Predictive Maintenance Using Machine Learning

Implementation Guide

Predictive Maintenance Using Machine Learning: Implementation Guide

Table of Contents

Deploy a solution that uses Amazon SageMaker to help automate the detection of potential equipment failures and provide recommended actions to take

July 2019 (last update (p. 15): June 2020)

This implementation guide discusses architectural considerations and configuration steps for deploying Predictive Maintenance Using Machine Learning in the Amazon Web Services (AWS) Cloud. It includes links to an AWS CloudFormation template that launches and configures the AWS services required to deploy this solution using AWS best practices for security and availability.

The guide is intended for developers and data scientists who have practical experience with machine learning and architecting in the AWS Cloud.

Overview

Many companies rely on people to perform routine diagnostic tests and preventive maintenance on fixed schedules. This can be a costly, labor-intensive process with little assurance that failures won't occur between tests.

Machine learning (ML) can provide a more reliable approach to preventive maintenance. ML models can help predict the likelihood of asset failure using sensor data, and optimize schedules for maintenance procedures. This predictive maintenance can help lower maintenance costs and reduce unscheduled downtime.

Amazon SageMaker is a fully managed service that enables developers and data scientists to quickly and easily build, train, and deploy machine learning models at any scale. Amazon SageMaker removes the barriers that typically slow down developers who want to use machine learning. This ability makes Amazon SageMaker applicable for a variety of use cases, including predictive maintenance.

To help customers more easily leverage Amazon SageMaker for predictive maintenance, AWS offers the Predictive Maintenance Using Machine Learning solution. This solution can help automate the detection of potential equipment failures, and provide recommended actions to take. The solution also includes an example dataset but you can modify the solution to work with any dataset. For more information on the example dataset, see Dataset (p. 4).

Cost

You are responsible for the cost of the AWS services used while running this solution. As of the date of publication, the one-time cost to train the solution's ML model in the US East (N. Virginia) Region is **$3** for the Amazon SageMaker ml.3p.2xlarge training instance. After the model is trained, the cost to process data from the example dataset is less than **$0.01 per hour**. Prices are subject to change. For full details, see the pricing webpage for each AWS service you will be using in this solution.

Architecture Overview

Deploying this solution builds the following environment in the AWS Cloud.

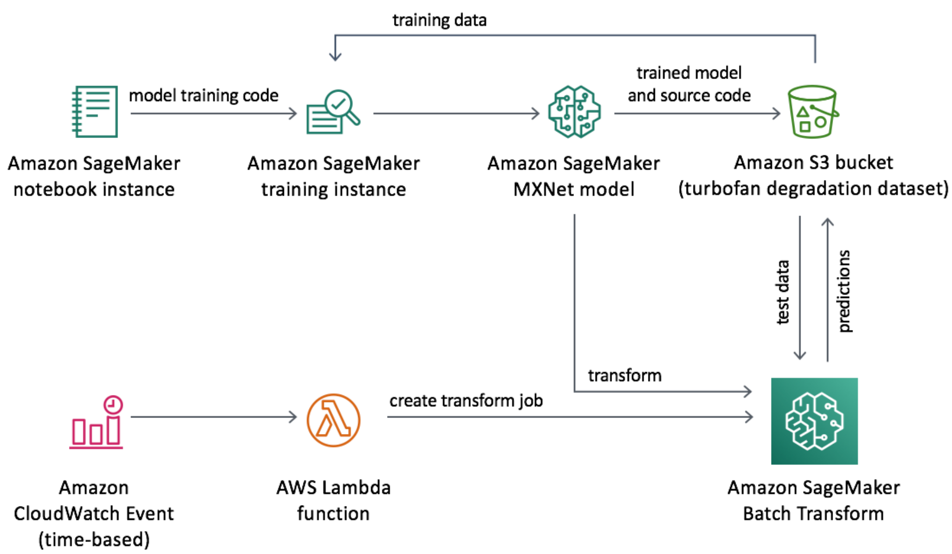

Figure 1: Predictive Maintenance Using Machine Learning architecture on AWS

The AWS CloudFormation template deploys an example dataset of a turbofan degradation simulation from NASA contained in an Amazon Simple Storage Service (Amazon S3) bucket and an Amazon SageMaker endpoint with an ML model that will be trained on the dataset to predict remaining useful life (RUL).

The solution uses an ml.t2.medium Amazon SageMaker notebook instance to orchestrate the model, but it uses an ml.3p.2xlarge SageMaker training instance to perform the training. The training code and trained model are stored in the solution's Amazon S3 bucket.

