Fraud Detection Using Machine Learning

AWS Implementation Guide

Soji Adeshina

Vishaal Kapoor

Thom Lane

Theodore Vasiloudis

May 2019

Last updated: May 2020 (see revisions)

Contents

About This Guide

This implementation guide discusses architectural considerations and configuration steps for deploying Fraud Detection Using Machine Learning on the Amazon Web Services (AWS) Cloud. It includes links to a 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 on the AWS Cloud.

Overview

Fraud is an ongoing problem that can cost businesses billions of dollars annually and damage customer trust. Many companies use a rule-based approach to detect fraudulent activity where fraud patterns are defined as rules. But, implementing and maintaining rules can be a complex, time-consuming process because fraud is constantly evolving, rules require fraud patterns to be known, and rules can lead to false positives or false negatives.

Machine learning (ML) can provide a more flexible approach to fraud detection. ML models do not use pre-defined rules to determine whether activity is fraudulent. Instead, ML models are trained to recognize fraud patterns in datasets, and the models are self-learning which enables them to adapt to new, unknown fraud patterns. In addition, unsupervised ML models allow us to extract knowledge from unlabeled data, flagging anomalous transactions for review.

Amazon SageMaker is a fully managed service that enables developers and data scientists to quickly and easily build, train, and deploy ML 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 fraud detection.

To help customers leverage Amazon SageMaker for real-time fraud detection, AWS offers the Fraud Detection Using Machine Learning solution. This solution automates the detection of potentially fraudulent activity, and flags that activity for review. This solution also includes an example dataset, but you can modify the solution to work with any dataset.

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 **$1.50** for the Amazon SageMaker ml.c4.large instance. The cost to process transactions using the example dataset is approximately **$0.65 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 and running the notebook builds the following environment in the AWS Cloud.

