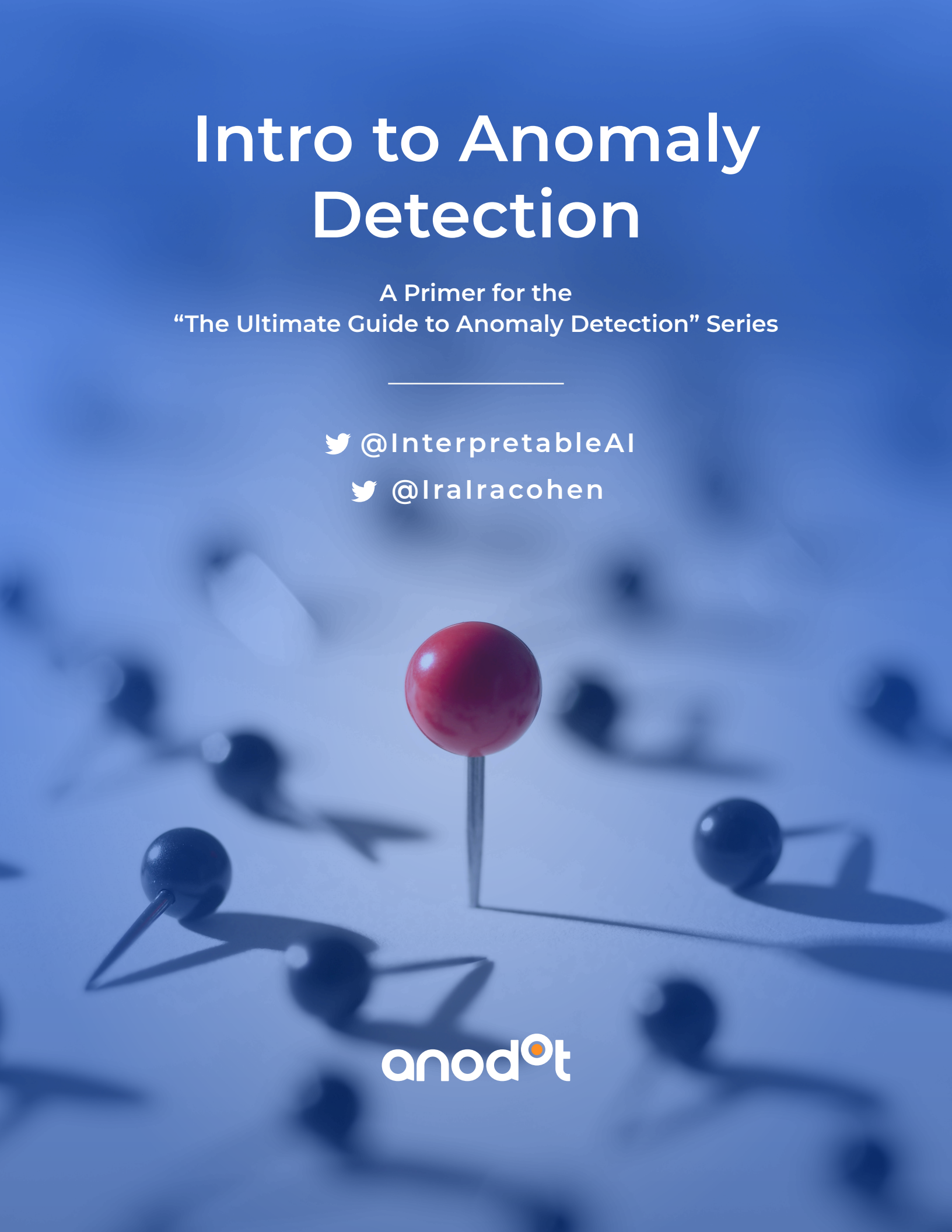
Intro to Anomaly Detection

A Primer for the
"The Ultimate Guide to Anomaly Detection" Series

@InterpretableAI

@IraIracohen



anodot

The Need for an Algorithmic Approach to Anomaly Detection

In recent years, data and analytics have rapidly reshaped organizations' core businesses. According to a recent survey from McKinsey, companies with high overall growth in revenue and earnings attribute a significant proportion of that boost to data and analytics. Further, the survey highlights:

> "... organizations are also making data a core part of employees' work flows and mind-sets by educating them as part of a broader effort to build a strong data-driven culture. All the while, they are ensuring that high-quality data and modern technological foundations are in place to support these efforts at scale."

