Guide to Real-time Anomaly Detection for Enterprise Data

Introduction

Detecting anomalous patterns in data can lead to significant actionable insights in a wide variety of application domains. Imagine being able to:

- Detect roaming abuse, revenue fraud, and service disruptions in the telecom industry
- Identify changes in employee behavior that signal a security breach
- Flag abnormally high purchases/deposits and detect cyber intrusions in banks
- Detect and prevent out-of-pattern medical spends and payments in incoming health insurance claims
- Detect compromised social media accounts and bots that generate fake reviews
- Detect intrusion into networks and prevent theft of source code or IP

These are only a few applications of anomaly detection; there are innumerable possibilities. However, detecting anomalies accurately can be difficult. The definition of an anomaly is continuously changing as systems evolve, behaviors change, and software gets updated over time. Furthermore, because anomalies by their nature are unexpected, an efficient detection system must be able to determine whether new events are anomalous without relying on pre programmed thresholds. Therefore, to detect an anomaly effectively, the system needs to be upgraded continuously.

In this paper, we will explore:

- What anomaly detection is and why it is important in enterprise data
- How to build an anomaly detection model
- A view of the platform approach to anomaly detection

What is an anomaly?

An anomaly is an observation that significantly deviates from most of the other observations, i.e., a data point/ behavior/pattern that appears to be statistically unusual or anomalous.

Anomaly detection is the identification of items, observations, or events that do not conform to a predictable pattern or other items in the dataset.

Importance of anomaly detection in real-time streaming data

Anomaly detection is influencing business decisions across verticals. Examples of how leading enterprises are using anomaly detection in real-time to gain deeper insights and respond more quickly to changing conditions and opportunities are as follows:

Financial services

Anomaly detection can flag abnormally big purchases/deposits and detect cyber intrusions. The signs or signals of risky and fraudulent transactions can be detected by in predictive intelligence models to alert firms to suspicious activities.

Manufacturing

Enterprises can detect abnormal machine behavior and communicate wear-and-tear before they become machine failures that interrupt or cripple operations, thereby preventing cost overruns. This can also help enhance preventive maintenance of equipment in areas such as transportation and energy generation.

Healthcare

In urgent healthcare situations, patient monitoring systems generate real-time responses from their monitoring of a large number of concurrent parameters to alert professionals to urgent issues that demand immediate attention or intervention.

Brand reputation

Given the wide adoption and influence of social media, brands are being discussed anywhere at any time. Anomaly detection can alert brand managers of the activity, and give them time to contain, control, or contribute to the brand conversation.

Anomaly detection techniques

There are three broad categories of anomaly detection techniques:

1. **Supervised** anomaly detection techniques require a dataset that has been labeled as "normal" and "abnormal" and involves training a classifier. This classifier can then be used to identify anomalies in new data using the patterns learned during training

2. **Unsupervised** anomaly detection techniques detect anomalies in an unlabeled test data set. This method assumes that majority instances in the dataset are normal, and looks for instances that least fit the normal dataset.

3. **Semi-supervised** anomaly detection techniques construct a model representing normal behavior from a given normal training dataset, and then look for anomalies based on the constructed model.

Traditionally, anomaly detection methods have been classified based on the characteristics of the data being analyzed:

Categorical and numeric attributes

- K-modes: Uses Hamming distance to measure distance for categorical features
- Generic mixture models: Extends the framework of Gaussian mixture models
- Robust SVM: A Kernel-based approach that identifies regions in which data resides in alternate feature space

Sequential data

- State space models: Model the evolution of data in time to enable forecasting and flag an anomaly if it exceeds a threshold
- Hidden Markov models (HMMs): Markov Chains and HMMs measure the probability of different events happening in some sequence
- Graph-based methods: Graphs capture interdependencies and allow the discovery of relational associations such as in fraud

More recent methods of anomaly detection include

- Deep learning (Auto Encoder): Auto Encoders can learn the latent representation of the data by using an encoder and a decoder together
- Deep learning (RNN-based): Recurrent Neural Network (RNN)-based architectures enable sequence prediction. The network can flag an anomaly when needed
- Generative Adversarial Nets (GANs): GANs combine two neural networks - a generator and a discriminator – and can be used to find anomalies

How to build an anomaly detection model

Building an anomaly detection model involves the following key steps:

