



Pavel Tiunov

[Follow](#)

CTO @ Statsbot

Jun 8 · 7 min read

# Time Series Anomaly Detection Algorithms

The current state of anomaly detection techniques in plain language



At Statsbot, we're constantly reviewing the landscape of anomaly detection approaches and refining our models based on this research.

**This article is an overview of the most popular anomaly detection algorithms for time series and their pros and cons.**

This post is dedicated to non-experienced readers who just want to get a sense of the current state of anomaly detection techniques. Not wanting to scare you with mathematical models, we hid all the math under referral links.

## Important Types of Anomalies

Anomaly detection problem for time series is usually formulated as *finding outlier data points relative to some standard or usual signal*.

While there are plenty of anomaly types, we'll focus only on the most important ones from a business perspective, such as unexpected spikes, drops, trend changes and level shifts.

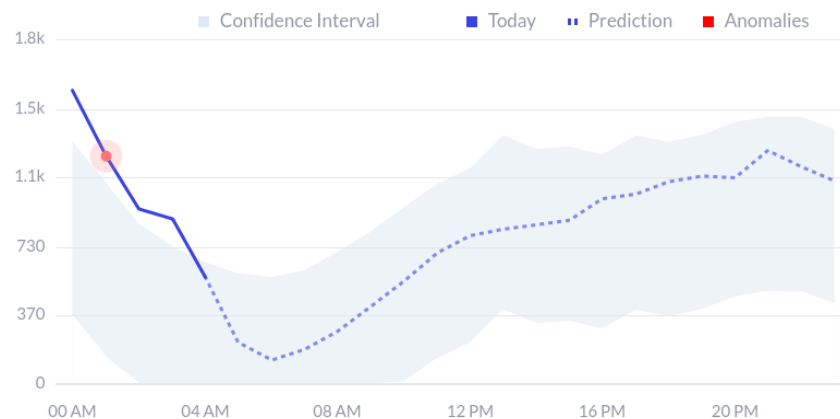
Imagine you track users at your website and see an unexpected growth of users in a short period of time that looks like a spike. These types of anomalies are usually called **additive outliers**.

Another example with the website is when your server goes down and you see zero or a really low number of users for some short period of time. These types of anomalies are usually classified as **temporal changes**.

In the case that you deal with some conversion funnel, there could be a drop in a conversion rate. If this happens, the target metric usually doesn't change the shape of a signal, but rather its total value for a period. These types of changes are usually called **level shifts or seasonal level shifts** depending on the character of the change.

Basically, **an anomaly detection algorithm should either label each time point with *anomaly/not anomaly*, or forecast a signal for some point and test if this point value varies from the forecasted enough to deem it as an anomaly.**

Using the second approach, you would be able to visualize a confidence interval, which will help a lot in understanding why an anomaly occurs and validate it.



Statsbot's anomaly report. Actual time series, predicted time series and confidence interval help understand why anomaly occurs.

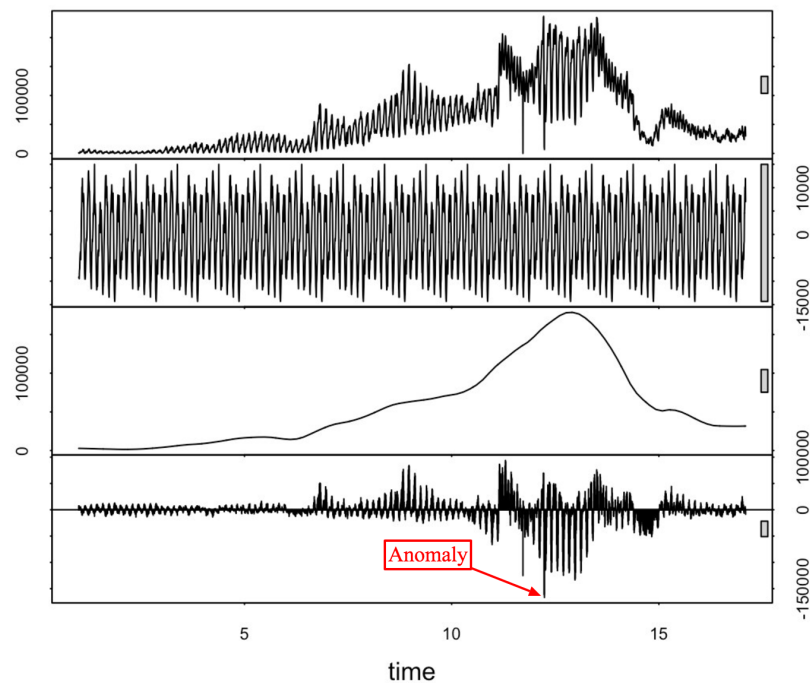
Let's review both algorithm types from the perspective of appliance to finding various types of outliers.

