AWS Logs Anomaly Detection

MSML 651 Project
Lohit Aryan Gopikonda
lgopik2@umd.edu

Contents

- Motivation and Dataset
- Hypothesis
- Predictions and Features
- Model and Algorithms
- Results

Motivation and Dataset

- Proactively detect abnormal patterns in AWS logs to enhance system reliability and reduce manual monitoring effort.
- Project goal: Develop an unsupervised anomaly detection model to spot abnormal patterns.
- Anomalies targeted: Potential security breaches, OS level issues and performance degradation.
- Dataset used:
  - Generated EC2 cloudwatch monitoring log
  - Anomalies were placed through trying to crash the system such as killing processes, simulating high resource loads and network calls.

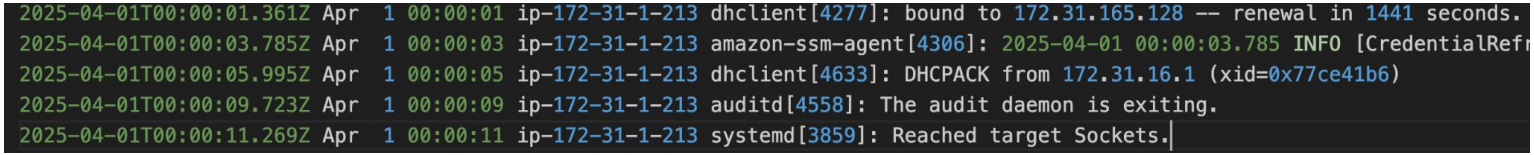
Hypothesis

The hypothesis is that detecting the information or patterns that are clear and interpretable enough to provide insights to characterize the normal operation and any deviation from these patterns can be flagged as anomalies.

- Enhanced Security
- Operational Efficiency
- Improved System Reliability
- Scalability

Predictions and Features

- Identify unusual AWS log entries that deviate from normal operational patterns
- Accurately flag outlier logs to enable rapid response to threats or operational issues
- Features: (OS level text tokens)



Logs have timestamp,ip, service and message

Pre Processing and Cleaning

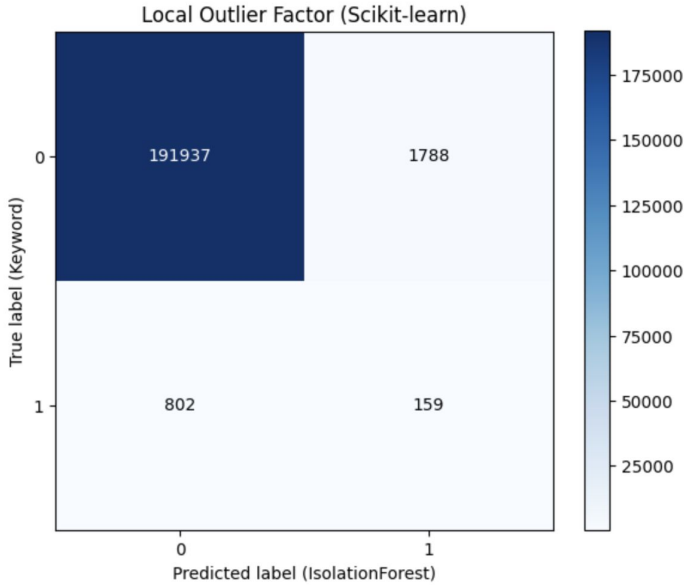
- Remove timestamp
- Remove stopwords and unwanted tokens from log message
- encode tokens with Tf-idf using spark's HashingTF for efficiency
- Using spark ML Library (Pipeline, HashingTF, PCA) for efficiency and parallelism
- Use mmlspark's Isolation forest to distribute the model training and testing

Model and Algorithms

- Local Outlier Factor - Density based anomaly detection algorithm.
  - Detects anomalies by comparing a point's local density to that of its neighbors (LOF > 1 ⇒ outlier)
  - Captures local deviations but sensitive to the choice of neighborhood size (k)
- Isolation Forest - It is an ensemble method fit for detecting anomalies in the high dimensional data such as logs.
  - Randomly partitions data: anomalies require fewer splits to isolate (shorter path lengths)
  - Scales efficiently to high-dimensional, large-scale datasets without computing distances

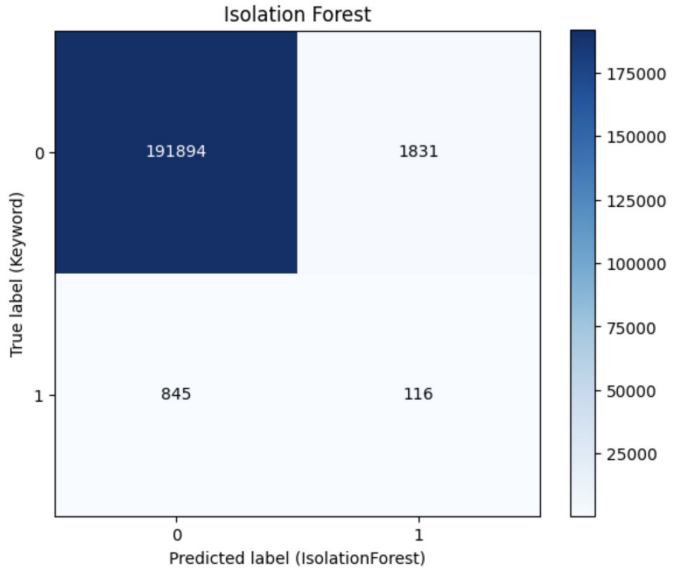
Results - Training(180k), Testing(20k)

Local Outlier Factor (Accuracy: 98.6% F1: 0.109)



Results

Isolation Forest (Accuracy: 98.7%, F1: 0.080)



Future Improvements

- Hyper parameter tuning
- Using different encoders such as Word2vec
- K-fold cross validation
- Incorporate Autoencoders