- **Identifying the problem and setting expectations:** One of the key aspects to consider while setting expectations is to minimize false positives. Any system that tries to detect some anomalies might flag some normal data as an anomaly, which needs to be considered while identifying the problem.
- **Defining the sources and schema:** Multiple data sources can come into play in a real-world setting while trying to establish an anomaly detection process. The schema and sources of these datasets need to be considered.
- **Parsing and pre-processing:** Before detecting an anomaly, you need to ensure the quality of the data. Data of poor quality can contribute to a skewed result. Therefore, it is important to analyze and pre-process the data before developing the anomaly detection model.
- **Model development:** Model development, which is the key step in anomaly detection depends on three components:

a. Type of anomaly detection

1. Point anomaly: If an individual data instance can be considered as anomalous with respect to the rest of the data (e.g., purchase with large transaction value)
2. Contextual anomaly: If a data instance is anomalous in a specific context, but not otherwise (an anomaly, if occurs at a certain time or in a certain region, for example, a large spike in the middle of the night)
3. Collective anomaly: If a collection of related data instances is anomalous with the entire set of data, but not individual values, like:
   a. Events in unexpected order (e.g., breaking rhythm in ECG)
   b. Unexpected value combinations (e.g., buying many expensive items)

b. Type of data available

1. Categorical – Nominal (named categories), ordinal (categories with an implied order), binary (variables with only two options)
2. Numerical – Discrete (only particular numbers), continuous (any numerical value)

c. If the data has labels

1. Model execution: Once the anomaly detection model is developed, it needs to be executed to ensure that the right anomalies are being flagged.
2. Investigation and feedback
3. Model updating: The model needs to be updated at regular intervals to ensure new data is being considered.
4. Operationalize model for scoring

Figure 1: Choice of Algorithm

The following figure depicts a typical flow of algorithm application for various anomaly detection use cases

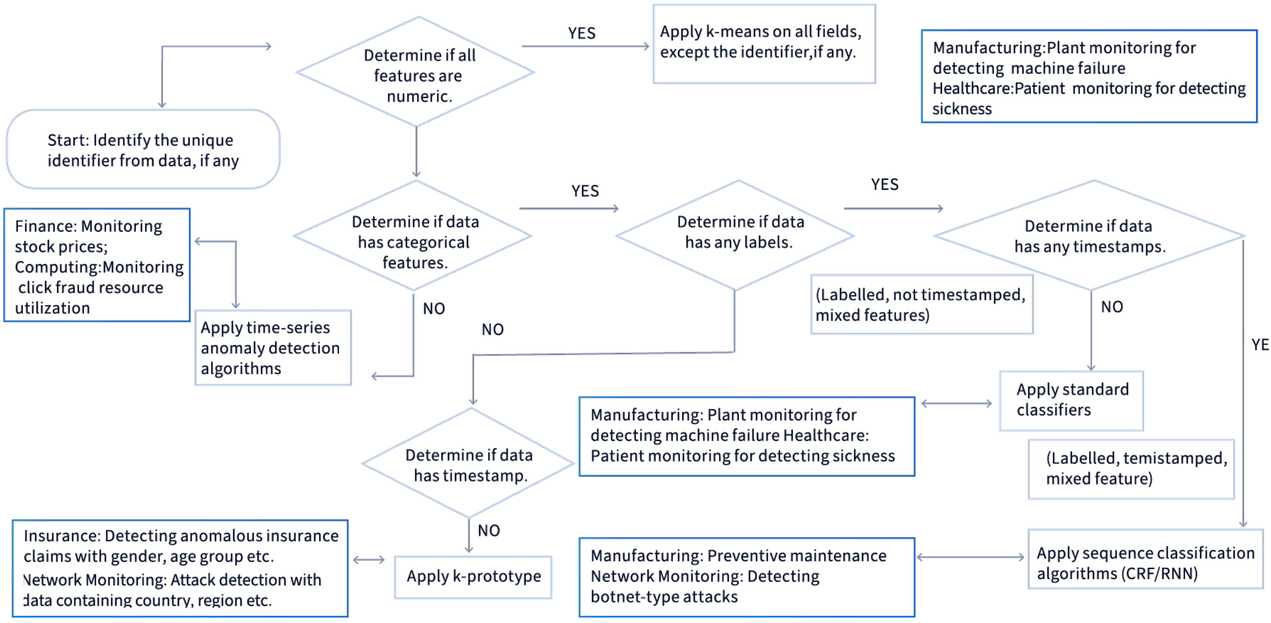
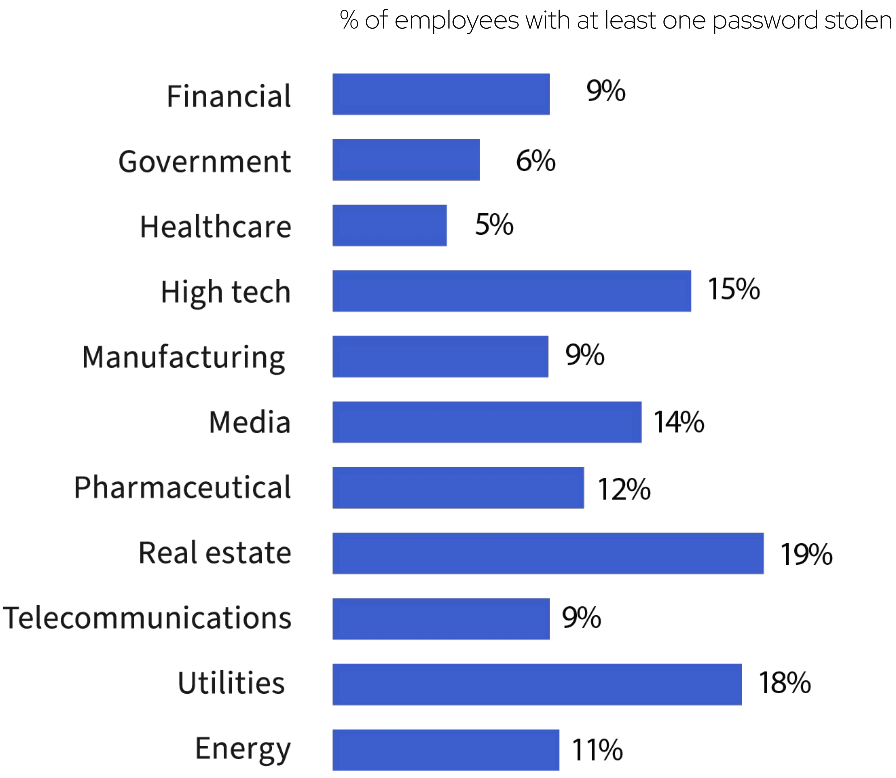