The survey also listed two key challenges to empowering employees with data-based decision making as low quality data and companies' key data-management processes - from ingesting and cleaning data to tracking data quality, reporting and visualization.

As much as it has become easier over the years to collect vast amounts of data across the entire stack (courtesy, availability of a plethora of open-source tools), but companies need to ensure that the data they're gathering actually matters [1, 2, 3]. Since many domains require real-time or near-real-time (i.e., under 10 ms) insights, an abundance of unwelcome data can create real issues. To aid insight collection from the data, machine learning (ML) has become a ubiquitous tool. In fact, there is an increasing trend of applying ML at the edge itself.

Before decisions are made, and critically, before actions are taken, we must ask: are there anomalies in our data that could possibly skew the results of algorithmic analysis? If anomalies do exist, it is critical that we automatically detect and mitigate their influence on the ML models. This ensures that we get the most accurate results possible before taking action.

For example, consider a doctor who runs an electrocardiogram (ECG) test and recommends a particular course of treatment to a heart patient. However, the doctor doesn't realize that there are anomalies in the ECG results, making the patient appear to be more ill than expected. The doctor misdiagnoses the patient's condition, prescribes the wrong treatment and does not address the patient's real needs.

The need for an algorithmic approach to anomaly detection (anomaly detection) is clear. It stems from the three Vs of data: volume, velocity and veracity. Unlike humans, algorithms can sift through a much higher volume of data (thanks to the availability of computing resources on public clouds) and handle different data types. Moreover, algorithms can potentially surface anomalous patterns which would not be detected by human domain experts.

This article is intended as a precursor to our three-part series "The Ultimate Guide to Anomaly Detection", and provides an introduction to monitoring time series data. While that guide is intended for more technical audiences, this primer is geared towards business and data leaders interested in the foundations of anomaly detection in a business context, specifically:

- **Types of anomalies**
- **Types of data used for anomaly detection**
- **Frequency of data**
- **Context awareness**
- **Centralized vs. decentralized anomaly detection**

Read on as we delve into anomaly types, and the different applications and techniques used for anomaly detection.

Types of Anomalies

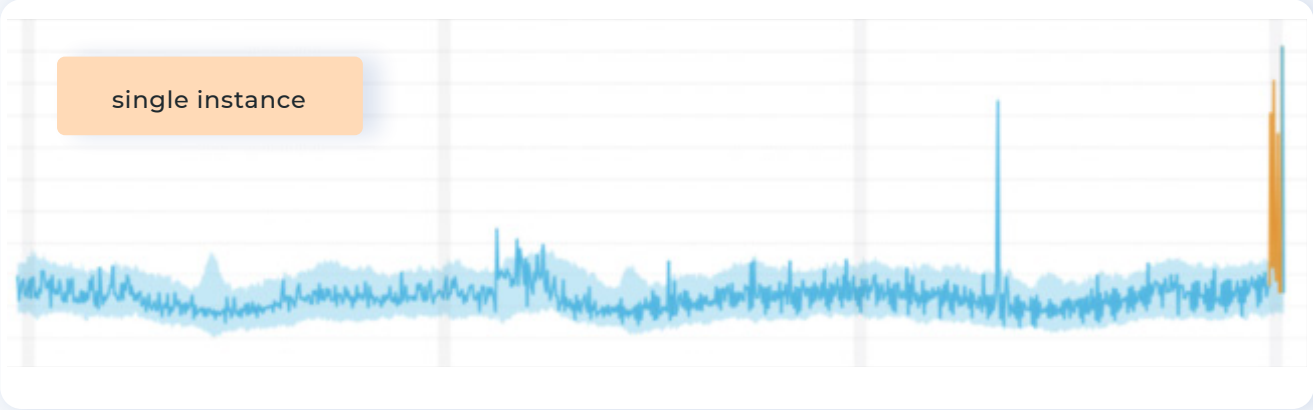
First, let's define anomalies. They can be anything that is different or abnormal and deviates in a substantial way from other data in the sample or from historical data.

