# Table of Contents

What is a Transparency Note? .................................................................................................................................................. 3

The basics of Anomaly Detector............................................................................................................................................... 3
   Introduction ..................................................................................................................................................................................................... 3
   Key terms .......................................................................................................................................................................................................... 4

Capabilities ..................................................................................................................................................................................... 4
   System behavior ............................................................................................................................................................................................. 4
   Use cases........................................................................................................................................................................................................... 4
      Intended uses............................................................................................................................................................................................ 4
      Considerations when choosing a use case ........................................................................................................................................... 5

Limitations ...................................................................................................................................................................................... 5
   Technical limitations, operational factors and ranges .................................................................................................................... 5

System performance...................................................................................................................................................................... 6
   Best practices for improving system performance ........................................................................................................................... 7
      Univariate Anomaly Detection .................................................................................................................................................... 7
      Multivariate Anomaly Detection (MVAD) ......................................................................................................................................... 7

Evaluating and integrating Anomaly Detector for your use ............................................................................................. 8

Learn more about responsible AI ............................................................................................................................................ 10

Learn more about Anomaly Detector .................................................................................................................................... 10

Contact us ..................................................................................................................................................................................... 10

About this document ................................................................................................................................................................. 10
What is a Transparency Note?

An AI system includes not only the technology, but also the people who will use it, the people who will be affected by it, and the environment in which it is deployed. Creating a system that is fit for its intended purpose requires an understanding of how the technology works, what its capabilities and limitations are, and how to achieve the best performance. Microsoft’s Transparency Notes are intended to help you understand how our AI technology works, the choices system owners can make that influence system performance and behavior, and the importance of thinking about the whole system, including the technology, the people, and the environment. You can use Transparency Notes when developing or deploying your own system, or share them with the people who will use or be affected by your system.

Microsoft’s Transparency Notes are part of a broader effort at Microsoft to put our AI Principles into practice. To find out more, see the Microsoft AI principles.

The basics of Anomaly Detector

Introduction

Anomaly Detector is an AI service that enables you to monitor and detect abnormalities in time series data without technical expertise of machine learning. The Anomaly Detector APIs adapt by automatically identifying and applying the best-fitting models to your data, regardless of industry, scenario, or data volume. Anomaly Detector service has two features, Univariate Anomaly Detection and Multivariate Anomaly Detection.

The Anomaly Detector RESTful API takes time series data as its input, the key parts of which are timestamps and the numerical values of metrics to be analyzed. The output of the API contains the anomalous status of each data point and, in the case of identified anomalies, the variables that contributed to the anomalous status.
### Key terms

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time series</td>
<td>A time series is a series of data points indexed (or listed or graphed) in chronological order. Most commonly, a time series is a sequence taken at successive, equally spaced points in time. It is a sequence of discrete-time data and usually contains timestamps, dimensions (optional), and measures.</td>
</tr>
<tr>
<td>Granularity</td>
<td>In Univariate Anomaly Detection, time series input data needs to be at equally spaced points in time. This means the data needs to be pre-aggregated to a specific granularity, like by year, month, week, day, hour, minute or second.</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>In Univariate Anomaly Detection, this is a system parameter indicated by a numerical value that adjusts the tolerance of Univariate Anomaly Detection. Lowering the sensitivity increases the range identified as normal which leads to fewer anomalies detected.</td>
</tr>
<tr>
<td>MaxAnomalyRatio</td>
<td>In Univariate Anomaly Detection, this indicates the max ratio of anomalies to be detected from a time series. For example, if it’s set to 0.25, for a time series with 100 points, the max anomaly points would be 25.</td>
</tr>
<tr>
<td>SlidingWindow</td>
<td>In Multivariate Anomaly Detection, this parameter indicates how many data points are used to determine anomalies. SlidingWindow must be an integer between 28 and 2,880. The default value is 300.</td>
</tr>
<tr>
<td>AlignMode</td>
<td>In Multivariate Anomaly Detection, this parameter indicates how to align multiple variables (time series) on timestamps. There are two options for this parameter, Inner and Outer, and the default value is Outer.</td>
</tr>
</tbody>
</table>

### Capabilities

#### System behavior

Using Anomaly Detector doesn’t require prior experience in machine learning. The RESTful API enables you to easily integrate the service into your applications and processes. Anomaly Detector API can be deployed using the cloud or the intelligent edge with containers.

