Detection of Security Attacks Using Time Series Analysis

Inês Moreira Alves
ines.s.m.alves@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

December 2021

Abstract

There are an increasing number of connected devices due to the evolution of the IoT. With this evolution, the Internet is now more exposed to security attacks. One of the ways to detect an attack is by analyzing the traffic, trying to distinguish regular traffic from the outliers caused by the attacks. This Msc Dissertation studies methods for detecting security attacks in time series based on a dataset to which the methods will be applied and analyse some algoritms to understand which is the best one for the dataset under study. After the dataset is introduced, an analysis of the performance of the algorithms is carried out by varying their parameters. Among the studied methods are an heuristic, Tukey's method, Distance Based-Outlier, SAX, and Tukey's method combined with PAA. Our results indicate that the latter method outperforms the remaining one for the detection of redirection attacks caused by BGP prefix hijacking.
Keywords: Anomaly, Anomaly Detection, Time Series, Internet Traffic, BGP.

1. Introduction

Today, the need to detect security attacks is increasing. Security attacks are increasingly due to the fact that more and more people have access to computers. Internet security is growing due to the growing number of devices connected to it as a result of the evolution of the IoT (Internet of Things).

In the area of telecommunications, network operators increasingly feel the need to use statistical methods capable of detecting anomalies, as well as a way to detect security attacks. However, this task is quite hard because there are limited means.

One type of attack that affects the Internet is traffic redirection, exploiting vulnerabilities in the BGP protocol. This document aims to exploit these vulnerabilities by detecting outliers. An outlier or anomaly is a value that is significantly far away from other observations.

One way to tell if there is a redirect attack is to measure the RTT between the source and destination of the traffic. When there is an attack, this RTT time will be higher than in a normal situation. However, this difference may not be easily detectable, especially when the attacker is close to the sender or receiver.

2. Anomaly Detection

With the detection of anomalies it is possible to detect patterns that vary from their regular behavior. A company usually generates large amounts of data, such as in commerce. Sometimes it is relevant to be able to analyze the sales data of certain products in a simple, cheap and fast way. Through anomaly detection methods businesses can become more profitable by analyzing current data compared to regular past data.

In the area of Internet security, the detection of anomalies helps to detect possible attacks. Attackers sometimes intend to change the route of the traffic and the user, whithout knowing, accesses his personal data. By detecting anomalies it is possible to detect these attacks.

In this chapter, initially there is a definition of what is an anomaly, as well as the types of learning that an algorithm can follow. It is also defined how an algorithm is categorized to understand its performance.

There are several methods for detecting anomalies that are mentioned in this report. In the section 2.5, we present distance-based algorithms such as the *Distance Based-Outlier*, the LOCI and the Nearest Neighbor Approach. The section 2.6 provides an explanation of time series. Finally, the section 2.7 mentions algorithms for the detection of anomalies in time series, such as a heuristic, the *Tukey* method and the SAX method.

2.1. Anomalies

Anomalies happen when a random phenomenon deviates from its regular behavior. When analyzing

a real problem, one of the main objectives focuses on discovering these difference . In other words it is important to discover observations that may deviate from the regular pattern of the generality of observations [8].

In data analysis, initially, it is considered that there will have to be a pattern within the randomness of the data. This pattern will be considered the regular pattern of the data.

2.2. Types of Learning Algorithms

To define the types of existing algorithms, it is necessary to first understand the concept of labeled data and unlabeled data. When the data is labeled, there is an indication of the existing classes, and it is possible to trust them as they come from true observations, and for each observation the class it belongs to is known.

There are three types of learning algorithms possible when analyzing data: supervised, unsupervised and semi-supervised.

In supervised learning algorithms, the learning process is done from a set of labeled training data. In this type of learning, the values that characterize each observation are known, as well as the true class of the observation. In other words, all dataset is labeled and the objective is to find a solution that can correctly predict the true class of a new observation [8].

When it comes to semi-supervised data, some data is labeled while other data is not. As such, we try to group the data by category or by similarity between them. Usually, among the labeled data, only regular observations are considered to estimate the classifier and identify as correctly as possible the unlabeled data [8].

In unsupervised algorithms, it is not known about the true class of each observation because the data is not labeled. In this case, one of the possible approaches focuses on grouping the data by similarities between observations to detect possible anomalies [8]. For this learning, the most common is to use clustering methods. Clustering is the creation of small subsets of data depending on the proximity between observations. This approach can allow the identification of areas with a high and low density of observations.

2.3. Metrics Used to Detect Anomalies

The evaluation of the performance of an anomaly detection method is extremely important to guarantee its practical use. When using unsupervised methodologies, it is more difficult to find a sample for which the true class (regular or anomalous) to which each observation belongs is known. However, to classify the performance of the algorithm it is necessary to have labeled data.

The usual procedure is to divide the sample into two subsets (chosen at random). One of these subsets is used to train the classifier (training sample) and the other to assess its performance (testing sample). The training sample is used to classify each observation that constitutes it. Once the true class is known, it is possible to cross the estimated class with the true class and build a table called a confusion matrix, which contains the four possible observation patterns:

- The true class and the estimated class indicate that the observation is anomalous. This observation is called the True Positive (TP);
- The true class and the estimated class indicate that the observation is regular. This observation is called the True Negative (TN);
- The true class indicates that the observation is anomalous, but the classifier wrongly indicates that the observation is regular. This observation is called False Negative (FN);
- The true class indicates that the observation is regular, but the classifier wrongly indicates that the observation is anomalous. This observation is called False Positive (FP).