The solution also deploys an Amazon CloudWatch Events rule that is configured to run once per day. The rule is configured to trigger an AWS Lambda function that creates an Amazon SageMaker batch transform job that uses the trained model to predict RUL from the example dataset.

By default, the solution is configured to predict RUL from the example dataset. To use your own dataset, you must modify the solution. For more information, see Customization (p. 5).

Solution Components

Amazon SageMaker

Predictive Maintenance Using Machine Learning uses an Amazon SageMaker notebook instance, which is a fully managed machine learning (ML) Amazon Elastic Compute Cloud (Amazon EC2) compute instance that runs the solution's Jupyter notebook. The notebook is used to train and deploy the solution's ML model. For more information on notebook instances, see Use Notebook Instances in the *Amazon SageMaker Developer Guide*.

By default, the solution uses an ml.t2.medium instance. But, you can modify the solution to use a different instance type based on your specific needs.

Algorithm

Amazon SageMaker enables you to train custom deep learning models using your preferred deep learning framework. This solution leverages a custom stack long short-term memory (LSTM) neural network for learning historical patterns from time-series data. The stacked LSTM neural network training and inference code is implemented with Apache MXNet deep learning framework. Apache MXNet is a fast and scalable training and inference framework with an easy-to-use, concise API for machine learning. For more information, see Apache MXNet on AWS.

Dataset

Predictive Maintenance Using Machine Learning contains a publicly available turbofan engine degradation simulation data set from NASA that is used to train the solution's machine learning (ML) model and run inference with the model. The dataset was carried out using commercial modular aero-propulsion system simulation (C-MAPSS). Four different sets were simulated under different combinations of operational conditions and fault modes. Several sensor channels were recorded for the dataset to characterize fault evolution. For more information, see the appendix (p. 13).

Considerations

Customization

By default, Predictive Maintenance Using Machine Learning uses a NASA dataset to train the machine learning (ML) model. However, you can customize the solution to use your own dataset. To train the model on your own dataset, you must modify the included notebook to point the model to your dataset and to convert your dataset to an Apache MXNet Gluon dataset. You must also modify the solution's AWS Lambda function to process and transform your sensor data during inference.

The solution's Amazon CloudWatch Events rule is configured to trigger once per day by default. But, you can modify the rule to trigger an interval for your specific needs.

Regional Deployment

Predictive Maintenance Using Machine Learning uses Amazon SageMaker which is currently available in specific AWS Regions only. Therefore, you must launch this solution in a region where Amazon SageMaker is available. For the most current service availability by region, see AWS service offerings by region.

AWS CloudFormation Template

This solution uses AWS CloudFormation to automate the deployment of the Predictive Maintenance Using Machine Learning solution in the AWS Cloud. It includes the following AWS CloudFormation template, which you can download before deployment:

predictive-maintenance-using-machine-learning.template: Use this template to launch the solution and all associated components. The default configuration deploys an Amazon CloudWatch Events rule, an AWS Lambda function, an Amazon SageMaker notebook instance, and an Amazon Simple Storage Service (Amazon S3) bucket, but you can also customize the template based on your specific needs.

Automated Deployment

Before you launch the automated deployment, review the considerations discussed in this guide. Follow the step-by-step instructions in this section to configure and deploy Predictive Maintenance Using Machine Learning into your account.

Time to deploy: Approximately five minutes

What We'll Cover

The procedure for deploying this architecture on AWS consists of the following steps. For detailed instructions, follow the links for each step.

Step 1. Launch the Stack (p. 7)

- Launch the AWS CloudFormation template into your AWS account.
- Enter values for the required parameters: **Stack Name**, **Model and Data Bucket Name**
- Review the other template parameters, and adjust if necessary.

Step 2. Run the Notebook (p. 9)

- Run the Jupyter Notebook to train the ML model.

Step 3. Enable the CloudWatch Events Rule (p. 9)

- Enable the Amazon CloudWatch Events rule.

Step 4. Verify the Lambda Function Is Processing Data (p. 10)

- Verify that the AWS Lambda function is processing data.

Step 1. Launch the Stack

This automated AWS CloudFormation template deploys Predictive Maintenance Using Machine Learning in the AWS Cloud.

> **Note**
> You are responsible for the cost of the AWS services used while running this solution. See the Cost (p. 2) section for more details. For full details, see the pricing webpage for each AWS service you will be using in this solution.

1. Sign in to the AWS Management Console and click the button below to launch the `predictive-maintenance-using-machine-learning` AWS CloudFormation template.

   Launch Solution

You can also download the template as a starting point for your own implementation.

2. The template is launched in the US East (N. Virginia) Region by default. To launch this solution in a different AWS Region, use the region selector in the console navigation bar.

