustrates the relationship between the size of the network telescope and the number of observed /24s. The first observation is that increasing the size of the network telescope increases the number of revealed /24s (the correlation seems sub-linear). However, this figure also illustrates the fact that two network telescopes of the same size can receive traffic from a significantly different number of /24s (as showed by the size of the boxplots). The quantitative analysis on figure 7 plots the proportion of countries and ASes that send traffic to the CAIDA UCSD network telescope [34]. This network telescope receives traffic from approximately 85% of countries over the course of a week (figure 7a). However, 85% of ASes do not send any traffic at all (figure 7b). Note that a method to collect more IBR traffic has been devised in [14], where Glatz and Dimitropoulos developed a way to record and analyze IBR on a live network by using flow-based classification.

Figure 7: CDF of the median number of unique IP addresses received every five minutes over the course of a week

(a) Countries



(b) Autonomous Systems



To summarize, IBR traffic is ubiquitous, to a point where it can be used to detect and analyze large-scale network events. Indeed, it is continually sent by a variety of sources all around the world, which makes it a great data source to make opportunistic worldwide Internet measurements. In particular, we aim to use it for efficiently detecting outages. Put in simple words, when this traffic decreases significantly, it is likely that an abnormal phenomena has occurred.

The next sections are going to describe the exact structure of the UCSD Network Telescope [34], and how we can analyze collected time series to detect outages.

3. Time series analysis and forecasting

This section gives an overview of the different methods that could be used to detect outages using IBR data using *time series analysis and forecasting*, a field focused on the extraction of statistics from time series. A time series is a sequence of data points indexed with time.

Time series are difficult to forecast because there are no causal variables associated with the data. Additionally, some components such as seasonality or random variation cannot always be easily identified.

But first, we need a way to evaluate the accuracy of our predictive model. Generally, the data is separated into two parts: the *training* set, which is used to create the model, and the *validation* set, which is used to evaluate the model. Models are evaluated using a regression

metric. There are a large number of different regression metrics, but the most well-known are the *Mean Forecast Error* (MFE), *Mean Absolute Error* (MAE), *Mean Absolute Percentage Error* (MAPE), *Mean Squared Error* (MSE), and the *Root Mean Squared Error* (RMSE). Their formal definitions are as follows:

$$MFE: \frac{1}{n}\sum_{t=1}^{n} e_t \quad MAE: \frac{1}{n}\sum_{t=1}^{n} |e_t| \quad MAPE: \frac{1}{n}\sum_{t=1}^{n} \left|\frac{e_t}{y_t}\right| \quad MSE: \frac{1}{n}\sum_{t=1}^{n} e_t^2 \quad RMSE: \sqrt{MSE},$$

Where e_t is the actual value minus the predicted value, and n is the size of the validation set. The MFE, also known as *forecast bias*, shows the direction of the error (i.e. if the prediction is above or below the actual value). The best possible error with MFE is 0, but a problem with this metric is that the error cancels itself out since positive and negative numbers are added. It results in a bias since an error of 0 does not necessarily mean that the predicted values were equal to the actual ones. Additionally, it makes the MFE score harder to interpret since the extent at which the effects of positive and negative errors canceled out is not known. In contrast, the MAE uses absolute errors instead of signed ones. As a result, an MAE score close to 0 will indicate that the predictions were very accurate. A MAE score of x is interpreted has the fact that, on average, a prediction had a distance of x to the actual value. However, the distance given the MAE score is relative to the scale of the signal, whereas the MAPE gives a percentage, which is useful is one wants to compare different signals at the same scale. The MSE is not scale independent but it adds a feature that can be beneficial: squaring e_t penalizes extreme errors, as a large distance between the real value and the prediction will be further emphasized by squaring. A problem is that a squared error is difficult to interpret, which is the reason why the RMSE is used. It keeps all the benefits of using MSE, while it keeps an interpretation similar to that of the MAE score. A summary of the differences is given in table 2.

There are several techniques that can be used to detect predict time series, including neural networks, Fourier transform, wavelet transform, and linear statistical methods. A technique that is computationally cheap is preferred over computationally expensive ones if one wants to analyze a great number of time series. As a result, neural networks are considered to be out of scope. Linear statistical methods have commonly been used for time series analysis due to their model simplicity and because of the relatively low computational power required. However, one of the necessary conditions for using linear statistical methods is that the time series needs to be stationary.

Section 3.1 is going to describe *stationarity* and Section 3.2 contains an overview of the different linear statistical methods that can be used to predict time series.

3.1 Stationarity

A stationary time series is one whose statistical properties such as mean and variance are constant over time. It can be assessed using the augmented Dickey-Fuller test. Stationarity is an assumption that is shared by multiple forecasting processes, as one can assume that

Table 2: Main regression metrics that are used to measure the accuracy of forecasts

	MFE	MAE	MAPE	MSE	RMSE
Shows the direction of the error	✓	✗	✗	✗	✗
Is scale indepedent	✗	✗	✓	✗	✗
Penalizes extreme errors	✗	✗	✗	✓	✓

its future statistical properties will be the same in the future. As a result, non-stationary time series need to be transformed into stationary ones to make accurate predictions and to be *inverted* (i.e. moved back to their original scale) to visualize the result. A conventional way to stationarize a time series is by applying seasonal differencing. A seasonal differencing process is applied when the difference between an observation and an observation from the past is computed. In other words,

$$y'_t = y_t - y_{t-d}, \tag{2.1}$$

Where y_t is the predicted value, and d indicates the order (i.e. how many observations ago) of the difference. Differencing can also be expressed with a lag (also known as *backshift*) operator:

$$\begin{aligned} Ly_t &= y_{t-1} \\ y'_t = y_t - y_{t-d} &= y_t - L^d y_t = (1 - L^d)y_t, \end{aligned} \tag{2.2}$$

Where L is a lag operator.

Once the data is differenced, it is possible to invert the operation by performing the opposite operation:

$$y_t = (1 + L^d)y'_t, \tag{2.3}$$

If the resulting time series if still not stationary it is possible to apply second-order differencing, which will result in the following:

$$y'_t = (1 - L^{d'})(1 - L^d)y_t, \tag{2.4}$$

Where d and d' are two different difference orders.

3.2 Linear statistical models

The following models are all assumed to have an independent and identically distributed error with zero mean and a constant variance σ^2, which corresponds to a typical normal distribution.

The two most common linear time series models in the literature are the *Autoregressive* (AR) model and the *Moving Average* (MA) model.

AR(p), or AR model of order p, estimates the future value of a variable to be a linear combination of p past observations and a random error in addition to a constant. Mathematically, an AR(p) model can be expressed as [2]:

$$y_t = c + \sum_{i=1}^{p} \varphi_i y_{t-i} + \varepsilon_t = c + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \cdots + \varphi_p y_{t-p} + \varepsilon_t, \tag{2.5}$$

Where y_t is the actual value, and ε_t is the random error at the time period t. $\varphi_i (i = 1, 2, \ldots, p)$ are the model parameters and c is a constant.

MA(q), or MA model of order q, use past errors as opposed to past values. An MA(q) model can be expressed as [2]:

$$y_t = \mu + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t = \mu + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t, \tag{2.6}$$

Where μ is the mean of the series, and $\theta_j (j = 1, 2, \ldots, q)$ are the model parameters.

AR and MA models can be combined to form a more general class of time series models known as ARMA models. The orders of an ARMA(p, q) model refer to p autoregressive terms and q moving average terms. An ARMA(p, q) model can be expressed as [2]:

$$y_t = c + \varepsilon_t + \sum_{i=1}^{p} \varphi_i y_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} \tag{2.7}$$

ARMA models are usually manipulated using the lag (or backshift) operator notation. It is defined as $Ly_t = y_{t-1}$. Polynomials of lag operators or lag polynomials can represent ARMA models as follows [2]:

$$\begin{aligned} \text{AR}(p) \text{ model: } & \varepsilon_t = \varphi(L) y_t \\ \text{MA}(q) \text{ model: } & y_t = \theta(L) \varepsilon_t \\ \text{ARMA}(p, q) \text{ model: } & \varphi(L) y_t = \theta(L) \varepsilon_t, \end{aligned} \tag{2.8}$$

Where $\varphi(L) = 1 - \sum_{i=1}^{p} \varphi_i L^i$ and $\theta(L) = 1 + \sum_{j=1}^{q} \theta_j L^j$.

The models described above can only be used for stationary time series. In practice, a lot of time series exhibit non-stationary behavior, which is the reason why the *Autoregressive Integrated Moving Average* (ARIMA) model was proposed. It is a generalization of an ARMA model that includes the case of non-stationary time series by applying seasonal differencing of the data points.

Mathematically, the formulation of an ARIMA(p, d, q) model using lag polynomials is as follows [2]:

$$\begin{gathered} \left(1 - \sum_{i=1}^{p} \varphi_i L^i\right)(1 - L^d) y_t = \left(1 + \sum_{j=1}^{q} \theta_j L^j\right) \varepsilon_t, \\ i.e. \varphi(L)(1 - L^d) y_t = \theta(L) \varepsilon_t \end{gathered} \tag{2.9}$$

Here, d controls the level of differencing. $d = 0$ reduces to an ARMA(p, q) model. Note that the seasonal difference applied in ARIMA is always of first order.

Finally, the *Seasonal ARIMA* (SARIMA) model generalized ARIMA to deal with seasonality. A time series that exhibits seasonality will contain variations that occur at specific regular time intervals. For example, temperatures have both a daily seasonality because it is colder during the night and a yearly seasonality because it is warmer during summer. SARIMA models are generally termed as SARIMA$(p, d, q) \times (P, D, Q)^s$, where:

- (p, d, q) are the parameters of the *non-seasonal* part of the model
- (P, D, Q) are the parameters of the *seasonal* part of the model
- s is the number of observations that corresponds to one season

Note that setting (P, D, Q) to 0 reduces to an ARIMA(p, d, q) model.