For simplicity of notation, the number of observations, of each type, in the training sample is represented by the abbreviation associated with each pattern.

So $TP + TN + FN + FP = n_T$, where n_T represents the size of the training sample. The confusion matrix is summarized in Table 1.

False Positives and False Negatives correspond to classifier errors. The first term refers to observations where the true class is the regular one. However, the method predicted the observation as anomalous. The second term corresponds to observations that are actually anomalous, but which the method classified as regular.

If the problem is a binary classification problem of two unbalanced classes, it is expected that the most predominant observations are not anomalous observations and that the least predominant observations are anomalous ones. In this case, the non-anomalous class is regular traffic.

The existence of unbalanced classes indicates that in addition to overall performance measures of the classifier, it is equally important to consider performance measures by class. For example, if the true percentage of anomalies is low and if a classifier indicates that all observations are regular, the overall percentage of misclassified observations coincides with the true percentage of anomalies, which is known to be low. However, the repercussions of ignoring the existence of anomalies can be catastrophic. For example, if there is a security attack and the classifier does not detect it correctly, the user continues to provide his personal data to the attacker without realizing it. In this

Table 1: Matrix of confusion.

		True Class	
		Anomalous	Not Anomalous
Estimated Class	Anomalous	True Positive (TP)	False Positive (FP)
	Not Anomalous	False Negative (FN)	True Negative (TN)

situation, the attacker can acquire the user's personal data such as access to bank accounts and this can have serious consequences.

After knowing these definitions and these values, the metrics are calculated so that it is possible to evaluate the classifiers under study.

The global measure of performance of a classifier is the percentage of observations in the training sample that are well classified, called in as *overall Accurancy* or simply *Accuracy*. Considering the Table 1, the *Accuracy* can be estimated by:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \quad (1)$$

There are three widely used metrics to assess how well anomalous observations are detected by the classifier: *Precision*, *Recall* and *F1-score* of the anomaly class.

Precision counts, among the detected anomalies, which percentage corresponds to true anomalies [13],

$$Precision = \frac{TP}{TP + FP}. \quad (2)$$

Adapting to the case of telecommunications, for example, if a classifier indicates that there is a communication failure in a certain region, the operator must be sure that the failure actually occurs. However, if there is no fault and the classifier mistakenly detects a anomaly (False Positive) this error may not be so unpleasant for the consumer. In this example, when the impact of a False Negative is much more relevant to an operator, it is more useful to use *Recall*. The *Recall* among the existing true anomalies, identifies the percentage of anomalies that were correctly detected [13],

$$Recall = \frac{TP}{TP + FN}. \quad (3)$$

Finally, the *F1-score* is the harmonic mean of the two metrics mentioned above. A low *F1-score* means that at least, one of the two metrics, *Precision* or *Recall*, also has a low value and therefore the classification method used is not the most suitable [13]:

$$F1 - score = 2\frac{Precision \times Recall}{Precision + Recall}. \quad (4)$$

2.4. Anomaly Detection Approaches

There are several possible approaches to detecting anomalies. The family of approaches that use distances is called the *distance-based* approach. Other possible approaches for detecting anomalies such as *density-based* and *rank-based* do not use distances between points to find anomalous observations and are little studied in this work [8].

The *distance-based* approach considers that closer observations are more similar to each other and observations further away from a centrality measure are possible anomalies.

In the *density-based* approach, a *cluster* is considered when there is a dense region of observations. In this case, observations located in regions with low observation densities are considered anomalous. This approach uses the local density, which can be defined as the number of observations in a certain area. The *rank-based* approach attests that the observations are anomalous based on *ranks* of the observations belonging to the neighborhood of the observation to be classified as anomalous or regular [8].

2.4.1 Distance Based Approaches

When defining observations as anomalous or regular, one usually hopes to be able to identify how dissimilar or similar two observations are. An observation is considered to be quite dissimilar from another if the distance or measure of dissimilarity between the two is high. Dissimilarity quantifies how different two objects are. The dissimilarity between two objects A and B is a function d_{AB} that verifies the following properties: $d_{AB} \geq 0$, $d_{AA} = 0$ and $d_{AB} = d_{BA}$, where d is the dissimilarity. A distance is a dissimilarity if, in addition to the mentioned criteria, the following conditions are also met: $d_{AB} = 0$ if and only if $A = B$ and if $d_{AB} \leq d_{AC} + d_{CB}$ (triangular inequality).

Some of the most popular measures of dissimilarity among observations that can be modeled as realizations of randomly continuous vectors, ie, $\mathbf{x}, \mathbf{y} \in R^p$, are the Mahalanobis distance, the Euclidean distance, the Minkowski distance and the Manhattan distance.

The Mahalanobis distance between two observations represented by $x = (x_1, x_2, \ldots, x_p)^T$, $y = (y_1, y_2, \ldots, y_p)^T \in R^p$ is defined by:

$$d_{Mahalanobis}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T \sum^{-1} (\mathbf{x} - \mathbf{y})}, \quad (5)$$

where $\sum$ is the sample covariance matrix estimated from the data of dimension $(p \times p)$ and $\sum^{-1}$

is its inverse.

