How to Use Anomaly Detection to Implement AI-Driven Use Cases

Published 23 June 2023 - ID G00787897 - 42 min read

By Analyst(s): Maryam Hassanlou

Initiatives: Analytics and Artificial Intelligence for Technical Professionals; Drive Quantifiable Value With D&A Solutions for the Business

Choosing the right anomaly detection techniques to implement AI use cases such as fraud detection or predictive maintenance is challenging. Data and analytics technical professionals should use this framework to evaluate use-case opportunities and choose an appropriate anomaly detection technique.

Overview

Key Findings

- Anomaly detection has significant practical applications in various industries, such as fraud detection in finance, fault detection for predictive maintenance in manufacturing, identifying network intrusions, detecting instrument failures and identifying tumor cells in healthcare.
- Different techniques work better for different types of data and use cases. Labeled data can be expensive and difficult to obtain for anomaly detection. This often makes unsupervised ML techniques a practical, effective option. Practically, ensemble methods can be effective in reducing false positives and increasing true positives.
- For some use cases, sophisticated techniques enhance anomaly detection beyond simple approaches. Deep learning techniques have shown promise in achieving high accuracy in anomaly detection. However, these methods tend to be more complex and require additional resources for both training and deployment.
- Interpretability can be a challenge with black-box ML techniques. Some ML techniques, such as deep learning models, can be difficult to interpret, which can make explaining the reasoning behind anomaly detection results to stakeholders and decision makers a challenge.
- Collaboration between data scientists and domain experts is crucial for effective anomaly detection. Their combined expertise helps select and interpret ML techniques tailored to the use case requirements and constraints.

Recommendations

Data and analytics technical professionals tasked with implementing anomaly detection should:

- Consider factors such as data type, quality and volume; performance expectations; and the complexity of the anomalies you are looking for when choosing an anomaly detection technique. For example, in the cybersecurity industry, techniques such as rule-based anomaly detection may be more appropriate for detecting known threats, whereas unsupervised techniques may be suitable for identifying unknown threats.
- Combine anomaly detection techniques (ensemble methods) to increase the accuracy and reliability of your results.

- Evaluate the potential impact of false positives and false negatives on your business operations. False positives can be costly in terms of time and resources, while false negatives can result in missed opportunities or potential risks.
- Involve domain experts actively in the process of anomaly detection. They can help interpret the results and identify the root cause of anomalies, which can lead to more effective and actionable insights.

Analysis

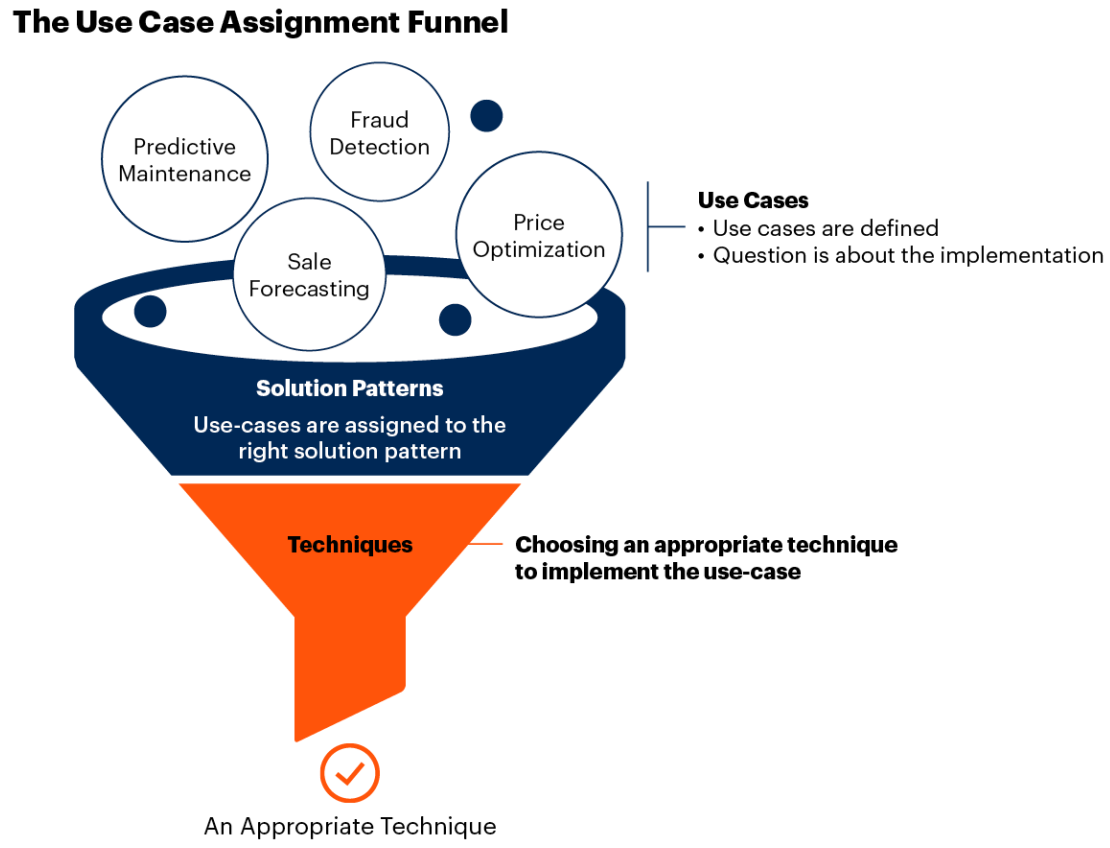
Many organizations are looking to implement AI technologies to deliver business impacts, but they are not sure which use cases to prioritize or how to implement them. The first step is building a portfolio of impactful, measurable and quickly solvable use cases.

This research assumes use cases are identified and prioritized based on the business preferences, feasibility and business impacts.

AI-driven use cases are defined for different industries and practices based on the nature of the business, values and competencies they are looking for, and market trends. Data and analytics technical professionals are confronted with a variety of use cases that need to be implemented. But in reality, they can be classified under a limited number of high-level categories that we call “solution patterns.”

Each solution pattern encompasses a category of capabilities and analytics to implement a portfolio of use cases that are similar methodologically or technically. Figure 1 shows use cases entering the funnel and then being assigned to an appropriate solution pattern.

Figure 1: The Use Case Assignment Funnel



Source: Gartner
787897_C

Gartner

Each solution pattern covers a category of tools and techniques that are different, but all provide the same service with different qualities and approaches in different circumstances. Some of the most common patterns identified by Gartner are optimization, anomaly detection, forecasting/prediction, segmentation, decision making, recommender systems, automation, autonomous systems, virtual assistants and content generation/moderation.

After assigning a use case to a particular solution pattern, the next step of the funnel is choosing the right technique to implement the use case. There are many factors that affect the technique selection that requires a structured framework to follow.

The scope of this document is the last step of the funnel. We focus specifically on anomaly detection and provide a practical framework to help data scientists choose an appropriate anomaly detection technique to implement an identified AI-driven use case.

Anomaly detection plays a crucial role in various industries. It enables the timely identification of abnormal or suspicious patterns within data. In finance, it detects fraudulent activities, safeguarding businesses from financial losses. In manufacturing, it identifies faults in machinery, enabling timely maintenance and preventing disruptions and increased costs. In healthcare, it detects abnormal patient conditions and disease outbreaks, facilitating early diagnosis and improving patient outcomes. Without effective anomaly detection, businesses risk financial losses, operational disruptions, compromised safety, decreased productivity and compromised customer care.

Anomaly detection is essential for detecting and mitigating risks, preventing fraudulent activities, ensuring operational efficiency, and maintaining the overall integrity of systems and processes. By effectively detecting anomalies, businesses can proactively address potential issues, minimize financial losses, and enhance the trust and satisfaction of their stakeholders.

Choosing the right anomaly detection technique is critical for achieving optimal performance and accuracy in anomaly detection tasks. Different industries have unique characteristics, data types and requirements, making it necessary to select a technique that aligns with their specific needs.

An inappropriate choice of technique may result in missed anomalies, false positives or inefficient resource utilization. The development of a framework is imperative to ensure the selection of an appropriate anomaly detection technique.

A framework provides a structured approach to consider the specific use case requirements, data features, performance needs and algorithmic considerations. By following a systematic process, businesses can evaluate and compare different techniques, assess their strengths and limitations, and ultimately choose the most suitable technique for their industry and use case.

Anomaly Detection

Anomaly detection is the process of identifying rare or unusual events, patterns or observations in data that deviate significantly from the expected or normal behavior. Anomaly detection uses the patterns discovered by statistical, probabilistic analysis, automated pattern discovery and prediction to determine a “normal” behavior and then continuously discerns departures from that normal behavior, both univariate and multivariate.

Transcending the mere detection of outliers, these discovered anomalies must then be correlated with business impact and other concurrent processes and activities, such as continuous delivery, to be fully useful. Thus, anomaly detection can greatly simplify root cause analysis and discover otherwise unnoticed issues that should be further investigated or remediated. Table 1 represents the introductory profile of the pattern anomaly detection.

