Platform for Anomaly Detection in Time-Series

October 22, 2018

Abstract: The goal of this paper is to present a platform that integrates a number of functionalities necessary in the process of anomaly detection, from preprocessing towards various anomaly detection techniques and visualization methods. The purpose of this tool is to allow a developer to test, select and fine tune different algorithms that best fit anomaly detection in a given domain. To demonstrate the utility of the platform, we present a series of experiments done with different methods for anomaly detection on time-series and evaluate their results.

Keywords: Anomaly Detection, time-series

1 Introduction

We rely on computer and automation systems to manage complex tasks. They are used in many fields and have many applications. Computer systems are used in industrial, academic, commercial and financial applications because of their speed and reliability. Because a defect in such a system can lead to large monetary loss and potentially even loss of life, we need systems that monitor other systems. These systems need to be able to detect faults and anomalous behavior in some way. Algorithms that detect anomalous behavior are therefore necessary and important.

Other applications of anomaly detecting systems include systems that monitor our health. We would like to identify as soon as possible the onset of any disease. By monitoring our health we may be able to spot individuals that have a high risk of contracting some disease and allow medical professionals to act in time to improve their quality of life.

Most anomaly detection algorithms were developed for a given problem or a given range of problems. While creating general algorithms is really hard and may not even be possible, it is best to try many different approaches. The problem we are trying to solve is the lack of a designated platform or tool that can aid in deciding which anomaly detection algorithm works best for a particular problem.

In this paper we propose a platform that enables a user to test a variety of anomaly detection algorithms, with emphasis on time series data. This data has the form $X = \{x_t \in \mathbb{R} : \forall t \geq 0\}$. Furthermore, we will use the following functional definition of what an anomaly is: An anomaly is a data point or set of data, which is significantly different from all other data points or sets. This means that in order to define an anomaly, one must first have a notion of nominal data. The term anomaly is relative and can not be applied to a data-point independently of any dataset.

The following chapters are structured as follows: In Section 2 we describe previous and related work on the subject. Next, in Section 3 we give a brief description of the developed platform and its features. In Section 4 we describe the types of anomaly detection methods provided by our platform. We will test these methods in Section 5. This will be followed by a discussion on the future research directions and implications in Section 6.

2 Related Work

Most methods for anomaly detection are developed for a given field or have a specific application. In [**nnfd1994**] and [**MMAD2006**] the authors developed methods to detect anomalies in airplane data. In [**ghad2018**] similar methods were developed for IoT. In [**ADDSCIL**] methods for detecting anomalies in Big Data are presented. In [**NDTSDII1996**] methods are used to detect anomalies in industrial machinery.

In [**outlierSurvey2014**] a host of algorithms and techniques are described and categorized. The authors found that while some algorithms in different fields are very similar, most algorithms are hard to generalize. They also note, that numerous formulations of anomaly detection problems are not sufficiently explored, i.e. it is not known how well some algorithms perform in a field that was not intended for that algorithm.

Efforts to bundle up different anomaly detection algorithms have already begun. In [**egads2015**] the authors introduced an open source, generic framework for detecting anomalies in large scale time-series data.

In [**tpad2018**] a platform is proposed, that offers tools for data visualization, filtering and classification for a variety of data formats, including but not limited to time series data.

We believe that there is a real advantage in a platform that offers a variety of anomaly detection methods to the user. One can test the performance of a number of anomaly detection algorithms for some given test data. By having as many implementations of these algorithms as possible, the user can easily test as many algorithms as she wants with minimal effort and cost.

3 The Platform

In this section we present the anomaly detection platform and it's functionalities.

4 Anomaly Detection Techniques

In this section we give short descriptions of the anomaly detection methods present in the platform.

4.1 Outlier Detection Methods

The first type of problem is concerned with classifying each element as anomalous or nominal. We could define a function $label$ that labels a data-point $x \in X$ either as anomalous or nominal:

$$c_x = label(x, X)$$

$$c_x \in \{Nominal, Anomaly\}$$

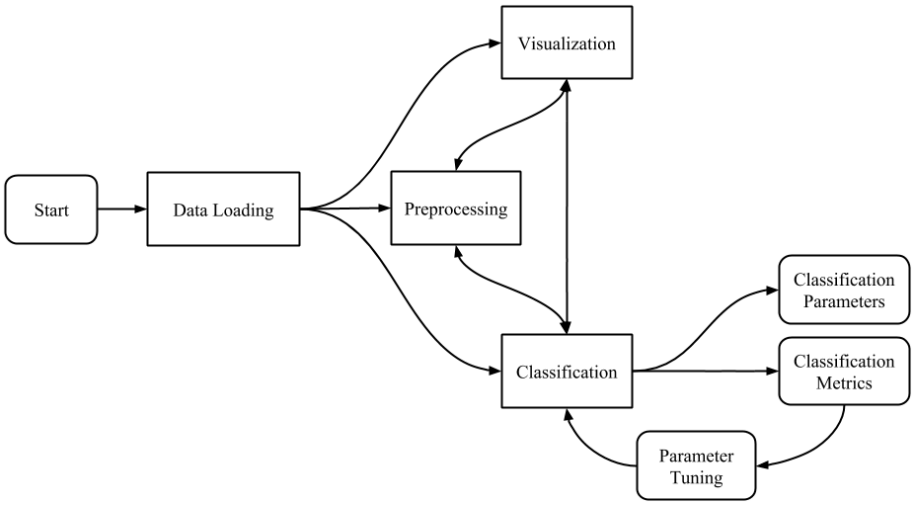

Figure 1: Workflow of the thing.

