BIG DATA ANALYTICS

Final presentation
25.02.2019

TU Berlin

Agenda

1 Use case presentation

2 System architecture

3 Game plan & Challenges

4 Data preprocessing

5 Anomaly detection models explanation and comparison

6 Prediction models explanation and comparison

7 Notification functions

8 Recap and Recommandations

8 Mio.

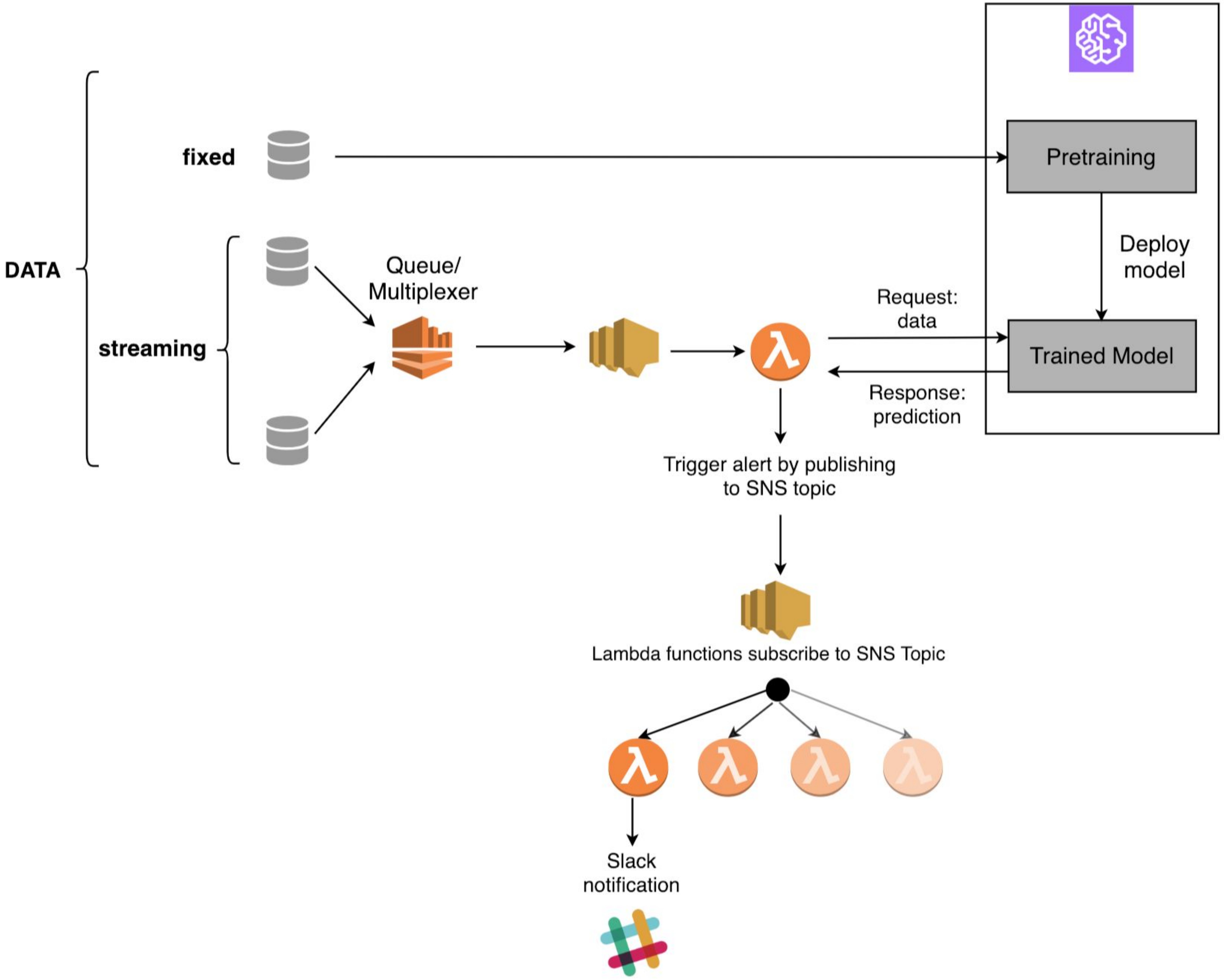
ConnectedDrive Cars

Use Case Presentation

Anomaly detection on time series data (requests per second)

10/22 00:00
10/22 08:00
10/22 16:00
10/23 00:00
10/23 08:00
10/23 16:00
10/24 00:00
10/24 08:00
10/24 16:00
MULTI-AZ-3: GTM per Second Max: 516 Avg: 188
CNGW-MULTI-AZ-2: GTM per Second Max: 2 Avg: 0