Table 1: Solution Pattern Profile for Anomaly Detection

(Enlarged table in Appendix)

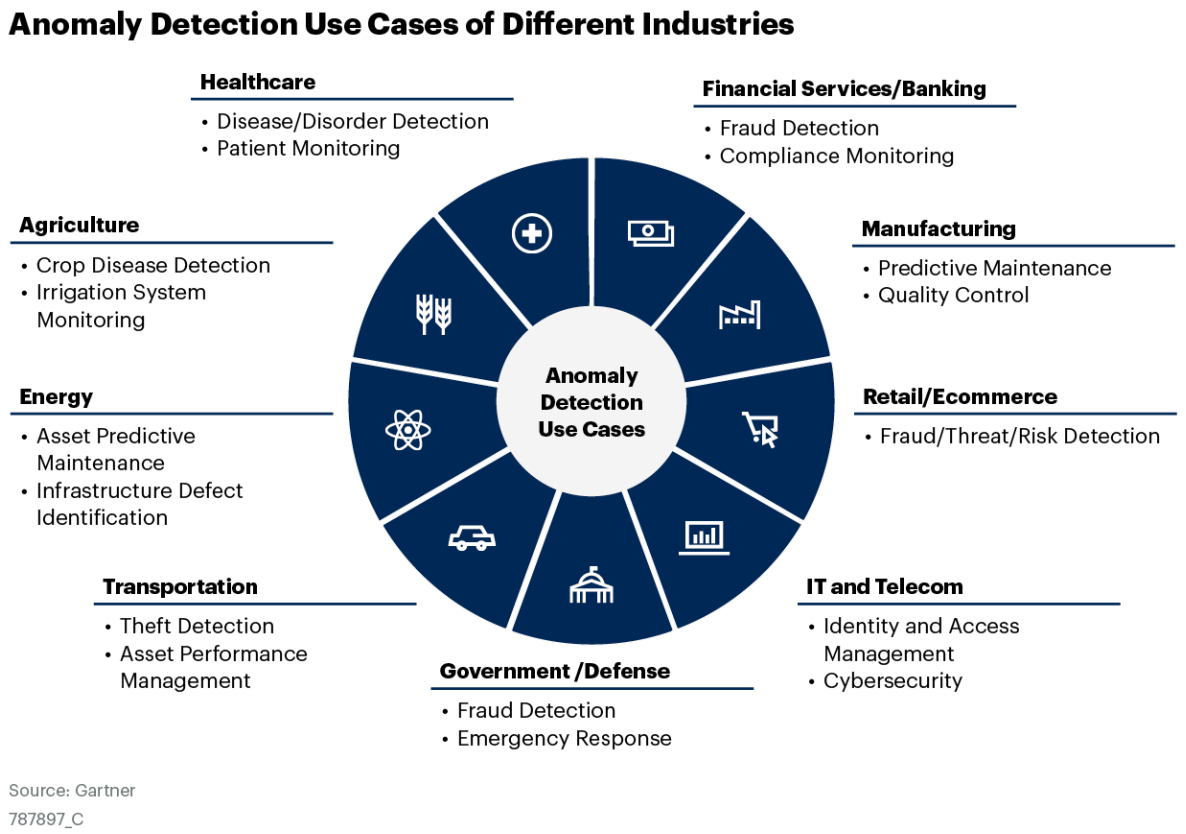
Pattern No. 1	Anomaly Detection
Description	A process used to identify data points that deviate significantly from the norm or expected behavior. It involves analyzing data to find unusual patterns or outliers that may indicate a potential problem or opportunity. It is a powerful tool for identifying and addressing potential issues in a proactive manner, leading to better decision making and improved outcomes in various industries.
Example Applied Areas	■ Fraud detection in finance ■ Fault/defect detection in manufacturing ■ Attack detection in cybersecurity ■ Disease outbreak detection and patient monitoring in healthcare
Typical techniques	■ ML ■ Expert system ■ Information theory
Data dependency	Historical time series data
Maturity	Medium to high
Challenge	■ False positive/negative detection ■ Lack of labeled data ■ Imbalanced data ■ Nonstationary data ■ Interpretability
Trends	■ DL ■ Graph ■ Causal AI ■ Ensemble techniques

Source: Gartner (June 2023)

Anomaly Detection Use Cases

Anomaly detection has significant practical applications in various industries, particularly in the realm of industrial processes. While credit card fraud detection is the most well-known example of such applications, anomaly detection is a crucial tool in many other scenarios, including fault detection for predictive maintenance in manufacturing, identifying network intrusions, detecting instrument failures, identifying tumor cells in healthcare and so on. Figure 2 represents some examples of AI use cases of anomaly detection in different industries. In all these cases, the ability to detect anomalies can be vital for ensuring smooth business operations, reducing risk and improving outcomes.

Figure 2: Anomaly Detection Use Cases of Different Industries



The core of a successful AI strategy includes a solid understanding of the different use cases and where AI can help your business best. But gathering, defining and prioritizing these use cases can be a daunting task due to the complexity of the work and the different views each stakeholder may have. A simple, easy-to-iterate approach can cope with these challenges.

In Table 2, Gartner identifies AI use cases — all implemented using anomaly detection techniques — for 15 industries and business practices.

Table 2: Anomaly Detection Use Cases

(Enlarged table in Appendix)

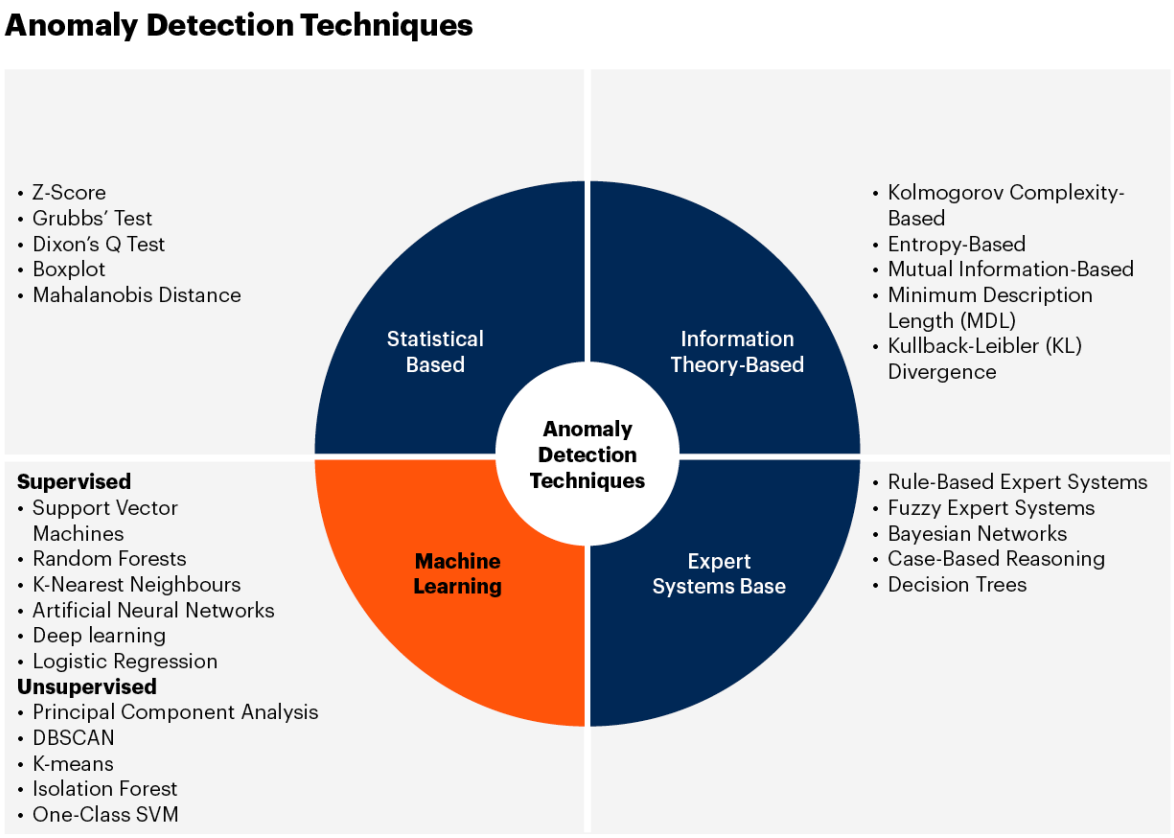
	Industry	Use Case
1	Banking	Customer transaction fraud detection
2	Finance	Anomaly and error detection
3	P&C and life insurance	Fraud detection
4	Retail	Risk/fraud/threat detection
5	Cybersecurity	Transaction fraud detection File-based malware detection Abnormal system behavior detection Text-based malicious intent detection
6	Digital commerce	Fraud detection
7	Defense and intelligence	Predictive equipment maintenance Early crisis detection
8	Software development and testing	Defect prediction Security/vulnerability detection
9	Healthcare provider	Digital speech analysis for clinical diagnosis
10	Supply chain	Asset performance
11	Automotive	Manufacturing quality control Machine predictive maintenance
12	Utilities	Asset predictive maintenance Infrastructure defect identification
13	AIOps	Anomaly detection
14	Transportation	Theft detection Asset performance management
15	Life science manufacturers	Safety signal detection Predictive maintenance

Source: Gartner (June 2023)

Anomaly Detection Techniques

A range of tools and techniques, from simple statistical techniques to very complex machine learning algorithms, are used to detect anomalies depending on the complexity and type of data and sophistication required. There are different classifications for anomaly detection techniques, but one of the most common classifications defines four categories of techniques based on the underlying principles and methodologies used by each (see Figure 3). This classification can help us select the appropriate technique for a particular use case and given application.

Figure 3: Anomaly Detection Techniques



Source: Gartner
787897_C

Gartner

Statistical-Based Methods

Statistical-based methods for anomaly detection are techniques that rely on statistical measures to identify anomalies in datasets. These methods assume that normal data follows a specific statistical distribution, and any data points that deviate significantly from this distribution are considered anomalies.

In these sets of approaches, a priori knowledge and assumption regarding the underlying distribution of the dataset is required, which is almost always unavailable when data evolves over time. Also, such methods may not be suitable for complex and high-dimensional data.

Univariate statistical-based methods assume that data in each variable is independent and use the Z-score method to identify anomalies. On the other hand, multivariate statistical-based methods consider the correlation and dependence among multiple variables, and the Mahalanobis distance method is commonly used to identify anomalies. Other multivariate methods include principal component analysis (PCA), linear discriminant analysis (LDA) and factor analysis.

Statistical-based methods are computationally efficient and require minimal training data, making them suitable for small to medium-sized datasets. These methods of anomaly detection are recommended for use cases where the data follows a specific distribution, such as Gaussian or Poisson distribution. They are also suitable for use cases where the expectation of normal behavior is well-defined, and deviations from this expectation can be easily quantified.

For example, using statistical-based techniques to detect anomalies in credit card transactions, financial institutions can prevent fraudulent activities and protect their customers from financial losses. Additionally, these techniques can help identify trends and patterns in credit card usage that can inform decisions regarding credit limits, rewards programs and other aspects of credit card services.