Anomalies are often divided into point and pattern anomalies. Point anomalies are single instances of something that is abnormal, while pattern anomalies represent clusters of data that is abnormal. The following examples throw light on the distinction.

Some point anomalies may seem like a case of bad luck. Something as simple as a web app taking much longer than usual to load can signify a point anomaly.

Point Anomalies

- An online retailer tracks sales on a daily basis. Sales seem to follow a steady pattern, except for one day when they are just a fraction of their normal value.
- An online travel site monitors bookings per destination and discovers that bookings spike to a certain destination in the middle of the night. They realize that a price glitch offered the trip for a fraction of the actual price.
- A bank customer has kept a fairly steady balance between $10,000 and $15,000 in his savings account. One day he receives a bank transfer deposit of $100,000 from a foreign bank.
- There is a video surveillance system that monitors activity outside of a cafe 24/7. One night the system shows that someone breaks in and steals equipment.
- A credit card processor is watching for fraudulent activity and notes a transaction in a geography other than where the cardholder lives.
- A web app is taking much longer than usual to load.



With a pattern anomaly, the data trend either goes or down, or it's more choppy, creating a pattern that looks very different from normal. Pattern anomalies are more difficult to detect because we must first be able to define a normal pattern, and then we need far more data to be able to classify that this pattern is anomalous compared to historical activity. (For further discussion of the types of anomalies and the properties of the various anomaly detection techniques, check out [4]).

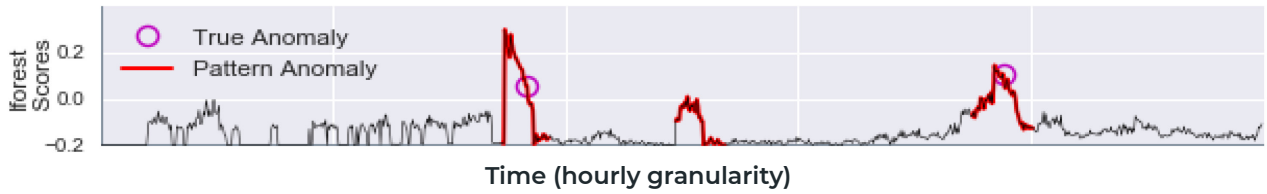
Pattern Anomalies

- A website that people use to renew their driver's license has a fairly steady amount of traffic during the typical business day. Suddenly and unexpectedly, on Monday between 9 AM and 4 PM, the traffic hitting the website spikes to ten times the normal level.

- A series of satellite photos taken over a specific region show that a large lake is shrinking in size, covering far less area than it did in previous months.

- A weather service tracks daily high and low temperatures for years. For three days in June, the temperatures are well below normal for that time of year.

- A bank operates a network of ATMs across a city. The typical amount of cash that is withdrawn from each machine on Monday averages around $10,000 per day. One Monday morning, dozens of ATMs in the network show a 3x spike in the amount of cash that is withdrawn.



Source: "On the Runtime-Efficacy Trade-off of Anomaly Detection Techniques for Real-Time Streaming Data".