System architecture



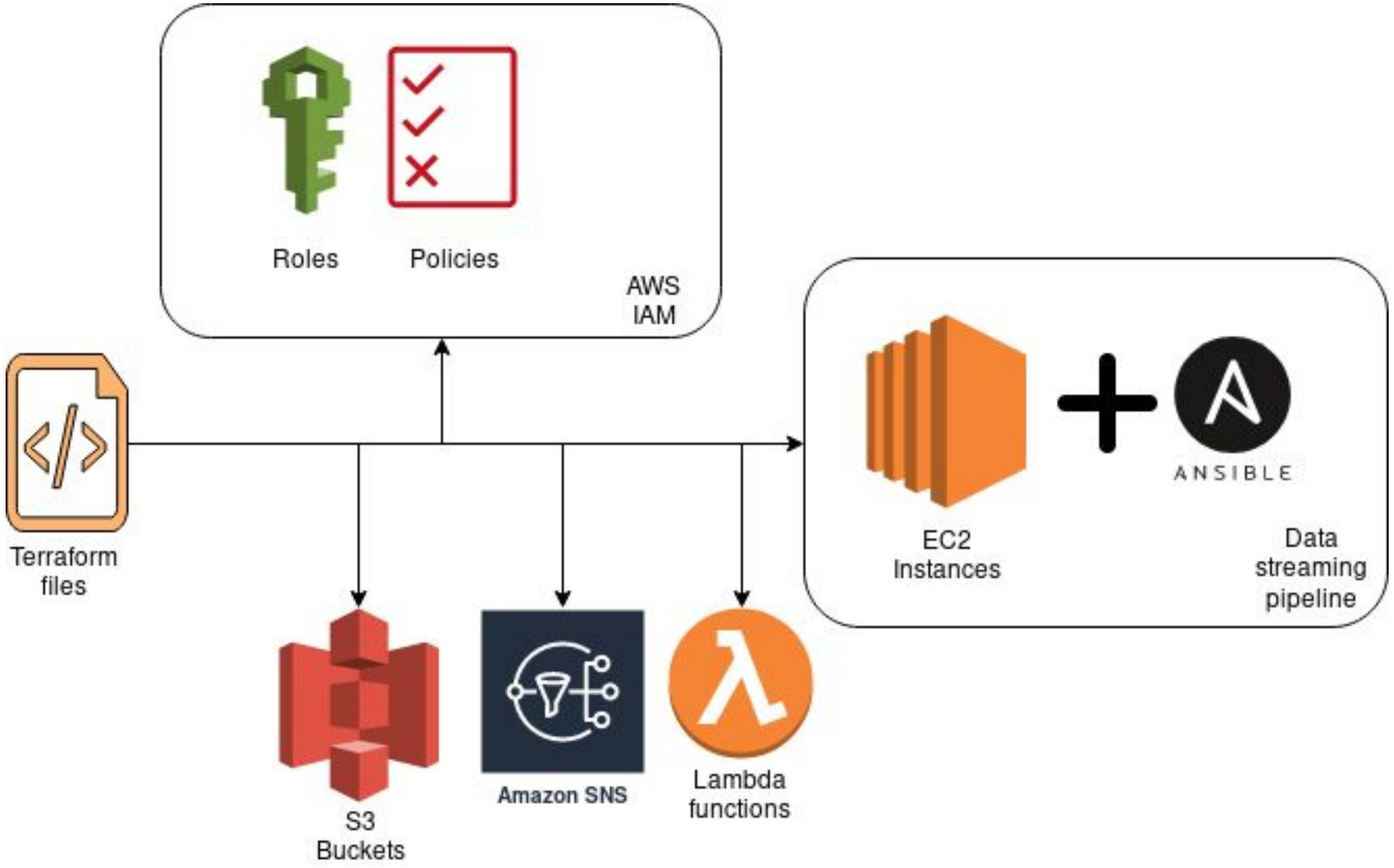
Deployment

Infrastructure managed with Terraform

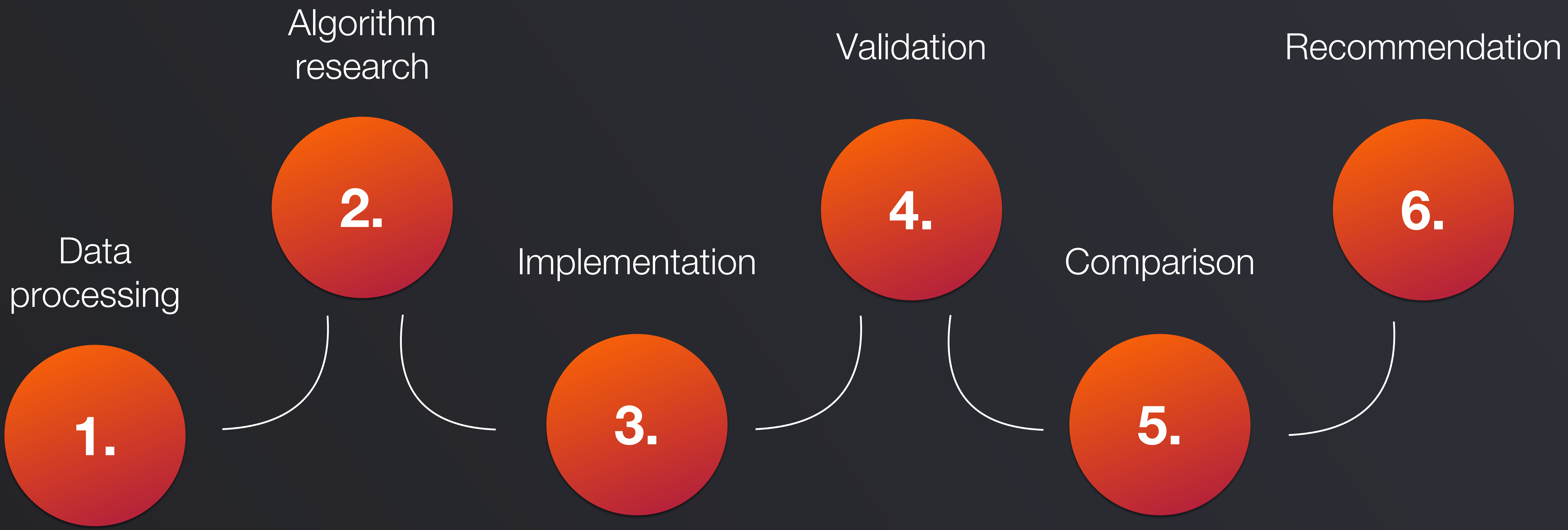
Provisioning with Ansible

Goal: get rid of most of the setup complexity

BUT some features are not supported yet



Game plan



1. Data processing

2. Algorithm research

3. Implementation

4. Validation

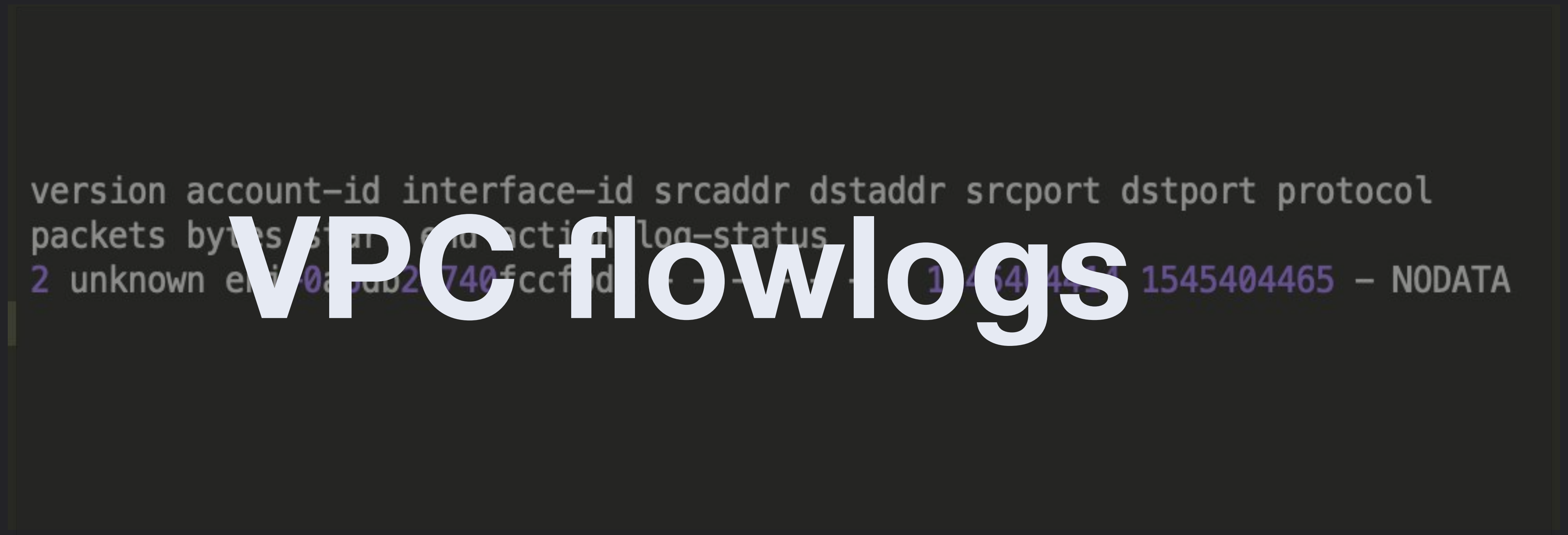
5. Comparison

6. Recommendation

Challenges	Solutions
1. Understanding and preprocessing data	1. Create 1 min and 5 min granularity buckets
2. Faking a streaming data pipeline	2. Feeding portions of data once every 15 min
3. Lack of extensive AWS knowledge	3. Organizing knowledge sharing sessions
4. Exploring Machine Intelligence solutions	4. Trying and comparing a couple algorithms
5. Small amount of data	5. Generating new data with our prediction models