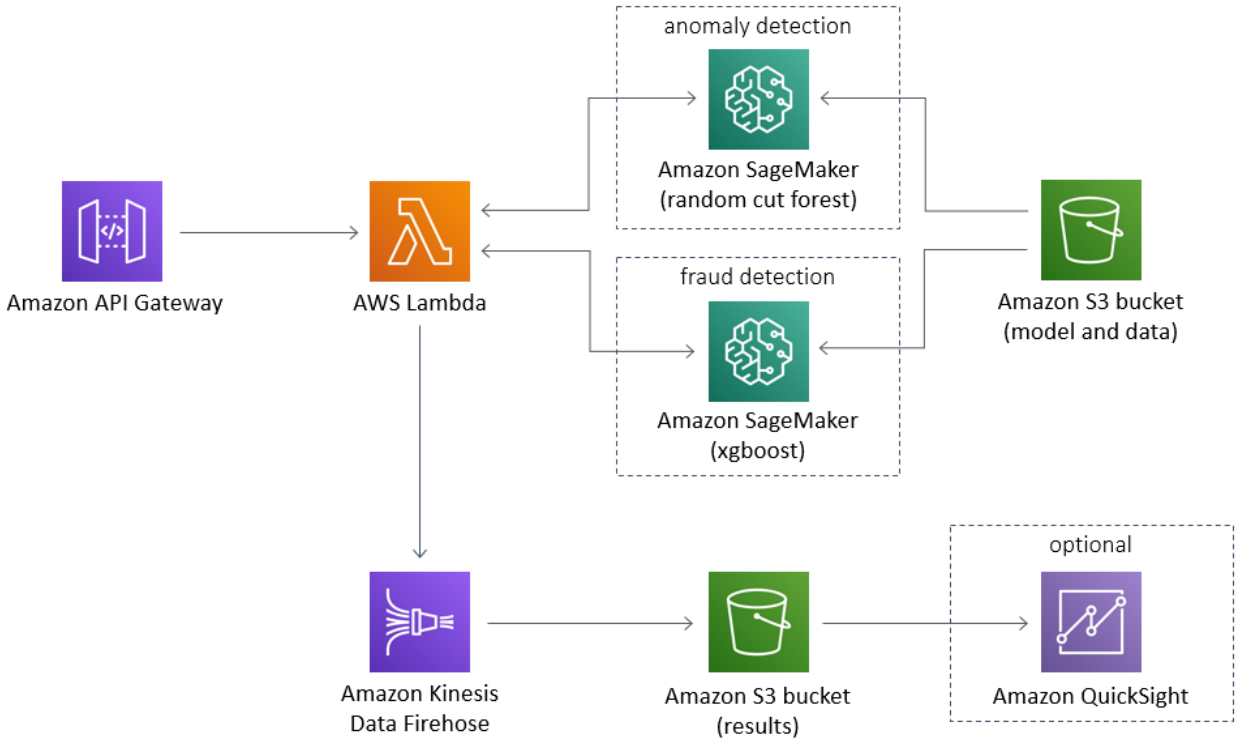

Figure 1: Fraud Detection Using Machine Learning architecture on AWS

The AWS CloudFormation template deploys an example dataset of credit card transactions contained in an Amazon Simple Storage Service (Amazon S3) bucket and an Amazon SageMaker notebook instance with different ML models that will be trained on the dataset.

The solution also deploys an AWS Lambda function that processes transactions from the example dataset and invokes the two Amazon SageMaker endpoints that assign anomaly scores and classification scores to incoming data points. An Amazon API Gateway REST API triggers predictions using signed HTTP requests, and an Amazon Kinesis Data Firehose delivery stream loads the processed transactions into another Amazon S3 bucket for storage.

The solution also provides an example of how to invoke the prediction REST API as part of the Amazon Sagemaker notebook.

Once the transactions have been loaded into Amazon S3, you can use analytics tools and services, including Amazon QuickSight, for visualization, reporting, ad-hoc queries, and more detailed analysis. For customers who want to use Amazon QuickSight to visualize the processed transactions, see Appendix A.

By default, the solution is configured to process transactions from the example dataset. To use your own dataset, you must modify the solution. For more information, see Customization.

Solution Components

Amazon SageMaker

Fraud Detection Using Machine Learning uses an Amazon SageMaker notebook instance: 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*.

Algorithm

Amazon SageMaker provides several built-in machine learning algorithms that you can use for a variety of problem types. This solution leverages the built-in Random Cut Forest algorithm for unsupervised learning and the built-in XGBoost algorithm for supervised learning. For more information, see How Random Cut Forest Works and How XGBoost Works in the *Amazon SageMaker Developer Guide*.

Dataset

Fraud Detection Using Machine Learning contains a publicly available anonymized credit card transaction dataset that is used to train the solution's machine learning (ML) model. The dataset was collected and analyzed during a research collaboration of Worldline and the Machine Learning Group of Université Libre de Bruxelles on big data mining and fraud detection. The dataset consists of anonymized credit card transactions over a two-day period in 2013 by European cardholders. In order to preserve the anonymity of the users, all features have been transformed using Principal Component Analysis (PCA), resulting in a dataset with 28 continuous PCA features, and two more features representing time and amount. Because the dataset is derived from real data, the distribution of fraud is low compared to legitimate transactions. Fraudulent transactions make up 0.172% of the total transactions. For more information, see Appendix B.

Considerations

Customization

By default, Fraud Detection Using Machine Learning uses a credit card fraud 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. You must also modify the solution's AWS Lambda function to process your events.

You can use the provided REST API to invoke the endpoints with incoming data points after you have trained on your historical data.

Regional Deployment

Fraud Detection Using Machine Learning uses Amazon SageMaker and Amazon Kinesis Data Firehose which are currently available in specific AWS Regions only. Therefore, you must launch this solution in a region where these services are 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 Fraud Detection Using Machine Learning solution on the AWS Cloud. It includes the following CloudFormation template, which you can download before deployment:

View template

fraud-detection-using-machine-learning.template: Use this template to launch the solution and all associated components. The default configuration deploys an Amazon API Gateway, an AWS Lambda function, an Amazon SageMaker notebook instance, an Amazon Kinesis Data Firehose delivery stream, and Amazon Simple Storage Service (Amazon S3) buckets, but you can also customize the template based on your specific needs.

Automated Deployment

Before you launch the automated deployment, please review the considerations discussed in this guide. Follow the step-by-step instructions in this section to configure and deploy Fraud Detection 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

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

Step 2. Run the Notebook

- Run the Jupyter Notebook to train the ML models.

Step 3. Verify the Lambda Function Is Processing Transactions

- Verify that the AWS Lambda function is processing transactions.

Step 1. Launch the Stack

This automated AWS CloudFormation template deploys Fraud Detection Using Machine Learning on the AWS Cloud.

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

1. Log in to the AWS Management Console and click the button to the right to launch the `fraud-detection-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 the solution in a different AWS Region, use the region selector in the console navigation bar.