The formulation of a SARIMA$(p, d, q) \times (P, D, Q)^s$ model using lag polynomials can be found in [2].

Chapter 3 Contributions

The goal of this chapter is to describe a novel outage detection technique that leverages Internet Background Radiation. This solution can study thousands of different time series in real-time to detect outages using statistical forecasting methods on the number of unique source IP addresses. The network telescope that was used to collect the IBR data used in this chapter is the UCSD network telescope[1], which is operated by CAIDA.

Section 1 is going to describe how this data set is structured, and will describe the data that can be extracted from it. Section 2 discusses the current method used to detect outages. Section 3 describes how I have modeled the data, and Section 4 explains how this model can be used to detect outages. Section 5 summarizes and evaluates the obtained results, and Section 6 critiques them.

1. Data set

CAIDA's data set contains more than 60 000 time series representing the number of unique source IP addresses that contacted their network telescope. These time series can be divided into four categories:

- Continents
- Countries
- Regions
- Autonomous systems

The first three have been gathered using the IP geolocation service Net Acuity [21], while the last one has been created using an ASN lookup. An important note is that some time series can be included inside others. For example, the time series of all the different regions of a country will be aggregated to form the time series of this country.

At the temporal level, they have been gathering data for more than 10 years. Data can also be sampled at different time intervals: one point of data every minute, 5 minutes, hour, or day. In [8], they describe how they used anti-spoofing heuristics and noise reduction filters. To remove large-scale spoofing, they looked for sudden spikes in the number of unique source IP addresses, and they applied a set of filters to remove what they identified as spoofed traffic. Some of these filters include unassigned protocol numbers, IP addresses ending in .0 or .255, or packets where the IP address originated from a network telescope.

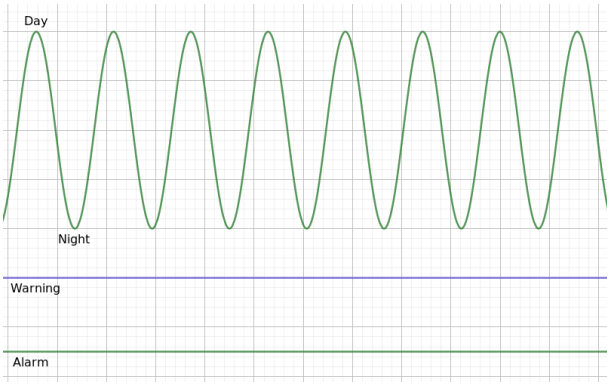
2. Current detection method

The method that CAIDA currently uses to detect outages is to use a history sliding window of a week to compute two linear thresholds: one for warnings (25%), and one for

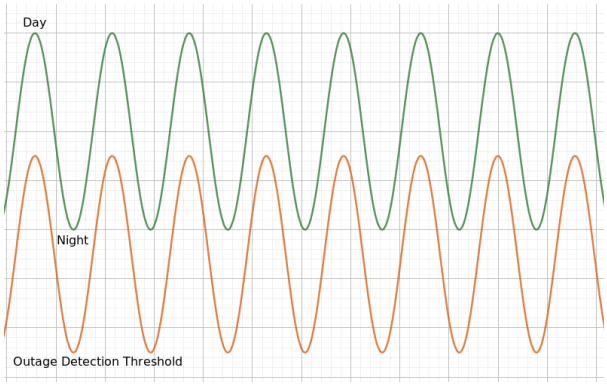
[1] https://www.caida.org/projects/network_telescope/

Figure 8: Overview of two different techniques that can be used to detect sudden drops on a time series

(a) Fixed thresholds



(b) Dynamic thresholds



critical errors (10%). This process is illustrated on figure 8a. The main drawback of this technique is that it does not adapt to the analyzed signal. On figure 8a, the green signal has a clear day and night seasonality, but the fixed thresholds that CAIDA use do not account for it. As a result, an outage occurring during the day (the local maxima) will be harder to detect that an outage occurring during the night.

The technique developed in this document aims at having a detection algorithm closer to the one presenter on figure 8b, in which the different characteristics of the time series will be accounted for. As a result, our technique will be able to use the different statistical properties of the time series to create a dynamic threshold instead of a constant one.

3. Time series modeling

This section describes how time series analysis and forecasting techniques were used to model time series of the number of unique source IP addresses.

The solution that I have chosen to use is SARIMA, since it is one of the more commonly used techniques for time series analysis and because its low computing time. Indeed, the objective of this solution is to analyze thousands of time series in real-time, and it needs to have a low computing time to achieve that. A viable alternative would have been to use signal processing techniques such as wavelet transforms but they are much more complex than ARMA-based techniques, which is why SARIMA has been chosen.

The time series on figure 9 will be used as a common thread to illustrate the different steps involved in time series modeling. It showcases how the amount of IP addresses from Egypt evolved during the Egyptian revolution in 2011. The dashed black vertical lines indicate when the government censored the Internet in response to the protests that were happening [9]. This case is also interesting because the outage is followed by a time period in which the statistical properties of the time series differ from normality before going back to their usual state. Here, it amounts to an increase in mean and variance.

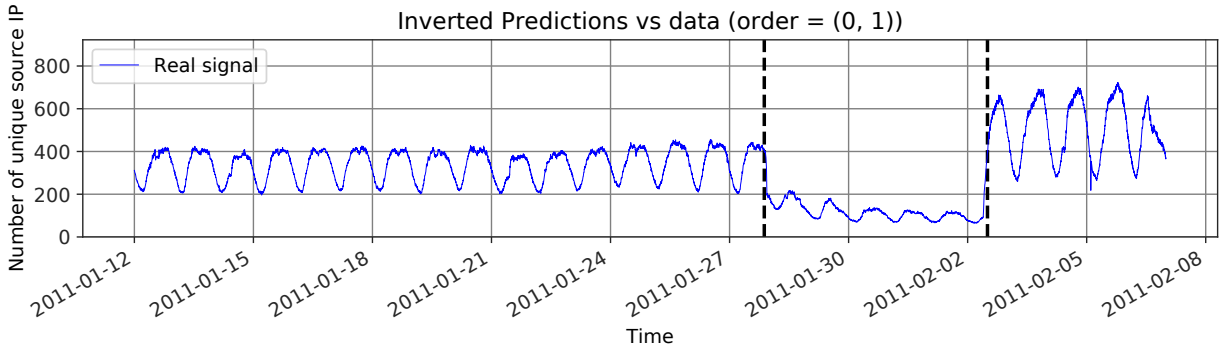
3.1 Preprocessing

The preprocessing part ensures that the time series is suited for predictions. The analyzed time series are separated into three different sets:

1. Training set: the set that will be fed to the ARMA model to create the model,

2. Validation set: the set that will be used to find the best p and q for the ARMA model,

Figure 9: Time series of the evolution of the number of unique IP addresses coming from Egypt over time



3. Test set: the set that will be analyzed using the model found during the validation period.

The separation of the different sets is illustrated by the blue dashed vertical lines on figure 10.

The training set serves as a baseline for our future predictions. As a result, we need to ensure that it is representative of the normal state of the time series, and that it does not contain any events that might bias the predictive model. We apply two different cleaning strategies to the training set.

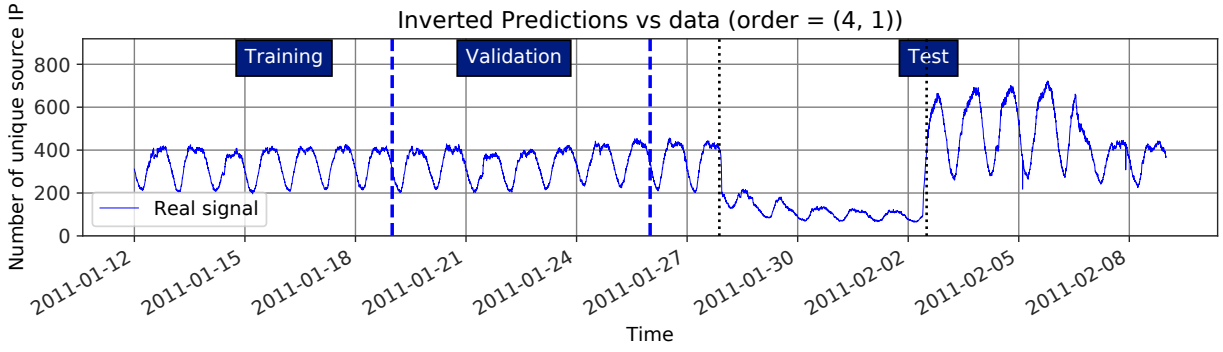
The first one consists of replacing the training set by the result of the *Median Absolute Deviation* (MAD) of the training set and of the previous weeks. MAD is a robust non-parametric measurement of the statistical dispersion that will remove outliers from the data. It also has the added benefit of filling missing data in the training set by using data from the time series, as the alternative would be to use interpolation methods that do not necessarily account for the different trends present in the data.

The second strategy is to normalize the median and the variance of the training set to correspond to that of the validation set. By doing so, we ensure that the evaluation of the quality of the ARMA model is going to be as accurate as possible. The normalization is executed on the validation set and not on the test set because the test set is not supposed to be known in advance, as any adjustment to resemble the test set would bias our results.