Data processing

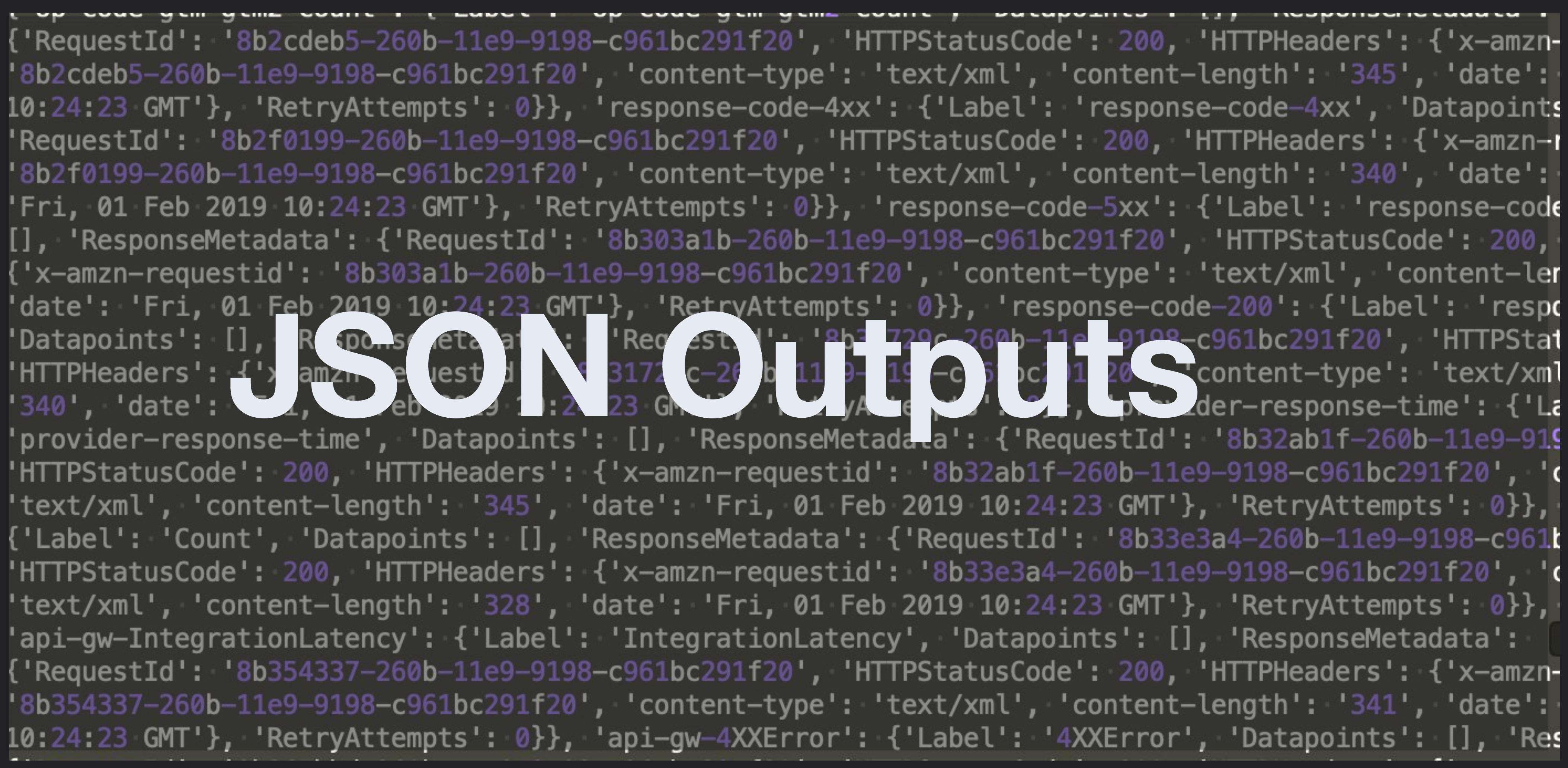
VPC flowlogs



```
version account-id interface-id srcaddr dstaddr srcport dstport protocol
packets bytes start end action log-status
2 unknown eni-0a[illegible]db2[illegible]740fccf0d - - - - - - - [illegible] 1545404465 - NODATA
```

1) Aggregate logs into **1 min** buckets

JSON Outputs

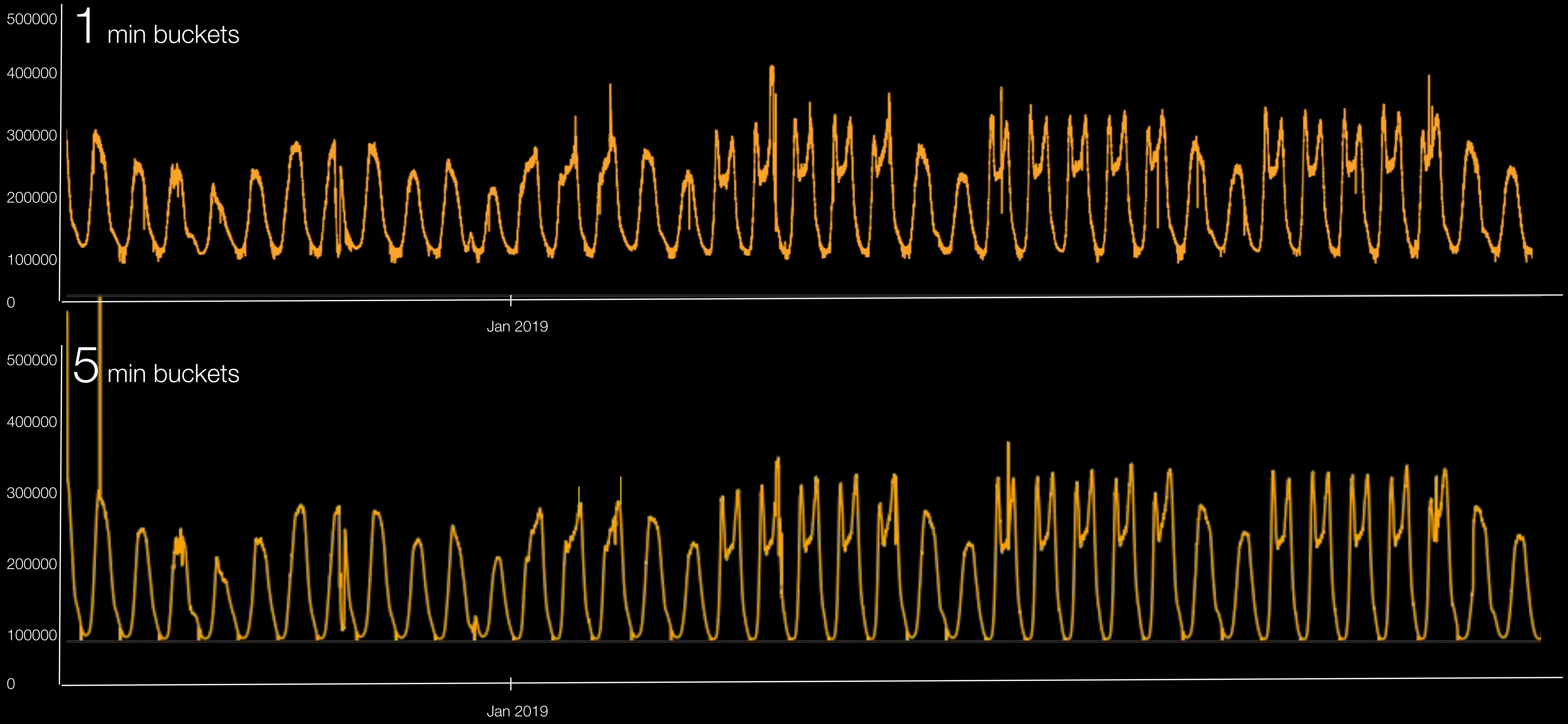


```
{'RequestId': '8b2cdeb5-260b-11e9-9198-c961bc291f20', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-
'8b2cdeb5-260b-11e9-9198-c961bc291f20', 'content-type': 'text/xml', 'content-length': '345', 'date':
10:24:23 GMT'}, 'RetryAttempts': 0}}, 'response-code-4xx': {'Label': 'response-code-4xx', 'Datapoints
'RequestId': '8b2f0199-260b-11e9-9198-c961bc291f20', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-r
'8b2f0199-260b-11e9-9198-c961bc291f20', 'content-type': 'text/xml', 'content-length': '340', 'date':
'Fri, 01 Feb 2019 10:24:23 GMT'}, 'RetryAttempts': 0}}, 'response-code-5xx': {'Label': 'response-code
[], 'ResponseMetadata': {'RequestId': '8b303a1b-260b-11e9-9198-c961bc291f20', 'HTTPStatusCode': 200,
{'x-amzn-requestid': '8b303a1b-260b-11e9-9198-c961bc291f20', 'content-type': 'text/xml', 'content-len
'date': 'Fri, 01 Feb 2019 10:24:23 GMT'}, 'RetryAttempts': 0}}, 'response-code-200': {'Label': 'respo
'Datapoints': [], 'ResponseMetadata': {'RequestId': '8b[illegible]-c961bc291f20', 'HTTPStat
'HTTPHeaders': {'x-amzn-requestid': '8b3172[illegible]', 'content-type': 'text/xml
'340', 'date': [illegible] 23 GMT'}, 'RetryAttempts': 0}}, 'provider-response-time': {'La
'provider-response-time', 'Datapoints': [], 'ResponseMetadata': {'RequestId': '8b32ab1f-260b-11e9-919
'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid': '8b32ab1f-260b-11e9-9198-c961bc291f20', 'c
'text/xml', 'content-length': '345', 'date': 'Fri, 01 Feb 2019 10:24:23 GMT'}, 'RetryAttempts': 0}},
{'Label': 'Count', 'Datapoints': [], 'ResponseMetadata': {'RequestId': '8b33e3a4-260b-11e9-9198-c961b
'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid': '8b33e3a4-260b-11e9-9198-c961bc291f20', 'c
'text/xml', 'content-length': '328', 'date': 'Fri, 01 Feb 2019 10:24:23 GMT'}, 'RetryAttempts': 0}},
'api-gw-IntegrationLatency': {'Label': 'IntegrationLatency', 'Datapoints': [], 'ResponseMetadata':
{'RequestId': '8b354337-260b-11e9-9198-c961bc291f20', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-
'8b354337-260b-11e9-9198-c961bc291f20', 'content-type': 'text/xml', 'content-length': '341', 'date':
10:24:23 GMT'}, 'RetryAttempts': 0}}, 'api-gw-4XXError': {'Label': '4XXError', 'Datapoints': [], 'Res
```