If the sample covariance matrix, $\sum$, is equal to the identity matrix then we are faced with the Euclidean distance [8]:

$$d_{Euclidean}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}. \quad (6)$$

The Minkowski distance is a generalization of the Euclidean distance and is defined by [8]:

$$d_{Minkowski}(\mathbf{x}, \mathbf{y}) = (\sum_{i=1}^{n} |x_i - y_i|^l)^{\frac{1}{l}}, \quad (7)$$

where $l \in N$ is the order of distance. If $l = 2$, then we are calculating the Euclidean distance, if $l = 1$, then the distance is the distance from Manhattan,

$$d_{Manhattan}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i|. \quad (8)$$

As the name implies, the distance of the observation **x** to all existing points in the data subset D is defined by [8],

$$d_{all}(\mathbf{x}) = \sum_{\mathbf{y} \in D} d(\mathbf{x}, \mathbf{y}), \quad (9)$$

where d is any distance chosen by the investigator. This is known as the distance to all points.

Methods based on distance to nearest neighbor are based on determining the distance between that observation and its nearest neighbor and are defined as [8]:

$$d_{nearest}(\mathbf{x}) = min_{\mathbf{y} \in D, \mathbf{x} \neq \mathbf{y}} d(\mathbf{x}, \mathbf{y}), \quad (10)$$

where D is the dataset.

Calculating the distance to the k^{th} nearest neighbor is similar to the previous method. However, the distances between $\mathbf{x} \in R^p$ and its k^{th} nearest neighbors are averaged, where k is smaller or equal to the total number of observations under study. Let $D_k(\mathbf{x})$ be the set of k observations belonging to the initial data, D, nearest to **x** then [8],

$$d_{k-nearest}(\mathbf{x}) = \sum_{\mathbf{y} \in D_k(\mathbf{x})} \frac{d(\mathbf{x}, \mathbf{y})}{k}. \quad (11)$$

If $k = 1$, then the result will be the same as that obtained in equation (10).

2.5. Algorithms Using Distance Based Approaches

In this section different approaches to detecting anomalies based on distance are mentioned. Initially there is the *Distance Based-Outlier* Approach and some tests are carried out to verify its performance. Then it is mentioned the LOCI Approach, where there is a explanation of how it works. Finally, a brief reference is made to the *Nearest Neighbor Approach*.

2.5.1 Distance Based-Outlier Approach

The *Distance Based-Outlier* approach is also known as $DB(\pi, r) - outlier$. This approach considers a point $\mathbf{x} \in R^p$ with a neighborhood centered on **x** and of radius r. If $N_{\mathbf{x}}(r)$ is the neighborhood, if a very low number of observations belongs to this neighborhood, then this point is isolated from most points and is considered an anomaly. The method has two parameters: r, the radius of the neighborhood and $(1 - \pi)$, the minimum percentage of points outside the neighborhood of **x** that takes **x** to be classified as an anomaly, called $DB(\pi, r) - outlier$ [8][5]. Where $D = \{x_1, x_2, \ldots, x_n\}$ is the set of collected observations and n the number of observations belonging to $N_{\mathbf{x}}(r)$, if $\frac{N_{\mathbf{x}}}{n} \leq (1 - \pi)$ then **x** is classified as anomalous.

Considering that the study population has a Normal distribution of expected value μ and variance σ^2, an anomaly is an observation x that satisfies the following condition: $|x - \mu| > 3\sigma$. In this case, the $DB(\pi, r) - outlier$ corresponding to this criterion has the parameters $\pi = 0.9988$ and $r = 0.13\sigma$, and we would obtain $DB(0.9988.0.13\sigma) - outlier$ [8].

There are several observations that cannot be easily detected, so the existence of a method that detects them effectively and efficiently is essential. If, on the one hand, there are observations that are confused with the regular dataset, on the other hand, there are also some observations that are so far removed from the regular that they are easily considered anomalous without requiring any excessively exhaustive approach.

2.5.2 LOCI

LOCI (*Local Correlation Integral*) is an approach to detecting anomalies that is easily adaptable and effective, as it does the necessary calculations in just one step [8][15].

The LOCI method uses values calculated using the MDEF (*Multi-granularity Deviation Factor*) to choose the observations that are anomalous. If **x** is an observation of the dataset, we can define the MDEF as:

$$MDEF(\mathbf{x}, r, \alpha) = 1 - \frac{n(\mathbf{x}, \alpha r)}{\hat{n}(\mathbf{x}, r, \alpha)}, \quad (12)$$

where $0 < \alpha < 1$ is a predetermined parameter, $n(\mathbf{x}, \alpha r)$ is the number of observations whose distance to **x** is less than or equal to αr. We consider $\hat{n}(\mathbf{x}, r, \alpha)$ as the mean of the observations of $\mathbf{y} : \mathbf{y} \in N_{\mathbf{x}}(r)$ and $N_{\mathbf{x}}(r)$ is the set of all observations belonging to the neighborhood centered on **x** of radius r.

The MDEF can be positive or negative and, depending on the result, conclusions can be drawn

regarding the irregularity of the observations. If the coefficient is positive, then the observation is a candidate to be classified as anomalous. If on the other hand the MDEF has a negative value, then **x** is classified as regular.

The value r belongs to the range $[r_{min}, r_{max}]$, where $r_max \approx \alpha^{-1} max_{\mathbf{x},\mathbf{y}\in D}\delta(\mathbf{x},\mathbf{y})$, where D is the dataset and $\delta(\mathbf{x},\mathbf{y})$ represents the distance between **x** and **y**. The value r_{min} is chosen so that the nearest neighbors contain about 20 observations.

