Detection of Internet-Scale Traffic Redirection

Filipa Piedade
filipa.piedade@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

December 2017

Abstract

Internet security has become a major concern amongst users and internet services providers, since successful attacks can cause serious damages. A world-wide redirection of target traffic allows attackers to intercept the information sent by the user before it reaches its destination. This action will compromise unencrypted communications and allow the deployment of various attacks on encrypted communications. Detecting these intrusions in real-time would allow the users to, not only be aware they were under attack, but also take protective measures to stop their malicious effects. In this paper we present a solution to this problem, which intends to identify if a certain time period, characterized by the traffic registered in that period, is the result, or not, of an attack. The first step is to identify the anomalous periods, and then detect if the machine that produces such traffic is or not under attack. It is expected that a compromised machine produces atypical traffic, when compared to the traffic the machine produces in a regular regime. The goal of this study is to analyse several classification methods (both supervised and unsupervised), as well as methods based in heuristic rules, in order to determine if we are facing a global attack or just legitimate irregularities caused by the traffic generated by the machine at study. This final decision is obtained using a latent class model, which combines the results of the classification of a certain supervised method, applied to traffic measurements with origin in the machine at study, but with different destinations, in the same time intervals.
Keywords: internet security, intrusion detection, outliers, latent class model, supervised methods

1. Introduction

As the number of internet worldwide users increases rapidly, internet security has become a major concern amongst users and internet services providers. The routing protocol that allows the worldwide propagation of information is the Border Gateway Protocol (BGP), which is subject to several forms of attacks. This leads to vulnerabilities in the transport of information via internet, namely, redirection of target traffic (Salvador and Nogueira, 2014; Goodell et al., 2003). Over the years, many techniques were employed to keep computers and information safe and private. However, as technology evolves, these techniques must keep being upgraded in order to face the new threats that recurrently appear. When it is not possible to prevent attacks (through firewalls or anti-viruses programs), or when attackers overcome these security measures, the first line of defence should be to detect attacks as they occur. This way, defence strategies can be employed.

This paper was based in the study made by Salvador and Nogueira (2014). The authors proposed an algorithm with the goal to detect when information sent via internet is intercepted by a third person, i.e., when there is an attack to the network. The authors measured the time needed for the information to go through a certain path, and simulated redirection attacks over these paths.

Similarly to the work of Salvador and Nogueira (2014), the goal of this paper is to detect when a network suffers an intrusion. Given measurements about the time the information needs to be received at a certain destiny and return, an increase on these measurements gives indication that the traffic is being captured by a third person and forced to be redirected from its usual path. In this situation, it is declared that the network is under attack.

Our first goal is to detect atypical true patterns in a given path, based on statistical methods. If these outliers happen simultaneously at several patterns, we believe that the network is under attack. Salvador and Nogueira (2014) establish an heuristic rule based on the number of detected outliers in the paths at study in order to declare that the network is under attack.

In this paper, we propose an alternative approach based on statistical methods (using supervised learning methods and the Latent Class Model).

Salvador and Nogueira (2014) proposed an algorithm to identify outliers in a set using the Round Trip Time (RTT) measurements between probes and targets in a certain interval of time. It was based on their work that this study was made.

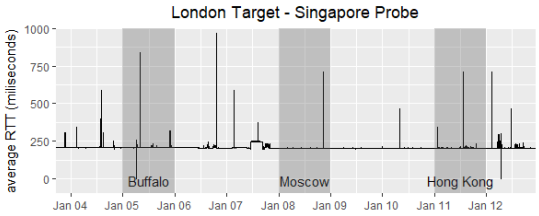
The methods used in this project were employed using the software R (R Core Team, 2017), and packages and functions implemented in it, as well as some created for the purpose of this study.

The following sections are organised as follows: Section 2 provides a description regarding the data used in the paper, as well as the approach made with the original algorithm. Section 3 provides some theoretical background about the methods employed to perform the analysis and the measures used to determine their efficiency. In Section 4 we show how we treated the problem, and the results obtained: the detection of atypical time periods is modelled by non-supervised and supervised methods. Since the days when the traffic is in fact redirected are known, we can evaluate the methods performance by constructing confusion matrices and estimating the measures described in Section 3: false positive rate, precision, recall, and ultimately, the Euclidean distance to the ideal values. Finally, in Chapter 5 we draw conclusions regarding said results, and provide some future work relevant for this theme.

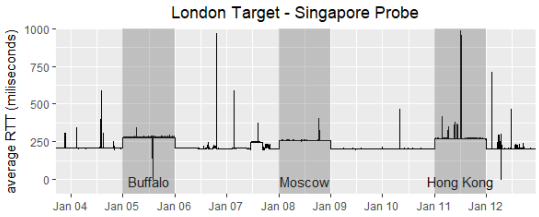
2. Data Description

The data used in this paper was generated by Salvador and Nogueira (2014) with the purpose of emulating traffic redirection attacks. In order to test their methodology, they implemented a monitoring infrastructure spread over four continents in order to monitor networks in London, New York, and Sydney, the so-called *targets*. A set of worldwide spread *probes* (entities that send the information to the targets, and receive it back) periodically measures the Round-Trip Time (RTT) to the targets, i.e, the time it takes for the information to be sent from the probe, received by the target, and sent back to the probe. To emulate traffic redirection attacks based on BGP route hijack, they used three *relay* agents located in the USA, Russia, and China, to where probes were forced to route the traffic. In Table 1 it is possible to see the geographical location of the four probes, three targets, and three relays.

The data obtained includes the minimum, average, and maximum RTT calculated every 180 seconds, between probes and targets. The observations began at 3rd January 2014 and ended at 12th January 2014, and in three of these nine days of measurement, a relay agent was introduced, in other words, an attack was emulated. In those days it is expected that the RTT values increase, because the traffic is forced to follow an alternate route, in most situations far from its expected course. This means that instead of following the usual path



(a) Before introducing the relays.



(b) After introducing the relays.