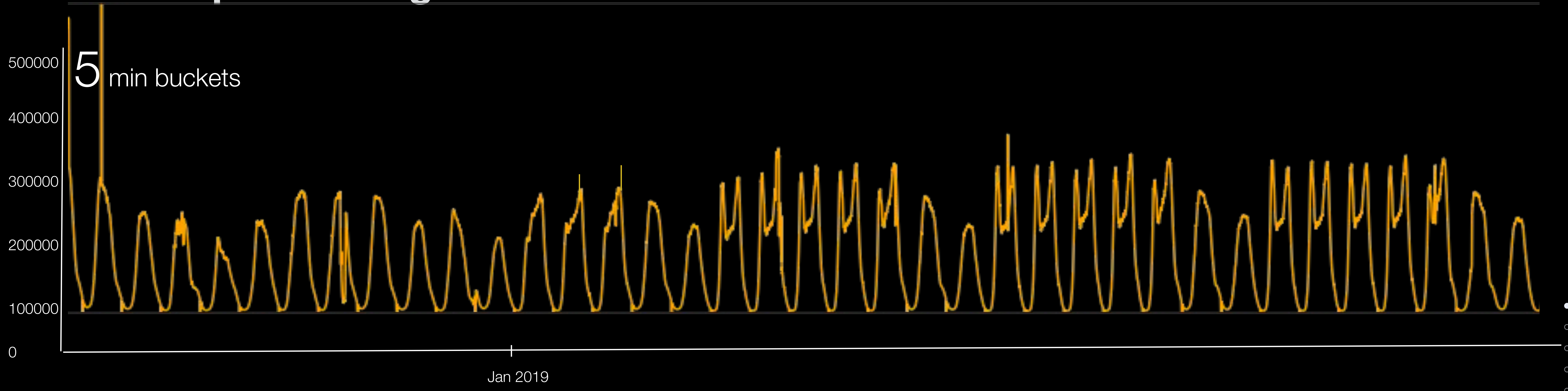
1) Extract the **Datetime & Sum**

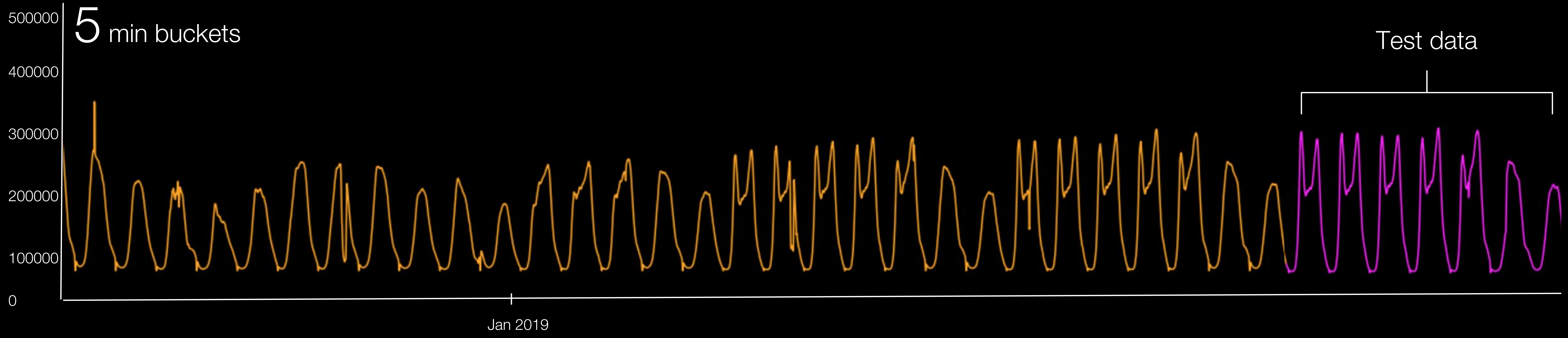
2) Aggregate results into **5 min** buckets