If the standard deviation of $n(\mathbf{x}, \alpha r)$ is defined by $\sigma_{\hat{n}}(\mathbf{x}, r, \alpha)$ then

$$\sigma_{MDEF}(\mathbf{x}, r, \alpha) = \frac{\sigma_{\hat{n}}(\mathbf{x}, r, \alpha)}{\hat{n}(\mathbf{x}, r, \alpha)}. \tag{13}$$

According to this method, **x** is considered to be anomalous if

$$MDEF(\mathbf{x}, r, \alpha) > k_\sigma \times \sigma_{MDEF}(\mathbf{x}, r, \alpha). \tag{14}$$

Authors in [11] suggest $\alpha = \frac{1}{2}$ and $k_\sigma = 3$, although k_σ may take other non-negative values and $0 < \alpha < 1$.

Figure 1 shows an example of the use of the algorithm to discover observations at a distance of **x** less than or equal to αr, this is a graphical display of how to use the formula $n(\mathbf{x}, \alpha r)$. In this case $\hat{n}(x_0, r, \alpha) = \frac{1+4+4+1+3}{5} = \frac{13}{5} = 2.6$ and $MDEF(x_0, r, \alpha) = 1 - \frac{1}{2.6} = 0.61538$. Note that the r neighborhood of x_0 contains 4 other observations, x_1, x_2, x_3 and x_4. The αr neighborhood of x_0 contains only 1 observation, which is x_0 itself. With respect to x_1, x_2, x_3 and x_4 contain 4, 4, 1 and 3 observations, respectively.

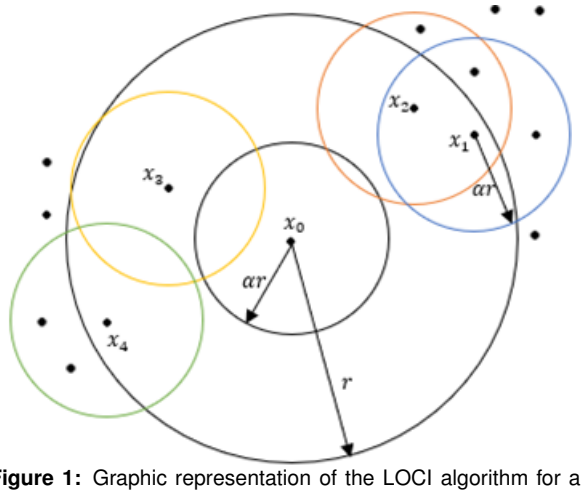

Figure 1: Graphic representation of the LOCI algorithm for a small dataset.

2.5.3 Nearest Neighbor Approach

An anomalous observation can be considered to be an observation whose nearest neighbor is at a considerably large distance [8]. Contrary to the approaches mentioned in 2.5.1 and 2.5.2, in which observations were classified as anomalous according to the number of observations in the neighborhood of the point with a given radius, this approach is based directly on distance between observations to decide which observations are anomalous.

More specifically, the distance between **x** and each of the remaining observations in the dataset is calculated. The minimum of these distances represents the distance between **x** and its nearest neighbor. This approach was developed for greater efficiency in calculating $D^k(\mathbf{x})$, which is the distance from the nearest k neighbor to **x**. If the value $D^k(\mathbf{x})$ is high, then the observation will be classified as anomalous [16].

It is

$$\alpha(\mathbf{x}) = min_{\mathbf{y}\in D\setminus\{\mathbf{x}\}} d(\mathbf{x}, \mathbf{y}) \tag{15}$$

the distance between **x** and its nearest neighbor. **x** is said to be an anomaly if $\alpha(\mathbf{x})$ is very high compared to $\alpha(\mathbf{y}), y \in D \setminus \{\mathbf{x}\}$. In the book [8] an example is given where an anomalous object or an anomalous point is considered to be the center of a well-defined cluster, which makes no sense. To conclude, the authors suggested considering the distance to the k nearest neighbor.

Let $\alpha_1(\mathbf{x}), \alpha_2(\mathbf{x}), \ldots, \alpha_k(\mathbf{x})$ be the k smallest distances between **x** and $\mathbf{y} \in D \setminus \{\mathbf{x}\}$. The idea is to use a location measure of $\alpha_1(\mathbf{x}), \alpha_2(\mathbf{x}), \ldots, \alpha_k(\mathbf{x})$ that allows us to decide if a observation is or is not anomalous. The most usual is to use the mean of $\alpha_1(\mathbf{x}), \alpha_2(\mathbf{x}), \ldots, \alpha_k(\mathbf{x})$ or the median of these distances, which it is known to be more robust.

However, this criterion does not work correctly if there is a large amount of clusters in the dataset, especially if their density is significantly different. For this, a measure based on the weight of **x** was proposed:

$$\sum_{i=1}^{k} d_i(\mathbf{x}), \tag{16}$$

where **x** is the observation, as mentioned earlier.

2.6. General Notions of Time Series

A dataset is assumed to be a time series when there is a sequence of chronologically ordered observations, in which each observation is associated with an instant of time [3]. In the case of a univariate time series, each instant corresponds to a univariate observation also known as a single value. In the multivariate case, each instant corresponds to a multivariate observation, which is, a vector of values. Note that consecutive observations do not necessarily have to occur at equally spaced times, although this happens in many cases.

An important objective of time series analysis is to find methods capable of describing the data. In the literature on the subject, several statistical methods appear to characterize a time series in terms of aspects such as dependence on past observations, trend, seasonal or cyclical behavior, and also to remove random noise present in the data [8] [12].