Anomaly Detector v1.1 supports two different features:

- **Univariate Anomaly Detection**: Detect anomalies in one variable, like revenue, cost, etc. The model is selected automatically based on your data pattern, without needing to be trained.
- **Multivariate Anomaly Detection**: Detect anomalies in multiple variables with correlations, which are usually gathered from equipment or other complex system. The underlying model used is graph attention networks, and the trained model performance varies based on your training data, fine tuning, and use cases.

#### Use cases

**Intended uses**

You can use Anomaly Detector for multiple scenarios, either with or without training depending on the Anomaly Detector feature you use. The system’s intended uses include:
- **Monitoring business metrics**: Track the performance of business metrics in real time to identify substantive or sudden changes. An example is an unexpected numerical pattern with daily active users of a website, stock inventory, or revenue.

- **Monitoring IT operations metrics**: Analyze and monitor telemetry data like performance counters from IT operations to detect anomalies that could impact the health and performance of an IT system. Examples include the CPU usage of VMs, the throughput of databases, or even the number of sign-ins and signups of a website.

- **Monitoring IoT data from sensors and predictive maintenance**: Sensor data read from IoT sensors that are deployed on factory floors, production lines, oil rigs, and pipelines, or even cars and drones could be analyzed and monitored with Anomaly Detector. Anomalies detected from the data might indicate an anomalous state of the machinery. The application you build can inform human reviewers, allowing for preventative maintenance actions before a bigger problem arises.

**Considerations when choosing a use case**

We encourage customers to leverage Anomaly Detector in their innovative solutions or applications. However, here are some considerations when choosing any use case:

- In use cases with potential to impact the physical safety of human beings, include human review of outputs prior to any decision regarding operations. Examples of such use cases include monitoring and detecting an anomalous state in the performance of machinery on factory floors or in production lines.

- Anomaly Detector is not suitable for decisions that may have serious adverse impacts. Examples of such use cases include health care scenarios like monitoring heartbeat or blood glucose levels. Decisions based on incorrect output could have serious adverse impacts. Additionally, it is advisable to include human review of decisions that have the potential for serious impacts on individuals.

- Because Anomaly Detector has no qualitative assessment capabilities, it is not suitable for monitoring or evaluating human activity, such as in the workplace. We do not recommend that it be used in this way. In addition, workplace monitoring may not be legal within your jurisdiction.

**Limitations**

**Technical limitations, operational factors and ranges**

There are several limitations of Anomaly Detector to be aware of:

- Anomaly Detector is stateless and doesn’t store any customer data or model updates. That means every API call is completely independent. Multiple API calls with a lot of data won’t change the accuracy of the service. To achieve better accuracy, tune within single API calls.

- Anomaly Detector doesn’t automatically tune parameters for customers. When customers want to rely on anomaly detection results from the service to trigger automated actions, we highly recommend that you perform an evaluation with real world data before you apply the settings to your scenario.

- Anomaly Detector only takes in time series data by using timestamps and numbers. The service has no knowledge of the context and surroundings where the data is collected. In production use, decision makers might need to consider knowledge beyond those measures.
System performance

