Anodot

ULTIMATE GUIDE TO BUILDING A MACHINE LEARNING ANOMALY DETECTION SYSTEM

DESIGN PRINCIPLES

PART
1 of 3

ULTIMATE GUIDE TO BUILDING A MACHINE LEARNING ANOMALY DETECTION SYSTEM PART 1: DESIGN PRINCIPLES

Anomaly detection is an imperative for digital businesses today, but it is a complex task to design and build a truly effective system in-house.

INTRODUCTION

It has become a business imperative for high velocity online businesses to analyze patterns of data streams and look for anomalies that can reveal something unexpected. Most online companies already use data metrics to tell them how the business is doing, and detecting anomalies in the data can lead to saving money or creating new business opportunities. This is where the online world is going; everything is data- and metric-driven to gauge the state of the business *right now*.

The types of metrics that companies capture and compare differ from industry to industry, and each company has its own key performance indicators (KPIs). What's more, regardless of industry, all online businesses measure the operational performance of their infrastructure and applications. Monitoring and analyzing these data patterns in real-time can help detect subtle – and sometimes not-so-subtle – and unexpected changes whose root causes warrant investigation.

Automated anomaly detection is a technique of machine learning, and it is a complex endeavor. There might be hundreds, thousands or even millions of metrics that help a business determine what is happening right now compared to what it has seen in the past or what it expects to see in the future. Data patterns can evolve as well as interact, making it difficult to understand what data

THERE MIGHT BE HUNDREDS, THOUSANDS OR EVEN MILLIONS OF METRICS THAT HELP A BUSINESS DETERMINE WHAT IS HAPPENING RIGHT NOW COMPARED TO WHAT IT HAS SEEN IN THE PAST OR WHAT IT EXPECTS TO SEE IN THE FUTURE.

models or algorithms to apply. Companies that use the right models can detect even the most subtle anomalies. Those that do not apply the right models can suffer through storms of false-positives or, worse, fail to detect a significant number of anomalies, leading to lost revenue, dissatisfied customers, broken machinery or missed business opportunities.

Anodot was founded in 2014 with the purpose of creating a commercial system for real-time analytics and automated anomaly detection. Our technology has been built by a core team of highly skilled and experienced data scientists and technologists who have developed and patented numerous machine learning algorithms to isolate and correlate issues across multiple parameters in real-time.

The techniques described within this white paper series are grounded in data science principles and have been adapted or utilized extensively by the mathematicians and data scientists at Anodot. The veracity of these techniques has been proven in practice across hundreds of millions of metrics from Anodot's large customer base. A company that wants to create its own automated anomaly detection system would encounter challenges similar to those described within this document series.

This document, Part 1 in a three-part series, covers the design principles of creating an anomaly detection system. Part 1 will explore various types of machine learning techniques, and the main design principles that affect how that learning takes place. Part 2 of the document series will explore a general framework for learning normal behavior of a time series of data. Part 3 will cover the processes of identifying and correlating abnormal behavior. Each of the documents will discuss the technical challenges and Anodot’s solutions to these challenges.

WHY COMPANIES NEED ANOMALY DETECTION

If anomaly detection is so complex and full of technical challenges, why do it? To detect the unknown.

In a high velocity business, many things occur simultaneously, and different people/roles are charged with monitoring those activities. For example, at the level of the underlying infrastructure, a technical IT group carefully monitors the operation and performance of the network, the servers, the communication links, and so on. At the business application level, an entirely different group monitors factors such as web page load times, database response time, and user experience. At the business level, analysts watch shopping cart conversions by geography and by user profile, conversions per advertising campaign, or whatever KPIs are important to the business.

Anomalies in one area can affect other areas, but the association might never be made if the metrics are not analyzed on a holistic level. This is what a large-scale anomaly detection system should do.

Still, the question arises, why care about anomalies, especially if they simply seem to be just "blips" in the business that appear from time to time? Those blips might represent significant opportunities to save money (or prevent losing it) and to potentially create new business opportunities. Consider these real-life incidents:

- An e-commerce company sells gift cards and sees an unexpected increase in the number of cards purchased. While that sounds like a great thing, there is also a corresponding drop in the revenue expected for the gift cards. Something strange is going on, and it turns out to be a price glitch—something quite common for e-commerce companies. Without looking at these two metrics together, it is hard to understand that there is a business incident that could cost the company a lot of money if not caught and addressed quickly. The relationship between these two metrics is shown in Figure 1.

