Random Cut Forest (RCF)

Random Cut Forest

- Unsupervised algorithm to detect outliers or anomalous data points
- Tree based ensemble method
- Support for Timeseries data
- Assigns an anomaly score for each data point

RCF Uses

Traffic spike due to rush hour or accident

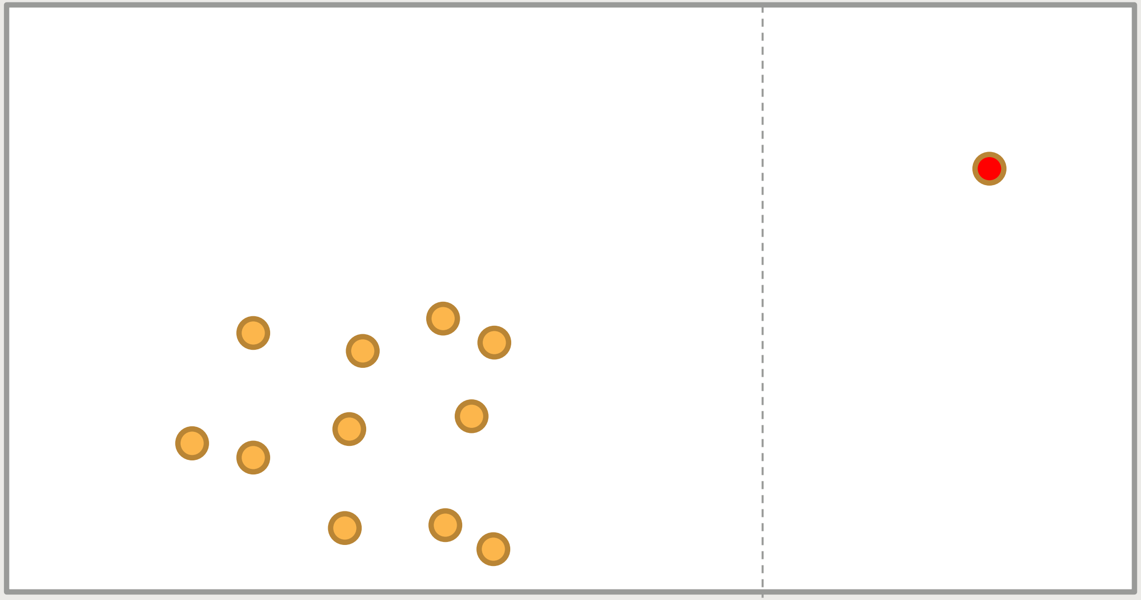
DDoS attack detection