Data Used in Anomaly Detection

The application of ML algorithms can broadly be classified across six data modalities:

- **Text**
- **Images**
- **Audio**
- **Video**
- **Numerical**
  - corresponding to operational metrics such as revenue, number of users, latency, error rate
- **Categorical**
  - such as online survey data

Data can be analyzed as a single modality – for example, looking at text-based social media posts – or in a multi-modal fashion. In the latter case, instead of analyzing each modality in isolation, we learn across multiple modalities (via Common Representation Learning [5]) - making our anomaly detection system more robust. An example of multi-modal analysis is out-of-sync audio and video in a livestream, which could potentially suggest that the livestream is doctored (For additional examples wherein cross-modal analysis is beneficial, take a look at [6]).

In many cases, one can model these data types as a time series where there is a data point for each individual unit of time. Even video can be thought of as a series of individual frames that happen sequentially. But not everything has to be a time series. When looking at images you might have a million images in random sequence.

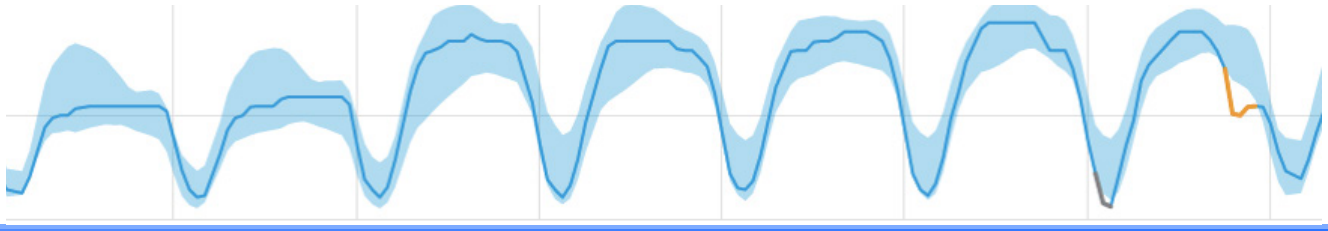
In the case of textual data, you can perform sentiment analysis, which is the process of algorithmically identifying the opinions expressed in a piece of text. This is useful when determining whether the writer's attitude towards a particular topic, product, etc. is positive, negative or neutral. Sentiment analysis is used in customer feedback, social media, political discourse, product reviews and more.

For example, medical researchers might use a text file to find aberrations in patient writing samples that may signify degradation of memory. In financial trading, the sentiment around a particular stock signals ML models to make buy/sell recommendations.

Anomaly detection in audio has a wide variety of practical applications, for instance, but not limited to, business monitoring, forensics, condition monitoring/machinery maintenance (e.g., finding faulty equipment), healthcare (e.g., using audio signal processing of the heart), audio surveillance, animal husbandry and product inspection [7, 8].

Detecting anomalies in images has important applications in medicine [9, 10, 11]. For example, finding tumors by leveraging deep learning for scanimage analysis.

Anomaly detection in video also has a wide spectrum of applications, including security, surveillance, health monitoring, autonomous driving and event detection. Unusual events of interest in long video sequences, e.g., surveillance footage, often have an extremely low probability of occurring. As such, manually detecting these rare anomalies is a very meticulous (and tedious) task that often requires more employee time than is generally available, thereby necessitating an algorithmic approach to anomaly detection.



The States of Data Feeds

Once you have your data, you need to establish how your anomaly detection system will ingest it. Depending on how fast you need results or how much data goes into the detection of an anomaly, anomaly detection systems can look very different.

To establish data ingestion, we'll consider three different states of the data: at rest, streaming and live. The state of the data has an impact on the speed and accuracy of the anomaly detection algorithm(s).

Data at rest is on a storage medium (e.g., a database or a data warehouse) and hence, the data processing speed does not have any bearing on data loss. The data might pass through several algorithms that progressively refine the search for patterns that are deemed to be anomalies.

On the other hand, streaming data is typically characterized by the continuous and unbounded nature of inflowing data (for example, YouTube, Spotify or Apple Music), where successive record chunks are sent simultaneously and in small sizes (order of Kilobytes). As the volume of data is huge over a considerable period of time, it is processed on a per-record basis over a sliding window [12].