As stated in Section 3.2, time series need to be stationary in order to obtain good predictions with ARMA-based models. As a result, we now need to work with two different time series: the *real* one, that contains the original data, and the *differenced* one, that is going to be stationarized. Here, the time series presented on figure 10 is not stationary for the following reasons:

1. Inconsistent mean over time

Figure 10: Training, validation, and test set separation on the Egyptian dataset

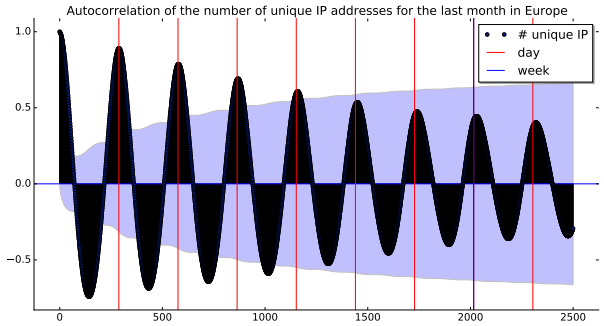


2. Inconsistent variance over time

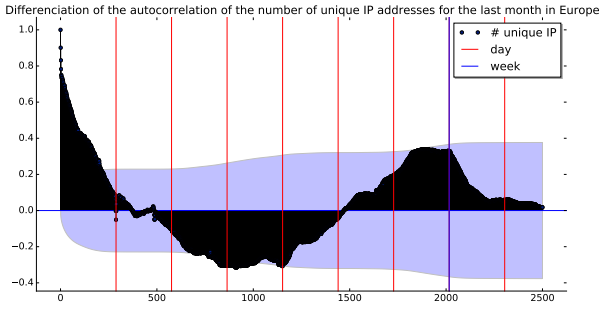
3. Presence of a trend in the data (day/night cycle)

Figure 11: Autocorrelation functions of the European time series [remove conf int]

(a) No finite difference



(b) Finite difference of a day



Stationarity can be achieved through finite differencing. A handy way to identify trends in the data is to look at the *Autocorrelation Function* (ACF) of a time series. An ACF plots the correlation between points separated by various time lags. As a result, a trend of n points in the original time series will be seen as a positive correlation on the ACF. Figure 11a is the ACF plot of 8 days of the European time series. The blue cone is a 95% confidence interval. The data is sampled at a granularity of 5 minutes, which means that a day amounts for 288 data points and that a week amounts for 2016 of them. The red vertical lines highlight when a day has passed, and the blue line highlights when a week passed. The fact that the autocorrelation is maximal after exactly one day teaches us that this time series contains a daily trend. To remove it, we can apply a finite difference of a day to the time series and plot a second ACF to see if a trend is still present in the data. Figure 11b shows this second ACF. The daily trend is gone since local maxima do not appear every day anymore, but this figure teaches us that a weekly trend is still present in the data. Once again, it is possible to apply a finite difference of a week to the original time series, and to plot its ACF. The correlation now oscillates around 0, which implies that there is no significant trend in the data anymore. It is fair to assess that there might be longer trends such as monthly or yearly. However, one of the underlying assumption of finite differencing is that the data from a cycle ago is similar to the one in the current period. That hypothesis seems reasonable for a duration of a week but the quick changes that networks and the Internet undergo makes it harder to justify the usage of a longer trend.

Let's now address the normalization of the mean and the variance. A finite difference of an order of a week often stabilizes both of these statistical properties. Indeed, moving a constant time series with a weekly trend by a week results in the same time series.

Once the finite difference has been applied, it is possible to determine if the resulting time series has become stationary using the augmented Dickey-Fuller test. If the time series is still not stationary, it is possible to apply additional finite differences until it becomes stationary.

At this point, the real time series has a clean training set, and the differenced time series is stationary. It is now possible to look for the best ARMA model.

3.2 Model evaluation and selection

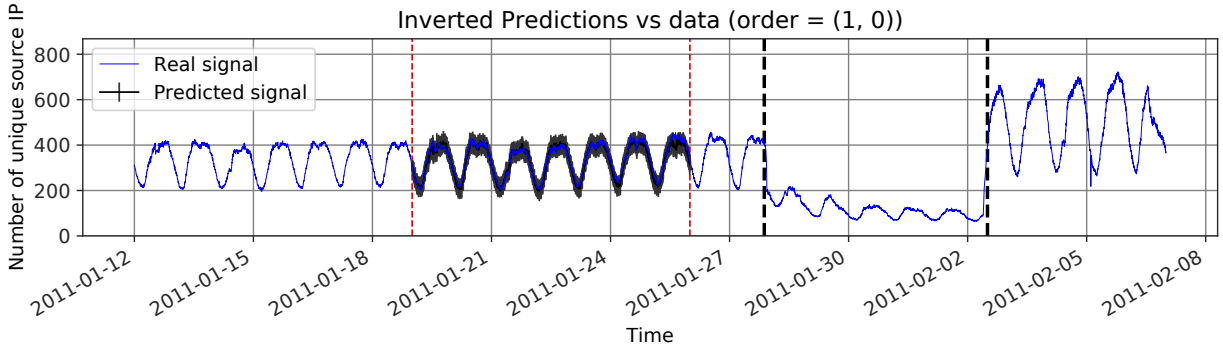
The goal of this section if to find the best p and q parameters to use for the ARMA model. As a reminder, increasing p or q will lead to an increase in the number of autoregressive or moving average that are going to be used to make an accurate prediction. Predictions are more accurate on a stationary data set, which means that the differenced time series is going to be used for this section.

The selected model is going to be the one with the lowest RMSE, which was described in Section 3. The reason why this regression metric has been chosen is because it penalizes extreme errors, which means that models with predictions far from the truth are going to have a worse result that models that are close to reality. Lastly, predictions are going to be computed using *walk-forward validation*, which is a process where the following steps are executed:

1. An $ARMA(p, q)$ model is going to be created with a list of values stored in a variables called *history*,
2. This model is going to be used to compute predictions for the next set of points,
3. The real values of this set of points are going to be included inside the *history* variable,
4. Repeat from step 1 until the entire data set has been iterated through, then compute the RMSE of the real values versus the predicted values.

Walk-forward validation is used because we want to evaluate the capacity of our model to predict a few values rather than its capacity to predict multiple days at a time. Indeed, having a good short-term prediction makes it possible to detect sudden changes, i.e. outages. Additionally, the fact that our model is constantly fed with new data (step 3) will ensure that our model will learn from the new values of the time series, and that the predictive process can be used for a very long time. Once the predictions are made, we end up with results like the ones presented on figure 12, where the black line and the gray interval represent the predicted values and their confidence interval. A better model with a lower RMSE will lead to a smaller confidence interval.

Figure 12: Predictions obtained on the validation set using walk-forward validation



3.3 Preserving the integrity of the model

As stated earlier, the training set has been cleaned using data from the preceding weeks. However, the following problems still remain for the validation and the test set:

1. Missing values need to be filled to have a valid history of values that will be used to create future models
2. Extreme values will bias future predictions
3. Incorporating outages into the history of the model will bias the future predictions

The solution that has been used to solve is three problems is *inpainting*. Inpainting is the action of adding the predictions of the ARMA model inside the history of values that are used to create to model as opposed to missing or extreme values. Indeed, doing so ensures that local biases will not affect future data.

Now that we know the best model for a given time series, it is possible to try to analyze it and to try to detect outages. We consider that an outage is occurring at a time t if the current point on the real time series is lesser than the lower bound of the confidence interval of the prediction. If an outage is detected, a vertical line is going to be displayed on the figure. The different colors of the vertical lines represent different thresholds that are relative to the following distance:

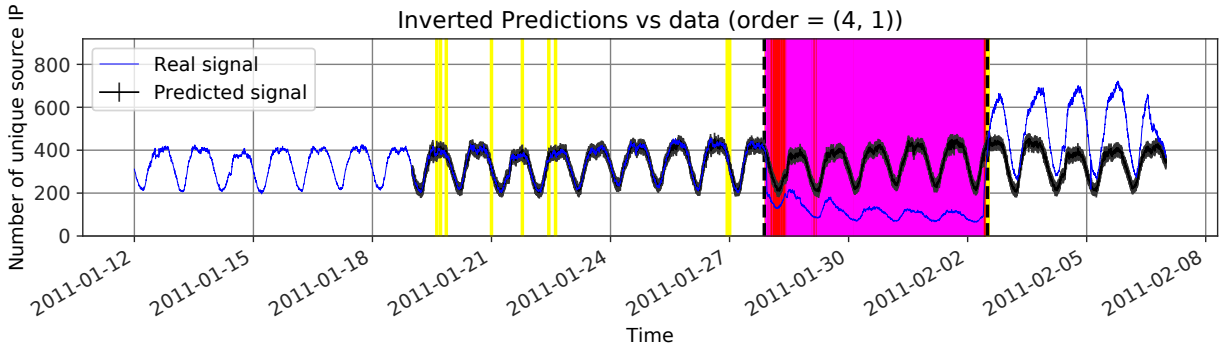
$$distance = (predicted - real)/(predicted - lower), \tag{3.1}$$

Where predicted is the predicted value, real is the actual value, and lower is the value of the lower bound of the confidence interval of the predicted value. A distance greater than 1 means that the real value is outside of the confidence interval of the predicted value. The different colors correspond to the following thresholds:

- Yellow: $1 < t < 1.5$
- Orange: $1.5 < t < 2$
- Red: $2 < t < 5$
- Magenta: $t > 5$

The result of our outage detection algorithm can be seen on figure 13.