Unauthorized data transfer detection

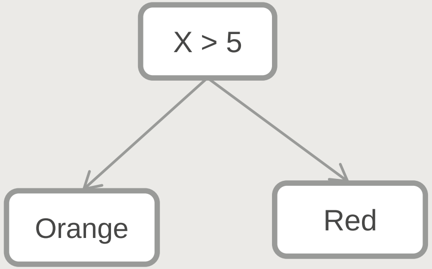
Intuition

Using Isolation Forest

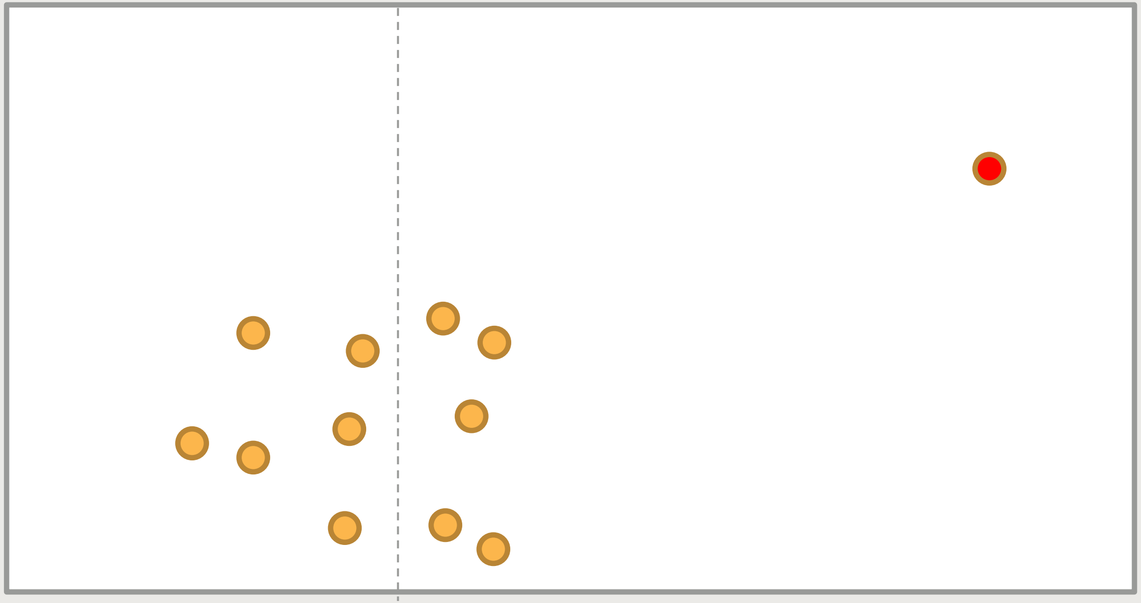
Tree based Classifier



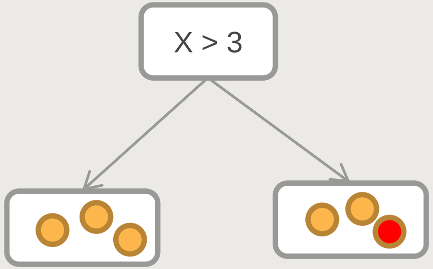
Supervised



Anomaly Detection – Random Cut



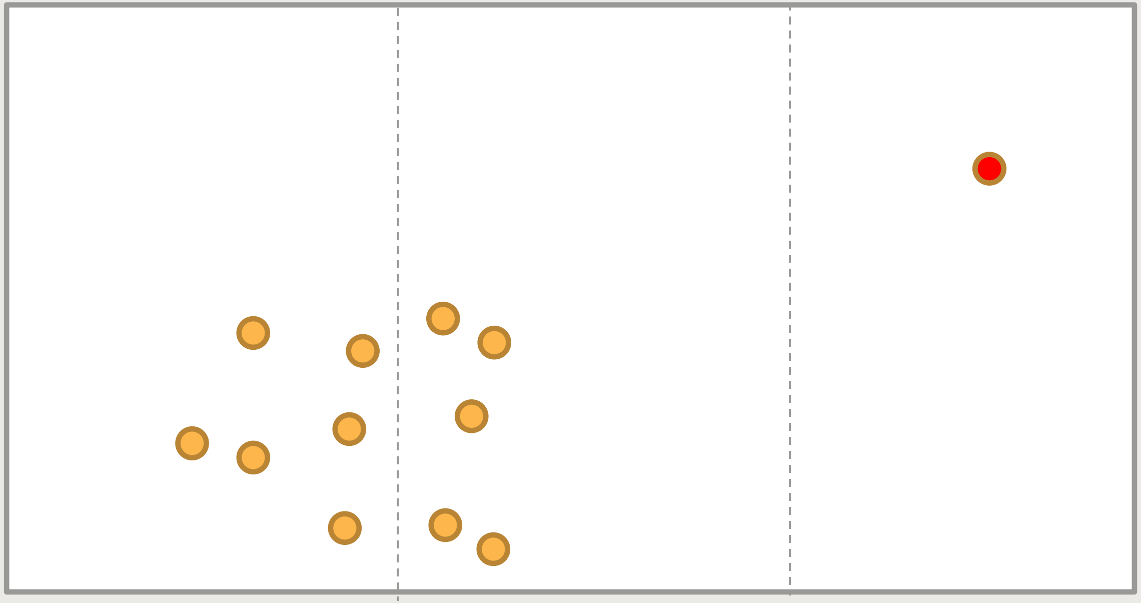
Unsupervised – Explain the data