Lastly, there is live data, which can flow in from social media, such as a Twitter firehose, or a live event, such as a concert or a sporting event. Example applications to this end include Facebook Live, Twitch and Periscope. The continuous data inflow calls for the design of a fast anomaly detection algorithm, which in turn requires an intrinsic trade-off between speed and accuracy. Detecting anomalies in live video feeds is particularly challenging owing to high compute requirements and stringent latency bounds.

More Data Characteristics to Consider

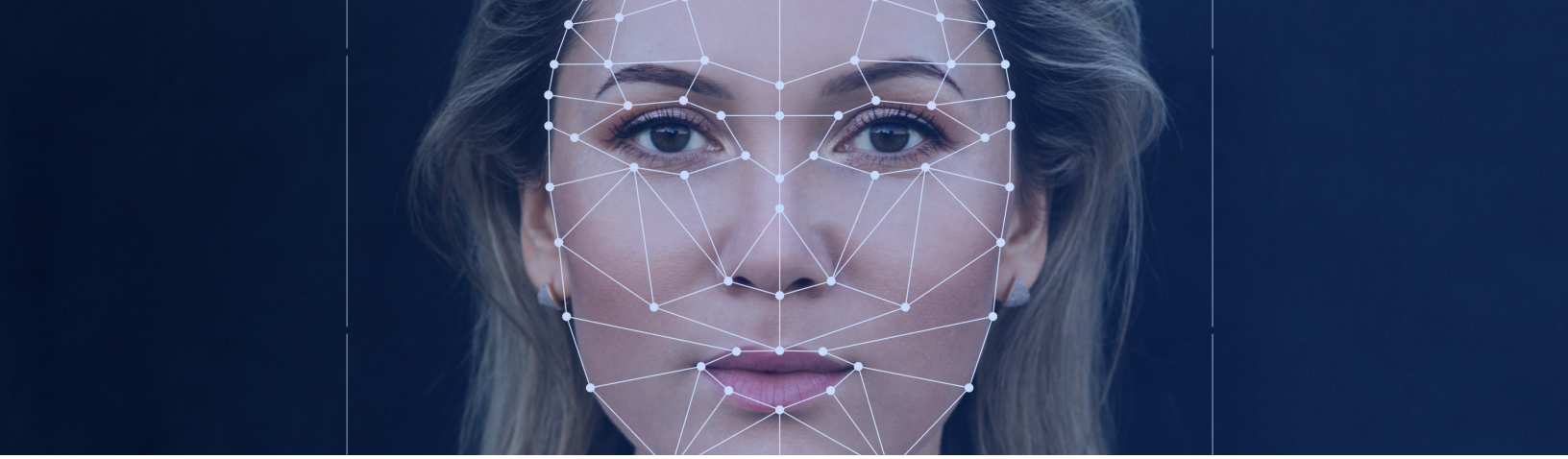
There are other data characteristics besides modalities to consider in anomaly detection. Knowing the temporality of your data — whether the data is part of a time series or an individual point — can radically adjust your anomaly detection architecture.

Data that collectively represents how something changes over time is considered part of a time series. Each data point includes a measurement of some sort taken at specific times. Some examples of time series data include steps in a manufacturing process, hourly sales of a retail company, daily atmospheric temperatures, stock market values, ocean tides and all audio and video samples.

Critically, the data must be processed in sequential order, which can be difficult with connection lags causing information to arrive at processing centers out-of-order. Ordered chronology is especially useful for predictive analytics, which is when anomaly detection helps forecast future conditions.

In most cases the measurements are taken at regular time intervals, but that is not guaranteed. Anomaly detection in non-equidistant time series poses a significant challenge to many off-the-shelf anomaly detection algorithms such as the Kalman filter [13], GARCH [14] and SDE [15] - which assume a regularly sampled time series [16, 17, 18, 19]).

If time is not consequential to the data, the measurements are simply individual data points that can be processed in any order; for example, images used in facial recognition, or text-based customer product reviews.