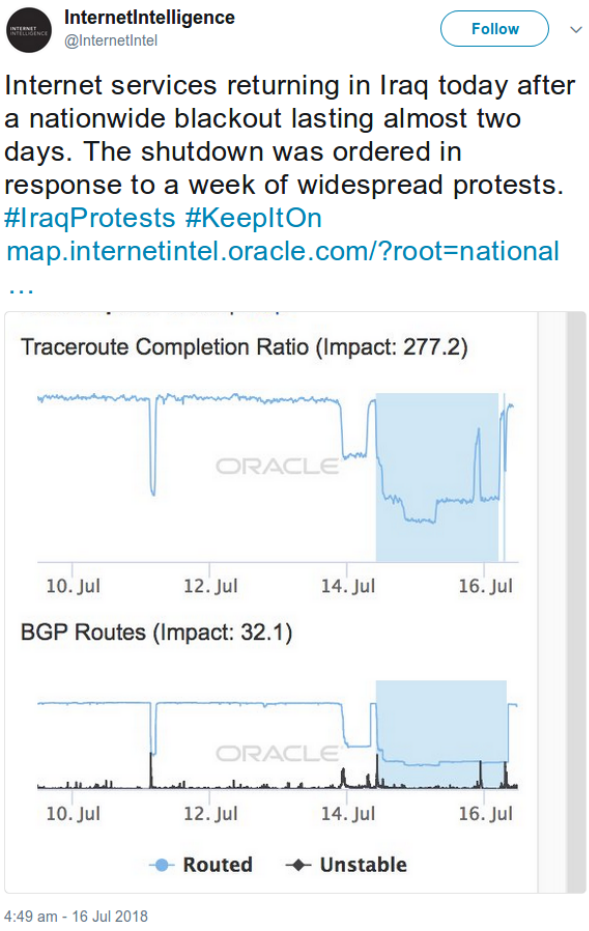
Figure 13: Outage detection during the Egyptian revolution



Here, the entirety of the outages is covered by alarms from our outage detection algorithm, as the predicted values are far above the real values. The period that follows the outage is not considered to be an outage because the real values are greater than the predicted ones. The few colored vertical lines that are outside of the outage period are considered to be false alarms. Lastly, it is interesting to note that the times where alarms were raised (i.e. alarms) will not be integrated into the history of the model, as the inpainting will integrate the predicted values instead. As a result, the outage is not integrated inside the data, which can be seen by the fact that the predicted time series does not decrease a week after the start of the outage.

Now that we know how to detect outages, it is possible to look at what information we extract from these time series and how their predictions are evaluated.

Figure 14: Example of a tweet announcing an outage (source)



5. Results

This section describes how we gathered outages to form a ground truth, and how these events were then analyzed to evaluate the quality of our outage detection algorithm.

Our ground truth is composed of 18 different test cases. Elements from the ground truth were gathered from the literature [9] and from Internet Intelligence [22], Oracle's twitter feed for network events. These feeds inform us where and when a potential outage is occurring. An example can be found on figure 14, where an analysis of traceroute measurements and of BGP routes revealed that the Iraqi government censored the Internet in response to protests.

An event can only be divided into ground truth events. For example, the Egyptian revolution is separated into several events that have different spatial granularities, namely country, regions, and autonomous systems, and that have a different number of unique source IP addresses per 5 minutes.

These different events try to evaluate our solution on different aspects, namely:

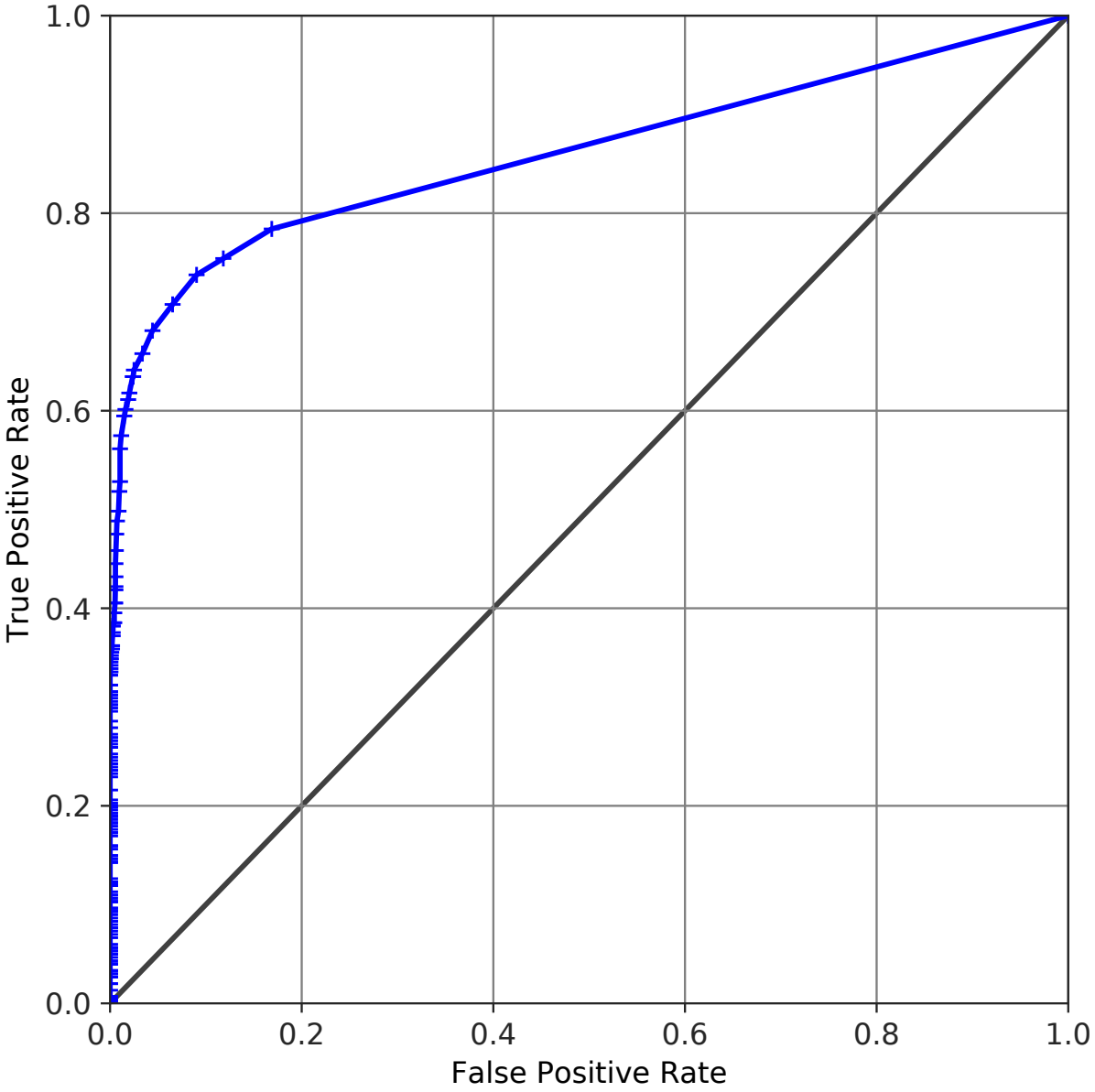
1. Time series with a low number of unique source IP addresses,
2. Time series with extreme values in the data,
3. Time series with missing values.

A complete breakdown of the different events and of their results can be found in Section 1 of the appendix.

Our solution is then evaluated using a *Receiver Operating Characteristic* (ROC) curve, which evaluates a binary classifier by plotting the *True Positive Rate* (TPR) over the *False Positive Rate* (FPR). Here, a true positive is defined as a 1 hour time bin that is part of an

outage and that contains at least one alarm. A false negative is a 1 hour time bin that is not part of an outage and that contains at least one alarm. In other words, we measure the ability of our solution to identify each hour of an outage as being part of an outage. The ROC curve is presented on figure 15. The reason why there are only thresholds on the leftmost part of the ROC curve is because I have only plotted thresholds that were already outside of the 99% confidence interval. Some interesting intervals include the fact that 63% of 1 hour time bins containing outages are identified as such with less than 1% of false positives, and that 79% of 1 hour time bins containing outages are identified as such with about 18% of false positives. These results depend on the way the ground truth has been constructed, which is going to be expanded upon in the discussion section.

Figure 15: ROC curve of the ability of our classifier to detect outages



6. Discussion

This section describes the limitations of our solution, and what could be improved.

The part that needs to be improved the most is the ground truth. I originally constructed this dataset for testing purposes, which means that I intentionally included edge cases that

are difficult to predict. As a result, the ROC curve presented on figure 15 might be biased by the fact that the outages contained inside the ground truth might not be representative of the majority of outages. For example, some examples have a number of unique source IP addresses received per 5 minutes that is lesser than 10, which is clearly a case where we have to wonder if there is enough data to make meaningful predictions. However, I included a lot of them because I have wanted to be able to determine what is 'too little', and I figured that the best way to quantify this would be to experiment with more time series. In order to create a realistic ground truth, I intend to systematically add every event from a dataset such as Internet Intelligence [22], and I plan to make multiple ROC curves with different classes of time series. Time series will be distributed into classes based on the median of the number of unique source IP addresses.

I also intend to analyze each outage from different spatial perspectives. The purpose is twofold. Firstly, it would allow me to be able to compare how the accuracy of my solution is impacted by a lower amount of unique source IP addresses. Secondly, it would allow me to try to localize outages more precisely. An example is the Brazilian electricity outage that is described in Section 1. The news outlets taught us that the outage occurred in the northern part of the country, and I would like to determine if my solution can precisely detect which regions were affected, and which were not.

Lastly, one of the current problems is that the evaluation of my solution did not come with a comparison to already existing outage detection tools. CAIDA recently provided us with a dataset that regroups outages captured from multiple data sources (namely pings, and BGP data) and I would like to systematize the analysis of all of these events to see which types of events are detected by my solution, as it might give us insight into what would be interesting to add to our solution in the future.