However, statistical-based methods may not be suitable for complex and high-dimensional data, where anomalies may not follow a distinct pattern or where the expectation of normal behavior is not well-defined. In such cases, more advanced techniques, such as machine learning-based methods, may be more appropriate.

Information-Theory-Based Methods

These methods assume that anomalies can be identified by measuring the amount of information that a data point or pattern contains relative to the information contained in other data points or patterns. In other words, anomalies are identified as points or patterns that have a significantly different information content than other points or patterns in the data.

These methods are suitable for use cases where the goal is to detect anomalies in large, complex datasets with high levels of noise or missing data. They are particularly effective at detecting anomalies in data streams or in datasets where the underlying distribution of the data may change over time. Overall, information theory-based methods can provide a powerful tool for detecting anomalies in complex datasets and identifying potential areas of concern in various industries.

For example, cybersecurity professionals can use these methods to analyze network traffic and detect anomalies that may indicate a potential security breach or cyber attack. By analyzing patterns in network traffic using techniques such as entropy, mutual information, or Kullback-Leibler divergence, anomalies can be detected that might not be apparent using more traditional techniques. If a particular device suddenly starts sending a large amount of data to an unknown IP address, this could indicate a potential security breach, and information-theory-based methods can help identify such anomalies.

Information-theory-based methods can be applied to a variety of data types, including numerical, categorical, and text data. They are particularly well-suited for data that can be represented as a probability distribution, such as time series data or data that can be modeled using a Gaussian distribution.

In some cases, information theory-based methods may outperform machine learning techniques, especially when the data has a high level of noise or is highly complex. However, in other cases, machine learning techniques may be more effective, especially when the data has a large number of features or complex interactions between features. Some of the most common techniques for information theory-based anomaly detection include entropy-based methods, Kullback-Leibler divergence, mutual information-based methods and Minimum Description Length (MDL) methods.

Expert System-Based Methods

Expert system-based methods for anomaly detection are based on the idea of using expert knowledge to identify anomalies in data. These methods typically involve developing a set of rules or heuristics that are used to identify anomalous behavior. The rules are developed based on expert knowledge of the domain and the behavior of the data.

Expert system-based methods assume that the knowledge of an expert in a particular domain can be used to identify anomalous behavior in that domain. The expert knowledge can be used to develop rules or heuristics that can be applied to data to identify anomalies. These methods are particularly useful in domains where there is a lot of prior knowledge and understanding of what is normal behavior. For example, in healthcare, expert knowledge can be used to develop rules to identify anomalous patient behavior that may be indicative of a particular disease or condition.

Some examples of techniques for expert system-based methods for anomaly detection include rule-based systems, fuzzy logic, decision trees and case-based reasoning.

Machine-Learning-Based Methods

Machine-learning-based methods for anomaly detection assume that anomalies can be identified by modeling the normal behavior of a system or dataset and identifying deviations from this model as anomalous. These methods use labeled or unlabeled data to train a model that can distinguish normal from anomalous behavior.

Machine-learning-based methods are suitable for a wide range of use cases, including fraud detection, intrusion detection, equipment failure prediction and anomaly detection in sensor data. They are particularly useful when there is a large volume of data and anomalies may be rare or difficult to define explicitly. Machine-learning-based methods have become increasingly popular for anomaly detection due to their flexibility, scalability and ability to handle complex and high-dimensional data.

Based on the nature of the data, the available labeled data (if any), the desired level of interpretability and the computational resources available, machine-learning-based techniques for anomaly detection can be classified as supervised, unsupervised or semisupervised.

Supervised

Supervised anomaly detection techniques are used when labeled data exists into two categories: normal and abnormal. The model is trained using data including examples of both labels, and it can learn how to detect anomalies in previously unseen data.

Using labeled data for training models in this category can be a limiting factor because labeled data is rare and/or expensive to collect. On the other hand, if supervised techniques are trained on a sufficient amount of labeled data, they can achieve a high level of accuracy in detecting anomalies. When the costs of missing an anomaly are high, and we are looking for more precise identification of anomalies, these techniques can be useful.

Some common techniques in the supervised anomaly detection category include decision trees, random forests, support vector machines (SVMs) and neural networks. Decision trees and random forests are popular due to their simplicity and ability to handle categorical data. SVMs are useful for separating data into multiple classes, while neural networks are highly flexible and can handle complex data.

Strengths:

- **High accuracy:** Supervised techniques can achieve high accuracy because they are trained on labeled data and can learn from known anomalies.
- **Flexibility:** Classification models can be applied to a variety of data types and are customizable based on the specific needs of the application. Can be used for both binary (fraudulent vs. nonfraudulent) and multiclass (e.g., different types of fraud) classification problems.
- **Fast detection:** Once the model is trained on a large dataset, it can be applied to new data very quickly. The model has already learned the patterns of normal data and anomalies, so it can easily differentiate between them. This makes these techniques useful for applications where real-time or near-real-time anomaly detection is required, such as in fraud detection, cybersecurity or equipment monitoring.
- **Labeled data-based specificity:** The ability to identify specific types of anomalies based on labeled data means that the algorithm is trained on a labeled dataset that contains examples of both normal and anomalous instances. This allows the algorithm to learn the specific characteristics and patterns associated with different types of anomalies, and to detect those same patterns in new data. As a result, supervised techniques can be more targeted in their detection of specific anomalies than unsupervised techniques, which may only identify general anomalies without specific knowledge of the anomalous patterns.

Weaknesses:

- **Data labeling requirement:** Supervised learning requires labeled data, which can be expensive and time-consuming to obtain.
- **Limited to known anomalies:** Supervised techniques can only detect anomalies that are similar to the labeled anomalies in the training data. This means that unknown or novel anomalies may not be detected.
- **Overfitting:** Supervised models can sometimes be prone to overfitting if the model is too complex or the dataset is imbalanced, which can lead to poor performance on new data.

Unsupervised

Unsupervised anomaly detection techniques are used when labeled data is scarce or unavailable. These techniques do not rely on prior knowledge of anomalies and detect them based on their deviation from normal patterns or behaviors. Engineers can apply unsupervised learning methods to automate feature learning and work with unstructured data. Some common techniques in this category include density-based techniques (such as DBSCAN and LOF), distance-based techniques (such as kNN and Isolation Forest), principal component analysis (PCA), autoencoders and clustering-based techniques (such as k-means and hierarchical clustering).

Unsupervised anomaly detection techniques are appropriate for industries and use-cases where labeled data is scarce or expensive to obtain, or where the types of anomalies are unknown or evolving. Examples include fraud detection in finance, intrusion detection in cybersecurity and equipment failure prediction in manufacturing.

Strengths:

- Unsupervised techniques can detect anomalies that have not been previously labeled, making them suitable for cases where labeled data is scarce or unavailable.
- They do not require prior knowledge of what constitutes an anomaly. They can identify anomalies based on the differences in the patterns of the data points.
- Unsupervised techniques are flexible and can be applied to various types of data, including high-dimensional and complex data, such as network traffic and images.

- These techniques are capable of detecting various types of anomalies, including point anomalies, contextual anomalies and collective anomalies. (Point anomalies, also known as global anomalies, are data points that significantly deviate from the normal behavior of the entire dataset. Contextual anomalies, also known as conditional anomalies, are data points that deviate from the normal behavior of the dataset within a specific context or condition. Collective anomalies, also known as group anomalies, are groups of data points that exhibit abnormal behavior collectively but may not be anomalous when considered individually.)

Weaknesses:

- Unsupervised techniques do not provide a clear labeling of detected anomalies, making it difficult to distinguish them from false positives or anomalies that are not of interest to the user.
- Unsupervised techniques may not provide a clear threshold for anomaly detection, which can make it challenging to determine the severity of detected anomalies.
- These types of techniques may have a high false positive rate, leading to a large number of detected anomalies that are not truly anomalous.
- Unsupervised techniques may lack interpretability, making it challenging to understand why a particular data point was flagged as anomalous.

Semisupervised

Semisupervised anomaly detection techniques use a combination of labeled and unlabeled data to detect anomalies. In these techniques, the algorithm learns from both labeled data, which provides information about the nature of the anomaly, and unlabeled data, which helps to capture the normal behavior of the system. These techniques combine the strengths of both supervised and unsupervised methods. Labeled data is used to train the model to detect specific types of anomalies, and unlabeled data is used to identify types of anomalies that may not be captured by the labeled data.

One common approach in semisupervised anomaly detection is to use a combination of a classifier and a clustering algorithm. The classifier is trained on the labeled data to identify known anomalies, and the clustering algorithm is used on unlabeled data to identify new anomalies not captured by the classifier. Another approach is to use a generative model, such as a variational autoencoder (VAE), to learn the underlying distribution of the data. The labeled data is used to train the VAE to reconstruct normal data, and the unlabeled data is used to identify any deviations from the learned normal distribution, which may indicate the presence of anomalies.

Some of the most common semisupervised anomaly detection techniques are: one-class SVM with partial labels, deep generative models (e.g., variational autoencoder with labeled and unlabeled data), transductive SVM, self-training and co-training.

Strengths:

- Semisupervised techniques can detect both known anomalies (through the labeled data) and unknown anomalies (through the unlabeled data).
- These techniques can achieve better performance than unsupervised techniques by leveraging information in the labeled data.
- They require less labeled data than supervised techniques, reducing the effort and cost required for labeling.

Weaknesses:

- The effectiveness of semisupervised techniques relies on the quality of the labeled data, and the model may perform poorly if the labeled data is incomplete or inaccurate.
- Semisupervised techniques may not be able to detect complex anomalies that are not represented in the labeled data. The labeled data used to train the semisupervised model may not sufficiently represent all types of anomalies that may occur in the data. In such cases, the model may struggle to detect complex anomalies that do not fit within the labeled data's representation of what is anomalous.
- They may require more computational resources than unsupervised techniques, as they involve both training a classifier and clustering algorithm or generative model.