Quantifying Non-Linear Data

When data isn't organized sequentially, anomaly detection algorithms use feature vectors to understand trends. A feature is an individual, measurable property or characteristic in a dataset. A set of numeric features can be conveniently described by a feature vector. In pattern recognition and ML, a feature vector is an n-dimensional vector of numerical features that represents some object. If this sounds abstract, don't worry, the examples help:

- **In-text recognition:** features may include histograms counting the number of black pixels along horizontal and vertical directions, number of internal holes, stroke detection and many others [20].

- **Speech recognition:** features for recognizing phonemes include noise ratios, length of sounds, relative power, filter matches and many others [21, 22].

- **Spam detection:** features include the presence or absence of certain email headers, the email structure, the language, the frequency of specific terms and the grammatical correctness of the text [23].

- **Image analysis:** features include the number of pixels, the RGB values and the coordinates for where the photo was taken [24, 25, 26].

You may have heard of "deep fake" videos. They are videos that have been digitally modified to appear like something they are not. Feature vectors can be useful in detecting that a video is a deep fake because there will be anomalies in the feature values [27, 28].

Handling Data Quantity

Detecting anomalies and responding quickly is critical, but more data isn't always the solution. Teams talk in terms of monitoring millions of time series, perhaps taking a measurement every second or millisecond. How often something is measured is called the frequency of data. The fact that something can be measured that often doesn't mean it should be.

Teams often don't use much of the data they collect; as much as 95% of collected data is never read. We recommend a top-down approach to deciding what data to collect; that is, collect just what is needed for specific use cases and business objectives.

Finding the right frequency is important. When data frequency is very high, you have more data but it will be more "noisy," and as your signal to noise ratio (SNR) goes down, it becomes very hard to detect anomalies.

Tests such as the *white noise* test and *red noise* test can be used to characterize background noise in times series [29]. Low pass filters can be used to smooth out noise before applying off-the-shelf anomaly detection techniques.

Consider weather data. We can get daily, hourly or even by-the-minute temperature readings. If we want to gauge long-term climate change, for example, if April 2020 was uncharacteristically warm, we need multiple years of weather data to establish a baseline. But we don't need to have temperatures at minute intervals; daily or even monthly will be sufficient if we have enough years of data in the sample.

Data frequency is especially important with video, especially high-resolution footage such as 8K video. 8K resolution refers to an image or display resolution with a width of approximately 8000 pixels, which is extremely high. Ingesting and analyzing 8K video requires large compute and networking resources. It also and adversely impacts latency (which adversely affects user experience). So, either you would have to lower the resolution or trade-off the accuracy of the anomaly detection.

Context Awareness

Context is very important in defining anomalies. It helps ensure that the anomalies revealed are actionable. A pattern in one domain, if applied to another, may potentially result in false positives or false negatives. Likewise, an anomaly detection algorithm that works for one domain might not work for another.

Thus, context-awareness makes it challenging to find a singular anomaly detection algorithm to fit every scenario; there is no “one size fits all”! This is also the reason why many off-the-shelf anomaly detection algorithms or solutions fail.

Centralized vs. Decentralized Anomaly Detection

The two key areas that drive discussions about centralized versus decentralized anomaly detection: data privacy and the Internet of Things (IoT).

As for data privacy, an increasing array of government regulations dictate where various types of sensitive information (e.g. medical data) can be stored and processed. Centralizing data storage may help with data protection and regulatory compliance.

In contrast, in the context of IoT, the large number of sensors typically used for collecting data pose a challenge to a centralized approach for anomaly detection.

Consider the case of a large manufacturing plant that has thousands of sensors. Piping the data to a centralized server for anomaly detection anomaly detection and correlation analysis would be prohibitively expensive from a communication standpoint. Additionally, the sensors typically tend to be of very low power. But when they must communicate with a server, they consume a lot of power.