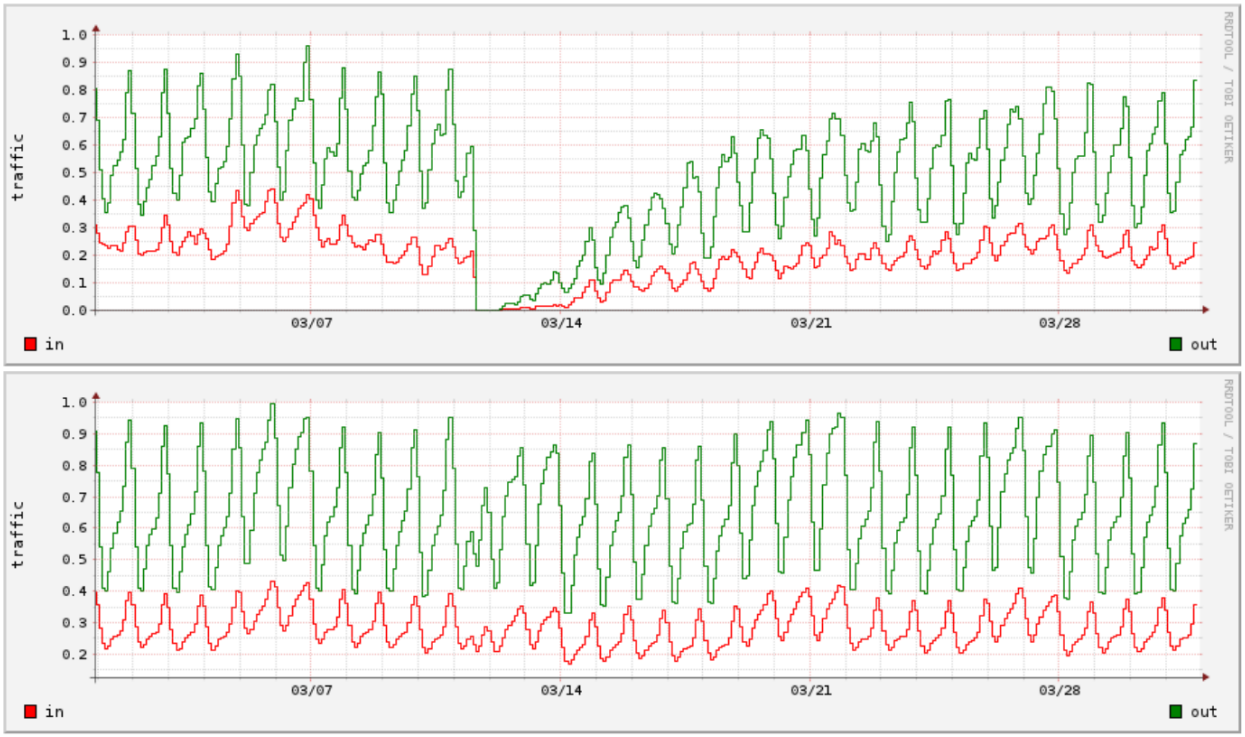
Chapter 4 Conclusion

The goal of this chapter is to discuss my internship as a whole. Section 1 will give a summary of my contributions. Section 2 will propose some perspectives for future additions. Finally, Section 3 will contain my personal feedback for this internship.

1. Contributions

This document introduces a technique that uses time series analysis to detect outages using *Internet Background Radiation*, a data set that records unsolicited traffic sent to unused collections of IP addresses. To do so, time series have been modeled using ARMA-based techniques, which are linear statistical methods. Robust statistical methods have then been used to get a prediction interval that was used to determine if an outage was occurring or not. However, this solution is so generic that it can be applied to any data set that can be modeled as a time series. For example, figure 16 illustrates how the Japanese Internet infrastructure was impacted by the 2011 earthquake. Indeed, the traffic variations can also be represented as time series with different trends that could be analyzed.

Figure 16: Japanese traffic for the March 2011 earthquake, Miyagi prefecture (top) and nationwide (bottom) [7]



2. Perspectives

Now that our outage detection model has been developed, It is expansively evaluate it by trying different scenarios. It will then be time to compare it with different outage detection techniques to see what kind of outages are detected by our solution, and how we can try to capitalize on it. Ideally, it would be nice to correlate our results with different data sources to create a monitoring system capable of leveraging multiple data sources to detect outages more efficiently. It might also be interesting to expand to different data sets such as the one presented on figure 16, or to active probing measurements such as latency or loss ratio.

3. Feedback

This internship has been enlightening at both a professional and a personal level. As an aspiring researcher, the opportunity to do research in a different team was extremely useful as it gave me a new look on what research could be. I was blessed with the chance to be supervised by very different researchers from very different worlds.

The only regret that I have is that I cannot help but feel like I could have done much more, and that the internship ended too soon. However, I realize that this feeling is very common in research. I would like to take this opportunity to renew my thanks to the many people that were involved in my internship.

Glossary

AR Autoregressive model. TODO description. 13, 14

ARIMA Autoregressive Integrated Moving Average: A statistical method used for time series analysis and forecasting.. 14

ARMA Autoregressive Moving Average. TODO description. 14, 16–19, 25

AS Autonomous System: A set of IP routers under a single administrative authority.. 10, 11

BGP Border Gateway Protocol: The *de facto* inter-domain routing protocol.. 8

DDoS Distributed Denial of Service: attack in which multiple compromised computer systems attack a target and cause a denial of service for users of the targeted resource.. 9

IBR Internet Background Radiation: A data source that contains a list of packets that reached a darknet.. 3, 8, 10, 11, 15

IGP Interior Gateway Protocol: A routing protocol that is used within an AS. Most common examples: RIP, OSPF, and IS-IS.. 8

IP Internet Protocol: The main communications protocol that allows a host to reach a destination solely based on its IP address.. 9

IS-IS Intermediate-systems to intermediate-systems: A link-state routing protocol used as an IGP.. 8

ISP Internet Service Provider: Company that provides customers with Internet access.. 8, 9

MA Moving Average model. TODO description. 13, 14

NLP Natural Language Processing: area of computer science and artifical intelligence concerned with the interactions between computers and human (natural) languages. (definition from Wikipedia). 6

OSPF Open Shortest Path First: A link-state routing protocol used as an IGP.. 8

RMSE Root Mean Squared Error: regression metric used to determine the accuracy of a predictive model.. 12, 19

SARIMA Seasonal Autoregressive Integrated Moving Average. TODO description. 14, 16

SLA Service Level Agreements: Contract between an Internet Service Provider and an end user that defines the level of service expected from the service provider.. 7

TCP Transmission Control Protocol: a connection-oriented transport layer protocol.. 9

Bibliography

[1] Giuseppe Aceto, Alessio Botta, Pietro Marchetta, Valerio Persico, and Antonio Pescapé. A comprehensive survey on internet outages. *Journal of Network and Computer Applications*, 2018.

[2] Ratnadip Adhikari and RK Agrawal. An introductory study on time series modeling and forecasting. *arXiv preprint arXiv:1302.6613*, 2013.

[3] Ritwik Banerjee, Abbas Razaghpanah, Luis Chiang, Akassh Mishra, Vyas Sekar, Yejin Choi, and Phillipa Gill. Internet outages, the eyewitness accounts: Analysis of the outages mailing list. In *International Conference on Passive and Active Network Measurement*, pages 206–219. Springer, 2015.

[4] Karyn Benson, Alberto Dainotti, Kimberly C Claffy, and Emile Aben. Gaining insight into as-level outages through analysis of internet background radiation. In *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*, pages 447–452. IEEE, 2013.

[5] Karyn Benson, Alberto Dainotti, Alex C Snoeren, Michael Kallitsis, et al. Leveraging internet background radiation for opportunistic network analysis. In *Proceedings of the 2015 Internet Measurement Conference*, pages 423–436. ACM, 2015.

[6] Randy Bush, Olaf Maennel, Matthew Roughan, and Steve Uhlig. Internet optometry: assessing the broken glasses in internet reachability. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement*, pages 242–253. ACM, 2009.

[7] Kenjiro Cho, Cristel Pelsser, Randy Bush, and Youngjoon Won. The japan earthquake: the impact on traffic and routing observed by a local isp. In *Proceedings of the Special Workshop on Internet and Disasters*, page 2. ACM, 2011.

[8] Alberto Dainotti, Karyn Benson, Alistair King, Michael Kallitsis, Eduard Glatz, Xenofontas Dimitropoulos, et al. Estimating internet address space usage through passive measurements. *ACM SIGCOMM Computer Communication Review*, 44(1):42–49, 2013.

[9] Alberto Dainotti, Claudio Squarcella, Emile Aben, Kimberly C Claffy, Marco Chiesa, Michele Russo, and Antonio Pescapé. Analysis of country-wide internet outages caused by censorship. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 1–18. ACM, 2011.

[10] Amogh Dhamdhere, Renata Teixeira, Constantine Dovrolis, and Christophe Diot. Netdiagnoser: Troubleshooting network unreachabilities using end-to-end probes and routing data. In *Proceedings of the 2007 ACM CoNEXT conference*, page 18. ACM, 2007.

[11] Claude Fachkha, Elias Bou-Harb, and Mourad Debbabi. Fingerprinting internet dns amplification ddos activities. In *New technologies, mobility and security (NTMS), 2014 6th international conference on*, pages 1–5. IEEE, 2014.

[12] Markus Fiedler, Tobias Hossfeld, and Phuoc Tran-Gia. A generic quantitative relationship between quality of experience and quality of service. *IEEE Network*, 24(2), 2010.