Accuracy, Complexity and Explainability

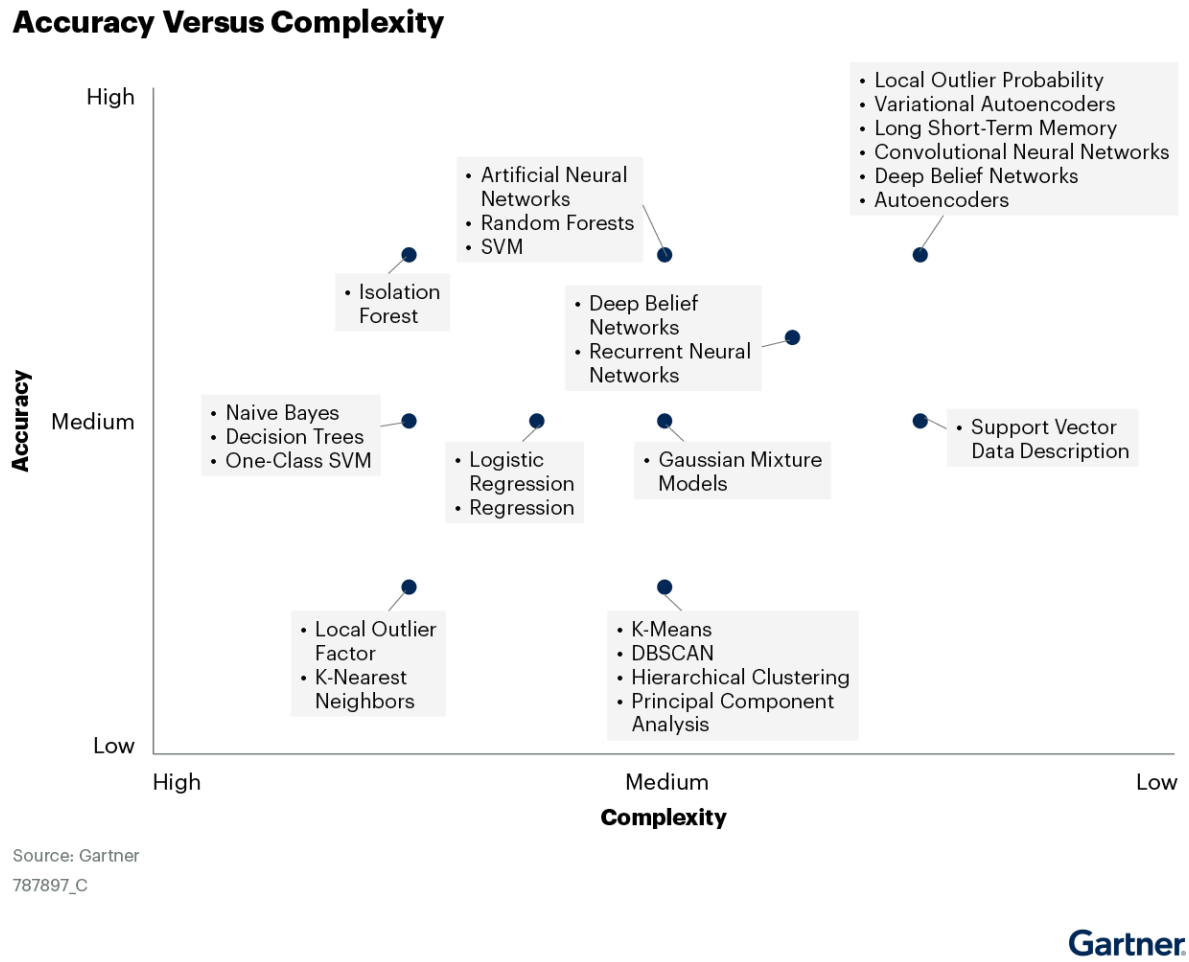
Anomaly detection techniques can be evaluated based on different criteria such as accuracy, complexity, and interpretability/explainability. These criteria are often interrelated, with more accurate models typically being more complex and less interpretable.

The complexity of an ML technique refers to the amount of computational resources and time required to train and deploy the model. Explainability in ML refers to the ability to understand and interpret the reasoning behind a model's predictions. An explainable model provides clear and understandable explanations of how it reached its conclusions. Interpretability is related to explainability but refers to the degree to which a human can understand and reason about the model's internal workings. An interpretable model provides insight into how it is making its decisions, allowing human experts to assess its accuracy and provide feedback.

In general, more accurate techniques tend to be more complex, and simpler techniques tend to sacrifice some accuracy in favor of ease of use and interpretability. However, the relationship between accuracy and complexity is not always straightforward. Some techniques may be able to achieve high accuracy with relatively low complexity, and others may be highly complex but still struggle to accurately detect anomalies. For example complex techniques such as deep learning models have the potential to achieve high accuracy, but can be computationally intensive and require large amounts of training data.

Figure 4 shows accuracy level versus complexity of anomaly detection techniques.

Figure 4: Accuracy Versus Complexity



Source: Gartner
787897_C

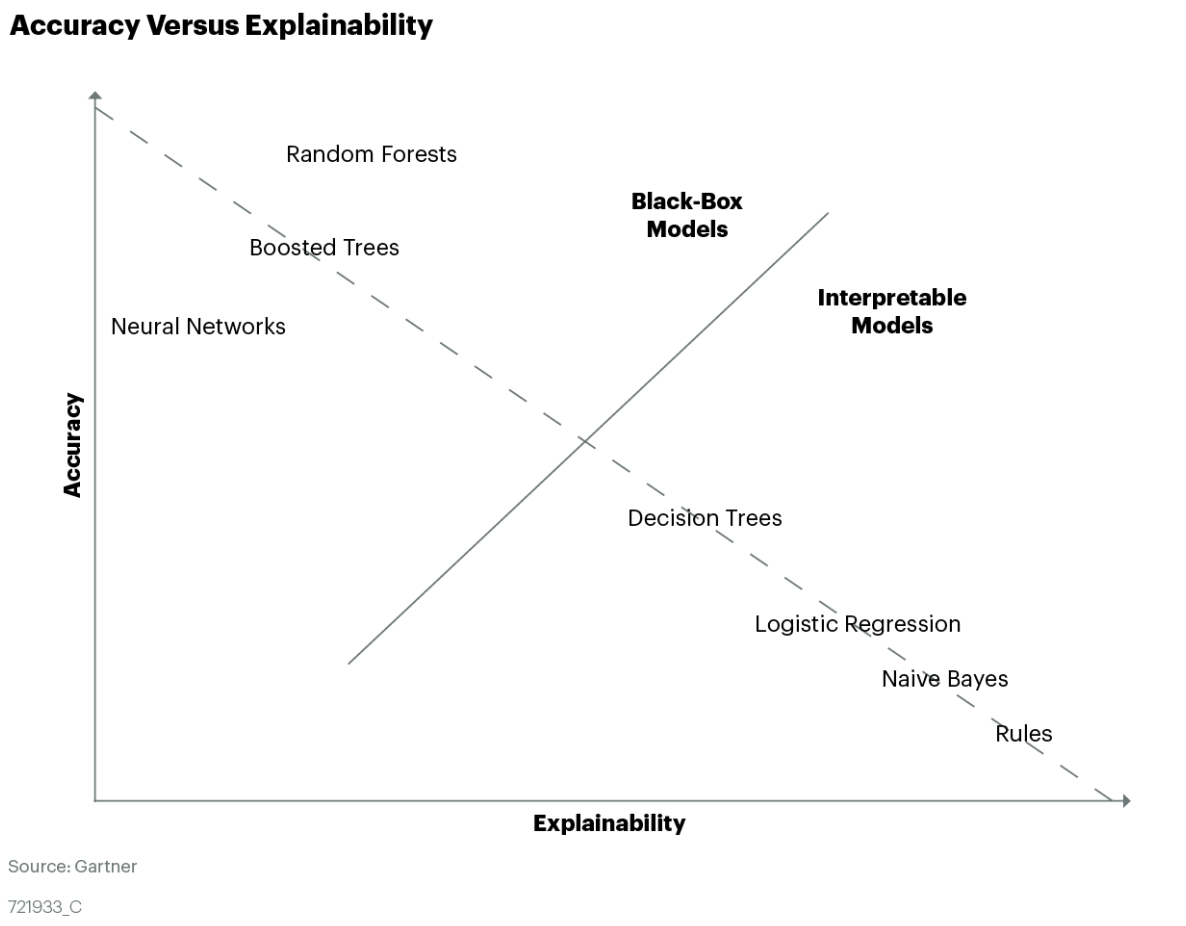
Gartner

The concept of explainability is not required for all types of AI techniques. Simpler, rule-based models such as decision trees and logistic regressions are easy to interpret and do not require extra features to be understandable. However, these models can quickly become complex if multiple variables are added to the algorithm.

Advanced AI techniques such as deep neural networks tend to lead to more accurate results but, by design, they tend to be opaque. Determining how a neural network model arrived at a certain output is difficult. This is why these advanced models require an extra layer of explainability (see Figure 5).

It's important to note that the choice of anomaly detection technique should be based on the specific needs and constraints of the problem at hand. For example, in some industries or applications, accuracy may be the top priority. In others, interpretability or speed may be more important. Ultimately, finding the right balance between accuracy and complexity is key to selecting an effective anomaly detection technique that meets the requirements of the specific use case.