Figure 1: RTT values for the London target with the Singapore probe before and after the redirection of traffic, i.e., before and after the insertion of the relays in days 5 (Buffalo), 8 (Moscow), and 11 (Hong Kong).

$$\textbf{Probe} \rightarrow \textbf{Target} \rightarrow \textbf{Probe}$$

the traffic is redirected to do

$$\textbf{Probe} \rightarrow \textbf{Relay} \rightarrow \textbf{Target} \rightarrow \textbf{Probe}$$

When we introduce a relay, there will be a change in the course. In cases where the relay is geographically close to the probes or targets, the redirection may not have a great effect in the values of the RTT between probe and target. On the other hand, when the relay is located far from the other entities, the change in the RTT values can be very noticeable. So, it is necessary that the probes are located relatively far from the monitored network and/or attack point, in order to have significant deviations of the RTT measurements that can be detected. Due to this issue, we can say that one of the challenges of this work is to identify if an atypical increase in the RTT values is in fact the result of a redirection attack or only the consequence of a local and isolated incident.

The data used consists of several datasets (all with a different number of observations), giving indication of the target, probe, and relay at study. The relays were introduced in days 5 (Buffalo), 8 (Moscow), and 11 (Hong Kong). In Figure 1 we can see that the differences in the average RTT values before and after the introduction of the relays (indicated by the darker stripes) are quite noticeable.

In this work, the dataset under study has the following four variables:

Table 1: Geographical location of the targets, probes, and relays.

Targets	Probes	Relays
London, UK	Los Angeles, CA, USA	Buffalo, NY, USA
New York City, NY, USA	Chicago, IL, USA	Moscow, Russia
Sydney, Australia	Amsterdam, The Netherlands	Hong Kong, China
	Singapore	

- `timestamp`: contains the hour and day in which each observation was measured. It has a frequency of 180 seconds;
- `mRTT`: minimum value of the RTT;
- `avgRTT`: average value of the RTT;
- `MRTT`: maximum value of the RTT.

Note that the sets have different sizes because they did not all began the measurements in the same time instant. These discrepancies can vary between a few seconds and, at most, two minutes. Hence, when we want to join all probes for the same target, it is necessary to choose a minimum and maximum timestamp. We do this by, given the lowest timestamp in each set, selecting the highest of all (and sum 1) and defining it as the first timestamp. We then increment it in 180 (the measurements occur every 180 seconds) and stop when we reach the first timestamp, considering all sets, that is immediately prior to the one corresponding to January 13.

2.1. Original Algorithm

In order to replicate the methodology used in Salvador and Nogueira (2014), we implemented the algorithm described in the article, which is composed by three steps. In the first step, the algorithm identifies regular time periods from atypical ones, per target and per probe. The periods tagged as outliers may be due to local and isolated incidents or due to traffic redirection. Thus, in the second step, the authors look for sequences of time periods where an atypical RTT pattern was identified. Isolated outliers are denominated *instantaneous*. The remaining, belonging to sequences of at least 10 outliers, are denominated *local*. Thus, after the second step, for each target and probe, the time periods are classified in one of the following categories: regular observation, instantaneous outlier, or local outlier.

Finally, in the third step, the four probes associated to a given target are aligned by period of time. And for a given period of time, if at least 2 out of the 4 probes are tagged as local outliers, then this period is classified as a *global* outlier. In this case, it is believed that the network is under attack and its traffic is being redirected to some relays, in other words, a global routing anomaly is identified.

For a given target, *n*, and probe, *p*, let $x_{n,p,i}$ represent the average RTT at the *i*-th period of time, where $\{n = 1\}$ represents the target in London, $\{n = 2\}$ New York, and $\{n = 3\}$ Sydney. The index p can take the values 1 to 4 where $\{p = 1\}$ refers to the probe located in Los Angeles, $\{p = 2\}$ Chicago, $\{p = 3\}$ Amsterdam, and finally $\{p = 4\}$ refers to Singapore. In this work, the number of periods of time under study, T, varies from set to set.

Being so, and according to Salvador and Nogueira (2014), the three steps can be formalized as:

1st Step

If $|x_{n,p,i} - \bar{x}_{n,p}| > \mathcal{E}s_{n,p}$, then the *i*-th period of time is labelled as outlier, and $\{y_{n,p,i} = 1\}$, with $\mathcal{E} = 1.2$.

Otherwise, it is labelled as regular, and $\{y_{n,p,i} = 0\}$, where:

$$\bar{x}_{n,p} = \sum_{i=1}^{T} \frac{x_{n,p,i}}{T}, \tag{1}$$

$$s^2_{n,p} = \sum_{i=1}^{T} \frac{x^2_{n,p,i} - T\bar{x}^2_{n,p}}{T-1}. \tag{2}$$

2nd Step

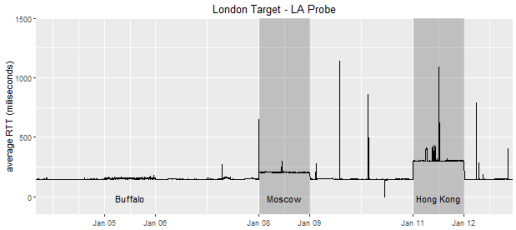
If $\{y_{n,p,i} = 0\}$, then $\{z_{n,p,i} = 0\}$, where $z_{n,p,i}$ is the label assigned to the *i*-th period of time on the 2nd step, of target n and probe p.

If there is a sequence of at least 10 consecutive outliers, i.e., $\{y_{n,p,i} = 1\}$, then all these time periods are labelled as $\{z_{n,p,i} = 1\}$, local outliers. The remaining time periods are labelled as $\{z_{n,p,i} = 0\}$, i.e., the class $\{z_{n,p,i} = 0\}$ contains regular and instantaneous outliers.

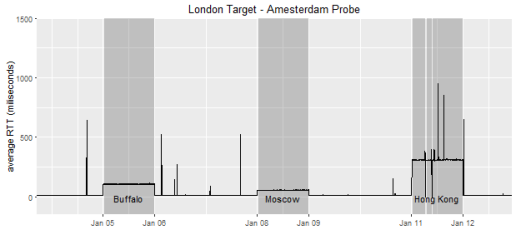
3rd Step

For target n, the *i*-th period of time is assigned as global outlier if and only if $\sum_{p=1}^{4} z_{n,p,i} \geq \rho H$, where $H = 4$ is the number of probes and $\rho = 0.5$. Otherwise, the *i*-th time period is assigned as non-global outlier.