[13] Romain Fontugne, Cristel Pelsser, Emile Aben, and Randy Bush. Pinpointing delay and forwarding anomalies using large-scale traceroute measurements. In *Proceedings of the 2017 Internet Measurement Conference*, pages 15–28. ACM, 2017.

[14] Eduard Glatz and Xenofontas Dimitropoulos. Classifying internet one-way traffic. In *Proceedings of the 2012 Internet Measurement Conference*, pages 37–50. ACM, 2012.

[15] Uli Harder, Matt W Johnson, Jeremy T Bradley, and William J Knottenbelt. Observing internet worm and virus attacks with a small network telescope. *Electronic Notes in Theoretical Computer Science*, 151(3):47–59, 2006.

[16] Félix Iglesias and Tanja Zseby. Entropy-based characterization of internet background radiation. *Entropy*, 17(1):74–101, 2014.

[17] Ethan Katz-Bassett, Harsha V Madhyastha, John P John, Arvind Krishnamurthy, David Wetherall, and Thomas E Anderson. Studying black holes in the internet with hubble. In *NSDI*, volume 8, pages 247–262, 2008.

[18] Jun Li and Scott Brooks. I-seismograph: Observing and measuring internet earthquakes. In *INFOCOM, 2011 Proceedings IEEE*, pages 2624–2632. IEEE, 2011.

[19] Maxmind geolite. https://www.maxmind.com/en/home. Accessed: 02/07/2018.

[20] Pascal Merindol, Pierre David, Jean-Jacques Pansiot, Francois Clad, and Stefano Vissicchio. A fine-grained multi-source measurement platform correlating routing transitions with packet losses. *Computer Communications*, 2018.

[21] Net acuity. https://www.digitalelement.com/geolocation/. Accessed: 02/07/2018.

[22] Internet intelligence. https://twitter.com/internetintel?lang=en. Accessed: 15/08/2018.

[23] Outages. https://puck.nether.net/mailman/listinfo/outages. Accessed: 22/05/2018.

[24] Ruoming Pang, Vinod Yegneswaran, Paul Barford, Vern Paxson, and Larry Peterson. Characteristics of internet background radiation. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 27–40. ACM, 2004.

[25] Cristel Pelsser, Luca Cittadini, Stefano Vissicchio, and Randy Bush. From paris to tokyo: On the suitability of ping to measure latency. In *IMC*, 2013.

[26] Lin Quan, John Heidemann, and Yuri Pradkin. Trinocular: Understanding internet reliability through adaptive probing. In *ACM SIGCOMM Computer Communication Review*, volume 43, pages 255–266. ACM, 2013.

[27] Ripe atlas. https://atlas.ripe.net/. Accessed: 27/04/2018.

[28] Ripe ris. https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris. Accessed: 27/04/2018.

[29] University of oregon route views project. http://www.routeviews.org/routeviews/. Accessed: 27/04/2018.

[30] Dominik Schatzmann, Simon Leinen, Jochen Kögel, and Wolfgang Mühlbauer. Fact: flow-based approach for connectivity tracking. In *International Conference on Passive and Active Network Measurement*, pages 214–223. Springer, 2011.

[31] Aaron Schulman and Neil Spring. Pingin'in the rain. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 19–28. ACM, 2011.

[32] Anant Shah, Romain Fontugne, Emile Aben, Cristel Pelsser, and Randy Bush. Disco: Fast, good, and cheap outage detection. In *Network Traffic Measurement and Analysis Conference (TMA), 2017*, pages 1–9. IEEE, 2017.

[33] Soon Tee Teoh, Supranamaya Ranjan, Antonio Nucci, and Chen-Nee Chuah. Bgp eye: a new visualization tool for real-time detection and analysis of bgp anomalies. In *VizSEC*, 2006.

[34] Ucsd network telescope. https://www.caida.org/projects/network_telescope/. Accessed: 04/07/2018.

[35] Qian Wang, Zesheng Chen, and Chao Chen. Darknet-based inference of internet worm temporal characteristics. *IEEE Transactions on Information Forensics and Security*, 6(4):1382–1393, 2011.

[36] Eric Wustrow, Manish Karir, Michael Bailey, Farnam Jahanian, and Geoff Huston. Internet background radiation revisited. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 62–74. ACM, 2010.