◦◦◦ Data visualisation



Data processing

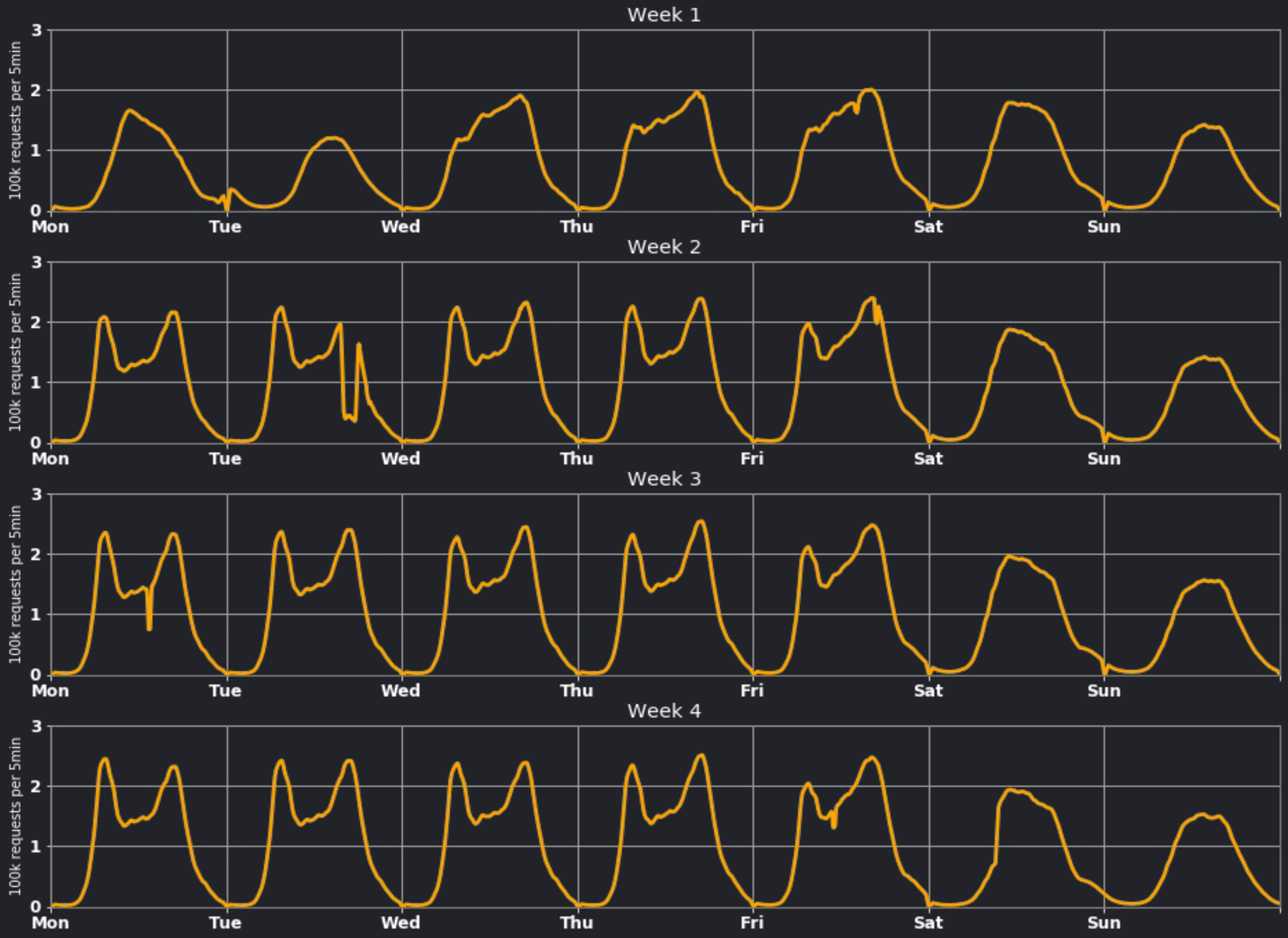




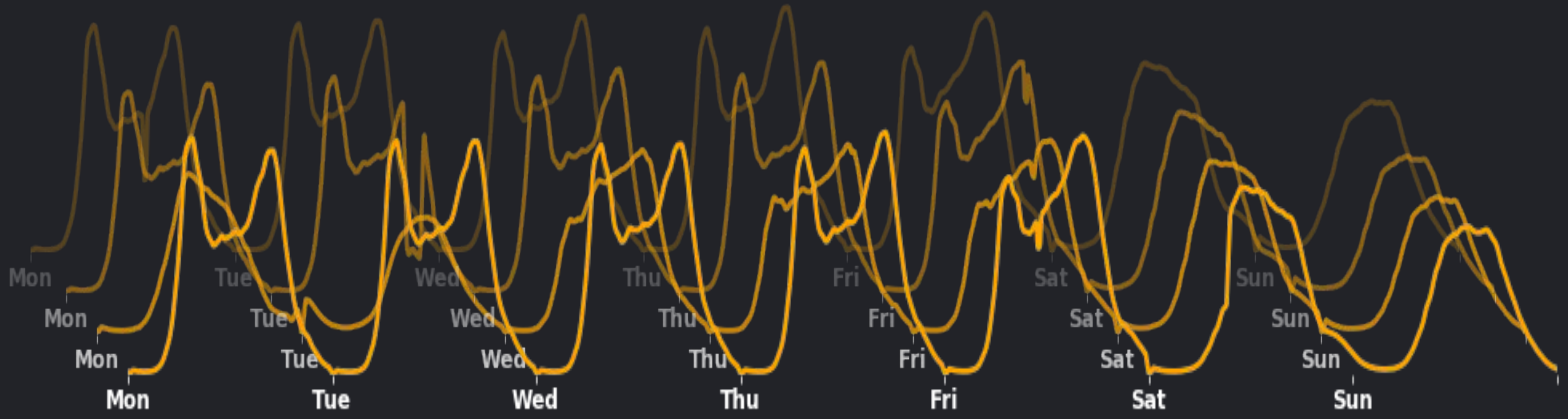
Anomaly detection models

1 - Mean Predictor

Data visualization



Mean predictor



Anomalies
Normal
Expected Value
Standard Deviation
Mon
Tue
Wed
Thu
Fri
Sat
Sun

Mean predictor

PROS

Requires very little data to train

Trains very fast

Requires very little computation power

Robust against outliers

Can be used both for forecasting as well as anomaly detection

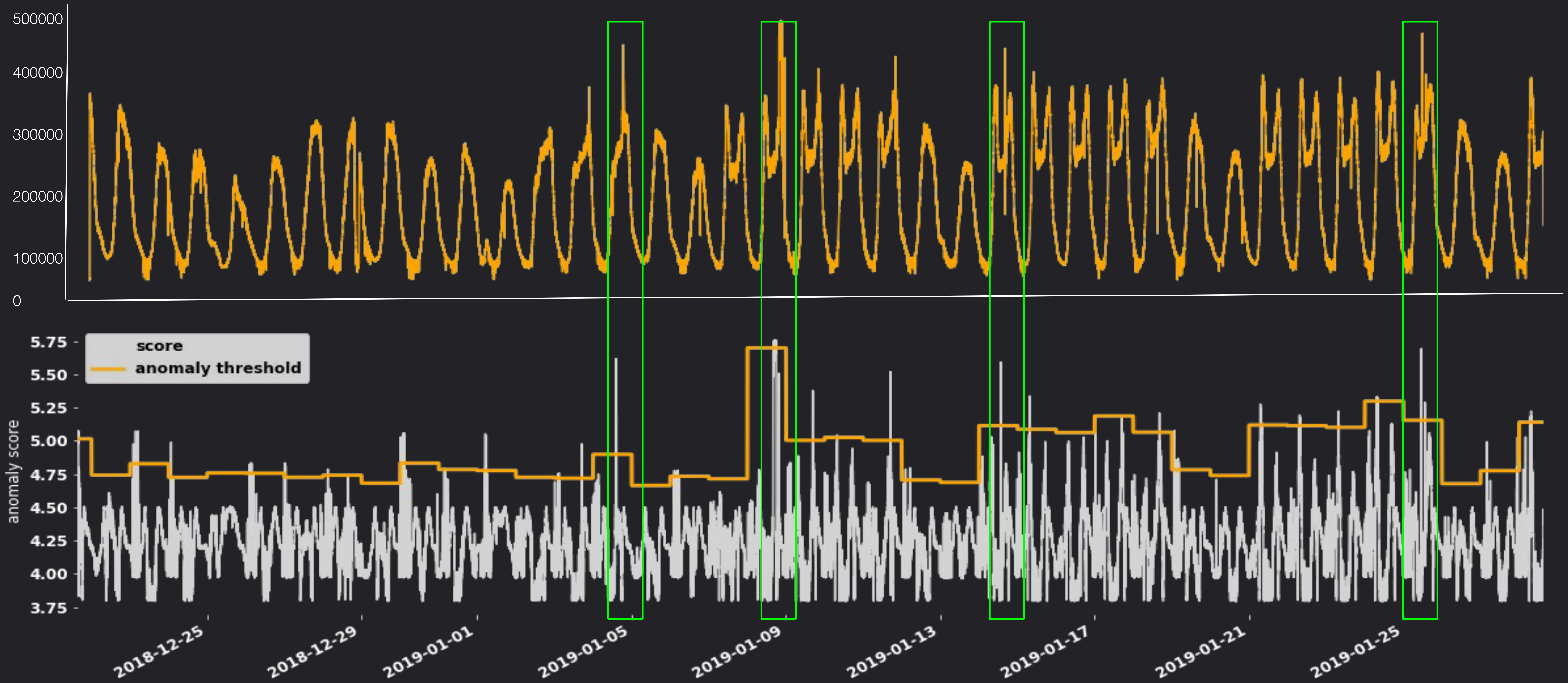
CONS

Can only take into account a single source of information

Not an Out-Of-The-Box solution from AWS

2- Random Cut Forest

Data visualization



Random Cut Forest

PROS

Good performance for long-lived data streams

Short training time

Relatively easy to create and deploy the model (Out-Of-The-Box solution from Amazon)