In Salvador and Nogueira (2014), the authors use a non-supervised approach where the true classifications are only used to evaluate the performance of the proposed algorithm.



(a) LA Probe.



(b) Amsterdam Probe.

Figure 2: Replication of the datasets for the London target with probes in LA and Amsterdam.

(a) London Target.

(b) New York Target.

(c) Sydney Target.

Figure 3: Number of probes that detect that a relay agent is redirecting traffic.

The choice of $\mathcal{E}$, the sequence size (10), and $\rho = 0.5$ are based on the datasets under analysis and no general recommendations are presented.

The authors show the results based on an interesting visualization plots, reproduced for the London target with probes in LA and Amsterdam, in Figure 2, where the darker stripes represent time periods labelled as local outliers in the 2^{nd} step.

Note that in Figures 2 (a) there is not a darker stripe on day 5, meaning that the algorithm did not detect the attack performed in this day, with the Buffalo relay. Recall that we know for a fact that, in days 5, 8, and 11, an attack, coming from different relay locations, was simulated. Still from Figure 2 we can say that, with the target in London, the LA probe could not detect the intrusions coming from the relay in Buffalo. This might happen because of the geographical proximity between Buffalo and LA.

Figure 3, shows the number of probes that detected traffic redirection via each relay, for each target. Observing this figure, we conclude that only the redirection via Buffalo for the New York target was not classified as a global anomaly event, for only one probe was able to detect the intrusion. Once again, this might be due to the geographical proximity between Buffalo and New York.

3. Background

3.1. Supervised and Unsupervised Methods

In order to classify a time period as regular or outlier, we used statistical methodologies, which can be separated into two groups of methods: the unsupervised methods, which estimate the classifier under study without using the true class (regular or anomaly) of the data, and supervised methods, which by contrast, use the class of the observations to estimate the classifier.

Unsupervised Methods

Let $\boldsymbol{X} = (X_1, ..., X_p)^t$ be a random vector with multivariate normal distribution with expected value $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Then, it is known that

$$(\boldsymbol{X} - \boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1} (\boldsymbol{X} - \boldsymbol{\mu}) \sim \chi^2_{(p)} \qquad (3)$$

is the squared of the Mahalanobis distance between $\boldsymbol{X}$ and $\boldsymbol{\mu}$.

Estimating $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ properly, it is expected that equation 3 is still approximately true when we replace $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ by their estimates. Let $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Sigma}}$ be the estimates of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, respectively. Thus, an observation, $\boldsymbol{x}_0$, is said to be an outlier if and only if

$$(\boldsymbol{x}_0 - \hat{\boldsymbol{\mu}})^t \hat{\boldsymbol{\Sigma}}^{-1} (\boldsymbol{x}_0 - \hat{\boldsymbol{\mu}}) \geq \tau_{1-\alpha,p}, \qquad (4)$$

where $\tau_{1-\alpha,p}$ is the $(1-\alpha)$-quantile of a χ^2 distribution with p degrees of freedom. α is referred as the false alarm rate. In case $\boldsymbol{X}$ does not follow (even approximately) a normal distribution, a transformation over $\boldsymbol{X}$ can be done, or the $(1-\alpha)$-quantile $\tau_{1-\alpha,p}$ may be estimated by bootstrap. In the present study, the location and scale parameters, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, are chosen among the Minimum Covariance Determinant (MCD) estimator (Rousseeuw and Driessen, 1999).

The second unsupervised learning method used was based on multivariate time series analysis. The general Autoregressive-Moving-Average (ARMA) model (Box et al., 2015) contains the autoregressive and the moving-average models, given by, respectively:

$$X_t = c + \sum_{i=1}^{p} \phi_i X_{t-i} + \epsilon_t, \quad (5)$$

where X_t is the data of the time series, $\phi_1, ..., \phi_p$ are the parameters, c is a constant, and ϵ_t is a random variable with a normal distribution with zero mean, also know as white noise, and

$$X_t = \mu + \epsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-i}, \quad (6)$$

where $\theta_1, ..., \theta_q$ are the parameters of the model, μ is the expectation of X_t, and $\epsilon_t, \epsilon_{t-1}$ are again white noise terms.

Therefore, the ARMA model is given by:

$$X_t = c + \epsilon_t + \sum_{i=1}^{p} \phi_i X_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i}. \quad (7)$$

In R, the functions used to perform the time series and the robust analysis were the `tsoutliers`, from the package *forecast* (Hyndman, 2017), and `Moutlier`, from the package *chemometrics* (Filzmoser and Varmuza, 2017), respectively.

Supervised Methods

The supervised methods used in this work can be divided into three groups: K-Nearest Neighbours (KNN), Decision Trees, and Ensemble Methods.

The KNN method classifies observations based on their k nearest neighbours by, first, determining which are the nearest neighbours, and then what is the most frequent class of those neighbours (Cunningham and Delany, 2007). In R, we used the function `knn` from the package *class* (Venables and Ripley, 2002).

According to Rokach and Maimon (2015), decision trees are used to classify observations into a predefined set of classes based on their attribute values. The tree is created by dividing the original dataset into subsets until it is not possible to separate the classes anymore. The functions that we used in *R* were the `rpart`, `C5.0`, and `evtree` from the pacakges *rpart* (Therneau et al., 2015), *C50* (Kuhn et al., 2015), and *evtree* (Grubinger et al., 2014), respectively.

An ensemble method consists of a set of individually trained classifiers whose predictions are combined when classifying novel instances, resulting in a higher accuracy value than any of the single classifiers in the ensemble (Opitz and Maclin, 1999). The ensemble methods used in this Thesis were the *Random Forest*, which combines several decision trees, and the *adaboost*, which combines the outputs of other learning algorithms (known as 'weak classifiers') into a weighted sum, in order to make a final prediction. In R, we used the functions `randomForest` and `ada`, from the packages *randomForest* (Liaw and Wiener, 2002) and *ada* (Culp et al., 2012), respectively.