Appendix

1. Results

Figure 17: Egypt - entire country

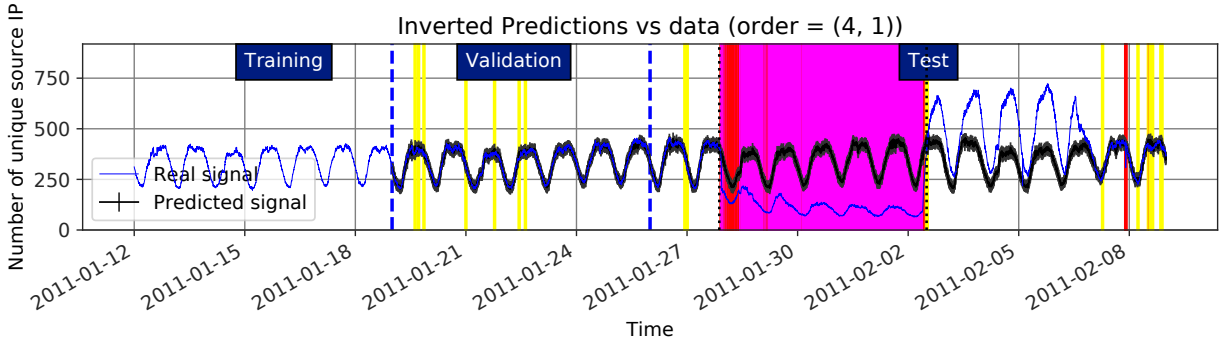


Figure 18: Egypt - AS 8452

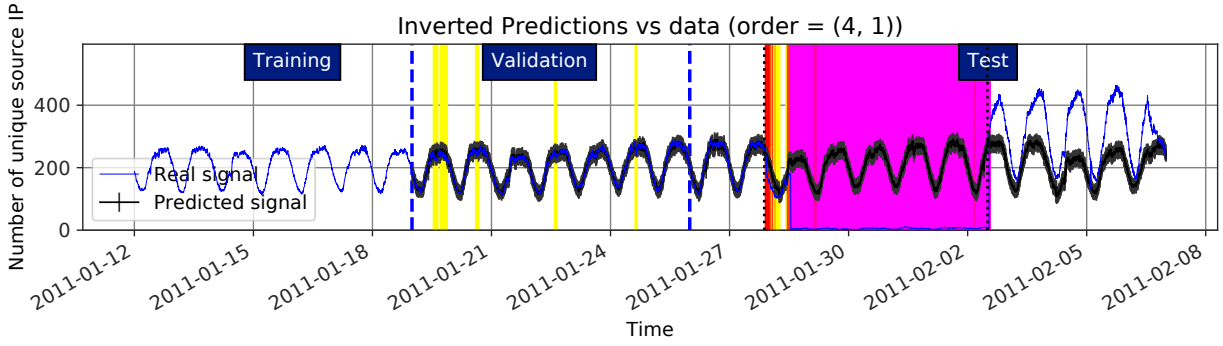


Figure 19: Egypt - Al Jizah

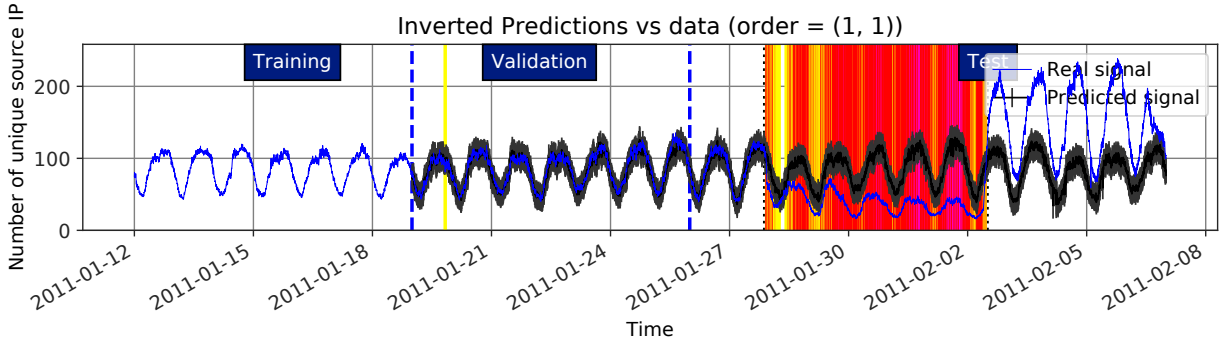


Figure 20: Egypt - AS 24863

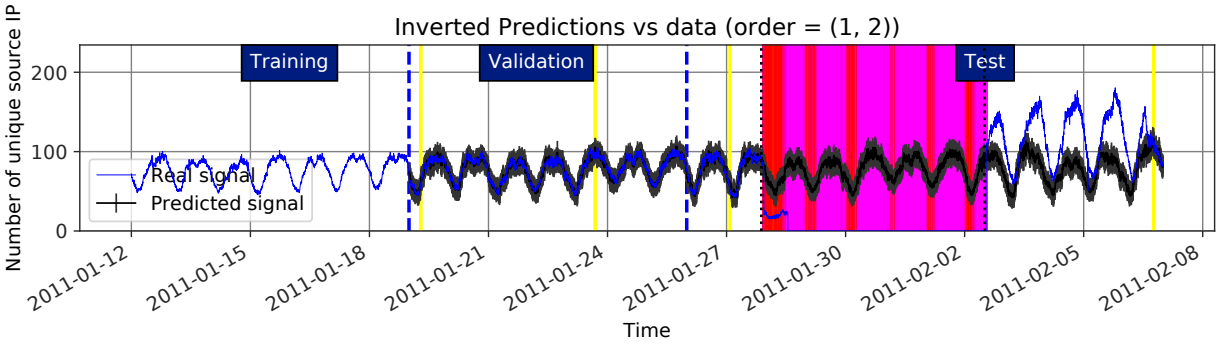


Figure 21: Egypt - AS 36992

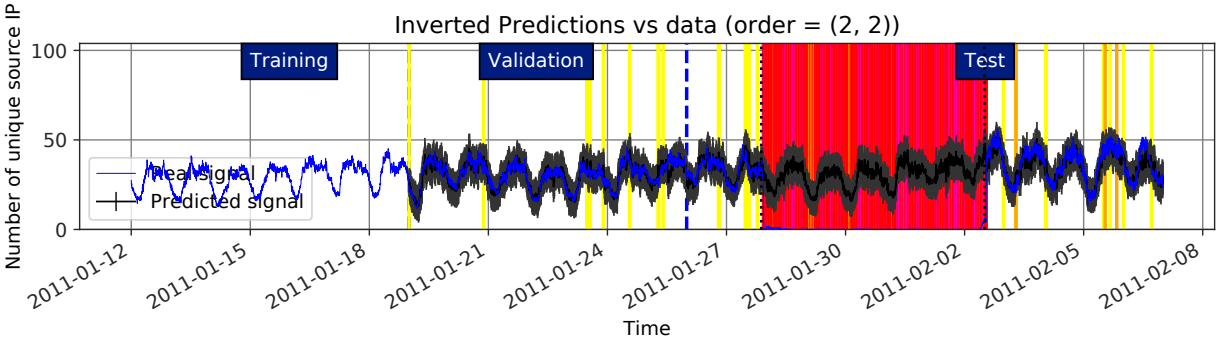


Figure 22: Egypt - Al Gharbiyah

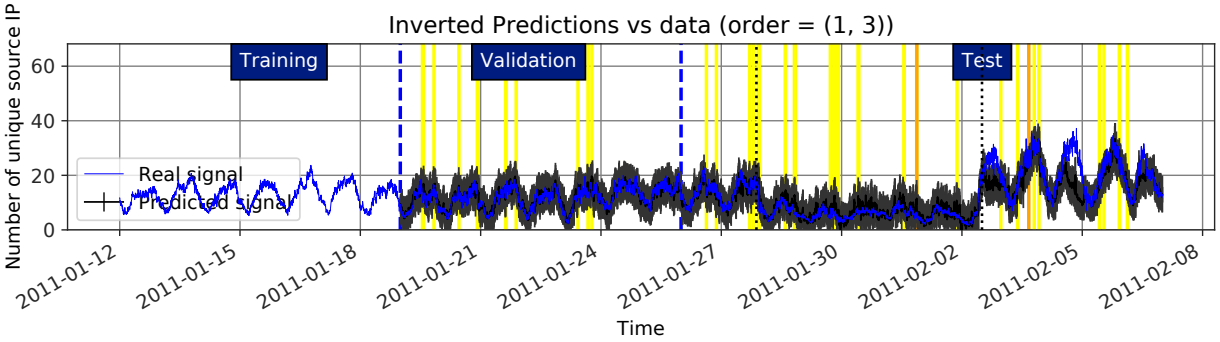


Figure 23: Egypt - AS 24835

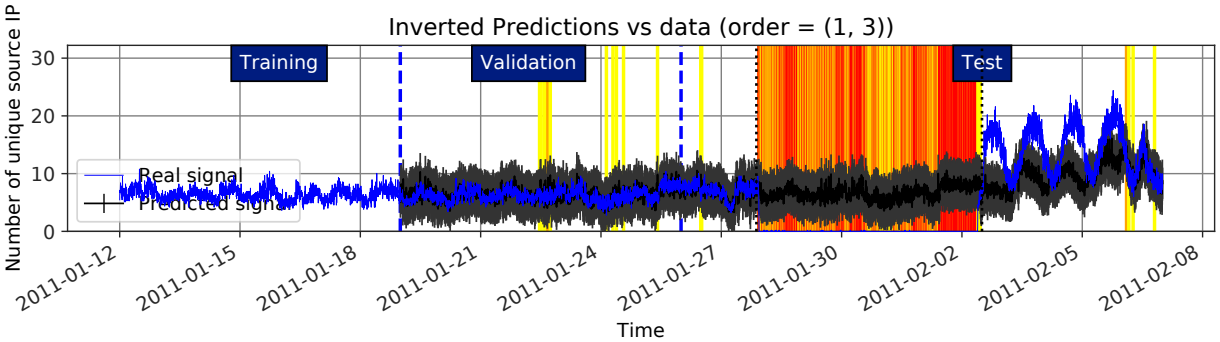


Figure 24: Egypt - Dumyat

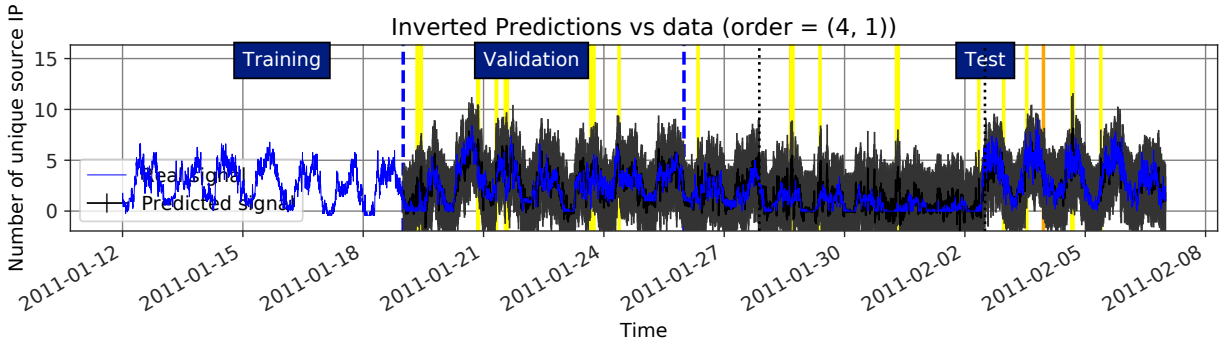


Figure 25: Egypt - Ban Sa'id