Figure 5: Accuracy Versus Explainability



Source: Gartner

721933_C

Gartner

How to Choose the Right Technique

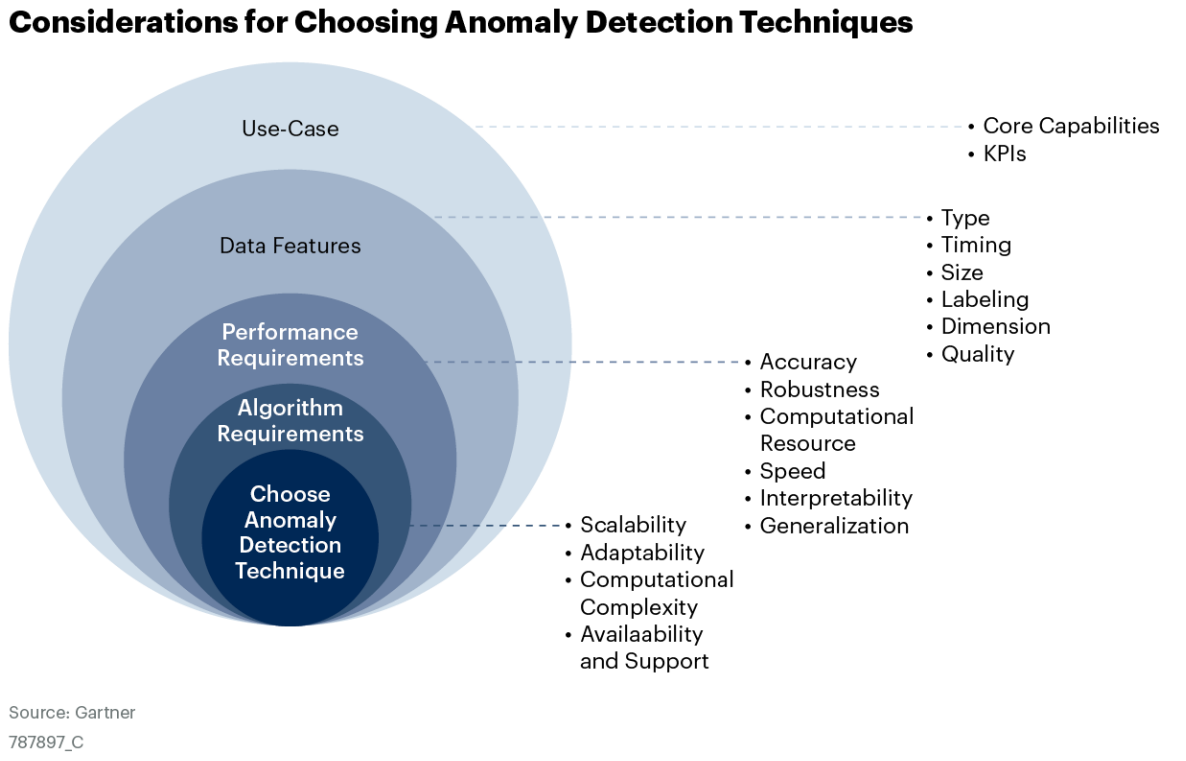
To choose an appropriate anomaly detection technique for a particular use case, several factors must be considered, including use-case requirements, data features, performance requirements and algorithm requirements:

- Use-case requirements involve understanding the problem being solved, the type of data being used and the desired outcome.

- Data features involve the size and complexity of the dataset, as well as the type and structure of the data.
- Performance requirements involve determining the desired level of accuracy, speed, and interpretability.
- Algorithm requirements involve understanding the adaptability, computational complexity, scalability, and availability of support for the chosen technique.

By carefully considering these factors, technical professionals can select an appropriate anomaly detection technique for the specific use case (see Figure 6).

Figure 6: Considerations for Choosing Anomaly Detection Techniques

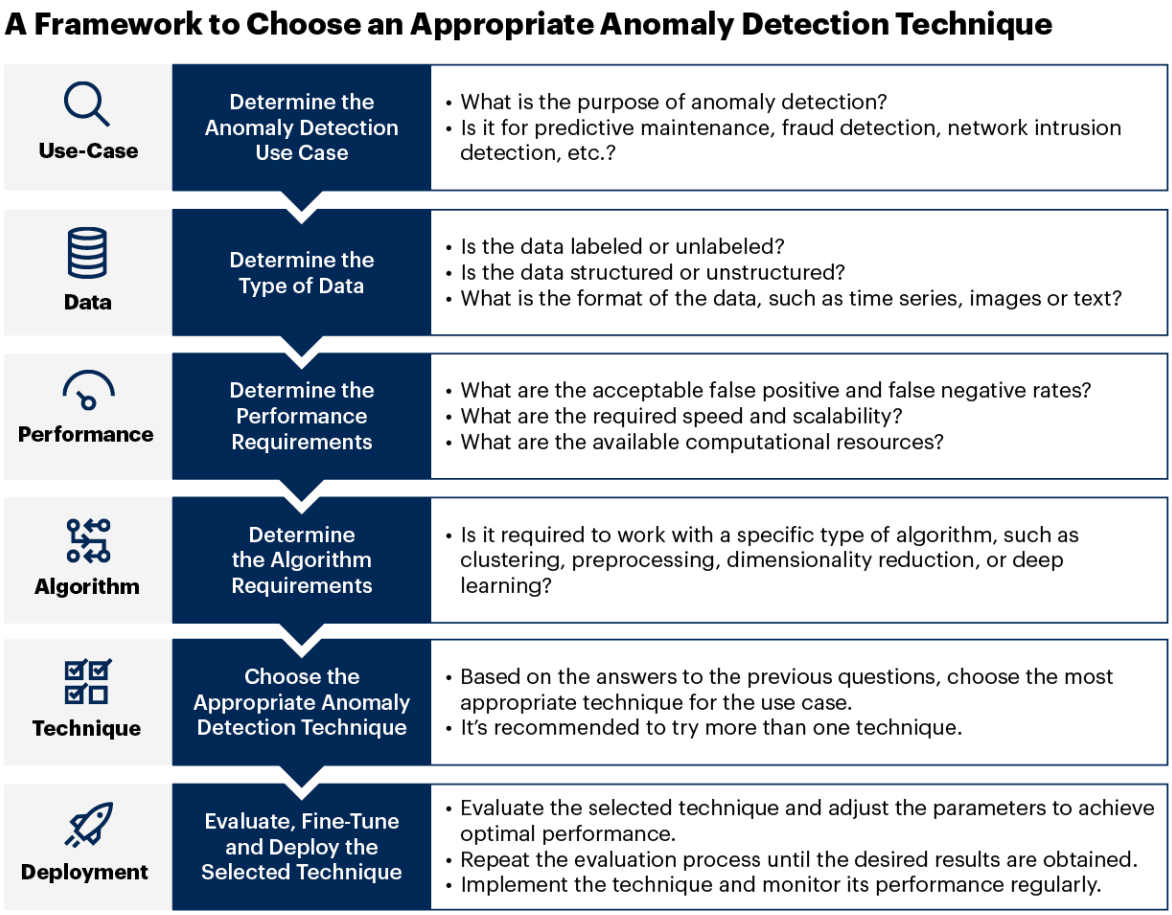


Choosing the right anomaly detection technique for a specific use case can be challenging. There are many factors to consider, including data characteristics, performance requirements and algorithm complexity.

To help guide the process, we present a six-step framework for choosing an appropriate anomaly detection technique (see Figure 7). The framework starts with defining the use case and selecting appropriate data features, followed by choosing a suitable technique based on performance and algorithm requirements. The selected technique is then fine-tuned and validated before being deployed into the production environment.

This framework provides a structured approach to help organizations make informed decisions when selecting an anomaly detection technique for their specific use case.

Figure 7: A Framework to Choose an Appropriate Anomaly Detection Technique



Source: Gartner
787897_C

Gartner

To provide a practical understanding of how to use this framework for selecting an appropriate anomaly detection technique, we will walk through the framework for two AI use cases in different industries (see the Guidance section):

1. Fraud detection in finance

2. Predictive maintenance in manufacturing

Trends in Anomaly Detection Techniques

Knowledge Graph

Knowledge graphs can be used in anomaly detection to improve accuracy and interpretability. By leveraging existing knowledge, such as ontologies and taxonomies, knowledge graphs can provide a structured representation of data and relationships between entities. This can help identify unusual patterns or relationships that might indicate anomalies. Knowledge graphs can be particularly useful in industries such as finance, healthcare and manufacturing where a lot of domain-specific knowledge is available. Additionally, knowledge graphs can be used to incorporate feedback from domain experts, making the anomaly detection process more accurate and effective over time.

To improve anomaly detection using knowledge graphs, it is important to ensure that the graph is accurate and up-to-date. This can be achieved by using data integration techniques to combine data from various sources and by using machine learning algorithms to automatically update the graph as new data becomes available. It is also essential to have a clear understanding of the data and the relationships between different data points to create an effective knowledge graph.

Generative AI

By leveraging generative AI, anomaly detection techniques can benefit from enhanced modeling capabilities, improved detection accuracy and increased resilience to variations in the data. This integration of generative AI with traditional ML techniques holds great promise in advancing the field of anomaly detection and addressing the challenges posed by complex and evolving datasets.

Generative AI can be used in data augmentation to create additional training samples, especially in cases where anomalous data is scarce. By learning from the training data, generative AI models can generate synthetic samples that closely resemble the normal data distribution. By generating synthetic anomalies, the training set can be enriched, improving the generalization and robustness of the anomaly detection model.

Furthermore, generative AI can be utilized to enhance the performance of traditional ML techniques for anomaly detection. One approach is to compare the generated synthetic data with real data to identify discrepancies or outliers that indicate anomalies. By capturing the complex relationships and patterns in the training data, generative models can effectively learn the normal behavior and identify deviations from it.

Strengths

Using anomaly detection in AI implementation can have significant business impacts across multiple industries. By enabling early detection, improving accuracy, handling big data and automating processes, anomaly detection can lead to improved outcomes, efficiency, productivity and risk mitigation as follows:

- Early detection of anomalies can have significant positive impacts on businesses across multiple industries by allowing organizations to respond quickly to potential issues, reduce losses and minimize disruption to their operations. For example, in the finance sector, early detection of fraudulent activities can help prevent significant financial losses. In the healthcare sector, accurate anomaly detection can help detect and diagnose diseases early, leading to better treatment outcomes and improved patient care.
- The improved accuracy of anomaly detection techniques can enable organizations to make more efficient and effective decisions, thus reducing the number of false positives or false negatives. This can have significant implications for businesses that rely on accurate data analysis, such as healthcare, finance and logistics. For instance, in healthcare, accurate anomaly detection can lead to better patient outcomes. In finance, it can help mitigate risks associated with fraudulent activities.
- The ability of anomaly detection techniques to handle big data can be beneficial for organizations across various industries. In manufacturing, transportation and logistics, where large amounts of data are generated daily, anomaly detection can help identify patterns, optimize processes and improve overall efficiency and productivity.
- Automation of anomaly detection can save time and resources, enabling organizations to focus on other critical tasks. In cybersecurity, for instance, automated anomaly detection can help organizations identify and respond to threats quickly, reducing the risk of a data breach. This can have significant cost savings and risk mitigation implications for businesses.