3.2. Performance Measures

The measures to assess the performance of a classifier are divided in two groups: measures of global performance and measures of class performance. The global measures are important but can be misleading when dealing with unbalanced classes. For example, in the case of anomaly detection where the probability of an anomaly to happen is low but has a drastic impact in the network, a classifier that assigns a new observation to the class of regular observations, with probability one, will assign correctly most of the observations (all the regular ones are correctly assigned), but will miss all the anomalies, that in fact are the primary concern of the network manager. Thus, per class measures might be of vital importance in the evaluation of a classifier.

In this Thesis, we consider the recall of class 1 (outliers), $Re(1)$, also known as *sensitivity*, which is the probability of an anomaly being correctly identified by the classifier as outlier. The probability that a time period assigned as outlier by the classifier be, in fact, under attack is called *precision* of class 1, $Pr(1)$. The last measure under study is the *false positive rate*, FPR, which is the probability of a regular time period being wrongly assigned as outlier. If Y is the class variable, assuming the value 1 in case a certain time period is under attack and zero otherwise, and X represents the result of a given classifier taking the value 1 for outlier and 0 for regular time periods, then the previous measures can be defined as:

$$Re(1) = P(X = 1|Y = 1), \quad (8)$$

$$Pr(1) = P(Y = 1|X = 1), \quad (9)$$

$$FPR = P(X = 1|Y = 0). \quad (10)$$

Given the estimated classifier, the test set must be used to classify its observations into one of the classes, and we end up with a predicted class and the true one, our reference. Being so, a cross-tabulation table can be obtained, denominated *confusion matrix*, as summarized in Table 2.

According to this table,

- *True Negatives (TN)*: number of observations classified with 0 and assigned, by the model, to class 0;
- *True Positives (TP)*: number of observations classified with 1 and assigned, by the model, to class 1;

Table 2: Confusion matrix.

		Reference	
		0	1
Predicted	0	True Negatives (TN)	False Negatives (FN)
	1	False Positives (FP)	True Positives (TP)

- *False Negatives (FN)*: number of observations with reference 1, but classified as 0 by the model;
- *False Positives (FP)*: number of observations with reference 0, but classified as 1 by the model.

The confusion matrix allows us to estimate the per class performance measures in the following way:

$$\widehat{Re}(1) = \frac{TP}{FN + TP}, \tag{11}$$

$$\widehat{Pr}(1) = \frac{TP}{FP + TP}, \tag{12}$$

$$\widehat{FPR} = \frac{FP}{TN + FP}. \tag{13}$$

In order to summarise these methods into one value, we calculated the Euclidean distance between the recall and precision of the outliers class and the false positive rate, and the ideal values (0 for the FPR, and 1 for the remaining two), as follows:

$$d = \sqrt{(\widehat{Re}(1) - 1)^2 + (\widehat{Pr}(1) - 1)^2 + \widehat{FPR}^2}. \tag{14}$$

The distances obtained were then used to compare how effective each classifier was.

For simplicity, we will refer to the estimated quantities as the same name of the population measures, and the "$\wedge$" representing the corresponding estimates will be dropped out to make the text easier to follow.

3.3. Latent Class Model

In order to classify the local outliers as global, the algorithm described in the previous Section, uses a heuristic rule: two out of the four probes must detect local outliers (third step).

In order to test a non heuristic rule to classify the global outliers, we employed a Latent Class Model (LCM). A Latent Class Analysis (LCA) is a measurement model in which individuals can be classified into mutually exclusive and exhaustive types, *latent classes*, based on their pattern of answers on a set of categorical variables. In other words, it is a technique, based on an explicit model of the data, that aims to recover hidden (non-observable) groups from observed data. True class membership is unknown for each individual. As categories of a latent variable, these classes cannot be directly measured other than through the patterns of responses on the indicator items. A variable that cannot be directly measured is called *latent*. For example, the quality of life or the level of happiness of an individual are latent variables. Latent Class Models can also be used to estimate diagnostic tests performance in the absence of a true gold standard (perfect reference test) (Subtil et al., 2012).

The LCM, in its simplest formulation, assumes that p manifest or observable (binary) variables, $X_1, X_2, ..., X_p$, give indication about a latent variable, Y, such that any two manifest variables are conditionally independent given Y - Hypothesis of Local Independence (HCI). The model can be formulated as:

$$\begin{aligned} P(X_1 = x_1, ..., X_p = x_p) = \\ = \sum_{j=0}^{1} \prod_{i=1}^{p} P(X_i = x_i | Y = j) P(Y = j) \\ = \sum_{j=0}^{1} \prod_{i=1}^{p} \pi_{ij}^{x_i} (1 - \pi_{ij})^{1 - x_i} P(Y = j), \end{aligned} \tag{15}$$

where $\pi_{ij} = P(X_i = 1 | Y = j)$, for $i = 1, 2, ..., p$ and $j = 1$.

The most common performance measures for the LCM are the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC), which according to Burnham and Anderson (2004), are defined, respectively, by:

$$\text{AIC} = -2\ln(\mathcal{L}) + 2K, \tag{16}$$

$$\text{BIC} = -2\ln(\mathcal{L}) + K\log(n). \tag{17}$$

where $\mathcal{L}$ is the likelihood function, K is the number of parameters to estimate, and n is the number of observations (sample size).

The lower the AIC and BIC values are, the better the model is. However, according to Nylund et al. (2007), usually the AIC is better for a higher number of classes and BIC provides better results for a

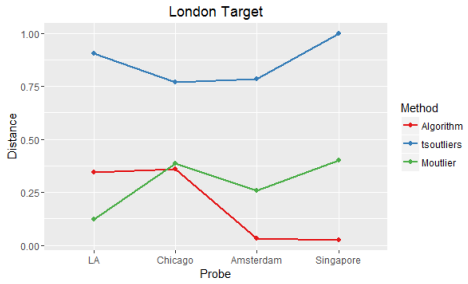


Figure 4: Distances to the ideal values obtained for the London target, per probe, with the unsupervised methods.

smaller number of classes. Thus, a study for comparing models fit obtained with different number of classes is going to be necessary.