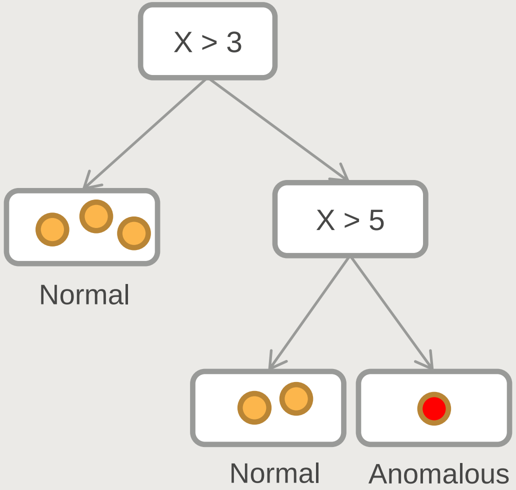
Normal points

Normal and anomalous points

Anomaly Detection – Random Cut



Unsupervised – Explain the data



X > 3

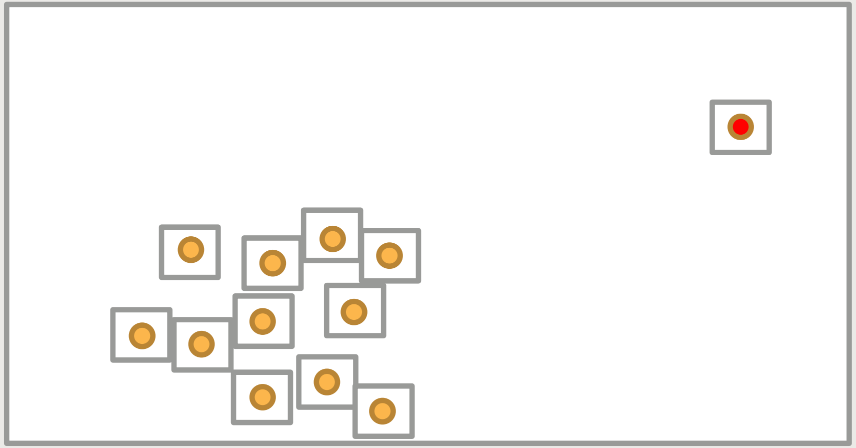
X > 5

Normal

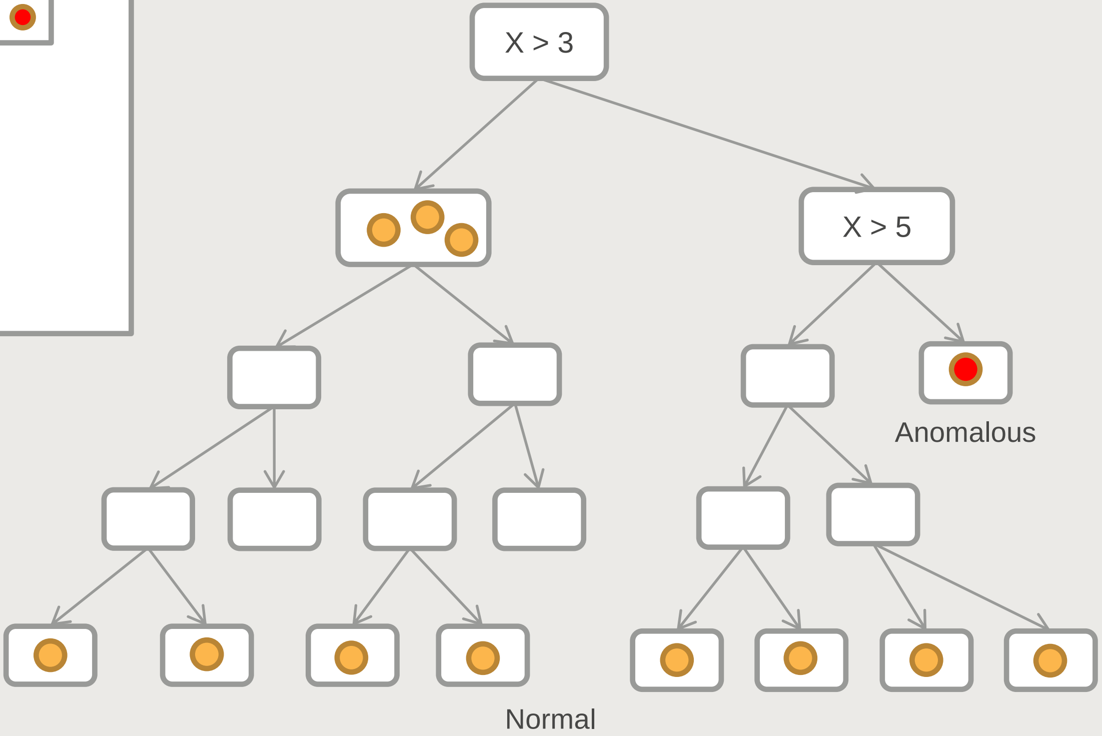
Normal

Anomalous

Anomalous points are closer to root (depth)