One of the models widely used in time series is the ARMA (*Autoregressive Moving Average*) whose designation identifies the characteristics of the model. "AR" indicates it is autoregressive, "MA" refers to moving averages (*Moving Arevage*). Let us assume that $x_1, x_2, \ldots, x_n$ are the n observations of a stochastic process, where x_t is the value observed at the instant $t(t = 1, 2, \ldots, n))$.

An autoregressive model (AR) of order p, AR(p), means that the current value of the series, x_t, can be explained as a function of $p(p > 0))$ past values [19],

$$x_t = \sum_{i=1}^{p} \phi_i x_{t-i} + \epsilon_t. \tag{17}$$

Since ϵ_t is the error, a white noise that follows a Normal Distribution with mean 0 and variance $\sigma^2(\epsilon_t\ N(0, \sigma^2))$ and $\phi_i(i = 1, 2, \ldots, p)$ are real constants, $\phi_p \neq 0$. The expression (17) corresponds to a series with a mean of zero.

For example, an AR(1) autoregressive model is written as

$$x_t = \phi_1 x_{t-1} + \epsilon_t. \tag{18}$$

In a moving average (MA) model it is assumed that the current value of the series, x_t, can be expressed as a linear combination of past noise. Thus, a moving average model of order $q(q > 0)$, MA(q), can be written as

$$x_t = \epsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-i}, \tag{19}$$

where $\epsilon_w\ N(0, \sigma_w^2)$, with $w = tq, \ldots, t, \theta_i(i = 1, 2, \ldots, q), \theta_q \neq 0$ are real constants. This series is stationary in covariance, this means that the mean and the autocovariance are constant, not varying over time. Thus, for $q = 1$ the MA(1) model is:

$$x_t = \epsilon_t + \theta \epsilon_{(t-1)}. \tag{20}$$

When an AR(p) model is combined with a MA(q) model, the ARMA(p,q) model is obtained in which the parameter p is the number of autoregressive terms, known by the number of delays that are necessary as predictors and q is the number of past prediction errors.

The ARMA model can be represented by:

$$x_t = \sum_{i=1}^{p} \phi_i x_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t, \tag{21}$$

with $\theta_q \neq 0$, $\phi_p \neq 0$, $\epsilon_w\ N(0, \sigma_w^2)$. This expression corresponds to a series with a mean of zero.

2.7. Time Series Anomaly Detection Approaches

In the detection of anomalies in time series, two different types of problems can be highlighted. Sometimes the aim is to distinguish anomalous time series between different series and other times the aim is to distinguish anomalous subsequences belonging to a single time series. In addition, we can also refer to the problem of online anomaly detection, in which the intention is to detect anomalies as they appear, assuming that the data generation process may vary over time.

When the objective is to distinguish anomalous series between several time series, we can consider the entire period of time in which data from multiple series are available and apply algorithms that allow to identify the anomalous series. For example, if the series can be satisfactorily represented by ARMA(p,q) models, distances between the parameters that define the series can be calculated and thresholds defined for these distances that allow identifying anomalous series.

To determine how similar two series are, certain aspects must be taken into account:

- Two series can be overlapping in different time intervals, considering for calculation purposes the intersection of the two time intervals;
- If there are missing values and there is no information in this respect, it is assumed that the missing values can be filled by an interpolation of observations from the existing data.

One of the plausible approaches to classifying the similarity between a time series and a set of other time series is to define the distance as the average of the point-to-point distances [8].

However, time series can be quite different from each other only in a certain period of time. In this case, one might want to identify either the series with anomalous behavior or the relevant time period in which the behavior is different from the remaining series [8].

Analyzing the behavior of the same time series, it is important to understand what varies from its regular behavior and if there is any apparent reason for this event. In these cases, an observation is a candidate for anomalous when it deviates significantly from the rest of the data in that same dataset. It is important to identify when the data began to vary from regular behavior and whether it was just a point anomaly or an anomalous sequence.

When trying to identify anomalies in the same time series, two types of anomalies can arise: rate anomalies and contextual anomalies [8]. A rate anomaly is considered to exist when the values,

if observed individually, appear to assume regular values, but the rate at which the change was made appears to be anomalous. On the other hand, in the contextual anomaly, the observations do not appear to be anomalous considering the whole range of possible values in the past, but only in relation to the immediately previous observations.

When dealing with an online anomaly detection algorithm, it is expected that it will be able to detect the presence of an anomaly as quickly as possible. As mentioned above, online anomaly detection algorithms must be able to detect new anomalies in the dataset, even if the behavior of the data varies over time. If the data varies substantially and a learning algorithm has been used to determine the model parameters, it is necessary to retrain the algorithm, adding new training data and discarding some of the old data. In this way, the model parameters are regularly updated so that new anomalies can be correctly detected [8].

2.7.1 Algorithms Applicable to Time Series

The detection of anomalies in time series, whatever the type of problem addressed, requires methods that take into account the time dependence of these data, which form a sequence of observations determined by the instants of occurrence. Therefore, anomaly detection methods applicable to time series are needed.