In R, the function used to estimate the LCM parameters and associated measures was `poLCA` from the package with the same name (Linzer and Lewis, 2011). This function assumes the hypothesis of conditional independence and estimates a mixture model of latent multi-way tables.

4. Data Analysis

Our initial goal is to tell, for each target and probe, which time periods are regular and which are atypical. These atypical periods, denominated outliers to respect the denomination of Salvador and Nogueira (2014), may be due to local events in the network or redirection attacks. We started by implementing the algorithm created by the original authors, described in Section 2, classifying each observation with 1 or 0 (outlier or not, respectively).

Unsupervised Methods

Apart from the original algorithm (from now on refereed to as simply *algorithm*), we also tried to detect outliers using methods based on the robust Mahalanobis distance (which, by its turn, is based on MCD estimators), using function `Moutlier` from package *chemometrics* (Filzmoser and Varmuza, 2017).

A second strategy was to understand the data as a multivariate time-series and use the function `tsoutliers`, from the package *forecast* (Hyndman, 2017), to automatically identify outliers in time-series.

For each target and probe, the three procedures were ran and the Euclidean distances to an ideal method were calculated and plotted in Figure 4, for the London target. As a global measure of performance, we averaged the Euclidean distances over all pairs of target-probe for each method. The results are listed in Table 3.

We conclude that the method based on time series lead to the worst results. For this reason, this method was abandoned and, even though improvements could have been made, this is left for future work.

Table 3: Average distances to the ideal values for the unsupervised methods, with the best result in bold.

Method	Algorithm	`tsoutliers`	`Moutlier`
Distance	**0.1911**	0.8654	0.2914

Supervised Methods

The supervised methods considered for this analysis are the KNN, Rpart, C5.0, Evtree, Random Forest, and adaBoost. Note that, regarding the KNN, we analysed, for each dataset, 20 neighbours and chose the smallest number of them that produced the lowest classification error.

The division of each dataset into training and test sets was done in several ways, with the intention to respect the structure of the data and, at the same time, to avoid bias in the estimation of the performance measures. Note that for now, we are assuming that each time period has equal probability of being under attack, regardless of the states of the previous periods (under attack or not).

In a first trial, we randomly chose 60% of the observations to train the classifiers and the remaining 40% to estimate the performance measures. The confusion matrices obtained for the adaBoost, applied to the dataset referring to the London target and the LA probe, are shown in Figure 5 (a). Even though we only show for the adaBoost (which reported the smallest number of wrongly classified observations - 6 out of 1776), all the methods lead to very good results when it comes to detecting the periods of time under redirection of traffic (outliers). The equivalent matrix based on the hole dataset for the algorithm (Figure 5 (b)) leads to the conclusion that all supervised learning methods present good results.

```
          Reference
Prediction    0    1
         0 1194    2
         1    4  576
```

(a) adaBoost.

```
          Reference
Prediction    0    1
         0 3000  498
         1    0  941
```

(b) Algorithm.

Figure 5: Confusion matrix obtained for the set concerning the London target and LA probe, for the adaBoost classifier and the algorithm, respectively. Here, *0* refers to the regular observations and *1* to the outliers.

In Figure 6, we have the distances to the ideal values for all the supervised methods, considering the London target. In this figure, there is a clear peak in the Chicago probe, meaning that this probe had difficulties detecting outliers. Even so, the values of the distances are all very low. Comparing with the unsupervised methods, the supervised are much better at classifying the observations in regular and outlier.

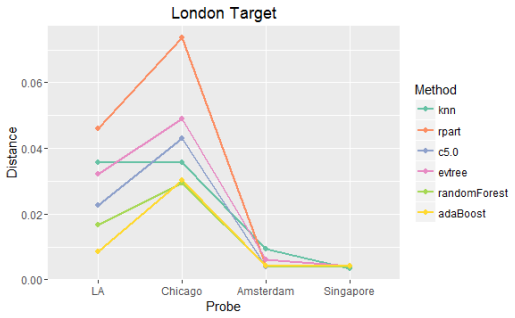


Figure 6: Distances to the ideal values obtained for the London target, per probe, for the supervised methods.

Performance stability under a smaller percentage of contamination

After performing the analysis for the original sets, with relays introduced in three of the nine days ($\simeq$33% outliers), we decided to see how the methods would behave for a smaller percentage of outliers. Hence, in addition to testing the methods for the original datasets, we also tested for sets after having removed some outliers. We studied the difference in the efficiency of the methods, for outliers in only one day, instead of three, and for outliers in only half of a day of one relay, and half of a day of all three relays. These groups are described below:

- Original set with three relays: $\simeq$33% of outliers
- Set with only one relay (only one day with outliers): $\simeq$11% of outliers
- Set with all three relays but only in 12h each: $\simeq$17% of outliers
- Set with only one relay only for 12h (only half of a day with outliers): $\simeq$6% of outliers

In order to take into account a possible temporal behaviour of the data, we also divided the dataset as follows: the first 60% observations were considered to be the training set and the remaining 40% the testing set.

In Table 4 we have the comparison for all the methods under study and all sets. Here we have the average results of the distances to the ideal values obtained with each method for all probes and targets. From the table, we conclude that the not randomized sets provided the highest values for the distance, with the exception that only a slight increase was registered. It can also be noted that the unsupervised methods are far from the ideal values, thus they will be disregarded; we dropped out the `tsoutliers` and `Moutlier`, but kept the algorithm method, for this is the original method. Finally, there is not a big difference in the values obtained for 24h and 12h (the latter are slightly higher) for all the relays, as well as for 33% and 17% of outliers.

In order to implement step 2 of the original algorithm, we need the sequential order of the test set observations, which was destroyed when the dataset was randomly divided in two. For comparison reasons with the algorithm (step 2), and in order to recover the temporal pattern in the test set, we decided to divide the original data in half as (i) the *odd* observations belong to the training set, and the *even* to the testing, and (ii) the *even* observations belong to the training set, and the *odd* to the testing.