> **Note**: This solution uses the Amazon SageMaker and Amazon Kinesis Data Firehose services, which are currently available in specific AWS Regions only. Therefore, you must launch this solution in an AWS Region where these services are available. For the most current availability by region, see 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
Results Bucket Name	*<Requires input>*	Specify a name for a solution-created S3 bucket where processed events will be stored
Amazon Kinesis Firehose Configuration		
Kinesis Firehose S3 Prefix	`fraud-detection/firehose/`	The Kinesis Data Firehose prefix for the delivery of processed events

6. Choose **Next.**
7. On the **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** 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.

> **Important:** If you delete the solution stack, you must manually delete the Amazon SageMaker endpoint (`fraud_detection_endpoint`).

Step 2. Run the Notebook

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

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

Step 3. Verify the Lambda Function Is Processing Transactions

1. Navigate to the AWS Lambda console.
2. In the navigation pane, select **Functions**.
3. Select the `fraud_detection_event_processor` Lambda function.
4. Select **Monitoring** and verify that the **Invocations** graph shows activity. You can also click the **View logs in CloudWatch** button to view the endpoint activity.

   After a few minutes, check the results Amazon S3 bucket for processed transactions.

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.

Amazon API Gateway

Amazon API Gateway requires that you authenticate every request you send by signing the request. In the included `generate_endpoint_traffic.py` file, you can find an example of how to sign requests.

To determine the model(s) used to craft the response, the API invocation can include the query string: `?model=`.

The valid values are `anomaly_detector` and `fraud_classifier`. The response will only include the prediction from the Random Cut Forest algorithm when you specify `?model=anomaly_detector` or the XGBoost algorithm when you specify `?model=fraud_classifier`. If the query string is not provided, the response will include the predictions of both models.

Amazon Kinesis Data Firehose

By default, the solution's Amazon Kinesis Data Firehose delivery stream is not encrypted because its destination bucket is encrypted. If you customize the solution to use your own dataset, we recommend encrypting the delivery stream using server-side encryption. For more information, see Using Server-Side Encryption with Amazon Kinesis Data Firehose in the *Amazon Kinesis Data Firehose Developer Guide*.

Additional Resources

- Amazon API Gateway
- AWS Lambda
- Amazon SageMaker
- Amazon Kinesis Data Firehose
- Amazon Simple Storage Service
- Amazon QuickSight
- AWS CloudFormation

Appendix A: Data Visualization

You can visualize the transactions this solution processes using Amazon QuickSight or your own visualization tools. Use the following procedure to configure Amazon QuickSight.

1. Open a text editor and copy the following code.

```
{
   "fileLocations": [
       {
                "URIPrefixes": [
                        "https://s3-us-east-1.amazonaws.com/bucket-
name/"
                ]
        }
   ],
   "globalUploadSettings": {
           "format": "CSV",
           "delimiter": ",",
           "textqualifier": "'",
           "containsHeader": "false"
   }
}
```

2. Modify the `URIPrefixes` URL.
   - Modify the region name to match the region where you deployed the solution, if necessary.
   - Modify the Amazon S3 bucket name to match the name you specified for **Results Bucket Name** AWS CloudFormation template parameter during deployment.
3. Save the following code as `manifest.json`.
4. Navigate to the Amazon QuickSight console.
5. Select **Manage Data**.
6. Select **New data set**.

7. Select **S3**.
8. For **Data source name**, enter a name. For example, `fraud_detection_events`.
9. For **Upload a manifest file**, select the **Upload** radio button and click the folder icon. Navigate to the `manifest.json` file you saved earlier.
10. Select **Connect**.

    After a few minutes, a success message should appear that shows that the S3 data imported into the Amazon QuickSight in-memory calculation engine (SPICE). SPICE acts as a cache for the data stored in your data source. For more information, see SPICE in the *Amazon QuickSight User Guide*.
11. When the **Finish data set creation** window appears, select **Edit/Preview data**.
12. On the **Edit** page, change the column headers to more meaningful names. For example, `Timestamp`, `ID`, `Source`, and `Fraud`.