Has a stream-friendly version

Supports streaming only on kinesis streams

Requires huge amounts of data

CONS

Technical decisions

We have:

- A lack of long-term data (FlowLogs collection started in December)
- A unique time series to analyze

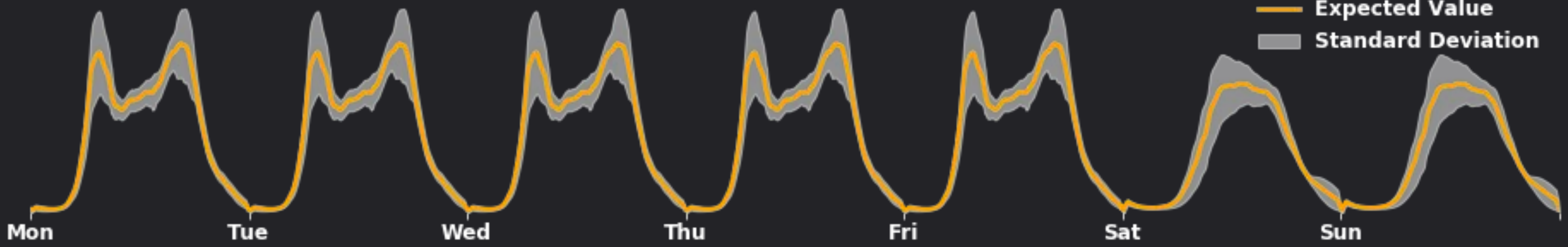
We need:

- A tweakable solution
- Short train times

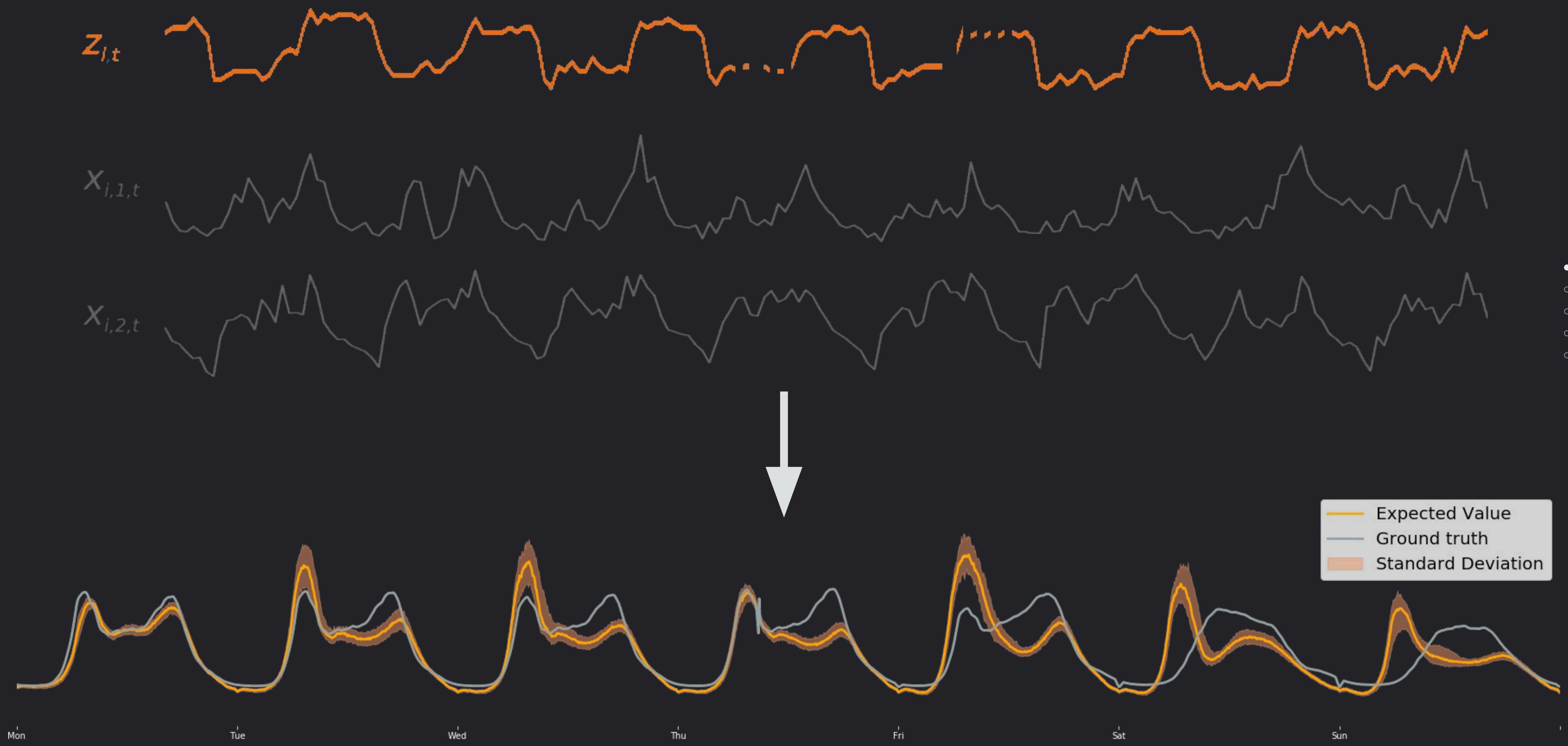
→ Explains why we built our demo pipeline with Mean Predictor