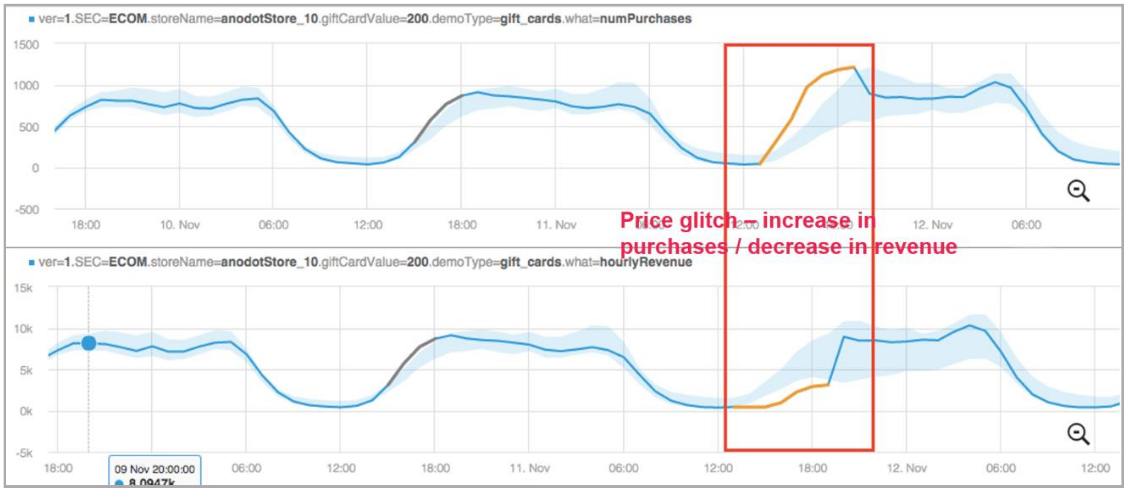

Figure 1 – Detecting business incidents

- A mobile game company notices a decrease in installations of one of its games, but may discover it several weeks in and not know why. With an anomaly detection system, it is easy to determine that a problem with the cross-promotion mechanism serving up the wrong ads or ads with the wrong link is what led to the decline.

As businesses grow, more incidents go undetected unless an anomaly detection system is directed to make sense of the massive volume of metrics available to every online business. Of course, not every metric is directly tied to money—but most metrics are tied to revenue in some way. Say an online news site counts visitors to its website. By itself, the visitor count doesn't lead to revenue, but the more visitors the news site gets, the more opportunity there is to generate revenue from ads on the pages, or to convert people who read the news site from free to paid subscribers.

AS BUSINESSES GROW, MORE INCIDENTS GO UNDETECTED UNLESS AN ANOMALY DETECTION SYSTEM IS DIRECTED TO MAKE SENSE OF THE MASSIVE VOLUME OF METRICS AVAILABLE TO EVERY ONLINE BUSINESS.

Most companies today tend to do manual detection of anomalous incidents. There are two main ways to do this:

- One is to create a lot of dashboards, create daily/weekly reports and have people monitor them to watch for spikes or dips. Then they investigate whatever looks strange. Obviously, this method isn't scalable to more than a dozen or so key metrics. A business could potentially detect major anomalies but would miss many smaller incidents. Moreover, the people would need to know what to look for, so they will miss things they didn't think to track or analyze. This process also has the business looking in the rear-view mirror at events that happened in the past, which delays the insights by days or weeks.
- The second method is to use a system that depends on setting upper and lower thresholds for each metric. An alert can be generated if a measurement goes outside those thresholds. The downside here is that setting these thresholds is complicated as it must be done for each KPI, which is difficult to do if there are many thousands of them. Setting thresholds requires an intricate understanding of the behavior of each metric over time. Also, if the thresholds are too high or too low, there could be a lot of false-positive alerts, or conversely, a lot of missed anomalies. In short, finding anomalies by setting thresholds is an impractical endeavor.

The solution, then, is automated anomaly detection, where computers look at this data, sift through it automatically and quickly, highlight abnormal behavior, and alert on it. This is far easier said than done, for the computers need to be taught what an anomaly is compared to normal activity. This is where machine learning comes in.

MACHINE LEARNING METHODS

Anomaly detection is a branch of machine learning, which itself is a type of artificial intelligence that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of computer programs that can teach themselves to grow and change when exposed to new data. The process of machine learning involves searching through data to look for patterns and adjusting program actions accordingly. There are two main types of machine learning methods: supervised and unsupervised.

MACHINE LEARNING METHODS:

SUPERVISED
UNSUPERVISED
SEMI-SUPERVISED

Supervised learning problems are ones in which the computer can be given a set of data and a human tells the computer how to classify that data. Future datasets can then be evaluated against what the machine has already learned. Supervised learning involves giving the machine many examples and classifications that it can compare against. This is not feasible for anomaly detection, since it involves a sequence of data where no one has marked where there are anomalies. The machine has to detect for itself where the anomalies are—if they even exist. Thus, a mathematical algorithm that processes the data must be able to detect an anomaly without being given any examples of what an anomaly is. This is known as unsupervised machine learning.

