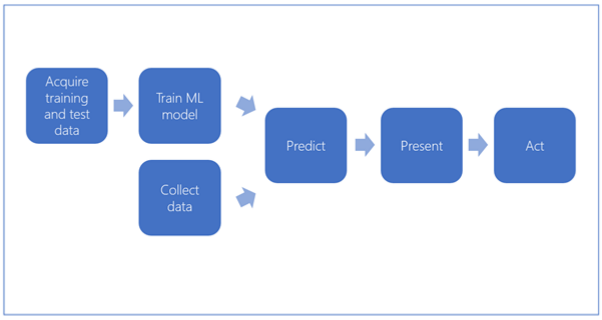
Predictive Maintenance Solution

To maximize equipment upkeep, predictive maintenance makes use of sensors, artificial intelligence, and data science. Manufacturers gain a lot of value by anticipating equipment maintenance needs to reduce costs and increase uptime.

The solution is based on data. Failure indicators and other aspects that describe the context must be present in the data. It can come from sensors, machine logs, and manufacturing application logs, among other places.

Options for developing a predictive maintenance solution are discussed in this article. To get you started, it provides a variety of viewpoints and references to existing resources. Equipment, environment, process, and organization all have different requirements for a predictive maintenance solution. We offer a variety of technologies and approaches to help you find a solution to your problems.

Let's begin with the fundamental parts of a predictive maintenance system.



The following high-level activities take place in this breakdown:

1. Gather data on training, including failure data.

2. Train a Machine Learning (ML) model using this training data to predict asset failures in the future based on a set of conditions.

3. Keep gathering data on a regular basis.

4. When you feed the ML model the data it has gathered, it will usually accurately predict failure. "There's an 85% probability that the machine will fail in the next 24 hours," for instance,

5. Identify the probable failure scenarios.

6. Plan and take action based on the data's insights.

Training the ML model

Data that is sufficient, accurate, and complete is needed to build an ML model. The availability of failure data is one of the unique obstacles that predictive maintenance faces. In high-capital equipment like computer numerical control (CNC) machines and oil refinery components, failures are relatively uncommon. We may not have sufficient failure data, even if we have collected sensor data for a considerable amount of time. Consider the definition of "failure." What exactly is meant by "failure"? Is it when the device suddenly stops working? Is it when the device deteriorates to the point where it no longer performs at the desired level? Is the failure case the destruction of a cutting machine as a result of a metal fatigue-related component failure or other warning signs before a catastrophe occurs?

Considering the data needed for ML

Also, consider whether we are collecting sufficient data to accurately record these failures. In many instances, sensor data alone may not be sufficient to locate a fault. To "flag" a machine's state as a failure state, we may occasionally require external data or a secondary source of information, such as an operator recording the failure case through a different system. This data may be in ERP, manufacturing execution systems (MES), historians, and other external systems. The fact that the data might cross the IT/OT divide, which is common in manufacturing businesses, makes it difficult to secure the necessary data.

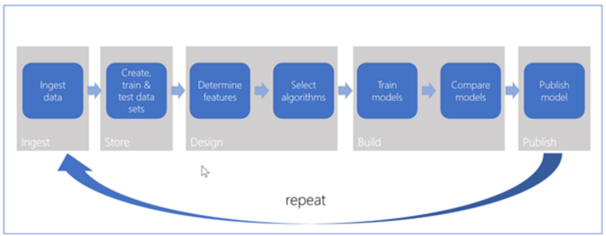
Predictive maintenance is a dynamic problem by definition, necessitating regular refreshment (or retraining) of the associated machine learning models. Predictive maintenance, if carried out correctly, should lessen the number of failures, which is a positive outcome; however, this also reduces the amount of failure data. Additionally, it is possible that the characteristics that influence failure will alter, rendering previous machine learning models invalid. When failure conditions change, we recommend periodically training models.

In addition, "fresh" data refer to new conditions added to the model in addition to those used for training. To put it another way, we might think of the failure as a function of the variables x1,x2,,xn, f(x1,x2,,xn), but we might find that the failure is also influenced by the variables x(n+1),,x(m+n), so we might need to adjust our model training to account for f(x1,x2,,x(m+n)). It's possible that the model won't be able to detect the failures well. For the subsequent iteration, the machine's MES log data points can be incorporated into a new model.