Weaknesses

Although anomaly detection techniques have significant strengths and can be a valuable tool for industries looking to identify and respond to anomalies quickly, there are some challenges that need to be considered when implementing them in different industries:

- The accuracy of anomaly detection depends heavily on the quality of data used. Incomplete or poor-quality data can lead to inaccurate results and false positives or negatives, affecting the overall effectiveness of the anomaly detection system. For example, in the retail industry, incomplete sales data can lead to inaccurate anomaly detection, resulting in missed opportunities for improving sales.
- Setting appropriate thresholds for anomaly detection can be challenging, especially in industries where normal behavior can vary significantly over time. It is essential to set thresholds that are flexible enough to adapt to changes in normal behavior while being robust enough to accurately identify significant anomalies. For instance, in the energy industry, setting appropriate thresholds for energy consumption can be challenging due to the seasonal and daily variations in energy usage patterns.
- Anomaly detection techniques may not be suitable for all applications, particularly those that require high precision or those that involve complex data structures. Organizations must carefully evaluate the suitability of anomaly detection techniques for their specific use cases and ensure that they are aligned with their business objectives and needs. For example, in the healthcare industry, anomaly detection techniques may not be suitable for detecting rare diseases that require highly specialized medical expertise.
- Some anomaly detection techniques, particularly those that use deep learning, can be difficult to interpret, making it hard to explain why certain anomalies were detected. Lack of interpretability can hinder decision-making and make it difficult to gain insights into the root cause of anomalies. For instance, in the finance industry, lack of interpretability of anomaly detection techniques can result in inaccurate fraud detection, leading to financial losses.

Guidance

In this section, we explore two real-world industry use cases that require the implementation of anomaly detection techniques. To recommend the appropriate anomaly detection techniques for each use case, we utilize the framework provided earlier, taking into account various factors discussed previously.

Example 1: Fraud Detection in Finance

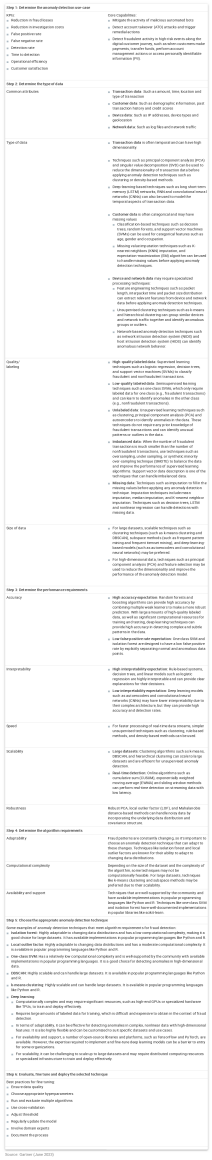
Fraud detection in finance is the process of identifying and preventing fraudulent activities related to financial transactions, such as credit card fraud, identity theft, money laundering, insider trading and other financial crimes.

Gartner defines the online fraud detection (OFD) market as the market for solutions that detect and prevent fraudulent actions within digital channels (browsers and mobile apps). OFD solutions provide a spectrum of capabilities within digital channels to prevent direct and indirect financial losses and to mitigate risks (see Market Guide for Online Fraud Detection).

Table 3 shows how to use the framework provided in Figure 7 to choose the appropriate anomaly detection technique for fraud detection.

Table 3: How to Choose an Anomaly Detection Technique for Fraud Detection

(Enlarged table in Appendix)



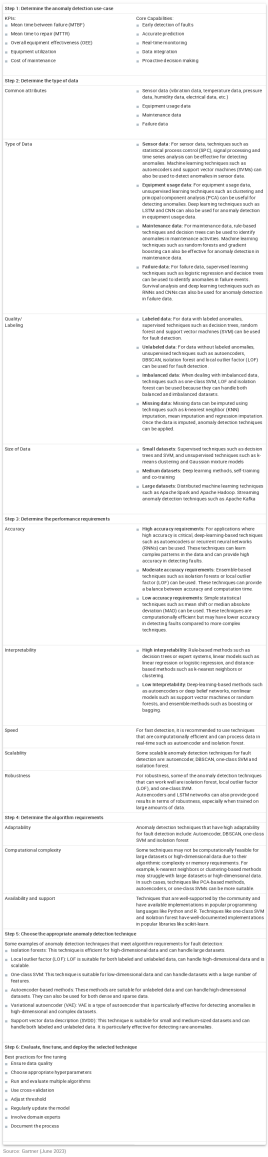
Example 2: Predictive Maintenance in Manufacturing

Predictive maintenance is a process of using data analysis techniques to predict when a machine or equipment is likely to fail, and then taking necessary actions to prevent the failure from happening. In the manufacturing industry, predictive maintenance is crucial to minimize downtime and prevent costly breakdowns. Anomaly detection techniques can be used to identify patterns and trends in machine data that indicate a fault or failure. By monitoring the machines and analyzing their data, the system can alert maintenance personnel of potential problems and schedule maintenance activities before a breakdown occurs. This helps manufacturers to save time and money by avoiding costly unplanned downtime, reducing maintenance costs and extending the life of equipment.

Table 4 shows how to use the framework provided in Figure 7 to choose the appropriate anomaly detection technique for fault detection in manufacturing.

Table 4: How to Choose Anomaly Detection Technique for Predictive Maintenance

(Enlarged table in Appendix)



Step 1: Determine the anomaly detection use-case

KPIs:	Core Capabilities:
▪ Mean time between failure (MTBF) ▪ Mean time to repair (MTTR) ▪ Overall equipment effectiveness (OEE) ▪ Equipment utilization ▪ Cost of maintenance	▪ Early detection of faults ▪ Accurate prediction ▪ Real-time monitoring ▪ Data integration ▪ Proactive decision making

Step 2: Determine the type of data

Common attributes	▪ Sensor data (vibration data, temperature data, pressure data, humidity data, electrical data, etc.) ▪ Equipment usage data ▪ Maintenance data ▪ Failure data
Type of Data	▪ **Sensor data**: For sensor data, techniques such as statistical process control (SPC), signal processing and time series analysis can be effective for detecting anomalies. Machine learning techniques such as autoencoders and support vector machines (SVMs) can also be used to detect anomalies in sensor data. ▪ **Equipment usage data**: For equipment usage data, unsupervised learning techniques such as clustering and principal component analysis (PCA) can be useful for detecting anomalies. Deep learning techniques such as LSTM and CNN can also be used for anomaly detection in this type of data. ▪ **Maintenance data**: For maintenance data, rule-based techniques and decision trees can be used to identify anomalies in maintenance records. Ensemble learning techniques such as random forests and gradient boosting can also be effective for anomaly detection in maintenance data. ▪ **Failure data**: For failure data, supervised learning techniques such as logistic regression and decision trees can be used to identify anomalies in failure events. Survival analysis and deep learning techniques such as RNNs and CNNs can also be used for anomaly detection in failure data.
Quality/ Labelling	▪ **Labeled data**: For data with labeled anomalies, supervised techniques such as decision trees, random forest and support vector machines (SVM) can be used for fault detection. ▪ **Unlabeled data**: For data without labeled anomalies, unsupervised techniques such as autoencoders, DBSCAN, isolation forest and local outlier factor (LOF) can be used for fault detection. ▪ **Imbalanced data**: When dealing with imbalanced data, techniques such as one-class SVM, LOF and isolation forest can be used because they can handle both balanced and imbalanced datasets. ▪ **Missing data**: Missing data can be imputed using techniques such as k-nearest neighbor (KNN) imputation, mean imputation and regression imputation. Once the data is imputed, anomaly detection techniques can be applied.
Size of Data	▪ **Small datasets**: Supervised tech niques such as decision trees and SVM, and unsupervised techniques such as k-means clustering and Gaussian mixture models ▪ **Medium datasets**: Deep learning methods, self-training and co-training ▪ **Large datasets**: Distributed machine learning techniques such as Apache Spark and Apache Hadoop. Streaming anomaly detection techniques such as Apache Kafka

Step 3: Determine the performance requirements

Accuracy	▪ **High accuracy requirements**: For applications where high accuracy is critical, deep-learning-based techniques (such as autoencoders or recurrent neural networks (RNNs)) can be used. These techniques can learn complex patterns in the data and can provide high accuracy in detecting faults. ▪ **Moderate accuracy requirements**: Ensemble-based techniques such as isolation forests or local outlier factor (LOF) can be used. These techniques can provide a balance between accuracy and computation time. ▪ **Low accuracy requirements**: Simple statistical techniques such as mean shift or median absolute deviation (MAD) can be used. These techniques are computationally efficient but may have lower accuracy in detecting faults compared to more complex techniques.
Interpretability	▪ **High interpretability**: Rule-based methods such as decision trees or expert systems, linear models such as linear regression or logistic regression, and distance-based methods such as k-nearest neighbors or clustering. ▪ **Low interpretability**: Deep-learning-based methods such as autoencoders or deep belief networks, nonlinear models such as support vector machines or random forests, and ensemble methods such as boosting or bagging.
Speed	For fast detection, it is recommended to use techniques that are computationally efficient and can process data in real-time such as autoencoder and isolation forest.
Scalability	Some scalable anomaly detection techniques for fault detection are autoencoder, DBSCAN, one-class SVM and isolation forest.
Robustness	For robustness, some of the anomaly detection techniques that can work well are isolation forest, local outlier factor (LOF), and one-class SVM. Autoencoders and LSTM networks can also provide good results in terms of robustness, especially when trained on large amounts of data.