A third category exists, known as semi-supervised machine learning, and this is where Anodot fits in. This is discussed in more detail in the section titled “Definition of Incidents.”

WHAT IS AN ANOMALY?

The challenge – for both machines and humans – is identifying an anomaly. Very often the problem is ill-posed, making it hard to tell what an anomaly is. Consider the set of images in Figure 2.

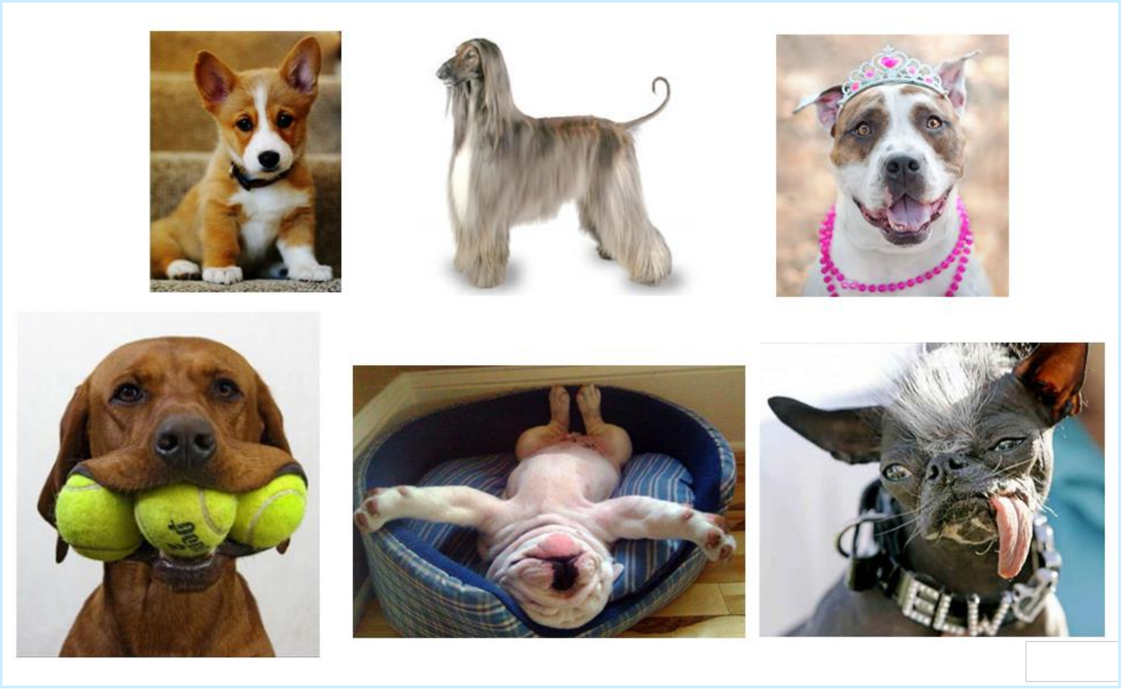

Figure 2 – What is an anomaly?

In this collection of dog pictures, which one is the anomaly? Is it the sleeping dog, because all the others are awake? Is it the dog with the very long fur? Is it the dog with balls in his mouth, or the one with a tongue hanging out? Depending on what criteria are used to define an anomaly, there could be many answers to the question. Thus, there must be some constraints on identifying an anomaly; otherwise almost any answer can be considered correct.

That is the philosophical part of anomaly detection. Fortunately, many metrics from online systems are expressed in time series signals rather than images. In time series signals, an anomaly is any unexpected change in a pattern in one or more of the signals. Figures 3, 4 and 5 present examples of unexpected changes in time series signals. In these examples, the anomalies are shown in orange to make them easier to visually identify.