The data needed to train the machine learning models might already be in your MES, historians, or other production systems, even if you don't have a modern IoT environment that streams data to the cloud. Preparing the data for use in the training of the machine learning models is all that is required.

The typical procedure for training a machine learning model is depicted in the figure below. This appears to be an iterative procedure from the arrow labeled "repeat." The models are constantly retrained as new data and conditions change. Depending on the specifics of the implementation, this loop must repeat when and how often. In order to identify when models are "aging" or "degrading," you must carefully monitor how well the models you've already built perform.

The difficulty of managing new models comes from constantly training and deploying them. The CI/CD (continuous integration and continuous delivery) of models can be accomplished with the help of Microsoft's Azure Machine Learning Model Management service.



A comprehensive guide on how to prepare the data and train the machine learning model has been published by Microsoft. There are three typical maintenance questions and machine learning algorithms associated with them.

What is the likelihood that the asset will fail within the next X hours? Answer: A machine-learning technique known as 0-100% binary classification employs data to classify an item or row of data according to its membership in one of the two classes. Microsoft released a set of classification algorithms that are available as Machine Learning Studio modules. There are many different kinds of classification algorithms.

What is the asset's remaining useful life? Answer: A subset of machine learning algorithms known as X hours Regression can predict a variable's value from a set of other variables. Regression algorithms are included as modules in Machine Learning Studio.

LSTM (Long Short Term Memory): Deep neural networks (DNNs) include LSTM networks. Modeling the behavior of individual brain neurons serves as the inspiration for DNNs. Microsoft has published a guide that explains how to use an LSTM for predictive maintenance step by step. Which asset needs to be serviced the most quickly? Answer: Asset X Multi-class classification is a machine learning technique that uses data to classify an item or row of data as belonging to more than two classes by determining its category, type, or class.

Again, bringing the data in might require utilizing a number of different channels. In order to predict failures, you would first bulk initialize it and then continue receiving streaming data. Additionally, you would use it for subsequent model builds.

Bringing data to Azure

The data can be ingested and stored using a variety of services offered by Microsoft Azure. If the data hasn't already been moved to Azure, we recommend using batch methods. if you are able to export your data as files into common formats like csv, json, xml, and so on. These are viable choices. You also have the option of compressing them prior to uploading and further processing them in the cloud.

- Upload to blob storage with AzCopy (Windows and Linux).
- Linux mounts blob storage as a file system.
- If the size of the data is large and uploading it takes too long, use the Import/Export service.
- Windows, Linux, and macOS all support mounting an Azure File share.

You can also use Data Migration Assistant to move data from a SQL Server database to an Azure SQL Database.

For extract, transform, and load (ETL) operations, the Azure platform offers a variety of tools and services. The Azure Data Factory is the most well-known service, offering a comprehensive set of data manipulation tools. Through the open source libraries, the numerous ML services that are accessible on Azure offer additional options for manipulating data.

Microsoft Azure offers a lot of options for training the ML mode, each of which can be used in different ways.

- Azure Machine Learning Services
- Azure Machine Learning Studio
- Data Science Virtual Machine
- Spark MLLib in HDInsight
- Batch AI Training Service

The complexity of the operations, team experience, and data size all play a role in determining which tool to employ.

In addition to the costs of cloud services, the cost equation for cloud solutions includes many other variables, such as additional engineering, administration, data transfers, and so on. Make an informed decision by taking into account these variables as you evaluate the cost. The total cost does not include the cost of services.

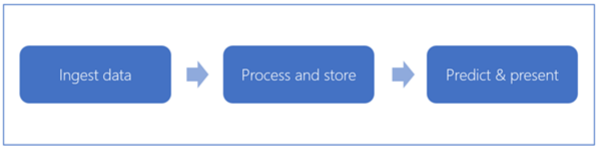
The technologies used differ in the specifics of the design of the data analysis and model publication processes. This article does not cover those subjects. There are a number of articles outlining the procedure and the Azure services that can be utilized to generate the model. Microsoft also offers a methodical approach to building solutions that makes it possible for teams of data scientists to work well together throughout the data's lifecycle.