The accuracy of anomaly detection can be measured by evaluating how well the system-detected anomalies correspond to actual anomalous events. An example is when an anomaly is captured by Anomaly Detector and at the same time an actual service outage is reported by a customer. To measure accuracy, the customer might use a set of historical data and let Anomaly Detector perform detection results. The customer could then compare that information with the record of real events and classify the detection results into two kinds of correct (or "true") anomalies and two kinds of incorrect (or "false") anomalies.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>True positive</td>
<td>The system-detected anomaly correctly corresponds to an actual anomalous event.</td>
<td>The system identifies an anomaly that corresponds to a real service outage.</td>
</tr>
<tr>
<td>TP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>True negative</td>
<td>The system correctly doesn’t detect any anomaly when metrics data follows a historical pattern that is within the range of normal operations.</td>
<td>The system doesn’t find an anomaly when there was no service outage.</td>
</tr>
<tr>
<td>TN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>False positive</td>
<td>The system incorrectly detects an anomaly when there’s no anomalous event.</td>
<td>The system detects an anomaly when there was no service outage.</td>
</tr>
<tr>
<td>FP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>False negative</td>
<td>The system fails to detect an anomaly when an anomalous event has occurred.</td>
<td>The system fails to capture an anomaly when a service outage occurred.</td>
</tr>
<tr>
<td>FN</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Precision and recall**

**Precision** is the proportion of true positives among all the positive results returned, regardless of their correctness (true positives and false positives). It indicates how many detected anomalies correspond to actual abnormal events. It can be calculated as follows:

\[
\text{Precision} = \frac{TP}{TP + FP}
\]

A precision score of 1.0 means that every anomaly detected corresponds to an actual abnormal event.

**Recall** is the proportion of true positives among all actual positives (both true positives and false negatives). It indicates how many of the actual abnormal events were detected. It can be calculated as follows:

\[
\text{Recall} = \frac{TP}{TP + FN}
\]

A recall score of 1.0 means that every actual abnormal event is detected.
Best practices for improving system performance

It can be difficult to optimize for both precision and recall at the same time. Depending on the specific scenario, customers might want to prioritize one over the other. For example, stakeholders who monitor specific KPIs might need a high recall rate so that no issues are missed. In contrast, stakeholders at the executive level might want to be notified only when something serious happens, which would require a high precision rate.

Several factors need to be balanced to get the accuracy needed to meet business requirements.

Univariate Anomaly Detection

Sensitivity

The sensitivity parameter (check API reference for more parameter descriptions) is the key to tuning the detection results. In general, a high sensitivity value means the model is more sensitive to outliers and is likely to identify more anomalies. A low sensitivity value usually means the model will tolerate minor anomalies.

Choose the right mode

There are two different detection modes offered by Anomaly Detector: /last mode and /entire mode. These two API operations look similar in terms of input parameters and output format. But there’s one key difference regarding the detection behavior which should be considered when choosing the mode to be used. The /entire operation uses all the data points from the request to create a single statistical model. The /last operation only looks at the window of data points that you’ve passed to the API, which is usually a subset of the entire dataset. In practice, the /entire operation is more likely to ignore random and small noise as it looks at the global statistical pattern. The /last operation only looks at the local window you indicated and is more likely to identify those noise points as anomalies.

Data granularity

When the raw data is at low granularity – e.g., one data point per second or even sub-second – a common practice is to aggregate (sum/average/sample) it to a higher granularity. The benefits of aggregation include:

- Easier alignment to even distribution on time, which is a protocol of the API input.
- Increased tolerance of random noise.
- Fewer data points and fewer API transactions.

The downside of aggregation is that subtle changes within lower granularity are not considered, which might lead to missing important anomalies.

Other factors can also be used to improve the accuracy of the model, such as specifying a known seasonal pattern. For more information on improving accuracy for Univariate Anomaly Detection, see Best practices when using the Univariate Anomaly Detection.

Multivariate Anomaly Detection (MVAD)

Data Quality and Quantity

The quality and quantity of training dataset is very important to the model performance.

- As the model learns normal patterns from historical data, the training data should represent the overall normal state of the system. It's hard for the model to learn these types of patterns if the training data is full of anomalies. An empirical threshold for the abnormal rate is 1% or below for good accuracy.
• In general, the missing value ratio of training data should be under 20%. Too much missing data may cause automatically filled values (usually linear values or constant values) being learned as normal patterns. That may result in real (not missing) data points being detected as anomalies.