   Note
   This solution uses the Amazon SageMaker service, which is currently available in specific AWS Regions only. Therefore, you must launch this solution in an AWS Region where Amazon SageMaker is available. For the most current service availability by region, see the AWS service offerings by region.

3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack.
5. Under **Parameters**, review the parameters for the template, and modify them as necessary.

   This solution uses the following default values.

Parameter	Default	Description
Amazon S3 Bucket Configuration		
Model and Data Bucket Name	`<Requires input>`	Specify a name for a solution-created Amazon S3 bucket where Amazon SageMaker model and training data will be stored.
SageMaker S3 Prefix	`pred-maintenance-artifacts/`	The Amazon S3 prefix Amazon SageMaker uses for training and transformation jobs.
Amazon SageMaker Model Configuration		
SageMaker Model Name	`pred-maintenance-mxnet-model`	The name of the model and training job.
Model Training Instance Type	`ml.p3.2xlarge`	The SageMaker instance type for the model training job.
Amazon SageMaker Batch Transform Configuration		
Prediction Input File	`pred-maintenance-artifacts/data/test-0.csv`	The input file to process and batch transform.
Batch Transform Input S3 Location	`pred-maintenance-artifacts/batch-transform-input`	The S3 location for the processed input file for batch transforms.
Prediction Transform Results S3 Location	`pred-maintenance-artifacts/batch-inference`	The S3 location for the batch transform output.
VPC Configuration		
Use VPC for SageMaker Notebook	Yes	Choose whether to create a VPC endpoint for the S3 bucket and a notebook instance in a VPC. This

Parameter	Default	Description
		parameter uses an existing VPC in the AWS account. Select No if you don't want to deploy the notebook instance in a VPC.
VPC Id	*<Requires input>*	Select the **VPC Id** that will deploy the SageMaker notebook instance. If the **Use VPC for SageMaker notebook** parameter is set to No, the solution ignores the VPC Id.
Subnet Id	*<Requires input>*	Select the **Subnet Id** for the notebook instance. If the **Use VPC for SageMaker notebook** parameter is set to No, the solution ignores the Subnet Id.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Be sure to check the box acknowledging that the template will create AWS Identity and Access Management (IAM) resources.
9. Choose **Create stack** to deploy the stack.

   You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should see a status of **CREATE_COMPLETE** in approximately five minutes.

Step 2. Run the Notebook

1. Navigate to the Amazon SageMaker console.
2. In the navigation pane, select **Notebook instances**.
3. Select **PredictiveMaintenanceNotebookInstance**.

   The notebook instance should already be running.
4. Select **Open Jupyter**.
5. In the Jupyter notebook interface, open the `sagemaker_predictive_maintenance.ipynb` file.
6. In the **Cell** dropdown menu, select **Run All** to run the file.

Step 3. Enable the CloudWatch Events Rule

1. Navigate to the AWS Lambda console.
2. In the navigation pane, select **Functions**.
3. Select the `predictive-maintenance-batch-transformer` Lambda function.
4. In the diagram in the **Designer** tab, select **CloudWatch Events**.
5. In the **CloudWatch Events** tab, select *<stackname>*`-ScheduledRule-`*<id>*.
6. Select **Actions** > **Enable**.
7. Select **Enable**.

Step 4. Verify the Lambda Function Is Processing Data

1. Navigate to the AWS Lambda console.
2. In the navigation pane, select **Functions**.
3. Select the `predictive-maintenance-batch-transformer` Lambda function.
4. Select **Monitoring** and verify that the **Invocations** graph shows activity.

   After a few minutes, check the results Amazon S3 bucket for batch transform output.

Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This shared model can reduce your operational burden as AWS operates, manages, and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the services operate. For more information about security on AWS, visit the AWS Security Center.

Additional Resources

- Amazon SageMaker
- AWS Lambda
- Amazon CloudWatch Events
- Amazon Simple Storage Service
- AWS CloudFormation

Appendix: Acknowledgements

Predictive Maintenance Using Machine Learning contains a publicly available turbofan degradation simulation dataset from NASA that is used to train the solution's machine learning (ML) model and run inference with the model.

- A. Saxena and K. Goebel (2008). Turbofan engine degradation simulation data set. NASA Ames Prognostic Data Repository, NASA Ames, Moffett Field, CA.

Source Code

You can visit our GitHub repository to download the templates and scripts for this solution, and to share your customizations with others.

Document Revisions

Date	Change
July 2019	Initial release
June 2020	Pinned the version of gluon-nlp in the requirements file to a specific version; added VPC support for a notebook instance

Notices