The problem of detecting temporal series anomalies can also be addressed using methods applicable to non-temporal data, such as the algorithms described in sections 2.5.1, 2.5.2 and 2.5.3, adapting these methods to time series. Another way to approach this problem is based on the construction of a specific model for time series such as ARMA, described in section 2.6, based on past values of the series. This model makes it possible to calculate residuals corresponding to the differences between the values observed and those estimated by the model and thus indicate anomalies based, for example, on a pre-established threshold [20]. Furthermore, according to [14], another possible approach aims to determine the instants in which changes occur in the series' characteristics, such as trends and seasonality, in order to detect anomalies. Additionally, the detection of subsequences of anomalous observations in time series has motivated the development of algorithms that identify anomalies based on the similarity between subsequences, such as the SAX method (*Symbolic Aggregate aproXimation*) [4][7].

Next, three particularly relevant algorithms are described, taking into account the data set that is analyzed in the Master's Dissertation. Thus, the heuristic proposed by Salvador and Nogueira in [11], the *Tukey* anomaly detection method and the SAX method are presented.

Heuristic Approach

This approach, described in [11], proposes the use of moving averages from past observations and the use of the average RTT (avgRTT). The RTT is defined as the round-trip-time between a source *host* and a destination *host* [17]. Taking this heuristic into account, an observation is declared to be anomalous if a certain number of $k = 10$ consecutive observations exceeds the limit ϵ, considering $\epsilon = 1.2$ multiplied by the mean of the $h = 480$ past observations.

***Tukey* Method**

The *Tukey* anomaly detection method uses the 1^{st} and 3^{rd} sample quartiles and the difference between them to define a lower and an upper limit, outside which observations are potential anomalies [2][6]. Let Q_1 and Q_3, respectively, be the 1^{st} and 3^{rd} sample quartiles and let the interquartile range (IQR) be the difference between them:

$$IQR = Q_3 - Q_1. \tag{22}$$

According to this method, an observation is potentially an anomaly when it is in the region $x : Q_3 + \delta IQR < x \vee x < Q_1 - \delta IQR$. The value of k can vary by dataset. It is usual for an observation to be considered a severe anomaly when $\delta = 3$ and to be considered a possible anomaly when $\delta = 1.5$.

This method also allows the upper limit and lower limit to be defined by $Q_3 + \delta IQR$ and $Q_1 - \delta IQR$ respectively, with δ being a value considered appropriate for the data set.

This method can be applied in the context of univariate time series considering, for example, a sliding window of n observations for which thresholds are calculated according to the method of *Tukey* and the observation at the instant following the interval is compared to these thresholds to decide whether it is an anomaly or not.

This procedure is analogous to the one adopted in the heuristic described in this work. However, the heuristic calculates the threshold for flagging anomalies based on the mean, an estimator that is not robust in the presence of anomalies. The *Tukey* method has advantages in terms of robustness, as it defines the thresholds based on the 1st and 3rd quartile and interquartile range.

SAX Method

The number of devices with Internet access is increasing and, as such, so is the amount of traffic. Therefore, when analyzing time series in the context of the study of Internet traffic, methods that allow the reduction of the series size are particularly interesting.

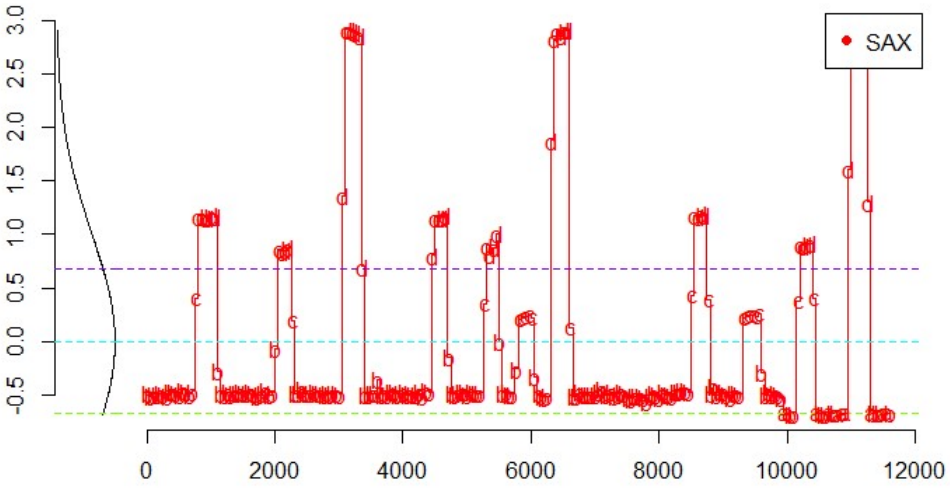

Figure 2: Illustrative figure related to the SAX method.

SAX (*Symbolic Aggregate aproXimation*) is an anomaly detection method that represents a time series through a set of letters [14][7]. The first step of SAX involves a transformation done by PAA (*Piecewise Aggregate Approximation*), an algorithm used to decrease, in time, the size of a time series by dividing it into equal-sized pieces. At this stage the number of observations are significantly reduced. In the second step, for each part is assigned a symbol that is an approximation of the original value [9].

Let $x_1, x_2, \ldots, x_n$ be a time series of size n. Typically, before applying the PAA, the series is normalized to have zero mean and unit standard deviation, making the standardization:

$$c_j = \frac{x_j - \mu_x}{\sigma_x}, \tag{23}$$

where c_j are the original observations of the time series $(j = 1, \ldots, n)$, μ_x is the mean of the observations and σ_x is the standard deviation of the comments. Normalization is essential in series comparison problems, as it is only normalized that the comparison makes sense [4].