If you want to learn more about building, deploying, and managing ML and AI models in the cloud, the Microsoft Machine Learning documentation is a good place to start.

There are numerous options for building ML models and processing large amounts of data on the Microsoft Azure platform. The construction of ML and AI models is made possible by the cloud platforms' almost limitless, scalable computing and storage resources. As a result, the most logical choice for putting this data flow into action is to build the models using Azure services.

Using the model

Once we have an ML model, we need a way to "use" it to predict when the equipment will need to be serviced. A processing layer processes the data after it is received from the equipment to anticipate future failures. It then provides the maintenance teams with a variety of options for taking action.



The data can be ingested either offline by periodically importing sensor data into the solution or online by streaming the live sensor data into the solution.

The data can be ingested, processed, and stored using a variety of services offered by the Microsoft Azure platform, including:

- Apache Kafka for HDInsight

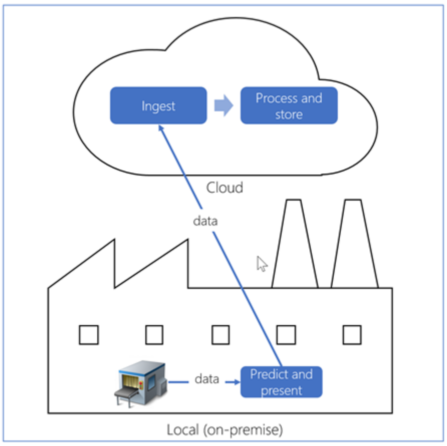
- Azure Event Hubs
- Azure Service Bus
- Azure IoT Hub

In contrast to the process of building the ML model, consuming it does not call for a lot of computational resources. The model can be installed locally on the factory floor or hosted as a service in the cloud, depending on your requirements.

The ML model can be executed in one of two primary locations: locally or just the cloud.

Local execution

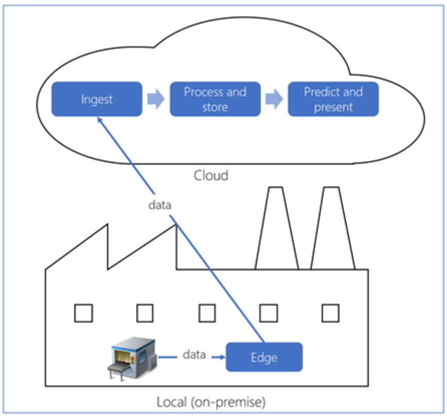
The data is sent to the cloud for ingestion, storage, and further processing while the ML model is used locally. In situations where prompt detection is absolutely necessary, this option is ideal.



Cloud execution

The Azure cloud can be used for the ML model's execution, processing, and ingest. If you share the results of the ML model's execution with multiple tenants or geographic regions (and if latency isn't a big deal), this option might be better for you. Locally, an optional component known as an "edge gateway" can carry out some of the work. Data projection and aggregation, stream analytics, and other tasks are all part of this work. It adheres to the "Ambassador" format.

On Azure, the model can be used in a number of different ways. The most straightforward Azure Machine Learning Web service makes use of Azure Machine Learning Studio as the model creation option. Another option is Azure Machine Learning Model Management, which features REST API endpoints with authentication, load balancing, automatic scale-out, encryption, and a comprehensive set of services for model management. Azure Container Service or a single machine (such as a Data Science Virtual Machine, an IoT device, or a local PC) can be used to deploy the model. The model can be used in a variety of ways, from custom applications to enterprise solution integration, once it is made available through a REST API.