Unsupervised – Explain every single point



Normal data points require a lot of splits

Anomalous points require fewer splits

Each point is assigned an anomaly score based on depth
Normal points = low score
Anomalous points = high score

Anomaly Detection with Isolation Forest

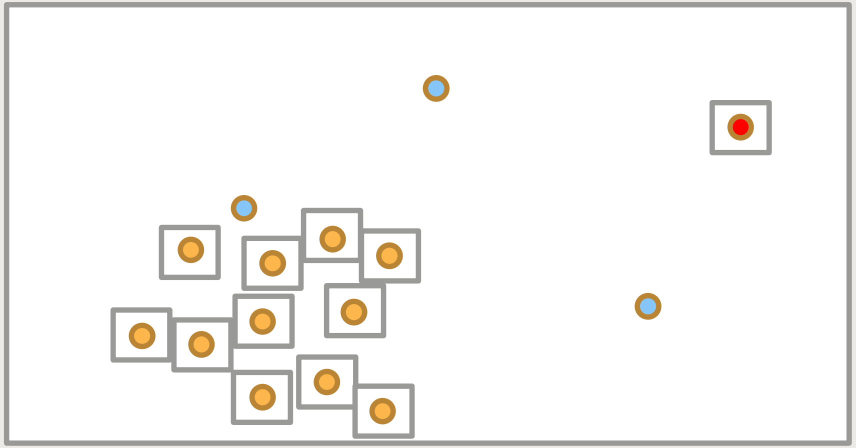
“*if I have a set of data points along the line and I choose an arbitrary split, there is going to be empty spaces between adjacent instances. And the algorithm allocates two additional patterns for that empty space*”

Dr. Thomas Dietterich

Anomaly Detection: Algorithms, Explanations, Applications | Microsoft Research

https://youtu.be/12Xq9OLdQwQ (40:00)

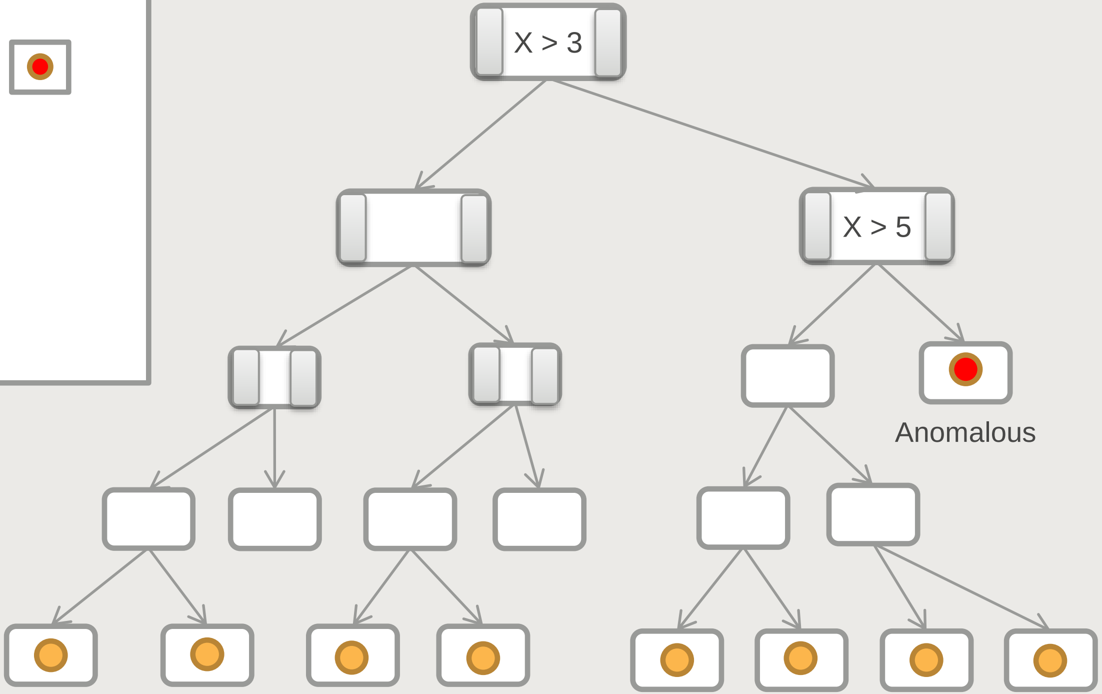
What about new data points?



Tree covers the gaps!