## STL decomposition

STL stands for **seasonal trend loess decomposition**. This technique gives you an ability to split your time series signal into three parts:

---

seasonal, trend and residue.



From top to bottom: original time series, seasonal, trend and residue parts retrieved using STL decomposition.

As the name states, it is suitable for seasonal time series, which is the most popular case.

*If you analyze deviation of residue and introduce some threshold for it, you'll get an anomaly detection algorithm.*

The not obvious part here is that you should use **median absolute deviation** to get a more robust detection of anomalies. The leading implementation of this approach is Twitter's Anomaly Detection library. It uses Generalized Extreme Student Deviation test to check if a residual point is an outlier.

## Pros

Pros of this approach are in its simplicity and how robust it is. It can handle a lot of different situations and all anomalies can still be intuitively interpreted.

It's good mostly for detecting additive outliers. To detect level changes you can analyze some rolling average signal instead of the original one.

## Cons

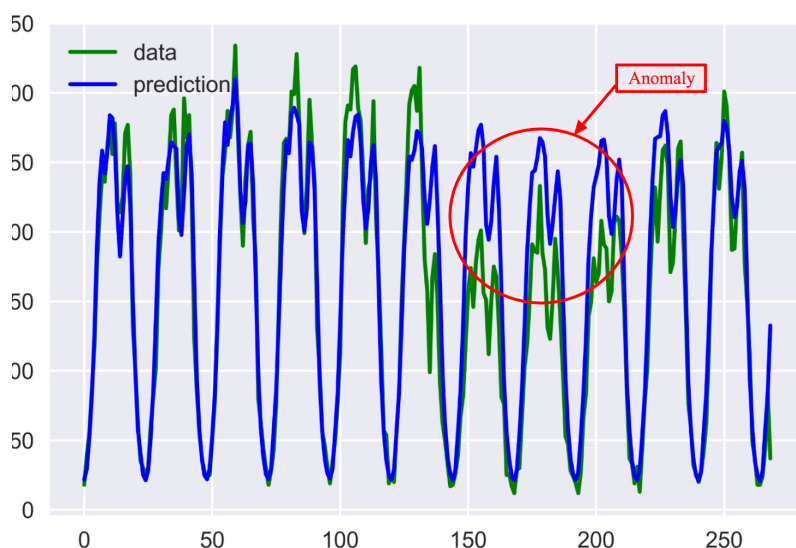
The cons of this approach are in its rigidity regarding tweaking options. All you can tweak is your confidence interval using the significance level.

The typical scenario in which it doesn't work well is when characteristics of your signal have **changed dramatically**. For example, you're tracking users on your website that was closed to the public, and then was suddenly opened. In this case, you should track anomalies that occur before and after launch periods separately.

## Classification and Regression Trees

Classification and regression trees is one of the most robust and most effective machine learning techniques. It may also be applied to anomaly detection problems in several ways.

- First, you can use **supervised** learning to teach trees to classify anomaly and non-anomaly data points. In order to do that you'd need to have labeled anomaly data points.
- The second approach is to use **unsupervised** learning to teach CART to predict the next data point in your series and have some confidence interval or prediction error as in the case of the STL decomposition approach. You can check if your data point lies inside or outside the confidence interval using Generalized ESD test, or Grubbs' test.



Actual time series (Green), predicted time series made using CART model (Blue), and anomalies detected as deviation from forecasted time series.

The most popular implementation to perform learning for trees is the xgboost library.

## Pros

The strength of this approach is that it's not bound in any sense to the structure of your signal, and you can introduce many feature parameters to perform the learning and get sophisticated models.

## Cons

The weakness is a growing number of features can start to impact your computational performance fairly quickly. In this case, you should select features consciously.

## ARIMA

ARIMA is a very simple method by design, but still powerful enough to forecast signals and to find anomalies in it.