Subsequently, the PAA is applied by dividing the dataset into segments of the same size to make the dataset smaller and simpler, as datasets are usually quite complex due to the high number of observations in them. Thus, the normalized series of size n will be transformed into a series of size $w < n$ (desirably $w \ll n$), by dividing it into segments with the same number of observations.

Let $c_1, c_2, \ldots, c_n$ be the normalized time series and w the number of divisions made in this series to obtain the reduced representation of the series, that is the number of segments in the PAA representation of the time series $\overline{c}_1, \overline{c}_2, \ldots, \overline{c}_w$, where $\overline{c}_i$ is given by [14] :

$$\overline{c}_i = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} c_j. \tag{24}$$

This means that each segment is represented by the average of the observations in each segment. For simplicity, assume that w is a divisor of n. After determining the segments of the PAA representation, a symbol will be assigned to each segment, according to the average calculated above. An alphabet $\alpha_1, \alpha_2, \ldots, \alpha_{af}$, of size $af > 2$ is used, where each α_j is a letter of the alphabet $(\alpha_1 = a, \alpha_2 = b, \ldots)$. The resulting series, $\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_n$ is such that: $\hat{c}_i = \alpha_j$, if and only if $\beta_{j-1} \leq c_i < \beta_j$, where $i = 1, \ldots, w, j = 1, \ldots, af$ and $\beta_1, \beta_2, \ldots, \beta_{af-1}$ is a sequence of points such that the area under the curve of the distribution N(0,1) between β_i and β_{i+1} is $\frac{1}{af}$ and β_0 and β_{af-1} are defined with $-\infty$ and $+\infty$. In [7], these points are defined based on the N(0,1) distribution.

The big difference between SAX and PAA is that SAX assigns a symbol to each segment, so it is easier to quantify in terms of similarity and dissimilarity, due to the ease of search of patterns into strings. Furthermore, with a SAX we reduce the size of the dataset, in amplitude, to just α possibilities. For example, a section where the first letter is the letter “a”, might be more similar to another section that also starts with the letter “a” [10]. Also, when dealing with large amounts of data, the time to look for patterns decreases and becomes simpler.

Figure 2 is an illustrative figure of the application of the SAX method. It is verified that, for each symbol there is some variety of possible values, something that differentiates it from PAA since with the use of SAX there are only 4 possible results, a, b, c and d. Note that Figure 2 uses normalized data.

3. Security Attacks

The attacks that this report is trying to detect are BGP redirect attacks. This attack is similar to a *man-in-the-middle*. In the MITM attack the communication between server-client is altered, starting to be diverted by an attacker. This way the attacker can transmit the data without making any changes, or can change information. In this attack, the objective is usually to intercept confidential information and use it for personal purposes.

BGP stands for Border Gateway Protocol and is a protocol that allows traffic to flow across the Internet from a source IP to a destination IP. Each BGP router stores a data forwarding table with the best routes between autonomous systems, AS. They are constantly updated, thus allowing traffic to always travel along the shortest and most direct route. BGP also enables the large-scale growth of the Internet.

BGP can be eBGP, when it happens between two BGP routers of different Autonomous Systems (AS), and iBGP, when BGP communication takes place between the same AS. eBGP is implemented in border routers and is responsible for the connection between different organizations, [1].

The *routing* information is kept in a routing table by each of the routers. This table contains all routes including static routes or BGP routes (iBGP or/and eBGP), [18]. The BGP routing table is also useful for address resolution, so it should always be up to date with new routes. If an entry in this routing table is wrong, that is, if another router incorrectly advertises a route, it can have catastrophic results. Therefore, it is important to detect routing table failures as soon as possible.

4. Dataset Analysis

In this work a dataset related to Internet traffic redirection attacks [11] is analyzed. These attacks are caused by the poisoning of the BGP (*Border Gateway Protocol*).

BGP is a protocol used on the Internet for routing between Autonomous Systems. This protocol has no security mechanisms. Using the BGP protocol, *routers* located on the border between Autonomous Systems announce network prefixes and routes for these prefixes, with a route being a sequence of Autonomous Systems.

A malicious *router* may advertise a network prefix that does not belong to its Autonomous System, such as the YouTube prefix, causing traffic destined for that prefix to be redirected to itself.

In [11] a methodology for detecting redirection attacks was proposed based on a set of probes (*probes*) spread across the globe. Based on this infrastructure, measurements were taken and the data set that will be presented in this chapter was obtained.

There are 4 *targets* placed in different locations, 12 *probes* and 4 *relays*. A *target* is a data receiver that later sends them to the respective *probe*, a *probe* is a place from which data packets are sent and received, to make calculations and if we decide whether an observation will be anomalous or not, and finally a *relay* is an attacker that does the data diversion.

In Table 2 we can find all *targets*, *probes* and *relays* existing in the dataset.

The detection of redirect attacks is based on RTT measurements (Round-Trip-Time). In this case, the RTT is the amount of time it took the packets to make the *probe-target-probe* path. 10 packets are sent every 2 min and it is intended, from the data, to distinguish what is abnormal from what is regular. Regular traffic is traffic that takes the *probe-target-probe* route and anomalous traffic is traffic that takes the *probe-relay-target-probe* route. Among these 10 packages, the minimum, maximum, mean, median and standard deviation were selected to be considered as observations for the dataset.

5. Results

In this work, several algorithms were analyzed and tested, including the heuristics proposed by Salvador and Nogueira, the method of *Tukey*, the *Distance Based-Outlier*, the SAX (*Symbolic Aggregate aproXimation*) and a variation with the PAA (*Piecewise Aggregate Approximation*) and the *Tukey* method.