Prediction models

Mean Predictor



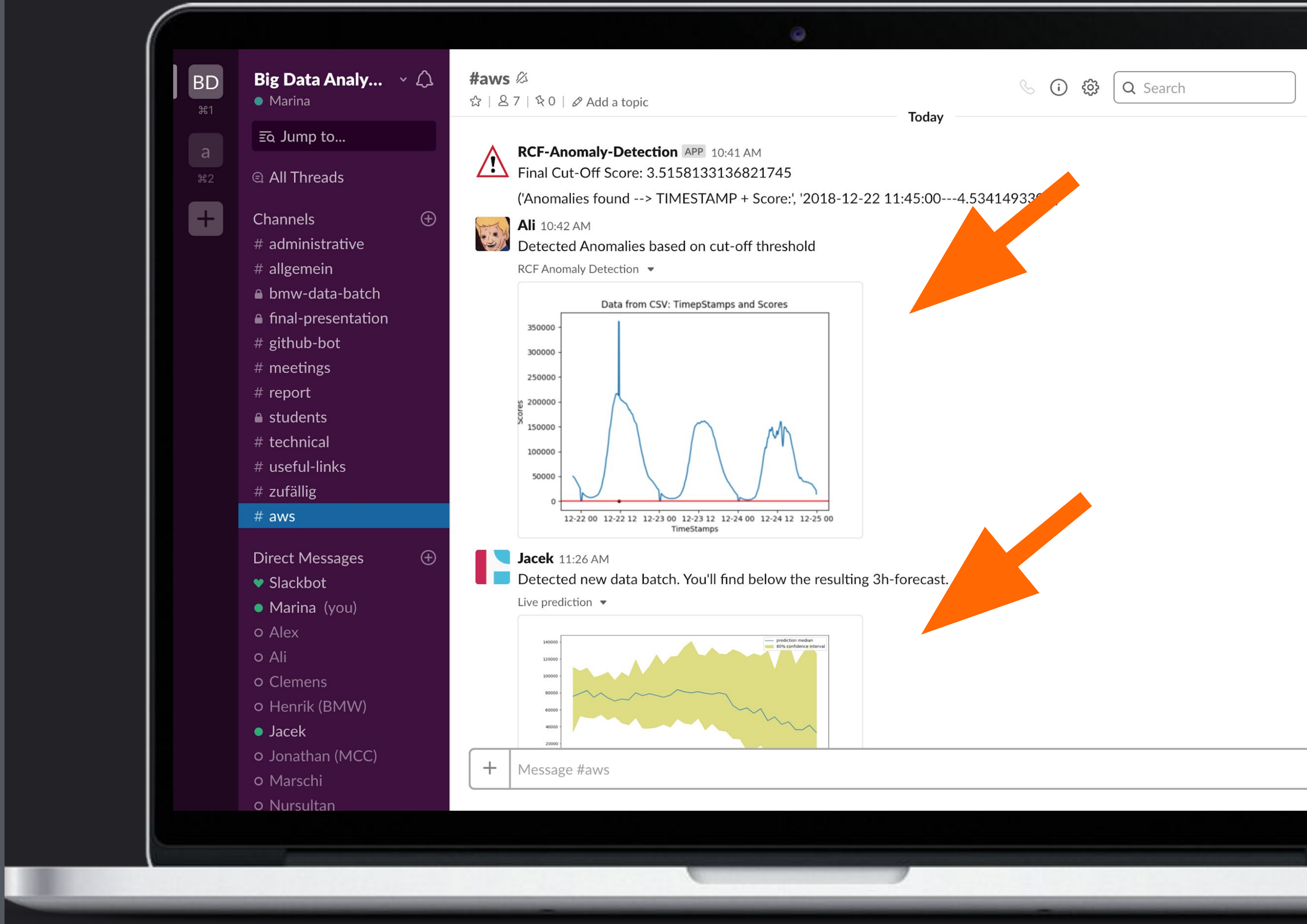
DeepAR


$z_{i,t}$
$x_{i,1,t}$
$x_{i,2,t}$
Expected Value
Ground truth
Standard Deviation
Mon
Tue
Wed
Thu
Fri
Sat
Sun

Notification functions

Anomaly detection notification

3 hour forecast notification



RCF-Anomaly-Detection APP 10:41 AM
Final Cut-Off Score: 3.5158133136821745

('Anomalies found --> TIMESTAMP + Score:', '2018-12-22 11:45:00---4.534149336

Ali 10:42 AM
Detected Anomalies based on cut-off threshold

RCF Anomaly Detection

Jacek 11:26 AM
Detected new data batch. You'll find below the resulting 3h-forecast.