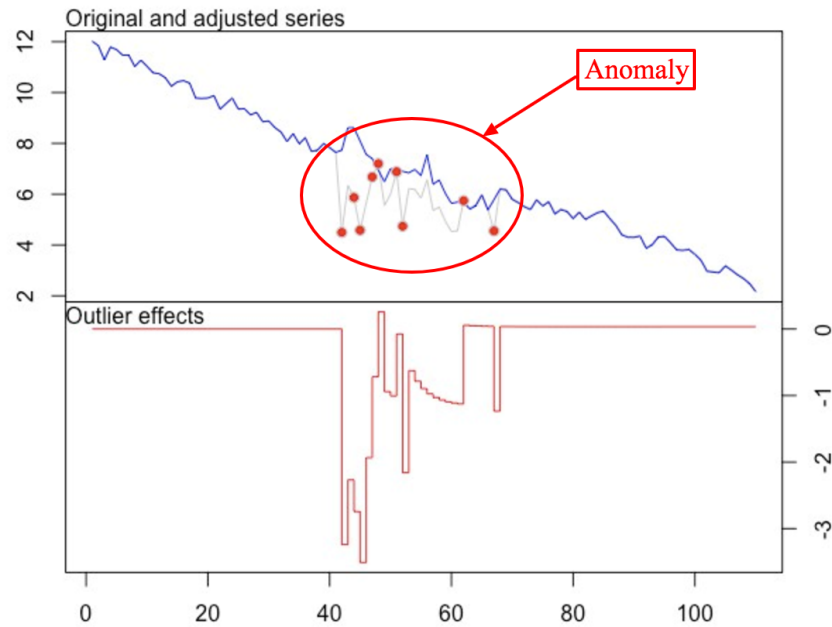
It's based on an approach that several points from the past generate a forecast of the next point with the addition of some random variable, which is usually white noise. As you can imagine, forecasted points in the future will generate new points and so on. Its obvious effect on the forecast horizon: the signal gets smoother.

The difficult part in appliance of this method is that you should select the number of differences, number of autoregressions, and forecast error coefficients.

*Each time you work with a new signal you should build a new ARIMA model.*

Another obstacle is that your signal should be stationary after differencing. In simple words, it means your signal shouldn't be dependent on time, which is a significant constraint.

Anomaly detection is done by building an adjusted model of a signal by using outlier points and checking if it's a better fit than the original model by utilizing t-statistics.



Two time series built using original ARIMA model and adjusted for outliers ARIMA model.

The favored implementation of this approach is tsoutliers R package. It's suitable to detect all types of anomalies in the case that you can find a suitable ARIMA model for your signal.

## Exponential Smoothing

Exponential smoothing techniques are very similar to the ARIMA approach. The basic exponential model is equivalent to the ARIMA (0, 1, 1) model.

The most interesting method from the anomaly detection perspective is Holt-Winters seasonal method. You should define your seasonal period which can equal to a week, month, year, etc.

In the case you need to track several seasonal periods, such as having both week and year dependencies, you should select only one. Usually, it'll be the shortest one: a week in this example.

This is clearly a drawback of this approach, which affects the forecasting horizon a lot.

Anomaly detection can be done using the same statistical tests for an outlier, as in the case of STL or CARTs.

## Neural Networks

As in the case of CART, you have two ways to apply neural networks: **supervised and unsupervised learning**.

As we're working with time series, the most suitable type of neural network is **LSTM**. This type of Recurrent Neural Network, if properly built, will allow you to model the most sophisticated dependencies in your time series as well as advanced seasonality dependencies.

This approach can also be very helpful if you have multiple time series coupled with each other.

This area is still on-going research, and it requires a lot of work to build the model for your time series. Should you succeed, you may achieve outstanding performance results in terms of accuracy.


## To Keep in Mind

1. **Try the simplest model and algorithm that fit your problem the best.**
2. **Switch to more advanced techniques if it doesn't work out.**
3. **Starting with more general solutions that cover all the cases is a tempting option, but it's not always the best.**

At Statsbot, to detect anomalies at scale we use different combinations of techniques starting with STL and ending with CART and LSTM models.

**Was it helpful? Please recommend and share this article to help other people find it.**

## Enjoyed the article?

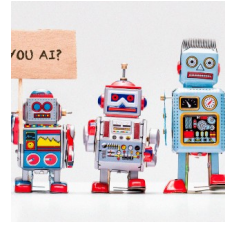
Press  below and join the other 13,000+ getting valuable updates from this blog.

## KEY TAKEAWAYS:

### A Big Data Cheat Sheet: From Narrow AI to General AI

Artificial Intelligence in 2017

[blog.statsbot.co](http://blog.statsbot.co)



### Which Metrics Matter The Most To You?

"Without a goal, analytics is aimless and worthless. A target should go with every goal. If a strategy meets a goal: It...

[blog.statsbot.co](http://blog.statsbot.co)



### Solve These 3 Common Problems to Save Your Analytics

It sometimes feels impossible to make sense or use of the vast amount of spreadsheets, reports, and other embodiments...

[blog.statsbot.co](http://blog.statsbot.co)