One might be able to label if an element is an anomaly by setting global lower and upper bounds. This can be used to detect obvious anomalies such as extreme temperature levels or very high blood pressure.

If such a predicate function is not possible, or doesn't meet the correctness requirements, a more complex approach can be used. Since the time series is generated by a generative process, one could hope to be able to accurately describe the underlying process, and create a model of the system:

$$x_t^* = f(t)$$

Given such a model, one can label the anomalies based on some threshold given a dstance metric $d : X \times X \to \mathbb{R}$. If the distance between the predicted value x_t^* and the actual value x_t is greater than some threshold d_{max}, x_t can be considered an anomaly:

$$p(x_t) = \begin{cases} \text{Anomaly} & \text{if } d(x_t, f(t)) \geq d_{max} \\ \text{Nominal} & \text{otherwise} \end{cases}$$

An ilustration of this can be seen in Figure 4.1.

In practice, this is almost never possible. Instead the model will predict the new values based on the old observed values: $f : x^n \to x^*, n \geq 1$. If $n < |X|$, we use a sliding window approach, where we generate a prediction for the next value based on the actual old values. This is useful if we can model the time series using an autoregressive model.

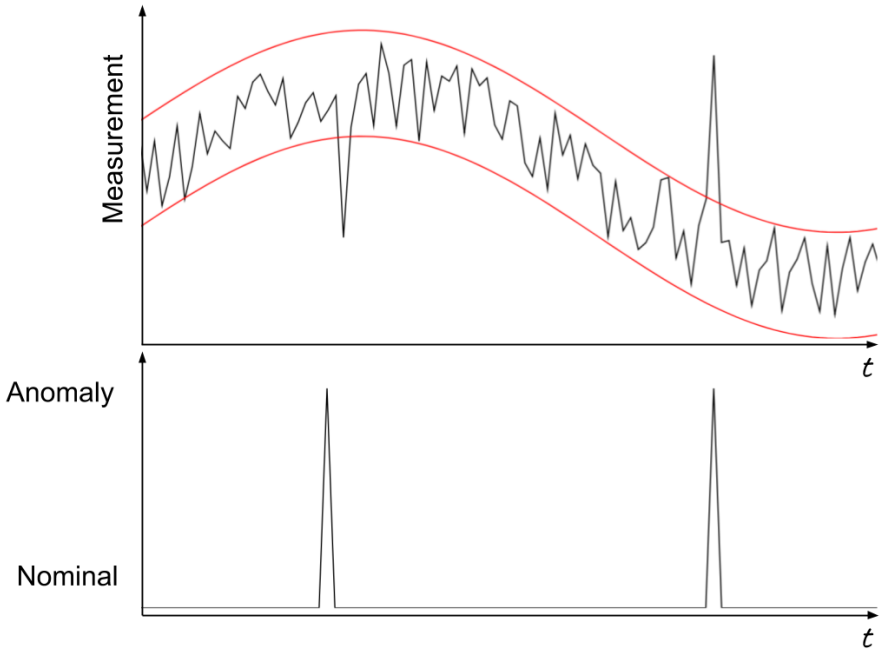

Figure 2: The upper graphs shows measured values from a process, and the lower figure is the classification. While the signal stays withing a certain range of the model, represented here by the upper and lower bounds, the signal is considered nominal. Points that are outside this "band of normality" are considered anomalies.

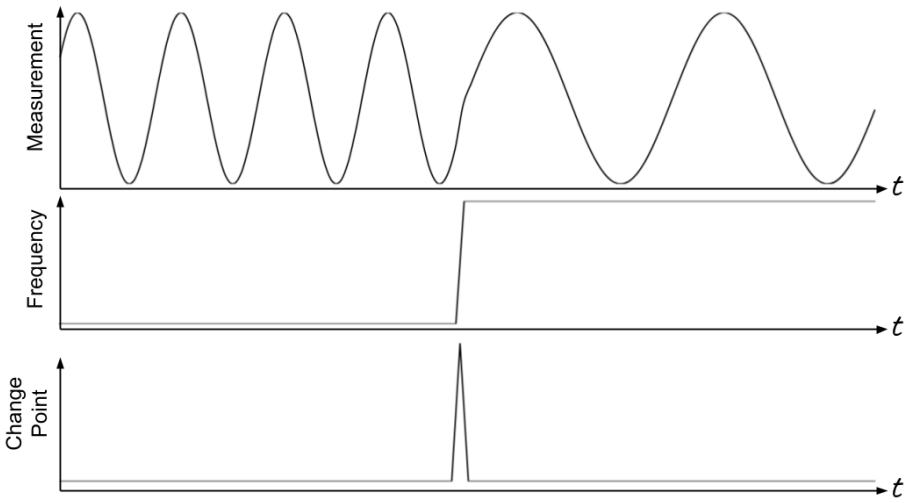

Figure 3: In the top most graph, we can see the observed process. In this case it is a pure sine wave. In the middle graphs we can see the coefficient of the model, which in this case is just a function of it's frequency, since it is enough to perfectly describe the process. We can consider as anomaly either the change point, either all the points where the model is outside some bounds.

4.2 Change Point Detection

4.3 Artificial Intelligence Classification

4.4 Anomalous Time Series Detection

5 Experiments

6 Discussion

Bibliography