We performed the analysis for each set with the methods used previously, and then we cross-validated the sets. As before, we tested for the different percentages of outliers: 33%, 17%, 11%, and 6%.

Table 5 shows the values obtained for the 50-50 sets, and for the 60-40 sets with 33% outliers, to compare. We conclude that, apart from the `rpart` and `C5.0`, the 50-50 set provided lower values than the 60-40, although very similar. The values for each relay are as well slightly better than the ones presented in Table 4, despite the best method still being the adaBoost. Only for the 17% outliers the results are not so good. This is due to the substantial difference in the values obtained when the odd observations are the training set and when they are the testing set (the latter provided high results).

4.1. Classification of the outliers

After having classified the observations as regular or outlier, we want to know which of the outliers are local and which are instantaneous. This corresponds to the second step of the algorithm, which, according to our proposal, is the only step that relies on a heuristic rule. Recall that, in order for an observation to be a local outlier, it has to belong to a sequence of at least ten consecutive outliers. Although this is the criteria used by the original algorithm, for the supervised methods with the 50-50 proportion, all the outliers in a test sequence with at least five outliers (instead of ten) are assigned as local outliers.

In order to implement the second step of the original algorithm, adapted for the output of the classification methods, we implemented the a procedure, based on the 50-50 splitting of each dataset, which starts by checking which observations are outliers, and then if these observations belong to a sequence of 5 outliers. If that is the case, then the observation is a local outlier, otherwise it is an instantaneous outlier.

Applying this function to our sets, we obtained figures 7 and 8, which show the classification of the observations, for the algorithm and the adaBoost

Table 4: Comparison of all methods under study. The best results are marked in bold.

	Relay								
	All	All	Buffalo		Moscow		Hong Kong		All (17%)
Method	(33% random)	(33% not random)	24h	12h	24h	12h	24h	12h	
Algorithm	0.2768	-	-	-	-	-	-	-	0.3010
`tsoutliers`	0.8698	-	0.7745	0.8157	0.6505	0.6791	0.6481	0.7388	0.6300
`rpart`	0.0189	0.2471	0.0408	0.0605	0.0053	0.0077	0.0204	0.0261	0.0127
`knn`	0.0255	0.1971	0.0533	0.0736	0.0105	0.0125	0.0164	0.0230	0.0256
`C5.0`	0.0169	0.2087	0.0462	0.0701	0.0098	0.0203	0.0182	0.0284	0.0198
`evtree`	0.0313	0.4276	0.0603	0.0862	0.0130	0.0139	0.0161	**0.0221**	0.0278
`randomForest`	0.0140	0.2961	0.0448	0.0558	0.0074	0.0096	0.0143	0.0806	0.0115
`ada`	**0.0110**	**0.0165**	**0.0234**	**0.0259**	**0.0037**	**0.0057**	**0.0089**	0.0819	**0.0102**
`Moutlier`	0.2843	-	0.4423	0.8985	0.7652	0.2593	0.8018	0.9222	0.6596

Table 5: Comparison of all methods tested for the 50-50 proportion, with the best results in bold.

	Relay								
	All	All	Buffalo		Moscow		Hong Kong		All
Method	(33% 60-40)	(33% 50-50)	24h	12h	24h	12h	24h	12h	(17% 50-50)
`knn`	0.0189	0.0168	0.0368	0.0552	0.0064	0.0085	0.0147	0.0235	0.1827
`rpart`	0.0255	0.0258	0.0508	0.0764	0.0059	0.0094	0.0150	0.0288	0.2025
`C5.0`	0.0169	0.0187	0.0389	0.0670	0.0056	0.0103	0.0164	0.0295	0.1970
`evtree`	0.0313	0.0240	0.0495	0.0750	0.0074	0.0114	0.0174	0.0302	0.1193
`randomForest`	0.0140	0.0128	0.0357	0.0515	0.0040	0.0070	0.0122	0.0195	0.1098
`ada`	**0.0110**	**0.0091**	**0.0205**	**0.0222**	**0.0029**	**0.0056**	**0.0078**	**0.0123**	**0.0950**

methods, using the sets concerning the London target and the LA probe. In Figure 7, the LA probe is not able to detect outliers via the Buffalo relay (day 5) - the observations in this day are classified as regular. The instantaneous outliers were also not detected. Comparing Figures 7 and 8, we conclude that the classification is much more accurate with the adaBoost, since all the local outliers seem to have been correctly detected.

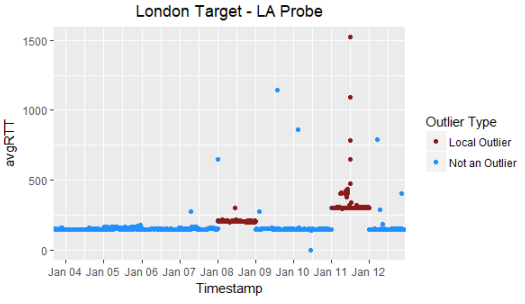


Figure 7: Classification of the observations, in local and non-local (second step), obtained with the algorithm method, for the London target and the LA probe.

4.2. Global outliers

Heuristic approach

Now that we have the outliers classified as local or instantaneous (second step), we want to know

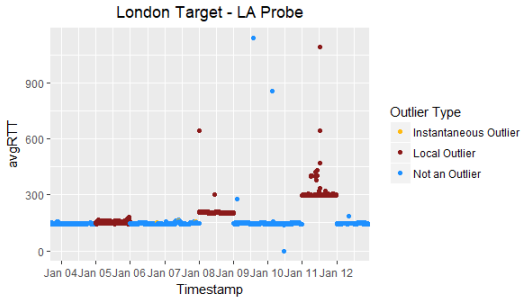


Figure 8: Classification of the observations, in local and non-local (second step), obtained with the adaBoost method, for the 50-50 proportion and the London target with the LA probe.

which of the local outliers are global outliers, i.e., we want to know if, at a certain time period, local outliers are detected by the several probes, thus having indication that at that period of time, the network is under a redirection attack. In order to do this, we created a second procedure which implements the third step of the Salvador and Nogueira (2014) algorithm: at least two of the four probes must detect local outliers, for the time period, for it to be classified as global outlier (in other words, for the global anomaly state be activated). As an alternative, we employed a Latent Class Analysis.

