**What is Anomaly Detection?**

Anomaly detection refers to using statistical models to identify unexpected or unusual events in data data. Anomaly detection is a well-studied academic field within statistics and machine learning, and in recent years has found many applications in industry – from system trading to fraud detection to observability and now, data observability.

**Why is Anomaly Detection Critical for Data Observability?**

Data observability consists of monitoring, alerting, metadata, and lineage.

Anomaly detection plays a critical role in the monitoring and alerting component of data observability. Since data observability is about knowing the state of your data systems at all times, and reacting appropriately when there is a problem, anomaly detection helps by identifying anomalies in data that might correspond to a data outage.

**Why manually set rules aren’t always sufficient**

The main alternative to anomaly detection for data observability are **simple, manual thresholds and rules**. For example, for a metric that tracks the percentage of successful Airflow jobs, we might set a manual threshold of 90%. If more than 10% of Airflow jobs failed over a certain period of time, then we’ll get an alert.

Simple, manual thresholds work for many scenarios, in particular in systems observability, where metrics are expected to remain constant over time. However, they also have a number of drawbacks:

1. They require manual setting and tuning, which is fine when you have one data metric but not when you have 1000.
2. They are inaccurate for metrics that are not constant.

**Why simple Anomaly Detection methods aren’t sufficient**

Some of the simplest anomaly detection methods are easy to grasp and implement, for example, **formulaic thresholds that are calculated using simple statistics**: if your current data point exceeds the mean + standard deviation of a certain lookback window, then let’s call the data point an anomaly.

This kind of proto-anomaly detection, while automatable, is limited in its efficacy. It doesn’t work on data that goes through phases or data where the current value is linked to the previous value. Moreover, most businesses are dynamic and changes are made incrementally. When these changes are rolled out, they can create huge jumps in data that render past means and standard deviations moot.

**Benefits of Bigeye’s anomaly detection for data observability**

Bigeye’s advanced anomaly detection goes beyond formulaic thresholds, using a mixture of sophisticated statistical models to pick up on anomalies in the data that would otherwise be missed. There are three main advantages: accurate detection, intelligent adaptation, and continuous improvement.

## Accurate Detection

Accurate detection is especially important in the data observability context because both false negatives and false positives are high-stakes: if your anomaly detection algorithm fails to detect an anomaly that turns out to be a data outage, you’ve potentially lost customers and revenue. And if your anomaly detection algorithm detects an anomaly that turns out not to be a data outage, but alerts on-call engineers anyways, over time, you erode their trust in observability systems.

Bigeye’s advanced anomaly detection uses both forecast and non-forecast models to detect hard-to-detect anomalies. This includes tricky cases like slow degradation, where a metric trends downwards slowly enough to go undetected. This can be problematic – imagine if the slowly downtrending metric is revenue, or number of requests!

Additionally, Bigeye’s anomaly detection understands trends and seasonality. If there’s a huge jump in the data, it can progress the history and the pattern by looking at user feedback.

## Intelligent Adaptation

Businesses are by nature dynamic, and their data reflects that: they might generate more data; they might begin to collect more user data, they might switch analytics systems. If data engineers set their thresholds too narrow, they could end up being overwhelmed by too many alerts. But by setting their thresholds too wide, they could potentially miss anomalies, some of which may be crucial to the business.

With Bigeye’s [Autothresholds](https://www.bigeye.com/product/autothresholds), dynamic boundaries have been put in place to examine historical data and identify what an actual anomaly is compared to what’s not. Using this, data engineers and scientists can rely on Bigeye to do most of the work, while retaining the ability to adjust the alerting sensitivity and thresholds as needed.

**One of the benefits of this kind of “intelligent adaptation” is that anomaly detection heps you do data observability at scale.**

In practice, if you were a data scientist or engineer, you wouldn’t want to spend hours and hours re-tuning data thresholds and adjusting everything manually. While you’ll still have the option to make manual changes through the app, you can rely on Bigeye to do the heavy lifting.

## Continuous Improvement

Since each business and their respective data environments are unique, any data observation platform, including ours, can still produce false alarms. What makes an anomaly detector effective, however, is how it adapts and learns from your team’s feedback and preferences.

With Bigeye, data teams have the option to tell the application about false positives through reinforcement learning. Bad values are also excluded from model training so that the system would not identify them as a normal state and adapt to them accordingly.

**Something is broken in the data - now what?**

Anomaly detection of data at rest is currently the prevailing philosophy of most data observability tools, but as a recent article pointed out, detecting anomalies alone is not sufficient. Once an anomaly has been identified, engineers want to be able to take action on them – to understand the root cause and assess the impact. What broke, why and where it broke – knowing the answers to these questions is critical for actually fixing the problem.

Bigeye is making anomaly detection proactive by providing a UI that aids your investigations, including:

* Graphs of the data metric that
* Copy-pastable queries that easily allow you to debug
* “Incident centric” UI that makes it easy for you to promote a specific allert into an issue