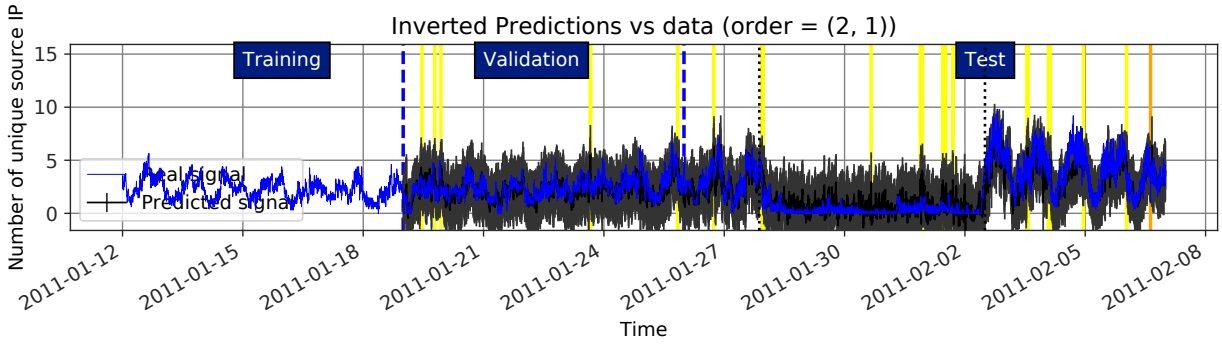


Figure 26: Brazilian power cut

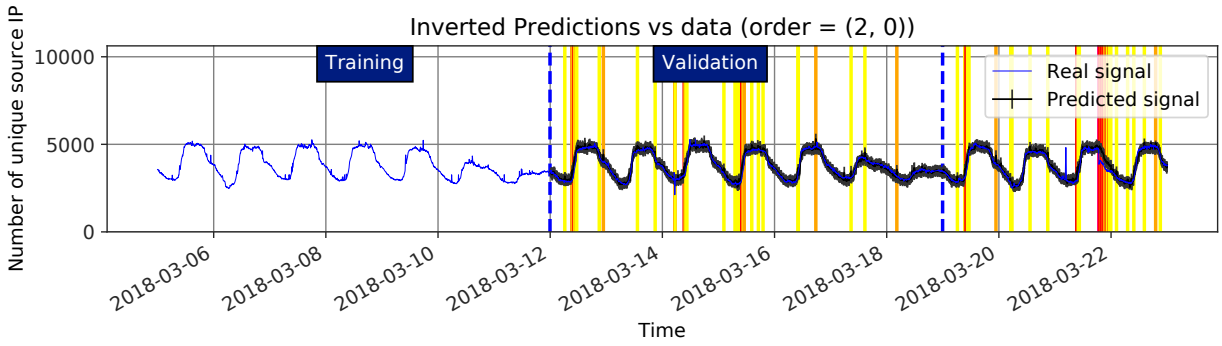


Figure 27: Brazilian power cut - Northern region

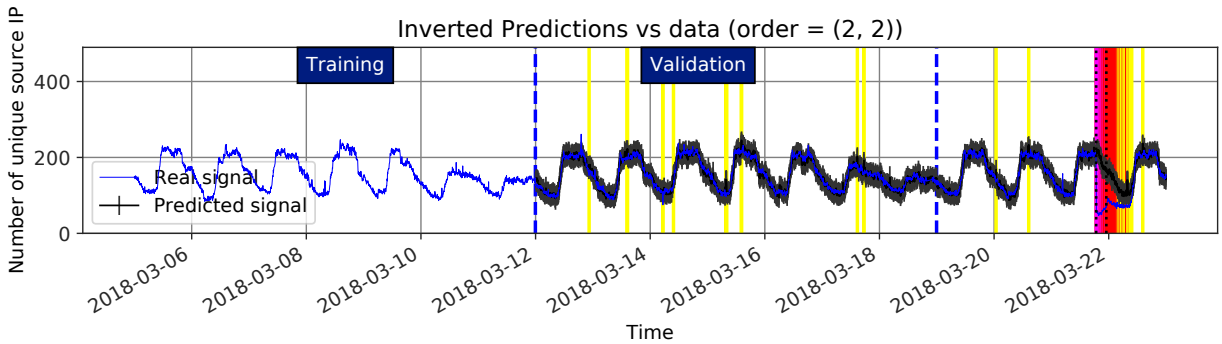


Figure 28: Syrian exams 2018

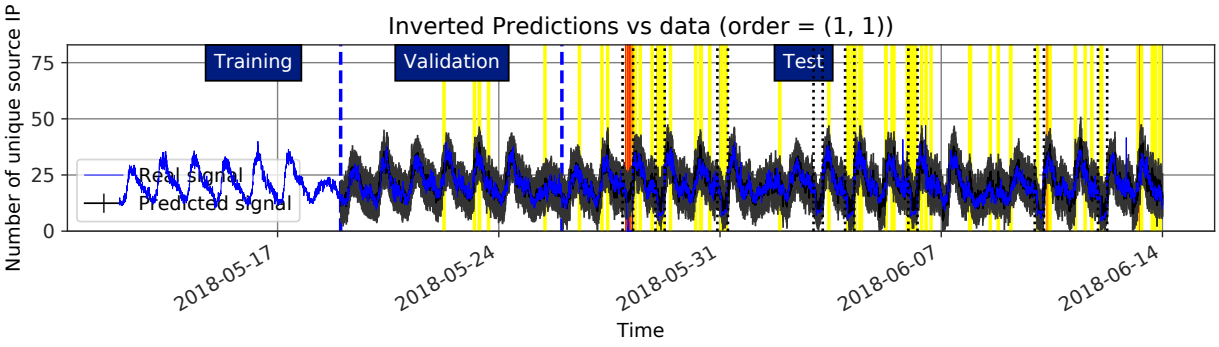


Figure 29: Syrian exams 2017

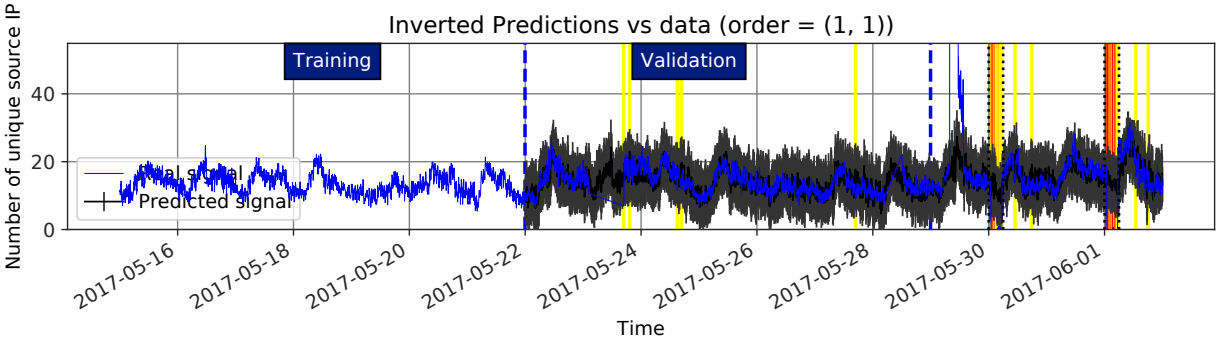


Figure 30: Iraq protests

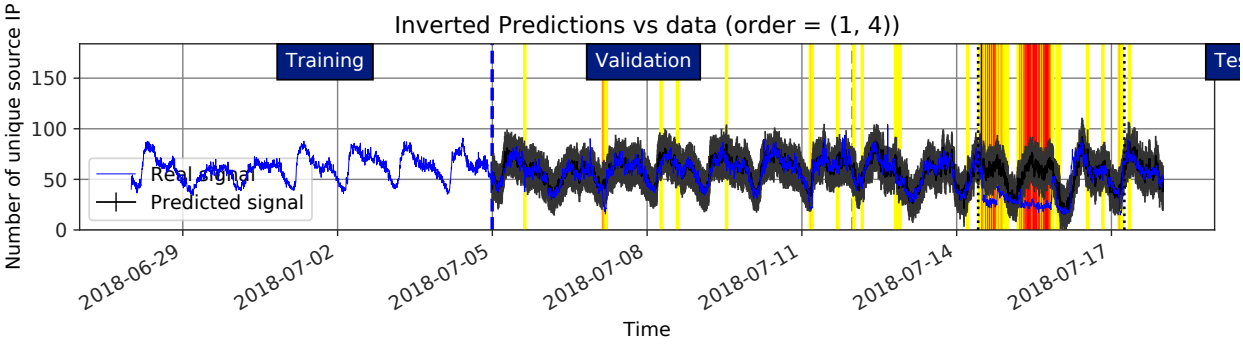


Figure 31: Azerbaijan

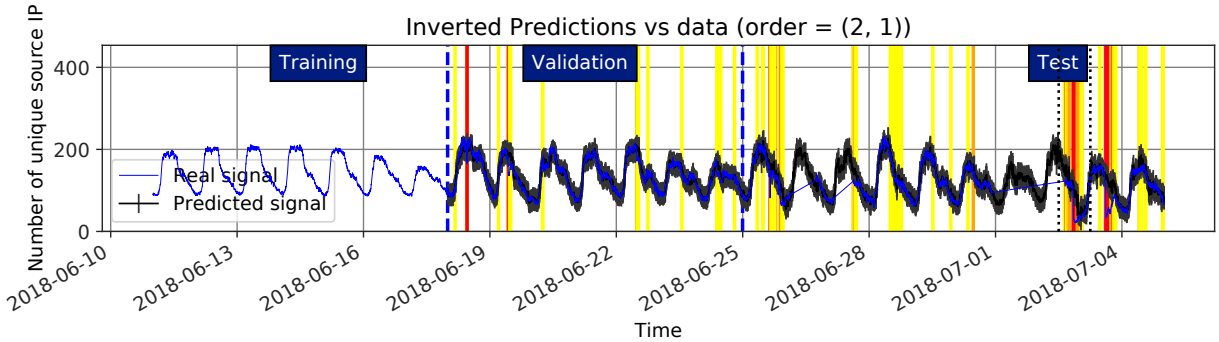


Figure 32: Democratic Republic of the Congo

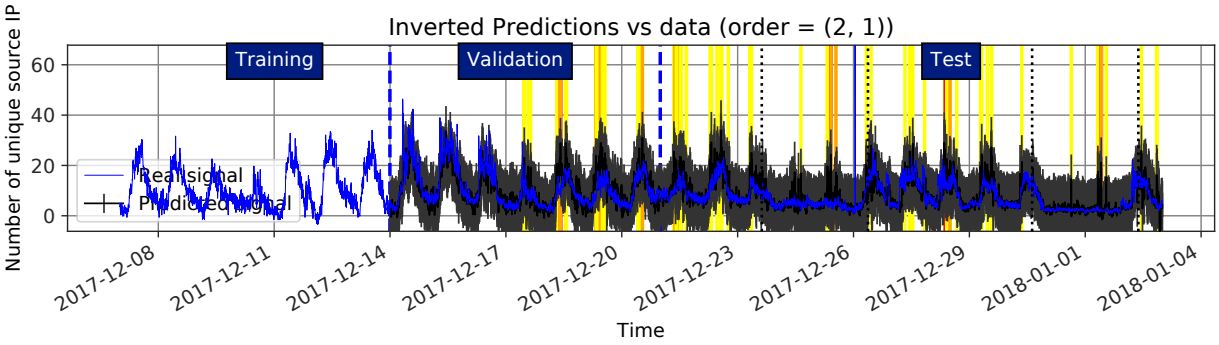


Figure 33: Gambia

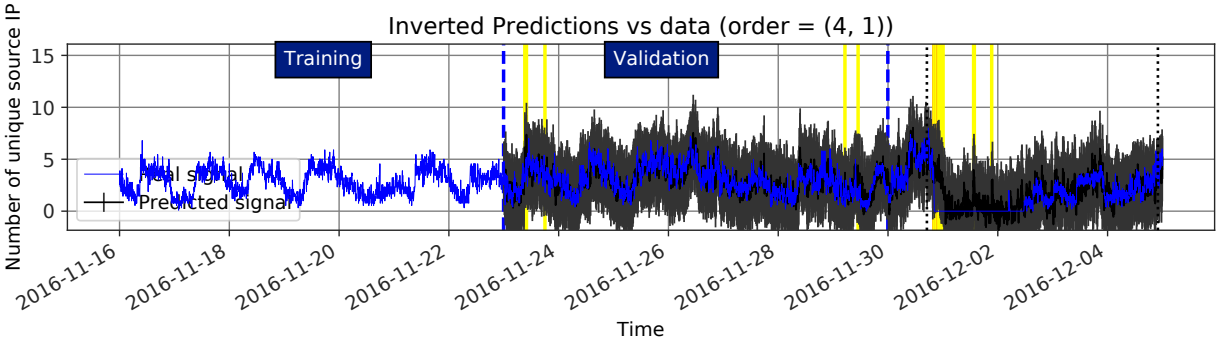


Figure 34: Gabon