The results obtained with the function we cre-

ated, which checks if at least two probes detected local outliers to activate the global anomaly state, can be seen, for the algorithm and the adaBoost methods with the London target, in figures 9 and 10, respectively. It is clear that the latter was much better at detecting global outliers, having in fact, detected all of them.

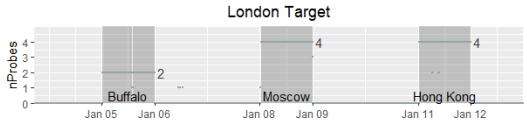


Figure 9: Detection of global outliers with the algorithm method, for the London target.

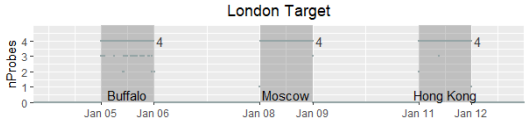


Figure 10: Detection of global outliers with the adaBoost method, for the London target, considering the 50-50 proportion.

Latent Class Analysis

We performed an LCA to the data in order to determine which observations were classified as global outliers without using the second heuristic rule, as well as discovering which patterns of local/non-local outliers originate, per probe, which latent class. In order to do this, we applied the function `poLCA`, from the package *poLCA* (Linzer and Lewis, 2011), implemented in *R*, to the datasets.

We began by creating matrices containing, for each time period, the respective classifications for the four probes, associated to a given target, as local (1) or non-local outlier (0). The LCM was applied with two and three classes. Thus, for three classes, we have the following codification: class 1 - regular observations, class 2 - non-global outliers (local and instantaneous outliers), and class 3 - global outliers. When in the case of only two classes, these are: class 1 - non-global outliers (regular observations and local and instantaneous outliers), and class 2 - global outliers.

Model with 3 latent classes:

After performing the analysis with 3 latent classes and the adaBoost classifier, we conclude that an observation is assigned as global outlier if one of the following cases happen:

- the LA probe detects a local outlier, with target in London;
- the Amsterdam probe detects a local outlier, with target in New York;
- the Singapore probe detects a local outlier, with target in Sydney.

For the algorithm we can say that for the London and New York targets, the LA, Chicago, and Amsterdam probes all need to detect a local outlier in order to activate the global anomaly state. For the Sydney target it takes both LA and Chicago probes to detect local outliers, for the observation to be classified as a global outlier. If the LA and Chicago probes do not detect a local outlier for the London target, then the network is not under a redirection attack, since this pattern is going to be assigned to the first latent class. Regarding again the Sydney target, it is required that all probes, except Chicago, do not detect a local outlier for an observation to be assigned to the first latent class. Considering the New York target, the characterization of the first latent class is not as clear: it is mandatory that the Chicago probe does not detect a local anomaly.

in Table 6, we show the distances to the ideal values obtained with these two methods.

Table 6: Distances to the ideal values for the adaBoost and the algorithm, obtained with the 3 classes LCM.

	London	New York	Sydney
adaBoost	0.0097	0.0007	0
Algorithm	0.3510	0.3636	0.0672

Model with 2 latent classes:

Regarding the LCM with 2 classes, we conclude the following, for the adaBoost classifier:

London Target: If both probes in LA and Chicago do not detect local outliers, then the observation is assigned by the LCM as a non-global outlier, except if both probes in Amsterdam and Singapore give indication of the local outlier.

New York Target: If the LA probe does not detect a local outlier, and at most one of the other probes (Chicago, Amsterdam, or Singapore) detect a local outlier, then the observation is declared as a non-global outlier, and the regular state is declared.

Sydney Target: If the Chicago and Singapore probes do not detect a local outlier, and at most one of the LA or Amsterdam probes detect a local outlier, then the time period is declared as non-global outlier.

Regarding the algorithm method, we can say that:

London Target: If both probes in LA and Singapore do not detect local outliers, and at most one of the probes in Chicago and Amsterdam detect a local outlier, then the observation is a non-global outlier.

New York Target: If the Amsterdam probe detects a local outlier, then the observation is a global outlier.

Sydney Target: If the Amsterdam probe does not detect a local outlier, and at most one of the other probes detect a local outlier, then the observation is a non-global outlier.

In Table 7, we show the distances to the ideal values obtained with these two methods.

Table 7: Distances to the ideal values for the adaBoost and the algorithm, obtained with the 2 classes LCM.

	London	New York	Sydney
adaBoost	0.0023	0.0014	0.0020
Algorithm	0.0118	0.1439	0.0189

Hypothesis of Conditional Independence

The Hypothesis of Conditional Independence (HCI) was tested for all the sets considered in this project, and for both two and three latent classes. After testing with the Log-Odds Ratio Check (LORC) and with the method proposed by Qu et al. (1996), we discovered that some of the probes were correlated.

Even though the traditional goodness-of-fit tests do not work well, the Bootstrap test indicates a poor fit, and the correlation plots and the LORC methods point to the existence of HIC violation, the results obtained with the LCM lead to good results and to an interesting interpretation. In fact, the LCM provides good performance measures for each classifier, a decision rule not based on a subjective heuristic, and an interesting interpretation of the patterns of probes which cause a traffic redirection attack. For all these reasons, we believe that the latent class models should not be disregarded.

5. Conclusions

Regarding the results obtained for the detection of local and instantaneous outliers, we can conclude that the efficiency of the methods varies from target to target and from probe to probe (regardless of belonging to the same target or not). Besides, the unsupervised methods provided much worse outcomes than the supervised ones.

In addition, we decided to test if for a smaller percentage of days with redirection attacks, the results obtained with the supervised methods would still be good. We discovered that the decrease in the number of outliers in the dataset does not affect severely the performance of the methods (apart from the 17% outliers with the 50-50 proportion), being the adaBoost the one that, overall, attained the best values. Also, the proportions used for the supervised methods seem to produce the same results, apart from the 60-40 random, whose results were very poor.