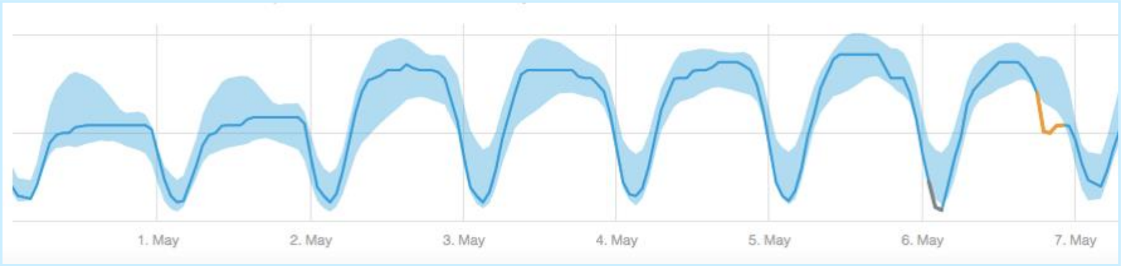

Figure 3 – Anomalies in a single time series signal

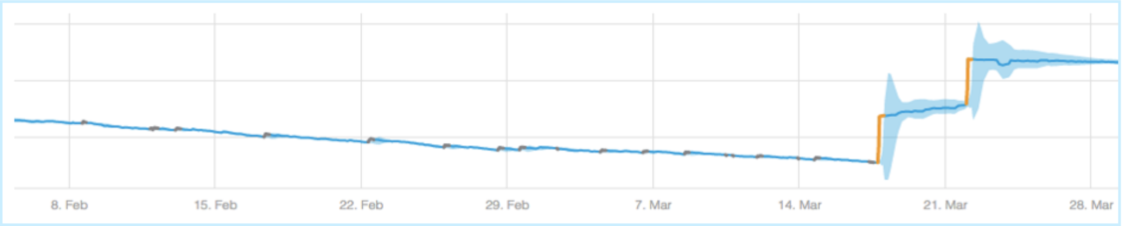

Figure 4 – Anomalies in a single time series signal

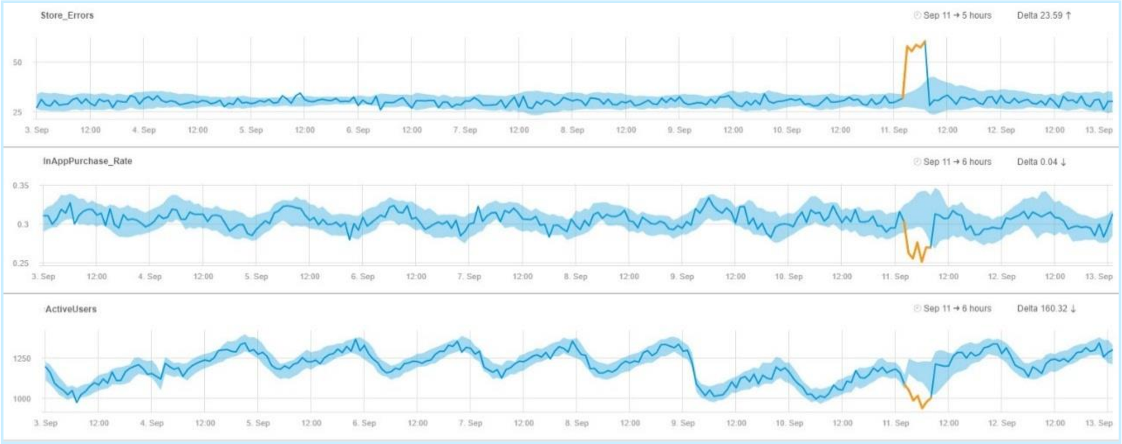

Figure 5 – Anomalies in multiple time series signals

Viewing these examples, humans would probably categorize the signal changes as anomalies because humans have some concept in mind as to what "unexpected" means. We look at a long history of the signal and create a pattern in our minds of what is normal. Based on this history, we expect certain things to happen in the future.

That is the crux of the question, "What is an anomaly?" It is still ill-defined.

Looking at the example in Figure 6 below, we could say that the two days highlighted in the red box are anomalous because every other day is much higher. We would be correct in some sense, but it is necessary to see more history to know if those days are real anomalies or not. There are a lot of definitions that go into an algorithm to determine with a high degree of confidence what an anomaly is.

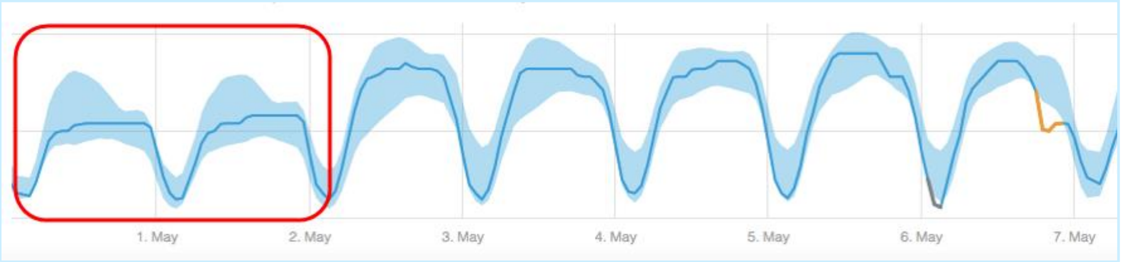

Figure 6 – What part of the signal is anomalous?

But anomaly detection is not impossible. There are obvious things that most people agree on of what it means to be the same, and that's true for time series signals as well—especially in the online world where companies measure number of users, number of clicks, revenue numbers and other similar metrics. People do have a notion of what "normal" is, and they understand it because, in the end, it affects revenue. By understanding "normal," they also understand when something is "abnormal." Thus, knowing what an anomaly is isn't completely philosophical or abstract.