Start: Identify the unique identifier from data, if any
Determine if all features are numeric.
YES
Apply k-means on all fields, except the identifier,if any.
Manufacturing:Plant monitoring for detecting machine failure
Healthcare:Patient monitoring for detecting sickness
Determine if data has categorical features.
YES
Determine if data has any labels.
YES
Determine if data has any timestamps.
Finance: Monitoring stock prices;
Computing:Monitoring click fraud resource utilization
Apply time-series anomaly detection algorithms
NO
NO
(Labelled, not timestamped, mixed features)
NO
YES
Apply standard classifiers
Manufacturing: Plant monitoring for detecting machine failure Healthcare: Patient monitoring for detecting sickness
Determine if data has timestamp.
(Labelled, temistamped, mixed feature)
NO
Insurance: Detecting anomalous insurance claims with gender, age group etc.
Network Monitoring: Attack detection with data containing country, region etc.
Apply k-prototype
Manufacturing: Preventive maintenance Network Monitoring: Detecting botnet-type attacks
Apply sequence classification algorithms (CRF/RNN)

Figure 2: Industries most exposed to compromised accounts



(Source: https://www.skyhighnetworks.com/cloud-security-blog/poll-9-top-questions-cios-are-asking-about-cloud-threat-detection/)

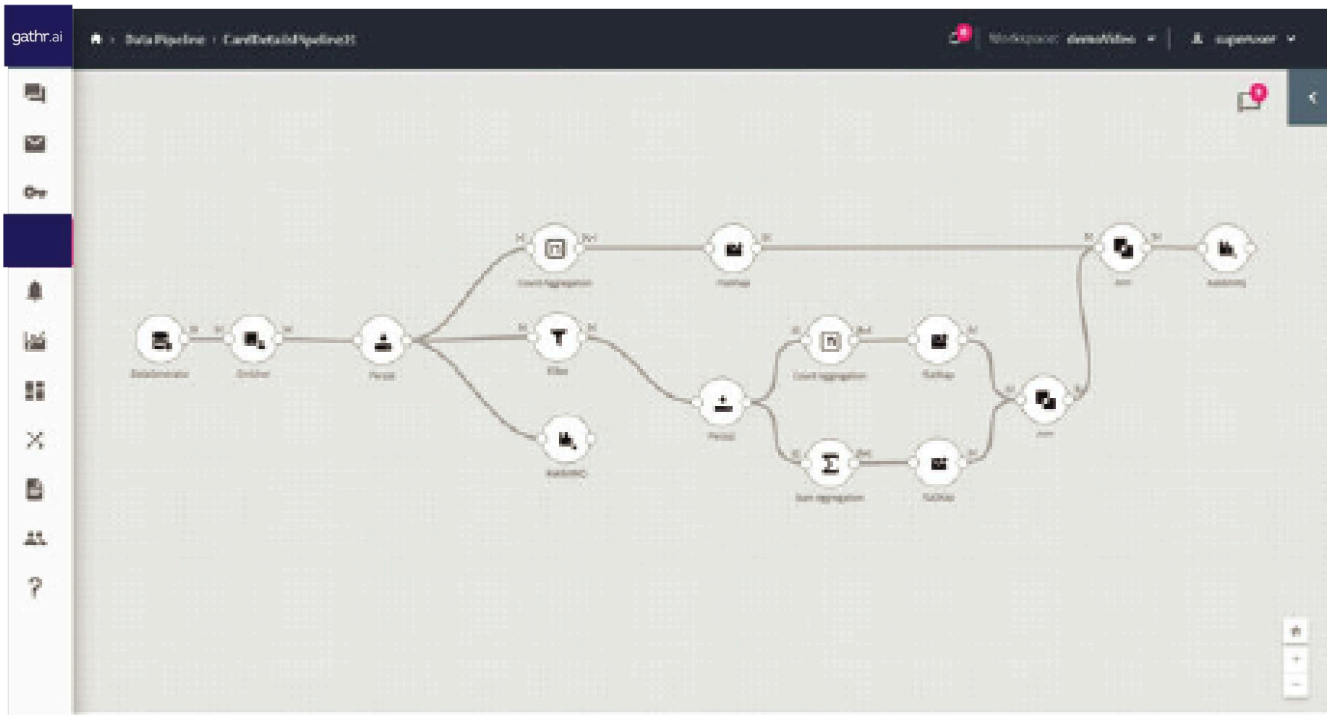
A platform approach to anomaly detection

One way to implement anomaly detection is to hand-code everything from scratch. But, developing a custom solution from scratch in this manner can have its own set of challenges:

- A very long implementation cycle
- Finding people with the right skill set
- Multiple QA cycles
- Once developed you need to monitor continuously, and scale with increasing loads