Step 4: Determine the algorithm requirements

Adaptability	Anomaly detection techniques that have high adaptability for fault detection include: Autoencoder, DBSCAN, one-class SVM and isolation forest
Computational complexity	Some techniques may not be computationally feasible for large datasets or high-dimensional data due to their algorithmic complexity or memory requirements. For example, k-nearest neighbors and density-based methods may struggle with large datasets or high-dimensional data. In such cases, techniques like PCA or deep learning autoencoders, or one-class SVMs can be more suitable.
Availability and support	Techniques that are well-supported by the community and have available implementations in popular programming languages like Python and R. Techniques like one-class SVM and isolation forest have well-documented implementations in popular libraries like scikit-learn.

Step 5: Choose the appropriate anomaly detection technique

Some examples of anomaly detection techniques that meet algorithm requirements for fault detection:

- Isolation forests: This technique is efficient for high-dimensional data and can handle large datasets.
- Local outlier factor (LOF): LOF is suitable for both labeled and unlabeled data, can handle high-dimensional data and is scalable.
- One-class SVM: This technique is suitable for low-dimensional data and can handle datasets with a large number of features.
- Autoencoder based methods: These methods are suitable for unlabeled data and can handle high-dimensional datasets. They can also be used for both dense and sparse data.
- Variational autoencoder (VAE): VAE is a type of autoencoder that is particularly effective for detecting anomalies in high-dimensional and complex datasets.
- Support vector data description (SVDD): This technique is suitable for small and medium-sized datasets and can handle both labeled and unlabeled data. It is particularly effective for detecting rare anomalies.

Step 6: Evaluate, fine tune, and deploy the selected technique

Best practices for fine tuning:

- Ensure data quality
- Choose appropriate hyperparameters
- Run and evaluate multiple algorithms
- Use cross-validation
- Adjust threshold
- Regularly update the model
- Involve domain experts
- Document the process

Source: Gartner (June 2023)

Recommendations for Techniques/Use Case Match Up

Selecting appropriate AI techniques that are linked to use cases, skills and data is an important step of AI implementation. The majority of AI techniques are mature and have been around for decades. Many are suited to specific problems, data and talents. For example, probabilistic reasoning techniques are particularly good for uncovering hidden patterns in a large amount of data, like fraud patterns, churn issues or risk variables. But ML techniques require a sufficient amount of very high-quality data, along with AI professionals who are conversant in analytical mechanisms and algorithms. When it comes to choosing AI techniques for various use cases, data scientists need to consider several factors to ensure effective and efficient solutions:

1. Gaining a deep understanding of the problem domain and specific requirements is crucial in identifying the most suitable techniques. This involves understanding the desired outcomes, data availability and any constraints.

2. Analyzing the characteristics of the data, such as volume, variety and patterns, helps in selecting appropriate techniques that can handle the data effectively. Considering factors like the scale of data, its complexity, and the need for real-time processing can guide the choice of techniques.

3. Evaluating performance metrics and considering algorithmic requirements, such as scalability and computational complexity, are important for selecting techniques that meet the desired performance goals.

4. Exploring a variety of techniques, including both traditional and cutting-edge approaches, and evaluating them on the specific data can provide insights into their strengths and limitations.

5. Stay updated with the latest research developments in the field also helps in keeping abreast of new techniques and methodologies.

6. Collaborating with domain experts and seeking advice from experienced practitioners can provide valuable insights and perspectives.

By following these recommendations, data scientists can make informed decisions and choose the most appropriate AI techniques for their use cases, leading to successful outcomes in their projects.

Acronym Key and Glossary Terms

PCA	principal component analysis
CNN	convolutional neural networks
RNN	recurrent neural networks
LSTM	long short-term memory
LDA	linear discriminant analysis
SVMs	support vector machines
DBSCAN	density-based spatial clustering of applications with noise
LOF	local outlier factor
KNN	k-nearest neighbors
VAE	variational autoencoder
SVDD	support vector data description

Recommended by the Author

Some documents may not be available as part of your current Gartner subscription.

5 Practical Steps to Implement AI Techniques

Uncovering Artificial Intelligence Business Opportunities in Over 20 Industries and Business Domains

Toolkit: Discover and Prioritize Your Best AI Use Cases With a Gartner Prism

Market Guide for Online Fraud Detection

Table 1: Solution Pattern Profile for Anomaly Detection

Pattern No. 1	Anomaly Detection
Description	A process used to identify data points that deviate significantly from the norm or expected behavior. It involves analyzing data to find unusual patterns or outliers that may indicate a potential problem or opportunity. It is a powerful tool for identifying and addressing potential issues in a proactive manner, leading to better decision making and improved outcomes in various industries.
Example Applied Areas	■ Fraud detection in finance ■ Fault/defect detection in manufacturing ■ Attack detection in cybersecurity ■ Disease outbreak detection and patient monitoring in healthcare
Typical techniques	■ ML ■ Expert system ■ Information theory
Data dependency	Historical time series data
Maturity	Medium to high

Challenge	■ False positive/negative detection ■ Lack of labeled data ■ Imbalanced data ■ Nonstationary data ■ Interpretability
Trends	■ DL ■ Graph ■ Causal AI ■ Ensemble techniques

Source: Gartner (June 2023)

Table 2: Anomaly Detection Use Cases

	Industry	Use Case
1	Banking	Customer transaction fraud detection
2	Finance	Anomaly and error detection
3	P&C and life insurance	Fraud detection
4	Retail	Risk/fraud/threat detection
5	Cybersecurity	Transaction fraud detection File-based malware detection Abnormal system behavior detection Text-based malicious intent detection
6	Digital commerce	Fraud detection
7	Defense and intelligence	Predictive equipment maintenance Early crisis detection
8	Software development and testing	Defect prediction Security/vulnerability detection
9	Healthcare provider	Digital speech analysis for clinical diagnosis
10	Supply chain	Asset performance
11	Automotive	Manufacturing quality control Machine predictive maintenance

12	Utilities	Asset predictive maintenance Infrastructure defect identification
13	AIOps	Anomaly detection
14	Transportation	Theft detection Asset performance management
15	Life science manufacturers	Safety signal detection Predictive maintenance

Source: Gartner (June 2023)

Table 3: How to Choose an Anomaly Detection Technique for Fraud Detection

Step 1: Determine the anomaly detection use-case	
KPIs: ■ Reduction in fraud losses ■ Reduction in investigation costs ■ False positive rate ■ False negative rate ■ Detection rate ■ Time to detection ■ Operational efficiency ■ Customer satisfaction	Core Capabilities: ■ Mitigate the activity of malicious automated bots ■ Detect account takeover (ATO) attacks and trigger remedial actions ■ Detect fraudulent activity in high-risk events along the digital customer journey, such as when customers make payments, transfer funds, perform account management actions or access personally identifiable information (PII).
Step 2: Determine the type of data	
Common attributes	■ **Transaction data:** Such as amount, time, location and type of transaction ■ **Customer data:** Such as demographic information, past transaction history and credit scores ■ **Device data:** Such as IP addresses, device types and geolocation ■ **Network data:** Such as log files and network traffic

Type of data	
	■ **Transaction data** is often temporal and can have high dimensionality.
	■ Techniques such as principal component analysis (PCA) and singular value decomposition (SVD) can be used to reduce the dimensionality of transaction data before applying anomaly detection techniques such as clustering or density-based methods. ■ Deep-learning-based techniques such as long short-term memory (LSTM) networks, RNN and convolutional neural networks (CNNs) can also be used to model the temporal aspects of transaction data.

	■ **Customer data** is often categorical and may have missing values
	  ■ Classification-based techniques such as decision trees, random forests, and support vector machines (SVMs) can be used for categorical features such as age, gender and occupation.
	  ■ Missing value imputation techniques such as K-nearest neighbors (KNN) imputation, and expectation-maximization (EM) algorithm can be used to handle missing values before applying anomaly detection techniques.

	■ **Device and network data** may require specialized processing techniques:
	  ■ Feature engineering techniques such as packet length, interpacket time and packet size distribution can extract relevant features from device and network data before applying anomaly detection techniques.
	  ■ Unsupervised clustering techniques such as k-means and hierarchical clustering can group similar devices and network traffic together and

	identify anomalous groups or outliers. ■ Network-based anomaly detection techniques such as network intrusion detection system (NIDS) and host intrusion detection system (HIDS) can identify anomalous network behavior.
Quality/ labeling	■ **High-quality labeled data:** Supervised learning techniques such as logistic regression, decision trees, and support vector machines (SVMs) to classify fraudulent and nonfraudulent transactions. ■ **Low-quality labeled data:** Semisupervised learning techniques such as one-class SVMs, which only require labeled data for one class (e.g., fraudulent transactions) and can learn to identify anomalies in the other class (e.g., nonfraudulent transactions). ■ **Unlabeled data:** Unsupervised learning techniques such as clustering, principal component analysis (PCA) and autoencoders to identify anomalies in the data. These techniques do not require any prior knowledge of fraudulent transactions and can identify unusual patterns or outliers in the data. ■ **Imbalanced data:** When the number of fraudulent transactions is much smaller than the number of nonfraudulent transactions, use techniques such as oversampling, under sampling, or synthetic minority over-sampling technique (SMOTE) to balance the data and improve the performance of supervised learning algorithms. Support vector data description is one of the techniques that can handle imbalanced data. ■ **Missing data:** Techniques such as imputation to fill in the missing values before applying any anomaly detection technique. Imputation techniques

	include mean imputation, median imputation, and K-nearest neighbor imputation. Techniques such as decision trees, LSTM and nonlinear regression can handle detections with missing data.
Size of data	■ For large datasets, scalable techniques such as clustering techniques (such as k-means clustering and DBSCAN), subspace methods (such as frequent pattern mining and frequent itemset mining), and deep-learning-based models (such as autoencoders and convolutional neural networks) may be preferred. ■ For high-dimensional data, techniques such as principal component analysis (PCA) and feature selection may be used to reduce the dimensionality and improve the performance of the anomaly detection model.
Step 3: Determine the performance requirements	
Accuracy	■ **High accuracy expectation:** Random forests and boosting algorithms can provide high accuracy by combining multiple weak learners to make a more robust prediction. With large amounts of high-quality labeled data, as well as significant computational resources for training and testing, deep learning techniques can provide high accuracy in detecting complex and subtle patterns in the data. ■ **Low false positive rate expectation:** One-class SVM and isolation forest are designed to have a low false positive rate by explicitly separating normal and anomalous data points.