That leads to a discussion of the main design principles when building an anomaly detection system. A company must look at why it needs such a system and then decide what methods are the right ones for its own use cases.

DESIGN PRINCIPLES OF ANOMALY DETECTION

Based on our experience, there are five main design considerations when building an automated anomaly detection system:

1. **Timeliness** – How quickly does the company need to an answer to determine if something is an anomaly or not? Does the determination need to be in real-time, or is it okay for the system to determine it was an anomaly after a day, week, month or year has already passed?
2. **Scale** – Does the system need to process hundreds of metrics, or millions? Will the datasets be on a large scale or a relatively small scale?
3. **Rate of change** – Does the data tend to change rapidly, or is the system being measured relatively static?
4. **Conciseness** – If there are a lot of different metrics being measured, must the system produce an answer that tells the whole picture, or does it suffice to detect anomalies at each metric level by itself?
5. **Definition of incidents** – Are the expected incidents well-defined? Is anything known about them in advance in terms of what types of things can be anomalous in the data? Can incidents be categorized over time?

TIMELINESS AND SCALE

Most online businesses need real-time decision-making capabilities, which means they have to know what's happening with their data right now. If something anomalous happens – for example, a sudden drop in site visitors or installs, or an unexpected spike in sales of a particular product – they need to know quickly in order to make and act on decisions *right away*. Each incident could lead to different types of issues or opportunities that are worthy of investigation. A sudden increase in product purchases might mean the company needs to increase its inventory because a celebrity endorsed this product and it is now wildly popular. Selling out without having new inventory coming in could mean significant lost revenue. Or, a sudden spike in purchases of a specific product could mean that the price was entered incorrectly, and shoppers are effectively getting a 90% discount so they are running up sales.

Non-real-time decision-making can be used for anything that relates to longer term planning, such as capacity planning, budget planning, marketing campaign analysis, A/B testing analysis, scheduled maintenance and data cleaning. The anomalies are used for a retrospective analysis of what happened, which helps in making decisions about the future. For example, an anomalous increase in the number of purchases after a marketing campaign can lead to a decision to invest in similar campaigns in the future. Real-time decision-making isn't necessary for those activities. It's sufficient to get yesterday's anomalies a week from now; i.e., doing anomaly detection on the data retroactively.

The distinction of when anomaly detection must take place – in real-time or not – is critically important in understanding the type of machine learning algorithms that can be used.

If real-time decision-making is required, the system must use online machine learning algorithms. Online machine learning algorithms process data sequentially, and the latest data point is used to update the best predictor for future data at each step. In other words, the machine learns as the data comes in piece by piece, rather than waiting for an entire dataset to be available from which to learn.

THE DISTINCTION OF WHEN ANOMALY DETECTION MUST TAKE PLACE – IN REAL-TIME OR NOT – IS CRITICALLY IMPORTANT IN UNDERSTANDING THE TYPE OF MACHINE LEARNING ALGORITHMS THAT CAN BE USED.

Some characteristics of online machine learning algorithms are that they scale more easily to more metrics and to large datasets, and it is not possible to iterate over the data—once data is read it is not considered again. This method is more prone to presenting false-positives because the algorithm gets a data point and has to produce a result; it cannot go back to fix that result at a later time. We will go into more detail about online machine learning in Part 2 of this document series.

If time is not a critical factor in decision-making, batch machine learning algorithms can be utilized. A company can collect data over a period of time and then apply the algorithm to that complete dataset for a list of anomalies that happened in that time period. There is no time urgency, so the algorithm can go over the data again and again to improve what the machine learns. A lot of algorithms do improve by repetitively going over the data, which ultimately results in fewer false-positives. However, this process is computationally expensive and scales poorly.

Consider the need to learn the average number of users that come to a particular website every day. There are two ways to learn this. One is to collect all the data over a period of several months and then compute that average. The alternative is to compute it as the data comes in. The latter method means that the company cannot go back and fix issues in what was previously learned about the average. The algorithm sees the data, uses it and then sets it aside; however, it does not go back and fix anything if there were anomalies that skewed the calculation of the average. This example illustrates why online machine learning algorithms tend to be more prone to false-positives. On the other hand, online learning methods tend to be more scalable; it is easier to scale them to a much higher number of data points/metrics to learn on.

At Anodot, we focus on real time anomaly detection at a massive scale, which determines the types of algorithms we can use. We, indeed, use online machine learning algorithms.

RATE OF CHANGE

In our experience, most of the businesses that we work with have constant change in their metrics. Their environment changes; they release new products or new versions of their applications, which in turn changes the patterns of how people use them.