Why Gathr for anomaly detection

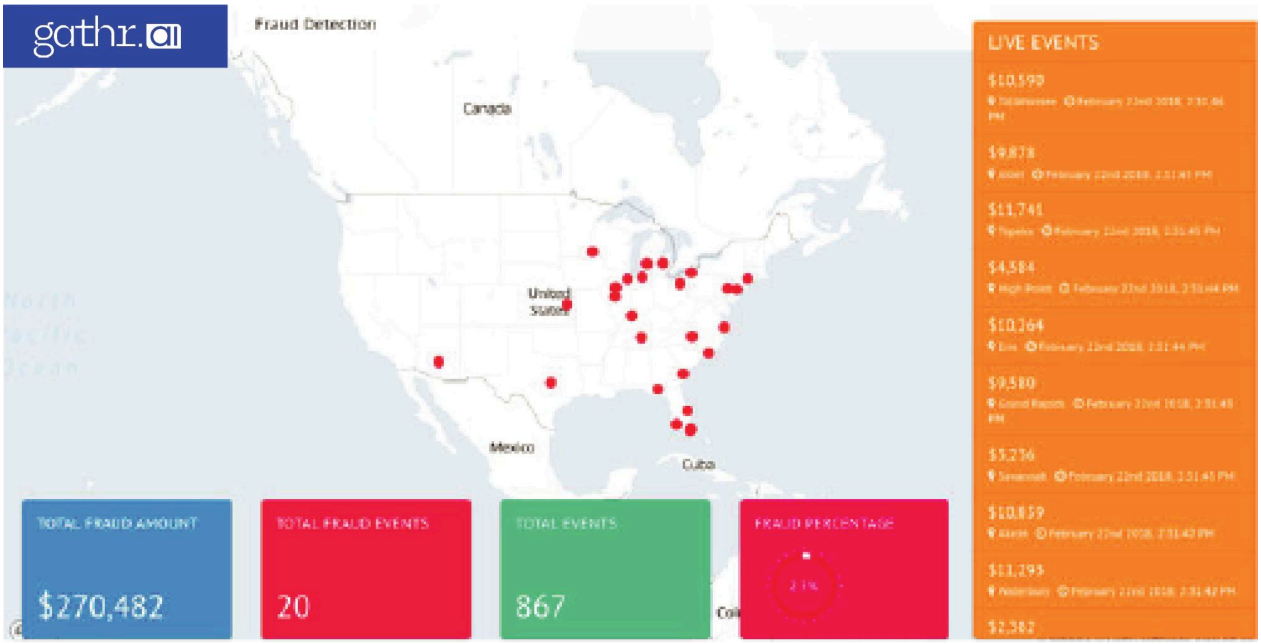
Imagine a platform that would let you achieve all these steps on one single interface. Gathr is a specialized platform to rapidly build and run big data analytics and machine learning applications. It leverages open source engines like Apache Spark to create analytics applications at big data scale and has an intuitive drag-and-drop interface to visually build and manage your application workflows.



These applications can be used to get real-time insights and can also be integrated with several third-party apps to provide business intelligence

For example, the image below depicts how a large US bank predicts and curbs credit card fraud by analyzing transaction data in real-time using Gathr.ai

- Identify fraudulent transactions in real-time with cluster models, based on historical data
- Update classification of fraudulent transactions in real-time
- Classify transactions by value and volume (low to high)
- Create custom alerts for various retail merchants and zip codes
- Provide summary report for any interval
- Enable persistence of identified patterns for future analysis and o line reporting
- Implement new models within a week



TOTAL FRAUD AMOUNT $270,482

TOTAL FRAUD EVENTS 20

TOTAL EVENTS 867

FRAUD PERCENTAGE 2.3%

LIVE EVENTS

$10,590
Tallahassee February 22nd 2018, 2:30:46 PM

$9,878
Joliet February 22nd 2018, 2:31:41 PM

$11,741
Topeka February 22nd 2018, 2:31:45 PM

$4,584
High Point February 22nd 2018, 2:31:44 PM

$10,264
Erie February 22nd 2018, 2:31:44 PM

$9,580
Grand Rapids February 22nd 2018, 2:31:48 PM

$5,236
Savannah February 22nd 2018, 2:31:45 PM

$10,859
Akron February 22nd 2018, 2:31:42 PM

$11,295
Waterbury February 22nd 2018, 2:31:42 PM

$2,382

Beyond anomaly detection

Gathr.ai is an integrated framework not just to create models but also provides an endto -end feature rich platform to build enterprise anomaly detection applications. It perfectly maps to the modern platform approach to anomaly detection by exposing features like:

- Multi-tenancy
- Read and process real-time data
- Rapid development and operationalizing applications
- A/B testing
- Monitor, debug, and diagnose at scale
- Version management
- Promoting workflows to di erent environments: Dev-Test-Prod

Conclusion

with Gathr.ai, we have taken a fresh approach to this problem and created what we believe is a world-class, real-time, anomaly detection platform. Our customers use Spark-based applications to detect and act on anomalies in streaming data faster and with greater accuracy than ever before. When you can see and act on any deviation as it happens, you realize the potential of streaming data to change how business is done. Anomalies provide us with the cues we need to help prevent fraud, predict mechanical failures, provide predictive healthcare, present the right offer at the right time, and make decisions in real-time with confidence.

Impetus' Gathr is an open-source enabled, enterprise-grade, multi-engine stream processing and machine learning platform. It's also the only platform of its kind that includes a visual integrated development environment (IDE) that makes it easy for developers to build, deploy, and manage Apache Spark applications in a matter of minutes. Apache Spark has emerged as the de facto choice for stream processing, real-time analytics, data science and machine learning applications.