Interpretability	■ **High interpretability expectation:** Rule-based systems, decision trees, and linear models such as logistic regression are highly interpretable and can provide clear explanations for their decisions. ■ **Low interpretability expectation:** Deep learning models such as autoencoders and convolutional neural networks (CNNs) may have lower interpretability due to their complex architecture, but they can provide high accuracy and detection rates.
Speed	■ For faster processing of real-time data streams, simpler unsupervised techniques such as clustering, rule-based methods, and density-based methods can be used.
Scalability	■ **Large datasets:** Clustering algorithms such as k-means, DBSCAN, and hierarchical clustering can scale to large datasets and are efficient for unsupervised anomaly detection. ■ **Real-time detection:** Online algorithms such as cumulative sum (CUSUM), exponentially weighted moving average (EWMA) and sliding window methods can perform real-time detection on streaming data with low latency.
Robustness	Robust PCA, local outlier factor (LOF), and Mahalanobis distance-based methods can handle noisy data by incorporating the underlying data

	distribution and covariance structure.
Step 4: Determine the algorithm requirements	
Adaptability	Fraud patterns are constantly changing, so it's important to choose an anomaly detection technique that can adapt to these changes. Techniques like isolation forest and local outlier factors are known for their ability to adapt to changing data distributions.
Computational complexity	Depending on the size of the dataset and the complexity of the algorithm, some techniques may not be computationally feasible. For large datasets, techniques like k-means clustering and subspace methods may be preferred due to their scalability.
Availability and support	Techniques that are well-supported by the community and have available implementations in popular programming languages like Python and R. Techniques like one-class SVM and isolation forest have well-documented implementations in popular libraries like scikit-learn.
Step 5: Choose the appropriate anomaly detection technique	
Some examples of anomaly detection techniques that meet algorithm requirements for fraud detection: ■ **Isolation forest:** Highly adaptable to changing data distributions and has a low computational complexity, making it a good choice for large datasets. It has available implementations in popular programming languages like Python and R. ■ **Local outlier factor:** Highly adaptable to changing data distributions and has a moderate computational complexity. It is available in popular programming languages like Python and R. ■ **One-class SVM:** Has a relatively low computational complexity and is well-supported by the community with available implementations in popular programming languages. It is a good choice for detecting anomalies in high-dimensional data. ■ **DBSCAN:** Highly scalable and can handle large datasets. It is available in popular programming languages like Python and R.	

- **k-means clustering:** Highly scalable and can handle large datasets. It is available in popular programming languages like Python and R.
- **Deep learning:**
  - Computationally complex and may require significant resources, such as high-end GPUs or specialized hardware like TPUs, to train and deploy effectively.
  - Requires large amounts of labeled data for training, which is difficult and expensive to obtain in the context of fraud detection.
  - In terms of adaptability, it can be effective for detecting anomalies in complex, nonlinear data with high-dimensional features. It is also highly flexible and can be customized to suit specific datasets and use cases.
  - For availability and support, a number of open-source libraries and platforms, such as TensorFlow and PyTorch, are available. However, the expertise required to implement and fine-tune deep learning models can be a barrier to entry for some organizations.
  - For scalability, it can be challenging to scale up to large datasets and may require distributed computing resources or specialized infrastructure to train and deploy effectively.

Step 6: Evaluate, fine tune and deploy the selected technique

Best practices for fine tuning:

- Ensure data quality
- Choose appropriate hyperparameters
- Run and evaluate multiple algorithms
- Use cross-validation
- Adjust threshold
- Regularly update the model
- Involve domain experts

- Document the process

Source: Gartner (June 2023)

Table 4: How to Choose Anomaly Detection Technique for Predictive Maintenance

Step 1: Determine the anomaly detection use-case	
KPIs: ■ Mean time between failure (MTBF) ■ Mean time to repair (MTTR) ■ Overall equipment effectiveness (OEE) ■ Equipment utilization ■ Cost of maintenance	Core Capabilities: ■ Early detection of faults ■ Accurate prediction ■ Real-time monitoring ■ Data integration ■ Proactive decision making
Step 2: Determine the type of data	
Common attributes	■ Sensor data (vibration data, temperature data, pressure data, humidity data, electrical data, etc.) ■ Equipment usage data ■ Maintenance data ■ Failure data
Type of Data	■ **Sensor data:** For sensor data, techniques such as statistical process control (SPC), signal processing and time series analysis can be effective for detecting anomalies. Machine learning techniques such as

	autoencoders and support vector machines (SVMs) can also be used to detect anomalies in sensor data. ■ **Equipment usage data:** For equipment usage data, unsupervised learning techniques such as clustering and principal component analysis (PCA) can be useful for detecting anomalies. Deep learning techniques such as LSTM and CNN can also be used for anomaly detection in equipment usage data. ■ **Maintenance data:** For maintenance data, rule-based techniques and decision trees can be used to identify anomalies in maintenance activities. Machine learning techniques such as random forests and gradient boosting can also be effective for anomaly detection in maintenance data. ■ **Failure data:** For failure data, supervised learning techniques such as logistic regression and decision trees can be used to identify anomalies in failure events. Survival analysis and deep learning techniques such as RNNs and CNNs can also be used for anomaly detection in failure data.
Quality/ Labeling	■ **Labeled data:** For data with labeled anomalies, supervised techniques such as decision trees, random forest and support vector machines (SVM) can be used for fault detection. ■ **Unlabeled data:** For data without labeled anomalies, unsupervised techniques such as autoencoders, DBSCAN, isolation forest and local outlier factor (LOF) can be used for fault detection. ■ **Imbalanced data:** When dealing with imbalanced data, techniques such as one-class SVM, LOF and isolation forest can be used because they can handle both balanced and imbalanced datasets.

	■ **Missing data:** Missing data can be imputed using techniques such as k-nearest neighbor (KNN) imputation, mean imputation and regression imputation. Once the data is imputed, anomaly detection techniques can be applied.
Size of Data	■ **Small datasets:** Supervised techniques such as decision trees and SVM, and unsupervised techniques such as k-means clustering and Gaussian mixture models ■ **Medium datasets:** Deep learning methods, self-training and co-training ■ **Large datasets:** Distributed machine learning techniques such as Apache Spark and Apache Hadoop. Streaming anomaly detection techniques such as Apache Kafka
Step 3: Determine the performance requirements	
Accuracy	■ **High accuracy requirements:** For applications where high accuracy is critical, deep-learning-based techniques such as autoencoders or recurrent neural networks (RNNs) can be used. These techniques can learn complex patterns in the data and can provide high accuracy in detecting faults. ■ **Moderate accuracy requirements:** Ensemble-based techniques such as isolation forests or local outlier factor (LOF) can be used. These techniques can provide a balance between accuracy and computation time.

	■ **Low accuracy requirements:** Simple statistical techniques such as mean shift or median absolute deviation (MAD) can be used. These techniques are computationally efficient but may have lower accuracy in detecting faults compared to more complex techniques.
Interpretability	■ **High interpretability:** Rule-based methods such as decision trees or expert systems, linear models such as linear regression or logistic regression, and distance-based methods such as k-nearest neighbors or clustering. ■ **Low Interpretability:** Deep-learning-based methods such as autoencoders or deep belief networks, nonlinear models such as support vector machines or random forests, and ensemble methods such as boosting or bagging.
Speed	For fast detection, it is recommended to use techniques that are computationally efficient and can process data in real-time such as autoencoder and isolation forest.
Scalability	Some scalable anomaly detection techniques for fault detection are: autoencoder, DBSCAN, one-class SVM and isolation forest.
Robustness	For robustness, some of the anomaly detection techniques that can work well are isolation forest, local outlier factor (LOF), and one-class SVM. Autoencoders and LSTM networks can also provide good results in terms of robustness, especially when trained on large amounts of data.
Step 4: Determine the algorithm requirements	

Adaptability	Anomaly detection techniques that have high adaptability for fault detection include: Autoencoder, DBSCAN, one-class SVM and isolation forest
Computational complexity	Some techniques may not be computationally feasible for large datasets or high-dimensional data due to their algorithmic complexity or memory requirements. For example, k-nearest neighbors or clustering-based methods may struggle with large datasets or high-dimensional data. In such cases, techniques like PCA-based methods, autoencoders, or one-class SVMs can be more suitable.
Availability and support	Techniques that are well-supported by the community and have available implementations in popular programming languages like Python and R. Techniques like one-class SVM and isolation forest have well-documented implementations in popular libraries like scikit-learn.

Step 5: Choose the appropriate anomaly detection technique

Some examples of anomaly detection techniques that meet algorithm requirements for fault detection:

- Isolation forests: This technique is efficient for high-dimensional data and can handle large datasets.
- Local outlier factor (LOF): LOF is suitable for both labeled and unlabeled data, can handle high-dimensional data and is scalable.
- One-class SVM: This technique is suitable for low-dimensional data and can handle datasets with a large number of features.
- Autoencoder-based methods: These methods are suitable for unlabeled data and can handle high-dimensional datasets. They can also be used for both dense and sparse data.
- Variational autoencoder (VAE): VAE is a type of autoencoder that is particularly effective for detecting anomalies in high-dimensional and complex datasets.
- Support vector data description (SVDD): This technique is suitable for small and medium-sized datasets and can handle both labeled and unlabeled data. It is particularly effective for detecting rare anomalies.

Step 6: Evaluate, fine tune, and deploy the selected technique
Best practices for fine tuning ■ Ensure data quality ■ Choose appropriate hyperparameters ■ Run and evaluate multiple algorithms ■ Use cross-validation ■ Adjust threshold ■ Regularly update the model ■ Involve domain experts ■ Document the process

Source: Gartner (June 2023)