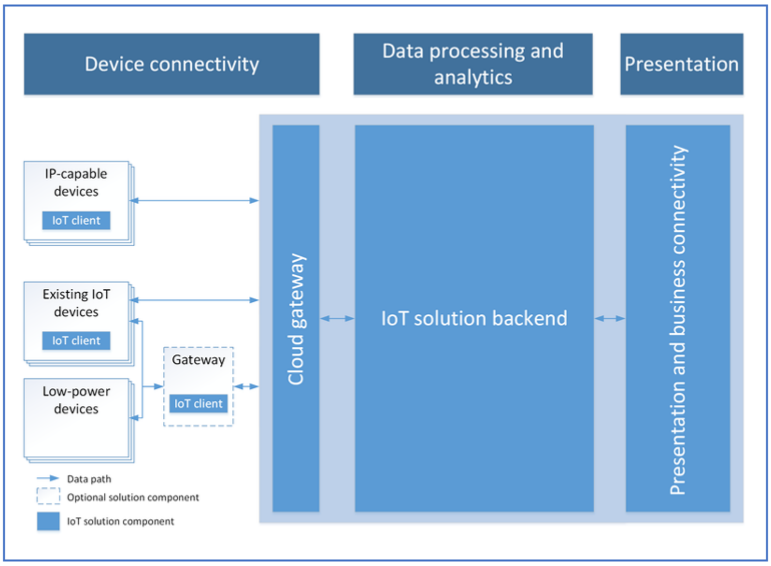
A deployment that only uses the cloud does not necessarily mean that data will be streamed live. It is possible to periodically import data from a local system (such as a historian or MES) into the cloud system, present the results, and so on. When devices are unable to directly push data into the system, when existing systems are already collecting data, or when near-real-time data processing is not required, this option may be a viable option. An edge gateway is not necessary in these circumstances.

Predictive maintenance in the IoT context

As part of their feature set, many IoT solutions ingest and store data. Predictive maintenance solutions can be a natural addition to Internet of Things solutions because they frequently rely on IoT data. In this context, it is essential to emphasize the significance of having failures recorded in the existing data in order to train a predictive model for the failures.

Processing data in near real time is required for some use cases. In these situations, we require a high data ingestion rate-capable scalable IoT solution. Many services on the Microsoft Azure platform can be used to create solutions for IoT requirements that are highly scalable. On the Azure platform, Microsoft's IoT solution architecture consists of three stages of logical components:

- Connectivity of devices
- data processing, and analytics
- presentation

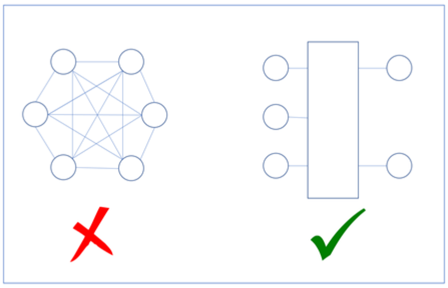


The Azure IoT Solution Architecture's specifics can be found online. However, the potentially significant number of devices connecting to backend services may present unique challenges.

Data ingestion and stream processing

A communication issue exists between two distinct services when it comes to importing data from the devices; i.e., systems that generate data (devices) and systems that process this data (for example, training a machine learning model and comparing the incoming data points to the trained model to predict how long it will last).

By definition, distributed systems are made up of distinct parts that have to talk to each other by default. Having related components communicate directly with one another is one way to facilitate communication. As a result, the system is difficult to manage and expand. The complexity of the communication links will increase to O(n2) as the number of components increases. Where data is posted and read to and from a common hub is a better approach.



The communication becomes more scalable by introducing a new component for data ingestion. With the option to geo-partition the data ingestion process, this component must be scalable, secure, and most likely globally accessible.

Considering the IoT solution's predictive maintenance feature. The data must be routed to services with predictive maintenance functionality as it flows through the gateway. Gateway routing is an additional pattern to think about.

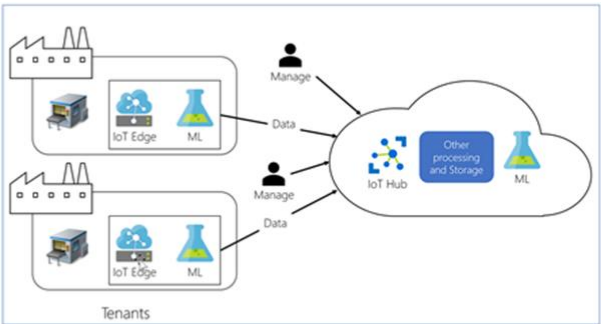
Utilizing the Azure service, IoT Hub, and Azure Stream Analytics, both patterns can be implemented.