The graph in Figure 7 below is an example from an Anodot customer; it represents a technical metric related to an application. It is obvious that the metric had a pattern that was fairly consistent for a while, and then it completely changed its behavior at some point in time, and it stayed this way for a long time.

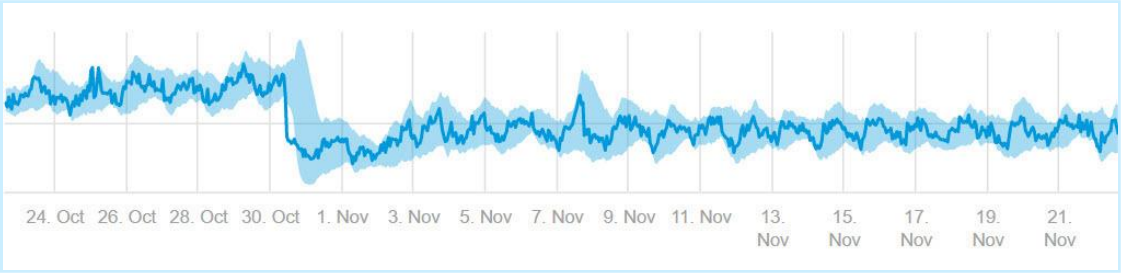

Figure 7 – This metric has a change in its pattern

Some systems change very slowly. They tend to be closed systems that are not impacted by outside events. For example, automated manufacturing processes tend to be closed systems that do not change much over time. When a company manufactures some sort of widget, the process is typically fairly steady. The machine operates with the same amount of force to stamp out each widget; the widget is heated in a furnace that is within a strict temperature range; the conveyor belt moves the widgets down the line at a constant speed; and so on throughout the manufacturing process. A metric with a slow rate of change might look like the graphic shown in Figure 8 below.

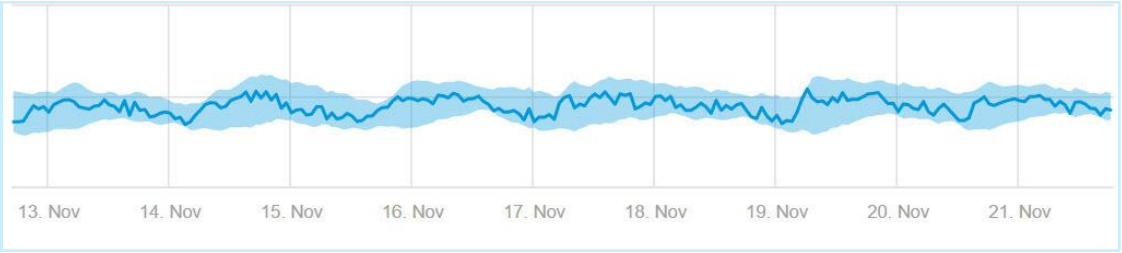

Figure 8 – This process has a slow rate of change

The rate of change has implications on the learning algorithms that an anomaly detection system should use. If a system has constant changes – which most online businesses do – then the system needs adaptive algorithms that know to take into account that things change. However, if the rate of change is very slow, the system can collect a year's worth of data and learn what is normal from that dataset. Then the model should not need to be updated for a long time. For example, the process to manufacture a widget is not going to change, so the anomaly detection system can apply an algorithm that does not have to be adaptive to frequent changes. On the other hand, data pertaining to an e-commerce website will change frequently because that is just the nature of the business.

At Anodot, the most interesting problem to solve is the one that changes frequently, so we utilize highly adaptive algorithms that take this into account.

CONCISENESS

Conciseness means the system takes multiple metrics into account at once for a holistic look at what is happening.

As an example of what is meant by conciseness: consider the human body as a type of system. It is possible to measure a person's vital signs, including blood pressure, body temperature, pulse rate and respiration rate. Assume that a person is wearing sensors for all those vital signs, and measurements are taken every minute. Under normal conditions – i.e., no sickness – measurement ranges of the vital signs are stable; a person's body temperature does not fluctuate by much throughout a normal day. However, if the person comes down with the flu, the signals from the body will appear anomalous: temperature goes up, pulse rate might change as the person becomes lethargic, respiration rate changes as breathing becomes more labored, and so on. All of these metrics, these vital signs, become anomalous more or less together.

Now the question is, how does a doctor look at all of these vital signs? If he or she looks at each measurement by itself, it will not be clear what is going on. When the pulse rate decreases, by itself, it does not tell the doctor very much. Only the combination of these metrics can begin to tell a story of what is going on.

And so it is with many business systems. Businesses measure many metrics, and all of them together tell a story about an issue. For example, a buggy deployment of a new version of a service may lead to high latency in page loads, high bounce rates, reduced number of visitors, and many more anomalous metrics. Viewed individually, each metric may not point to the root cause, but together they tell a clear story. For proper root cause analysis on most incidents, a company needs concise anomalies.