We need to define a threshold to classify if a score is anomaly or normal

Additional patterns to cover for adjacent gaps



Useful resources – Isolation Forest

These resources helped me gain insight into Random Cut Forest (you don’t have to watch it; this is an acknowledgement of people who helped me)

- Elena Sharova: Unsupervised Anomaly Detection with Isolation Forest | PyData 2018 - https://youtu.be/5p8B2Ikcw-k
- Jan van der Vegt: A walk through the isolation forest | PyData 2019 https://youtu.be/RyFQXQf4w4w
- Dr. Thomas Dietterich - Anomaly Detection: Algorithms, Explanations, Applications | Microsoft Research 2018 https://youtu.be/12Xq9OLdQwQ

Random Cut Forest

- Build several trees (forest)
- Each tree is a given several random sample of instances drawn from the training dataset
- RCF uses reservoir sampling to draw random samples from large dataset
    - Works efficiently when size of the data set is too large to fit in memory
    - Or when we don’t know the training set size
- Final Anomaly score = Average of anomaly scores of all trees

Random Cut Forest Prediction

RCF predicts an anomaly score for the data point

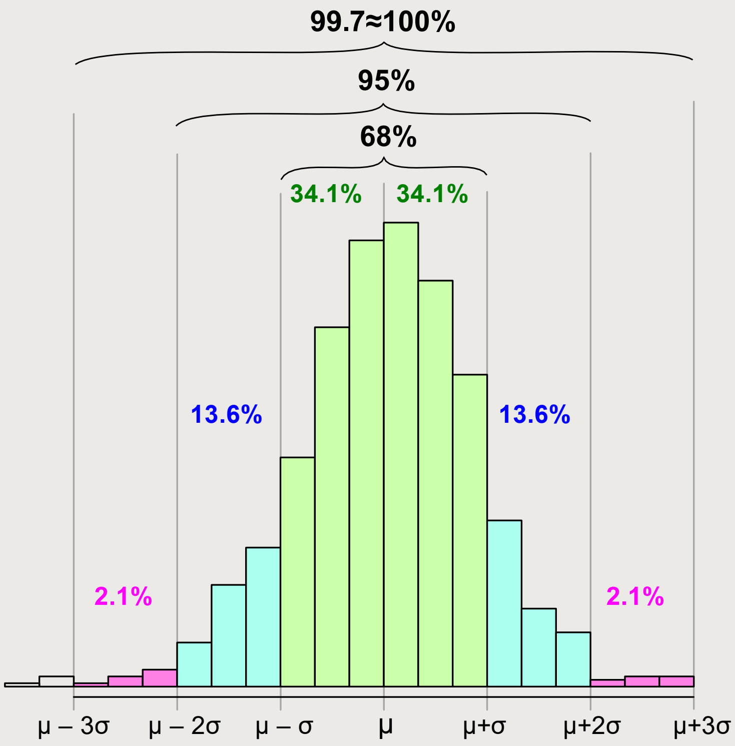
- Score varies inversely with depth
- Low Score is considered “normal”
- High Score indicates “anomaly”

Definition of Low and High score depends on application

Common practice: Scores beyond three standard deviations from mean score are considered anomalous

Reference: https://docs.aws.amazon.com/sagemaker/latest/dg/randomcutforest.html

Distribution (68-95-99.7 rule)



"For approximately normal dataset, 99.7% of the datapoints fall within three standard deviations from mean"

By Melikamp - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=65001875
https://en.wikipedia.org/wiki/68%E2%80%9395%E2%80%9399.7_rule

RCF Supported Data Formats

- Training, Test channels - CSV, RecordIO
- Test – Optional. First column in each row represents the anomaly label
  - “1” – anomalous data point
  - “0” – normal data point
  - RCF computes accuracy, precision, recall, F1-score for test data
- Inference format: JSON, CSV, RecordIO

RCF Hyperparameters

Hyperparameter	Description
feature_dim	Number of features in the data set. SageMaker RCF Estimator automatically computes this
eval_metrics	Test data evaluation metrics. Default: accuracy, precision, recall, f1 score
num_trees	Number of trees in the forest
num_samples_per_tree	Number of random samples given to each tree from the training set

num_trees, num_samples_per_tree are tunable parameters using automatic hyperparameter tuning

Lab – Taxi Passenger (AWS Example)

Analyze anomalies in NY Taxi usage timeseries data

Optimization Techniques: Shingling, number of trees, sample size, cutoff for anomaly score