Live prediction

RECOMMENDATIONS

Implement the Mean Predictor Model

Switch to RCF and DeepAR

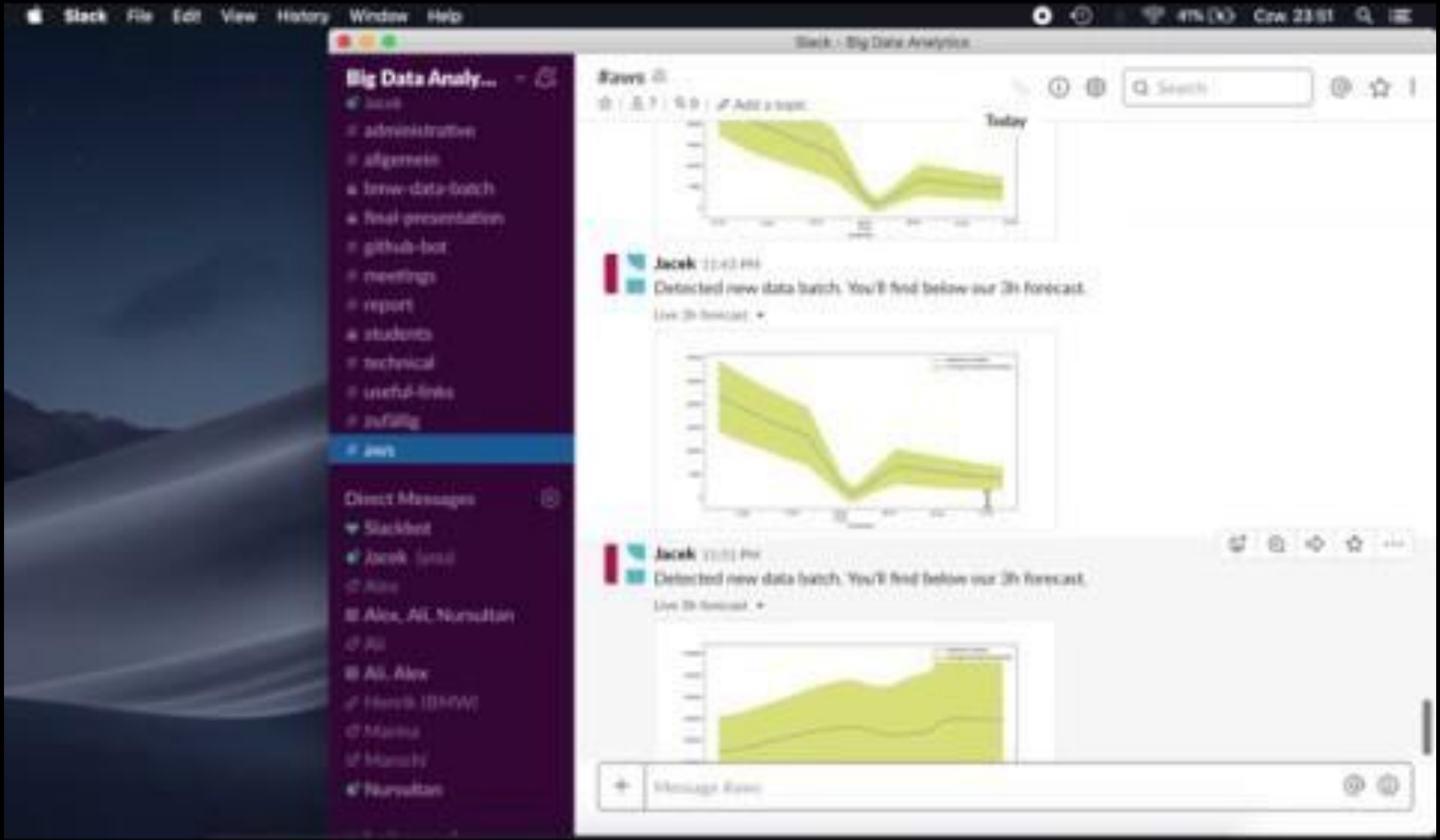
Attach geolocation to data points

Gather 2-3 years of data

Look for seasonal trends

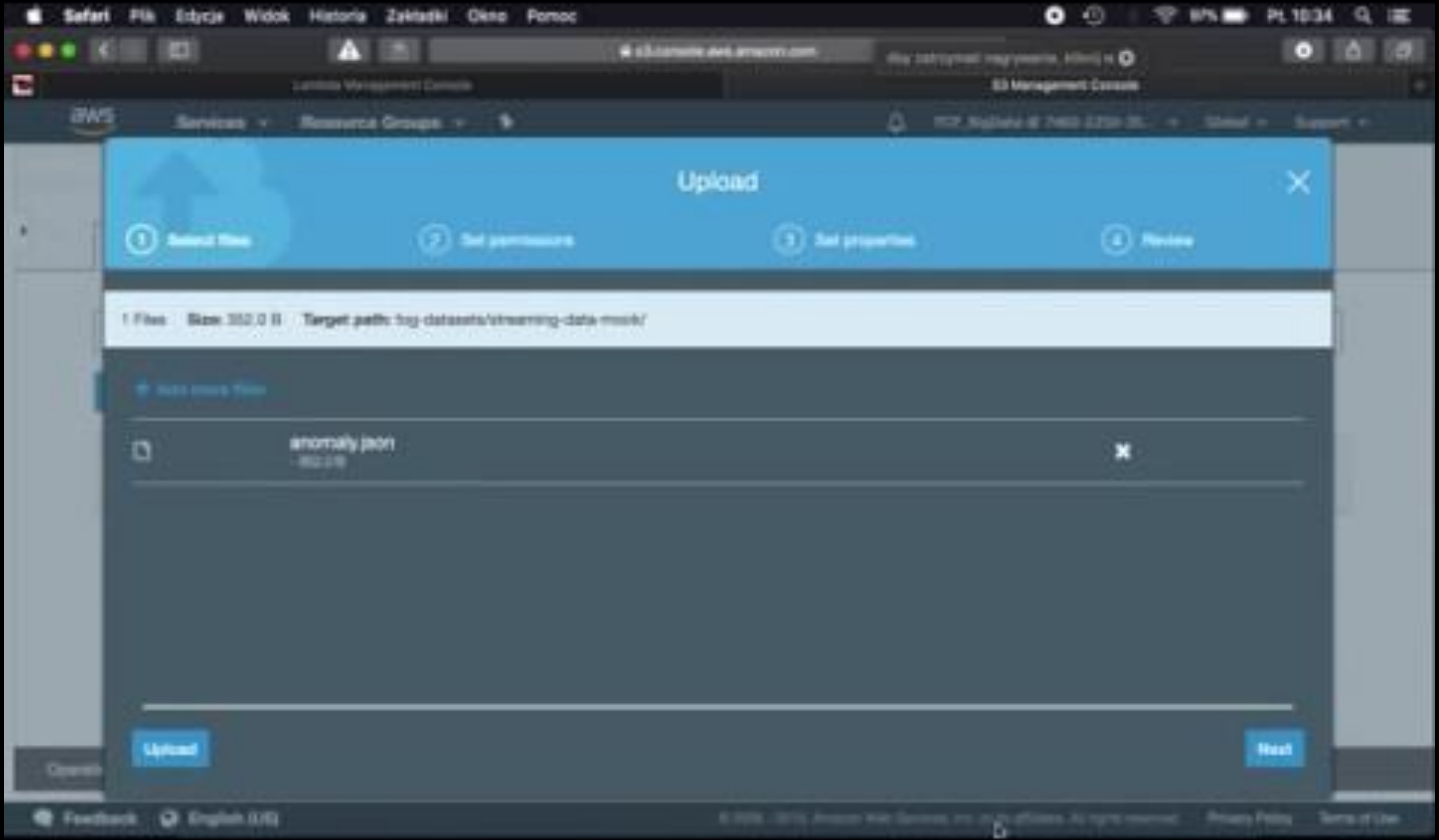
Explore the holidays periods

Thank you for your attention

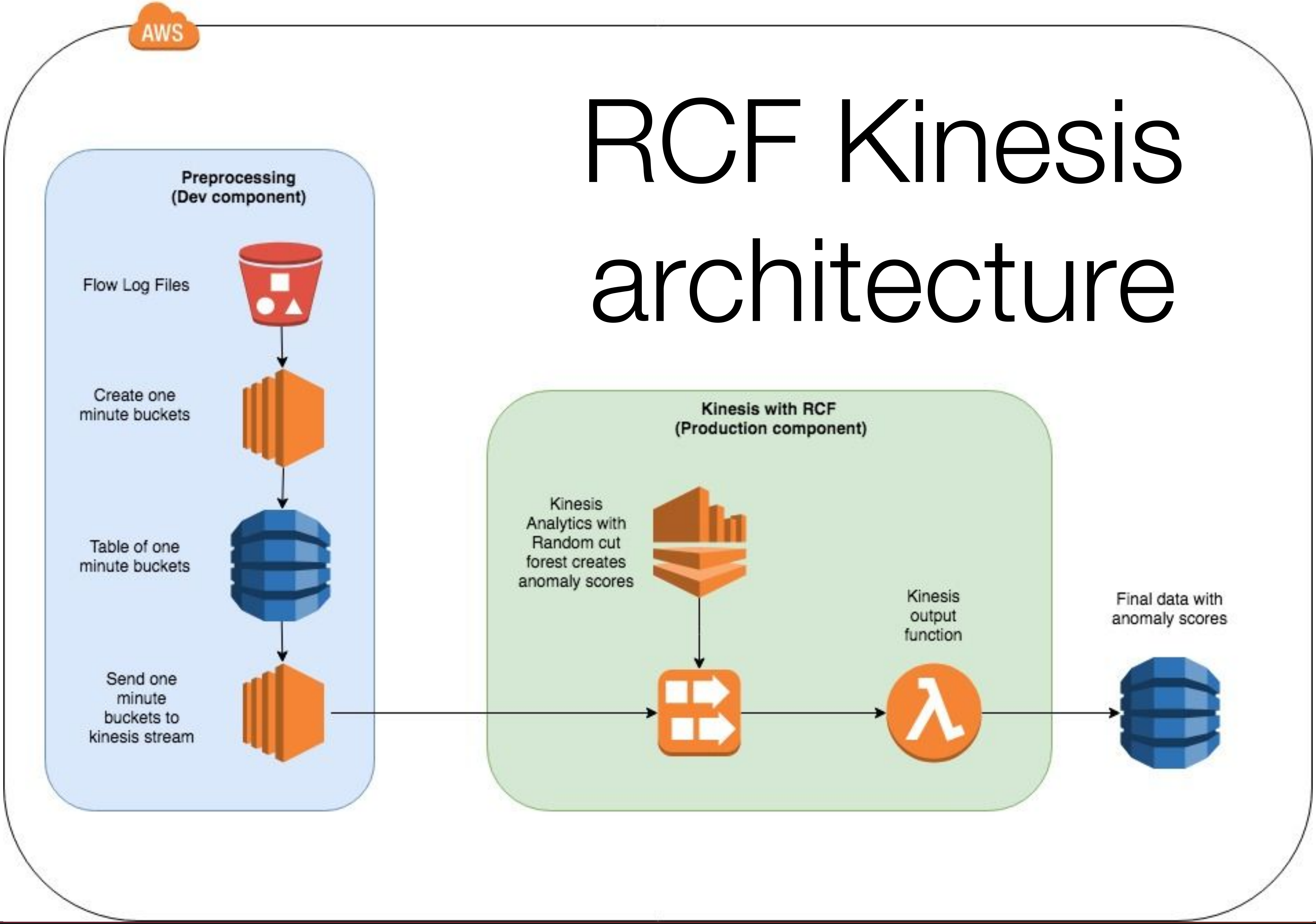


Jacek 11:43 PM
Detected new data batch. You'll find below our 3h forecast.

Jacek 11:51 PM
Detected new data batch. You'll find below our 3h forecast.



RCF Kinesis architecture



AWS

Preprocessing (Dev component)

Flow Log Files

Create one minute buckets

Table of one minute buckets

Send one minute buckets to kinesis stream

Kinesis with RCF (Production component)

Kinesis Analytics with Random cut forest creates anomaly scores

Kinesis output function

Final data with anomaly scores