BUSINESSES MEASURE MANY METRICS, AND ALL OF THEM TOGETHER TELL A STORY ABOUT AN ISSUE.

In terms of design of the anomaly detection system, there are two methods that take conciseness into consideration: univariate anomaly detection and multivariate anomaly detection.

UNIVARIATE ANOMALY DETECTION

With univariate anomaly detection, the system looks at each metric by itself, learning its normal patterns and yielding a list of anomalies for each single metric. Oftentimes, it is difficult to perform root cause analysis of an issue because it is hard to see the forest for the trees. The advantage of univariate anomaly detection is that it is a lot easier to do than other methods. It is easier to scale in terms of computation. Less data is needed to learn what is normal because the system looks at each metric by itself, as opposed to looking at combinations of metrics. It is possible to model a lot of different types of metric behaviors. However, when something unexpected happens that affects a lot of metrics, the system yields a storm of anomalies. Now someone has to sift through them to understand what is happening.

MULTIVARIATE ANOMALY DETECTION

Multivariate anomaly detection techniques take input from all the signals together as one, without separating them out. In the example of the human body, take all of a person's vital signs and put them into a black box that outputs a single model of what is normal for all of them together. When the person has the flu, the anomaly describes the flu based on all the vital signs together.

There are downsides to using multivariate anomaly detection techniques. For one thing, these methods are very hard to scale. They are best when used with just several hundred or fewer metrics. Also, it is often hard to interpret the cause of the anomaly. All of the metrics are taken as input but the output simply says there is something strange—an anomaly, without identifying which metric(s) it is associated with. In the healthcare analogy, the doctor would put in the vital signs and receive “the patient is sick,” without any further explanation about why. Without having insight into what is happening with each metric, it is hard to know which one(s) affect the output, making it hard to interpret the results.

Another technical issue with these multivariate techniques is that they require all the measured metrics to be somewhat homogeneous in their behavior; i.e., the signal type must be more or less similar. If the set of signals or metrics behave very differently from each other, then these techniques tend to not work well.

A HYBRID APPROACH

The univariate method causes alert storms that make it hard to diagnose why there is an anomaly, and the multivariate methods are hard to apply. Anodot utilizes a hybrid approach to take advantage of the good aspects of each method, without the technical challenges they present. Anodot learns what is normal for each one of the metrics by themselves, and after detecting anomalies the system checks if it can combine them at the single metric level into groups and then give an interpretation to that group.

We never have a model that indicates how all the metrics should behave together. Instead we have a model for each metric by itself, but when some of them become anomalous, we look for smart ways to combine related anomalies into a single incident. This hybrid approach offers a practical way to achieve very good results. The main challenge in this approach is how to know which metrics are related to each other. We will describe how to use machine learning algorithms to automatically discover these relationships in the third part of this series.

Table 1 summarizes the characteristics of all three approaches to conciseness of data.

Univariate Anomaly Detection	Multivariate Anomaly Detection	Hybrid Approach
➢ Learn normal model for each metric ➢ Anomaly detection at the single metric level ➢ Easier to scale to large datasets and many metrics ➢ Causes anomaly storms—can't see the forest from the trees ➢ Easier to model many types of behaviors	➢ Learn a single model for all metrics ➢ Anomaly detection of a complete incident ➢ Hard to scale ➢ Hard to interpret the anomaly ➢ Often requires metric behavior to be homogeneous	➢ Learn normal model for each metric ➢ Combine anomalies to single incidents if metrics are related ➢ Scalable ➢ Make interpretation from groups of anomalies ➢ Can combine multiple types of metric behaviors ➢ Requires additional methods for discovering the relationships

Table 1 – Characteristics of univariate, multivariate and hybrid anomaly detection methods

DEFINITION OF INCIDENTS

The last design principle asks the question, “Are incidents well-defined?” While the answer to this question is typically “no,” as incidents for an online business are almost never well-defined, we will cover it because it provides the opportunity to further discuss supervised versus unsupervised learning and apply it to the design principle. In addition, it may be that over time, a business can define some incidents—leading to semi-supervised learning techniques.

A well-defined incident is one in which all (or at least most) of the potential causes of anomalies can be enumerated. This typically applies to a closed system with a very limited number of metrics. It might apply, for example, to a relatively simple machine where the product designers or engineers have written documentation of what could go wrong. That list could be used to learn how to detect anomalies. However, for an e-commerce website, it would be a Sisyphean task to try to list what could go wrong and break those things down to tangible incidents where mathematical models could be applied.