Measuring performance: Labeled Test Data (binary classification)

https://github.com/awslabs/amazon-sagemaker-examples/blob/master/introduction_to_amazon_algorithms/random_cut_forest/random_cut_forest.ipynb

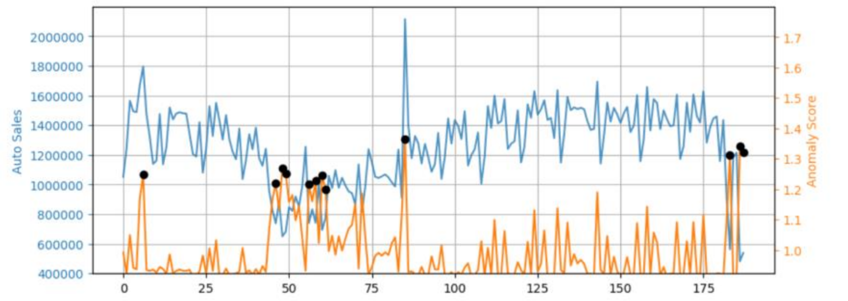
Lab – Auto Sales Analysis

- Analyze 15 years of monthly auto sales in the USA
- Verify how RCF Score varies for change in volume:
    - Housing Crisis
    - Recovery
    - COVID
- Data Source:

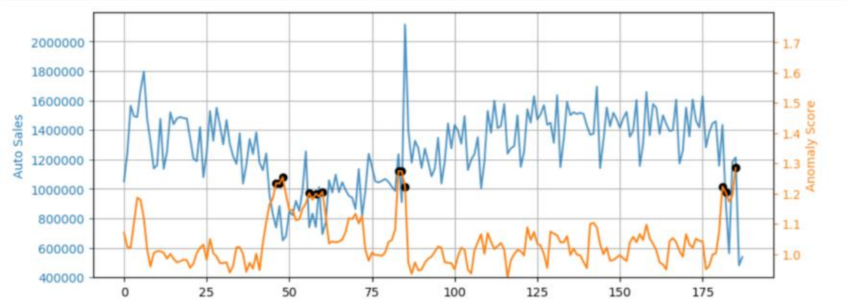
  https://www.goodcarbadcar.net/usa-auto-industry-total-sales-figures/,
  http://www.bea.gov/
- Shingle sizes: 1-month, 3-months, 12-months

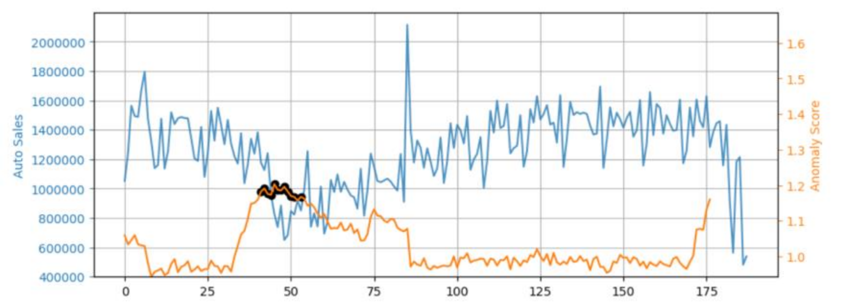
Auto Sales – RCF Anomaly Scores

Shingle size = 1

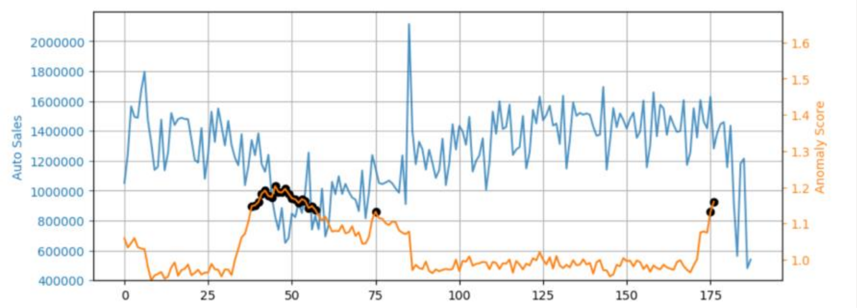


Shingle size = 3





Shingle size = 12. Cutoff = 2 SD



Shingle size = 12. Cutoff = 1.5 SD

Chandra Lingam

57,000+ Students

For AWS self-paced video courses, visit:

https://www.cloudwavetraining.com/

Cloud Wave LLC