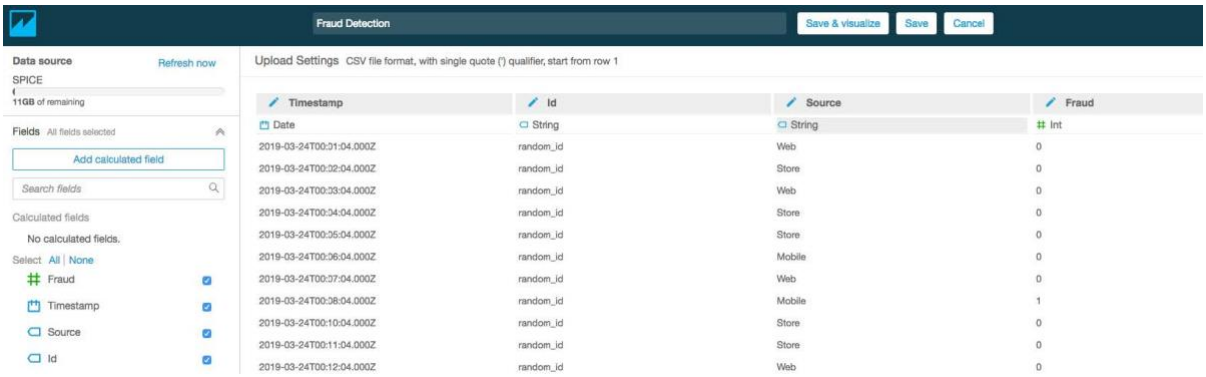

Figure 2: Amazon QuickSight SPICE dataset

13. Select **Save & Visualize**.

    Now, you can create graphical representations of your data. For more information, see Working with Amazon QuickSight Visuals in the *Amazon QuickSight User Guide*.

Appendix B: Acknowledgements

Fraud Detection Using Machine Learning contains a publicly available anonymized credit card transaction dataset that was collected and analyzed during a research collaboration of Worldline and the Machine Learning Group of Université Libre de Bruxelles on big data mining and fraud detection. For more details on current and past fraud detection projects, see ResearchGate and DEFEATFRAUD.

- Dal Pozzolo, Andrea, Caelen, Olivier, Johnson, Ried A., & Bontempi, Gianluca (2015) Calibrating probability with undersampling for unbalanced classification. In *Symposium on Computational Intelligence and Data Mining (CIDM)*, IEEE
- Dal Pozzolo, Andrea, Caelen, Olivier, Le Borgne, Yann-Aël, Waterschoot, Serge, & Bontempi, Gianluca (2014). Learned lessons in credit card fraud detection from a practitioner perspective. *Expert Systems With Applications*, *41*(10), 4915-4928, Pergamon
- Dal Pozzolo, Andrea, Boracchi, Giacomo, Caelen, Olivier, Alippi, Cesare, & Bontempi, Gianluca (2018). Credit card fraud detection: A realistic modeling and a novel learning strategy. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(8), 3784-3797, IEEE
- Dal Pozzolo, Andrea. Adaptive machine learning for credit card fraud detection. ULB MLG PhD thesis (supervised by Bontempi, Gianluca)
- Carcillo, Fabrizio, Dal Pozzolo, Andrea, Le Borgne, Yann-Aël, Caelen, Olivier, Masser, Yannis, & Bontempi, Gianluca (2018). Scarff: A scalable framework for streaming credit card fraud detection with Spark. *Information Fusion*, *41*, 182-194, Elsevier
- Carcillo, Fabrizio, Le Borgne, Yann-Aël, Caelen, Olivier, & Bontempi, Gianluca (2018). Streaming active learning strategies for real-life credit card fraud detection: Assessment and visualization. *International Journal of Data Science and Analytics*, *5*(4), 285-300, Springer International Publishing

Appendix C: Collection of Operational Metrics

This solution includes an option to send anonymous operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. When enabled, the following information is collected and sent to AWS:

- **Solution ID:** The AWS solution identifier

- **Unique ID (UUID):** Randomly generated, unique identifier for each Fraud Detection Using Machine Learning deployment
- **Timestamp:** Data-collection timestamp

Note that AWS will own the data gathered via this survey. Data collection will be subject to the AWS Privacy Policy. To opt out of this feature, modify the AWS CloudFormation template mapping section as follows:

```
"Send" : {
"AnonymousUsage" : { "Data" : "Yes" }
},
```

to

```
"Send" : {
"AnonymousUsage" : { "Data" : "No" }
},
```

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
May 2019	Initial publication
May 2020	Changed the supervised model to XGBoost. Added unsupervised learning with Random Cut Forest, SMOTE resampling, model evaluation, and REST API for predictions. See the list of v2 updates on our GitHub repository.