• The underlying model of MVAD has millions of parameters. It needs a minimum number of data points to learn an optimal set of parameters. The empirical rule is that you need to provide 5,000 or more data points (timestamps) per variable to train the model for good accuracy. In general, the more the training data, the better the accuracy. However, in cases when you're not able to accrue that much data, we still encourage you to experiment with less data and see if the accuracy is still acceptable.

**SlidingWindow**

Multivariate Anomaly Detection takes a segment of data points to decide if the next data point is an anomaly. The length of the segment is a slidingWindow. Please keep two things in mind when choosing a slidingWindow value:

1. The properties of your data: whether it's periodic and the sampling rate. When your data is periodic, you could set the length of 1 - 3 cycles as the slidingWindow. When your data is at a high frequency (small granularity) such as minute-level or second-level, you could set a relatively higher value of slidingWindow.

2. The trade-off between training/inference time and potential performance impact. A larger slidingWindow may cause longer training/inference time. There is no guarantee that larger slidingWindows will lead to accuracy gains. A small slidingWindow may make it difficult for the model to converge to an optimal solution. For example, it is hard to detect anomalies when the slidingWindow has only two points.

**Severity and Score**

We recommend using severity as the filter to sift out ‘anomalies’ that aren't important to your business. Depending on your scenario and data pattern, anomalies that are less important often have relatively lower severity values or standalone (discontinuous) high severity values, like random spikes.

In cases where you need more sophisticated rules than thresholds against severity or duration of continuous high severity values, you may want to use score to build more powerful filters. Understanding how MVAD is using score to determine anomalies may help: We consider whether a data point is anomalous from both the global and local perspective. If the score at a timestamp is higher than a certain threshold, then the timestamp is marked as an anomaly. If the score is lower than the threshold but is relatively higher in a segment, it's also marked as an anomaly.

For more information on improving accuracy for Univariate Anomaly Detection, see [Best practices when using the Multivariate Anomaly Detection](#).

**Evaluating and integrating Anomaly Detector for your use**

Anomaly Detector’s performance will vary depending on the real-world uses that customers implement. To ensure optimal performance in their scenarios, customers should conduct their own evaluations of the solutions they implement using Anomaly Detector. As you do so, consider the following:

- Customers might have their own preferences for viewing the anomaly results. If so, they would need to develop additional interface elements to display the results according to their preference.

- Anomaly Detector deals with various metrics. Some of these metrics might be sensitive and shouldn't be accessed publicly. When you plan to implement a solution that uses Anomaly Detector, consider access control on different metrics data. Ensure that your solution is secure and has adequate controls to preserve the integrity of your content and prevent unauthorized access.
• Before deploying Anomaly Detector into your scenario, test how it performs using real world data and make sure it can serve the goal you are trying to achieve.
• Validate accuracy of the results generated by the service, do some fine-tune according to best practices for improving accuracy and make sure it delivers the accuracy you need.
• Anomaly Detector will not be 100% accurate, so consider how you will identify and respond to any errors that may occur.
• Anomaly Detector offers data insights, not recommendations on actions to take, so when responding to anomalies identified by this service, you should act based on your domain expertise.
Learn more about responsible AI
Microsoft AI principles
Microsoft responsible AI resources
Microsoft Azure Learning courses on responsible AI

Learn more about Anomaly Detector
Anomaly Detector - Anomaly Detection System | Microsoft Azure
Identify abnormal time-series data with Anomaly Detector - Training | Microsoft Learn
What is Anomaly Detector? - Azure Cognitive Services | Microsoft Learn

Contact us
Give us feedback on this document by email to: anomalydetector@microsoft.com

About this document
© 2022 Microsoft Corporation. All rights reserved. This document is provided “as-is” and for informational purposes only. Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred.
Published: 2022/11/1
Last updated: 2022/11/1