If a system has well-defined incidents, it is possible to apply supervised learning techniques. There is a well-defined set of incidents; now the system simply must classify the data into these sets. It requires labeled examples of anomalies, but it will be limited to the types of things the company is trying to use its system to find. If there is a list of a hundred different things that could go wrong, the system is trained about all of them, and then tomorrow the 101st thing happens that was not previously considered, the system will not be able to find it because the system is not trained to do so.

Supervised learning methods are very powerful because when they are properly trained on a set of well-defined incidents, they will catch those incidents and provide an exact definition for them. In terms of learning algorithms, they are usually more accurate than unsupervised learning methods. There are fewer false-positives, but a prerequisite is having a well-defined set of incidents to start with, and that is usually where companies encounter problems. It is very difficult to prepare a list of well-defined incidents and every possible thing that could happen.

SUPERVISED LEARNING METHODS ARE VERY POWERFUL BECAUSE WHEN THEY ARE PROPERLY TRAINED ON A SET OF WELL-DEFINED INCIDENTS, THEY WILL CATCH THOSE INCIDENTS AND PROVIDE AN EXACT DEFINITION FOR THEM.

The flipside is unsupervised learning methods, where a system learns what is normal over time. Anomalies are detected whenever the data that is now presented deviates from that normal model. The good thing is that such a system can detect any type of incident, known or unknown. The disadvantage is that the system is dependent on how "normal" has been defined. If the learning system does not do a good job of defining what is normal, it will get poor results with its anomaly detection. The system needs to know what the criteria are for being normal, and the detection technique is sensitive to that.

Then there are semi-supervised learning methods. Sometimes people can categorize examples where "this is a real anomaly, but that is not a real anomaly." It usually covers a very small subset of all the examples that can be identified. But getting even a few can be helpful. The question then becomes, can the unsupervised techniques be improved to be a little bit more supervised? For example, can it help a company choose what the normal model is that it needs to use? Can it help in choosing other things in the algorithms that learn what normal is and detect what is abnormal?

The answer is yes. There is a whole field called semi-supervised learning, and this is a technique that Anodot uses. We collect some feedback in our system to improve how we learn what is normal and how we detect anomalies based on a few examples that we get from our users. This helps in making the assumption of what is normal.

Table 2 summarizes the characteristics of these learning methods.

Supervised methods when incidents are well-defined	Unsupervised methods when incidents are not well-defined	Hybrid Approach
➢ Requires a well-defined set of incidents to identify ➢ Learning a model to classify data points as normal or abnormal ➢ Requires labeled examples of anomalies ➢ Cannot detect new types of incidents	➢ Learning a normal model only ➢ Statistical test to detect anomalies ➢ Can detect any type of anomaly, known or unknown	➢ Use a few labeled examples to improve detection of unsupervised methods ➢ -OR- ➢ Use unsupervised detection for unknown cases, supervised detection to classify already known cases

Table 2 – Characteristics of supervised and unsupervised learning, and a hybrid method

SUMMARY

Building an automated anomaly detection system for large scale analytics is a tremendously complex endeavor. The sheer volume of metrics, as well as data patterns that evolve and interact, make it challenging to understand what data models to apply. Numerous algorithm design principles must be considered, and if any of them are overlooked or misjudged, the system might overwhelm with false-positives or fail to detect important anomalies that can affect business. Data scientists must consider, for example:

- Timeliness – how quickly a determination must be made on whether something is an anomaly or not
- Scale – how many metrics must be processed, and what volume of data each metric has
- Rate of change – how quickly a data pattern changes, if at all
- Conciseness – whether all the metrics must be considered holistically when looking for anomalies, or if they can be analyzed and assessed individually
- Definition of incidents – how well anomalous incidents can be defined in advance

Anodot has built an anomaly detection system with these design considerations in mind. In parts 2 and 3 of the series, we will explain how these design considerations played into building the Anodot system. Part 2 looks at various ways that an anomaly detection system can learn the “normal” behavior of time series data. The concept of what is normal is a critical consideration in deciding what is abnormal; i.e., an anomaly. And part 3 explores the processes of identifying and correlating abnormal behavior.

For more information, please contact Anodot:

North America
669-600-3120
info.us@anodot.com

International
+972-9-7718707
info@anodot.com

ABOUT ANODOT

Anodot was founded in 2014, and since its launch in January 2016 has been providing valuable business insights through anomaly detection to its customers in financial technology (fin-tech), ad-tech, web apps, mobile apps, e-commerce and other data-heavy industries. Over 40% of the company’s customers are publicly traded companies, including Microsoft, VF Corp, Waze (a Google company), and many others. Anodot's real-time business incident detection uses patented machine learning algorithms to isolate and correlate issues across multiple parameters in real time, supporting rapid business decisions. Learn more at http://www.anodot.com/.