Edge and cloud processing cooperation

Some equipment and devices are unable to consistently and directly connect to the internet. Sometimes, a common gateway needs to be used to get their data. For instance, the only data extraction interface offered by MTConnect agents is a REST interface.

Latency, the requirement to scrub device data locally before sending it to the cloud (in multitenant cases), and the requirement to perform projections or aggregations on device data are additional considerations. To address these requirements, the Ambassador pattern is an effective strategy. An implementation called Microsoft Azure IoT Edge can serve as a proxy for Microsoft Azure IoT Hub and provide local processing capabilities with remote management.

While still scrubbing and posting data to a multitenant solution in the cloud for archival, model training, and non-time-critical reporting, a typical deployment might include near-real-time alerts on the shop floor. Customers can interact with other shop-floor systems to deliver alerts and control the data-filtering options on the edge device with the capabilities of Azure IoT Edge and IoT Hub.



Multitenant perspective

Predictive maintenance services may be desired by some manufacturers or third parties, as previously mentioned. Multitenant cloud deployments are likely to provide these services, which come with their own set of difficulties:

Data security and isolation

The party providing a service must ensure that customers' confidential information is identified, appropriately secured, or scrubbed. Depending on the storage service used, Microsoft Azure offers encryption capabilities.

Utilizing well-known techniques like per-device certificates, per-device enable/disable, TLS security, X.509 support, IP allowlists or blocklists, and shared access policies, the devices' data generation and submission processes must also be protected. The party providing a service needs to make sure that customers' confidential information is found, properly secured, or deleted. Services that can be used to encrypt data at rest include Azure Data Lake Store, Azure Storage, Azure Cosmos DB, and Azure SQL Database. The solution providers should also think about how to divide the data between multiple resources—such as a database—or within the same resource.

Geographical considerations

The devices that generate data will most likely be spread out geographically. An ingestion point that is closest to the data source needs to be included in the solution. Continuous connectivity may also be a problem at times, necessitating the bulk ingest of data or the use of a local store-and-forward mechanism.

Scalability

The ML models can only be built using compute resources that are scalable. Processes that make good use of compute resources and are used by the solution provider to scale the solution as needed are essential.

Provisioning tenants and secure access

The service provider must come up with ways to effectively onboard new tenants and provide them with tools for managing their own accounts. Deployments to exclusive or shared resources are also decided at this point.

Pillars of software quality review

Beyond meeting the functional requirements, complex systems require additional scrutiny. Scalability, availability, resilience, management, and security are the five pillars of successful cloud solutions. In addition to the five pillars, we would like to highlight the solution's cost effectiveness.

For more information, please refer to the Azure Well-Architected Framework pillars of software quality.

Pillar

Scalability

The majority of Azure services allow for both vertical and horizontal scaling. Exploit on-request organization of assets on the Sky blue stage and control their scale (size and number of occurrences) by means of computerized administrations.

Availability and resiliency

Using numerous Azure services, compute and storage resources can be elastically provisioned as needed. The solutions must take into account and make use of the various SLA levels offered by the Azure services by adhering to appropriate design principles.

Management

ARM templates, PowerShell cmdlets and command line tools, Azure Management APIs, and other options are available for deploying and managing Azure resources. Instead of relying on UI-based tools to manage Azure resources, think about developing automated solutions.

Security

Over TLS, symmetrical and asymmetrical keys (such as X509 certificates and TPM) are supported by Azure IoT Hub. Encryption of data at rest is supported and the data stores are secured by Identity and Access Management (IAM) settings. Consider reviewing the auditing, authorization, authentication, transport and at rest encryption mechanisms as part of a general high-level security checklist.

Cost effective

Consider automating the provisioning and disposal of resources when they are not in use.

Conclusion

Predictive maintenance has been an ongoing topic of discussion. Implementers of predictive maintenance can now overcome numerous data-handling challenges thanks to recent advancements in cloud platforms like Microsoft Azure. Cloud platforms present new opportunities for implementing predictive maintenance and new revenue opportunities due to the elastic scaling of the compute and storage capacity. In order to achieve the business objectives of a predictive maintenance solution, Microsoft's Azure platform offers a variety of services with distinct capabilities. In addition to using the trained model to act on the outcomes predicted in the preceding sections, this article offered a vision for how to collect data and train data models.