The question becomes, how can we do anomaly detection on a localized dataset? Local processing won’t be as robust as doing it in a centralized fashion. However, instead of sending all the data – which can be tens of MBs – to a central server, you would send only the set anomalies. Concretely, instead of sending a day’s worth of minutely data, one would send `<timestamp, value>` corresponding to the perceived anomalies. Thus, the amount of data to be communicated over the network would be much lower, which helps to preserve the sensor’s battery.

In this example with hundreds or thousands of sensors, all sending small amounts of data, you may still end up with too much data to do a correlation analysis. You might notice that, out of 100 sensors, 75 report the same timestamp with the same anomalous value. In such a scenario, you can then say with confidence that there was something wrong, or at least there is something there worth looking into. Maybe there’s a water quality issue, or the air pollution index has gone up, because in three months of observing hundreds of sensors, we got a high-strength signal that something isn’t right.

When well trained, decentralizing data processing makes it more scalable when you carry out anomaly detection locally and then carry out correlation analysis in a centralized fashion (the associated overhead can be minimized by employing a hierarchical approach) - and it helps you surface the insights that matter.

Carrying out anomaly detection in a distributed fashion is an up-and-coming field that certainly isn't limited to IoT. Hopefully, in the near future, the device hardware will become more powerful, where you can do anomaly detection on the device itself and then pipe the results to the server.

Summary

In this introductory article on automatic anomaly detection, we looked at the common types of anomalies, the typical types of data used for anomaly detection, what frequency of data means, the importance of context awareness, and the issues pertaining to centralized vs. decentralized anomaly detection.

To learn more about the essential components and design considerations of an anomaly detection system for business use, we recommend that you continue on to the 3-part series "The Ultimate Guide to Anomaly Detection".

Citations

[1] "Measure What Matters: How Google, Bono, and the Gates Foundation Rock the World with OKRs"
[2] "Metrics That Matter"
[3] "Always Measure One Level Deeper"
[4] "On the Runtime-Efficacy Trade-off of Anomaly Detection Techniques for Real-Time Streaming Data"
[5] "CM-GANs: Cross-modal Generative Adversarial Networks for Common Representation Learning"
[6] "See, Hear, and Read: Deep Aligned Representations"
[7] "Unsupervised Detection of Anomalous Sound Based on Deep Learning and the Neyman-Pearson Lemma"
[8] "How Can We Detect Anomalies From Subsampled Audio Signals?"
[9] Image Anomaly Detection With Capsule Networks and Imbalanced Datasets"
[10] "Robust Anomaly Detection in Images using Adversarial Autoencoders"
[11] "Unsupervised Anomaly Detection for X-Ray Images"
[12] "Real-Time Streaming and Anomaly Detection Pipeline on AWS"
[13] "A New Approach to Linear Filtering and Prediction Problems"
[14] "GARCH modelling in continuous time for irregularly spaced time series data"
[15] "Estimation of a Time series Model From Unequally Spaced Data"
[16] "A Framework for the Analysis of Unevenly Spaced Time Series Data"
[17] "A Note on Trend and Seasonality Estimation for Unevenly Spaced Time Series"
[18] "Studies in Astronomical Time Series Analysis. VI. Bayesian Block Representations"
[19] "Discrete-Time Autoregressive Model for Unequally Spaced Time-series observations"
[20] "Review of Feature Extraction Techniques for Character Recognition"
[21] "Speech Signal Processing and Feature Extraction"
[22] "Some Commonly Used Speech Feature Extraction Algorithms"
[23] "Email Spam Filtering: A Systematic Review"
[24] "Unmasking DeepFakes with Simple Features"
[25] "FaceForensics++: Learning to Detect Manipulated Facial Images"
[26] "Learning to Detect Fake Face Images in the Wild"
[27] "In Ictu Oculi: Exposing AI Generated Fake Face Videos by Detecting Eye Blinking"
[28] "FakeCatcher: Detection of Synthetic Portrait Videos Using Biological Signals"
[29] "Detecting and Classifying Events in Noisy Time Series"

www.anodot.com | EMEA +972-54-522-4085 | USA +1-408-306-1612 | info@anodot.com