Regarding the third step of the original algorithm, we can say that the results attained with the supervised methods tested (C5.0, Random Forest, and adaBoost) with the heuristic ruel are very satisfying, considering that all of them detected the global outliers, being able to tell the difference between these observations and the regular ones. When it comes to the LCM, we obtained interesting results with both number of classes. The LCM suggests an interesting and differentiating role for each probe in detecting traffic redirections for each target. In practice, this is an important contribute with additional value to the network manager. For example, according with the LCM with 3 latent classes, for the London and New York targets with the adaBoost method, if the LA, Chicago, and Amsterdam probes gives indication of a local outlier, then whatever is the indication of the Singapore probe, the network is declared as being under a traffic redirection.

In the future we suggest the following developments to this work:

1. If at a given time period, t, the network is under attack, it is expectable that the next period of time, $t+1$, has a higher probability of continuing to be under attack, when compared with a scenario where at time t the network is not under a redirection attack. This assumption is supported by the fact that the attack duration in our dataset is of one day, at a time. Thus, the realizations of Y_t (the state of the network at time period t) cannot be considered independent random variables. This fact is not taken into account in this paper, therefore tackling this question is left for future work.

2. Another problem that deserves a deep investigation is the second step of the original algorithm. We believe that fitting a LCM, with the proper number of classes, to the output of the first step (outlier or non-outlier) for each probe of a given target may also lead to good results. Unfortunately, in this paper we did not have the time to explore this conjecture. Nevertheless, even if a procedure to the second step needs to be performed, an alternative to the heuristic rule (of 10 consecutive outliers) should be explored.

3. One more issue left to future work is the violation of the HIC. The results from the LCM lead to good results, and to an interesting explanation of the differentiated role of the probes, in the identification of global anomalies. However, the model assumptions are not verified.

Finally, we propose the creation of a real-time application that could, indeed, help the users to know when an attack is happening in their ma-

chines. We expect that the LCM results are implemented in the prototype under development by the Telecommunication experts.

References

George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. *Time Series Analysis: Forecasting and Control*. Wiley, 5^{th} edition, 2015.

K.P. Burnham and D.R. Anderson. Understanding aic and bic in model selection. *Sociological methods and research*, 33(2):261–304, 2004.

Mark Culp, Kjell Johnson, and George Michailidis. *ada: ada: an R package for stochastic boosting*, 2012. URL `http://CRAN.R-project.org/package=ada`. R package version 2.0-3.

P'adraig Cunningham and Sarah Jane Delany. k-nearest neighbour classifiers. *Technical Report UCD-CSI-2007-4l*, 2007. URL `http://csiweb.ucd.ie/UserFiles/publications/UCD-CSI-2007-4.pdf`.

Peter Filzmoser and Kurt Varmuza. *chemometrics: Multivariate Statistical Analysis in Chemometrics*, 2017. URL `https://CRAN.R-project.org/package=chemometrics`. R package version 1.4.2.

Geoffrey Goodell, William Aiello, Timothy Griffin, John Ioannidis, Patrick McDaniel, and Aviel Rubin. Working around bgp: An incremental approach to improving security and accuracy of interdomain routing, February 2003. URL `https://www.internetsociety.org/doc/working-around-bgp-incremental-approach-\improving-security-and-accuracy-\interdomain-routing`.

Thomas Grubinger, Achim Zeileis, and Karl-Peter Pfeiffer. evtree: Evolutionary learning of globally optimal classification and regression trees in R. *Journal of Statistical Software*, 61(1):1–29, 2014. URL `http://www.jstatsoft.org/v61/i01/`.

Rob J Hyndman. *forecast: Forecasting functions for time series and linear models*, 2017. URL `http://github.com/robjhyndman/forecast`. R package version 8.0.

Max Kuhn, Steve Weston, Nathan Coulter, and Mark Culp. C code for C5.0 by R. Quinlan. *C50: C5.0 Decision Trees and Rule-Based Models*, 2015. URL `http://CRAN.R-project.org/package=C50`. R package version 0.1.0-24.

Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002. URL `http://CRAN.R-project.org/doc/Rnews/`.

Drew A. Linzer and Jeffrey B. Lewis. poLCA: An R package for polytomous variable latent class analysis. *Journal of Statistical Software*, 42(10):1–29, 2011. URL `http://www.jstatsoft.org/v42/i10/`.

Karen L. Nylund, Tihomir Asparouhov, and Bengt O. Muthén. Deciding on the number of classes in latent class analysis and growth mixture modeling: A monte carlo simulation study. *STRUCTURAL EQUATION MODELING*, 14(4):535–569, 2007.

David Opitz and Richard Maclin. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169–198, 1999. URL `https://www.d.umn.edu/~rmaclin/publications/opitz-jair99.pdf`. doi:10.1613/jair.614.

Yinsheng Qu, Ming Tan, and Michael H. Kutner. Random effects models in latent class analysis for evaluating accuracy of diagnostic tests. *Biometrics*, 52(3):797–810, 1996. doi:10.2307/2533043.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017. URL `https://www.R-project.org/`.

Lior Rokach and Oded Maimon. *Data Mining with Decision Trees: Theory and Applications*. World Scientific, 2^{nd} edition, 2015.

Peter Rousseeuw and Katrien Driessen. A fast algorithm for the minimum covariance determinant estimator. 41:212–223, 08 1999.

Paulo Salvador and António Nogueira. Customer-side detection of internet-scale traffic redirection. *Telecommunications Network Strategy and Planning Symposium (Networks), 16th International*, 2014. 10.1109/NETWKS.2014.6958532.

Ana Subtil, M. Rosário de Oliveira, and Luzia Gonçalves. Conditional dependence diagnostic in the latent class model: A simulation study. *Statistics and Probability Letters*, 82(7):1407–1412, 2012. DOI: 10.1016/j.spl.2012.03.030.

Terry Therneau, Beth Atkinson, and Brian Ripley. *rpart: Recursive Partitioning and Regression Trees*, 2015. URL `http://CRAN.R-project.org/package=rpart`. R package version 4.1-10.

W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. URL `http://www.stats.ox.ac.uk/pub/MASS4`. ISBN 0-387-95457-0.