The first algorithm analyzed in this work was the heuristic proposed by Salvador and Nogueira. Initially it proved to be quite promising due to the fact that it uses sliding windows. However, it was necessary to adjust several parameters to achieve a favorable result. In this sense, it became a very sensitive algorithm to changes in the different parameters studied.

Then the method of *Tukey* was studied, which also uses sliding windows. It was found that this method is more robust and is not as susceptible to changes in the values of the parameters under study. The *Tukey* method achieves high values for all metrics, being one of the methods to consider for future experiments. The metrics studied to evaluate the performance of the algorithm were *Accuracy*, *Precision*, *Recall* and *F1-Score*.

The *Distance Based-Outlier* turned out to be better than expected, as it is not a method for time series. It was a method that was adapted taking into account the needs of the dataset. However, the metrics obtained were very similar to the metrics of the *Tukey* method and the computational time of the *Distance Based-Outlier* is higher than the *Tukey* method. As such, the best algoritm up to

Table 2: Dataset targets, probes and relays.

Targets	Probes			Relays
Chicago1	Amsterdam	Iceland	SaoPaulo2	LA1
Frankfurt1	Chicago2	Israel	Johannesburg1	Madrid
HongKong	VdM	LA2	Johannesburg2	Moscow
London	Frankfurt2	Milan	Sweden	SaoPaulo1

this point remains the *Tukey* method.

The algorithm that was analyzed next was SAX. SAX is best suited for a daily or even weekly comparison and doesn't use sliding windows, something that is quite important for this dataset. Sliding windows attenuate small level changes. On the other hand, they are useful when the objective is to detect attacks in real time, because it is possible to compare the current observation with a small set of past observations. After this analysis, we concluded that it is not the most suitable method. However, SAX includes a transformation done by PAA, which helps in dimensioning the dataset. Therefore, we later adapt PAA to the *Tukey* method, which was considered the best so far.

The adaptation of PAA with the *Tukey* method did not show significant differences in terms of performance, compared to the use of the *Tukey* method without PAA. PAA makes the algorithm faster as it significantly reduces the data set. Thus, it was concluded that of all the methods analyzed in this work, the best for the dataset under study is the *Tukey* method with the reduction made by PAA.

6. Conclusions

The detection of anomalies is extremely important in several areas such as telecommunications, security, commerce, management, among others. In certain cases it may be possible to use simpler anomaly detection methods based on distances, or dissimilarities between objects. In other cases, a simpler approach is not enough and more exhaustive algorithms are needed to be able to detect anomalies. The dataset studied in this work have temporal characteristics, as it corresponds to round-trip-time measurements over time. In this case, algorithms based on time series models are particularly promising for detecting anomalies.

References

[1] Symbolic aggregate approximation, August 2020.

[2] L. C. V. T. W. S. K. S. Chengwei Wang, Krishnamurthy Viswanathan. Statistical techniques for online anomaly detection in data centers. 2011.

[3] D. S. M. David Ruppert. *Statistics and Data Analysis for Financial Engineering with R examples*. Series Editor. Springer, 2015.

[4] A. F. Eamonn Keogh, Jessica Lin. Hot sax: Finding the most unusual time series subsequence: Algorithms and applications. 2005.

[5] R. T. N. Edwin M. Knox. Algorithms for mining distance-based outliers in large datasets. pages 392–403, 1998.

[6] L. C. P. S. A. U. N. V. Georgy Shevlyakov, Kliton Andrea. Robust versions of the tukey boxplot with their application to detection of outliers. pages 6506–6510, 2013.

[7] L. W. S. L. Jessica Lin, Eamonn Keogh. Experiencing sax: a novel symbolic representation of time series. pages 107–144, 2007.

[8] H. H. Kishan G. Mehrotra, Chilukuri K. Mohan. *Anomaly Detection Principles and Algorithms*. Series Editor. Springer, 2017.

[9] V. Krish. Piecewise aggregate approximation, February 2018.

[10] V. Krish. Symbolic aggregate approximation, June 2018.

[11] A. N. Paulo Salvador. Customer-side detection of internet-scale traffic redirection. 2014.

[12] D. S. S. Robert H. Shumway. *Time Series Analysis and Its Applications with R Examples*. Series Editor. Springer, 2017.

[13] V. Rodrigues. Métricas de avaliação: acurácia, precisão, recall... quais as diferenças?, April 2019.

[14] M. Silva. Modelação da incerteza e deteção de outliers para melhoria do diagnóstico de perdas em sistemas de abastecimento de Água, 2016.

[15] P. B. G. C. F. Spiros Papadimitriou, Hiroyuki Kitagawa. Loci: Fast outlier detection using the local correlation integral. pages 315–326, 2003.

[16] K. S. Sridhar Ramaswamy, Rajeev Rastogi. Efficient algorithms for mining outliers from large data sets. pages 427–438, 2000.

[17] A. Subtil. Latent class models in the evaluation of biomedical diagnostic tests and internet traffic anomaly detection, 2020.

[18] T. L. E. S. H. E. Y. Rekhter, Ed. A border gateway protocol 4 (bgp-4). January.

[19] T. Yiu. Understanding arima (time series modeling), April 2020.

[20] S. L. D. Yufeng Yu, Yuelong Zhu. Time series outlier detection based on sliding window prediction. 2014.