KubeCon | CloudNativeCon

OPEN SOURCE SUMMIT

China 2019

KubeCon | CloudNativeCon

OPEN SOURCE SUMMIT

China 2019

Tune Your Microservices by Learning from Traces

Zhang Wentao, Yang Yang

Agenda

- About us
- Background story
- Brief about Distributed Tracing & Kubeflow
- Architecture Overview
- Training & Modeling
- Tune microservices based on result

About Us

张文涛

zwtzhang@cn.ibm.com

Zhang WenTao is advisory software engineer in IBM. He is experienced in system/Cloud monitoring, DevOps, big data and kubernetes. He is interested in container orchestration in clusters, Service Mesh and AI.

杨洋

bjyangyy@cn.ibm.com

Yang Yang is advisory software engineer in IBM. She's been working on monitoring for cloud platform over 5 years, and has a lot experience on large scale and dynamic environments. Besides cloud related, she is also very interested in front-end technologies.

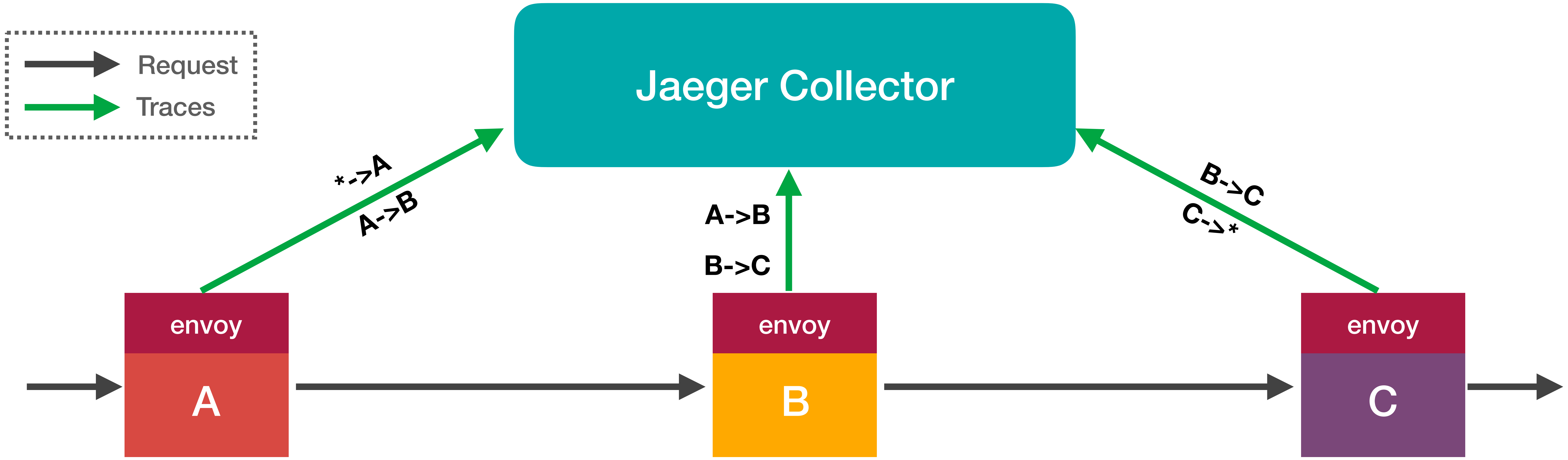
Background Story…

- How to track down problems in cloud world easily?
- Traces are very helpful, but **one** request result in **tens of traces** — how to work with them efficiently?
- Is the **pre-set static threshold** can really identify abnormal in a **constantly changing** cloud environment?

- **What we're trying to do:**
  - Leverage ML to help us understand the huge amount of traces
  - Use the model to help us tune and refine services:
    - Anomaly detection
    - Scaling guidance

Distributed Tracing

- Distributed tracing is a **super powerful tool** to help with trouble shooting and performance analysis in real world operation
- Jaeger is inspired by Dapper and Zipkin, initiated by Uber
- Implemented by following OpenTracing https://opentracing.io/docs/overview/
- We use **Istio** to help us gather traces. In Istio, spans will be generated by each envoy sidecar and send to Jaeger by default

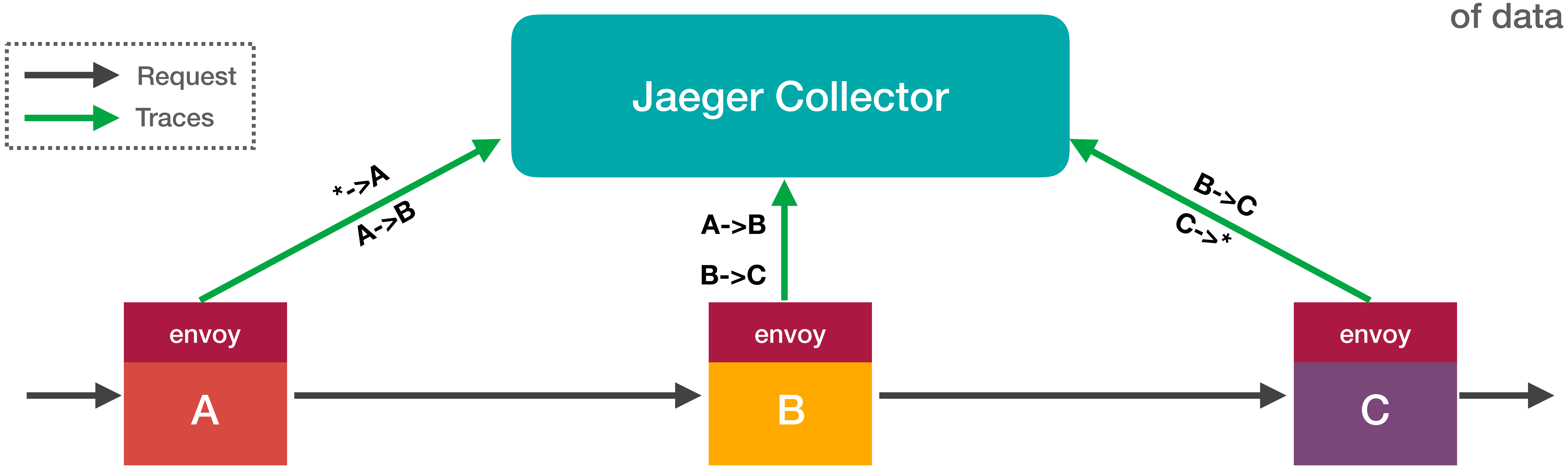


Distributed Tracing

- Distributed tracing is a **super powerful tool** to help with trouble shooting and performance analysis in real world operation
- Jaeger is inspired by Dapper and Zipkin, initiated by Uber
- Implemented by following OpenTracing https://opentracing.io/docs/overview/
- We use **Istio** to help us gather traces. In Istio, spans will be generated by each envoy sidecar and send to Jaeger by default

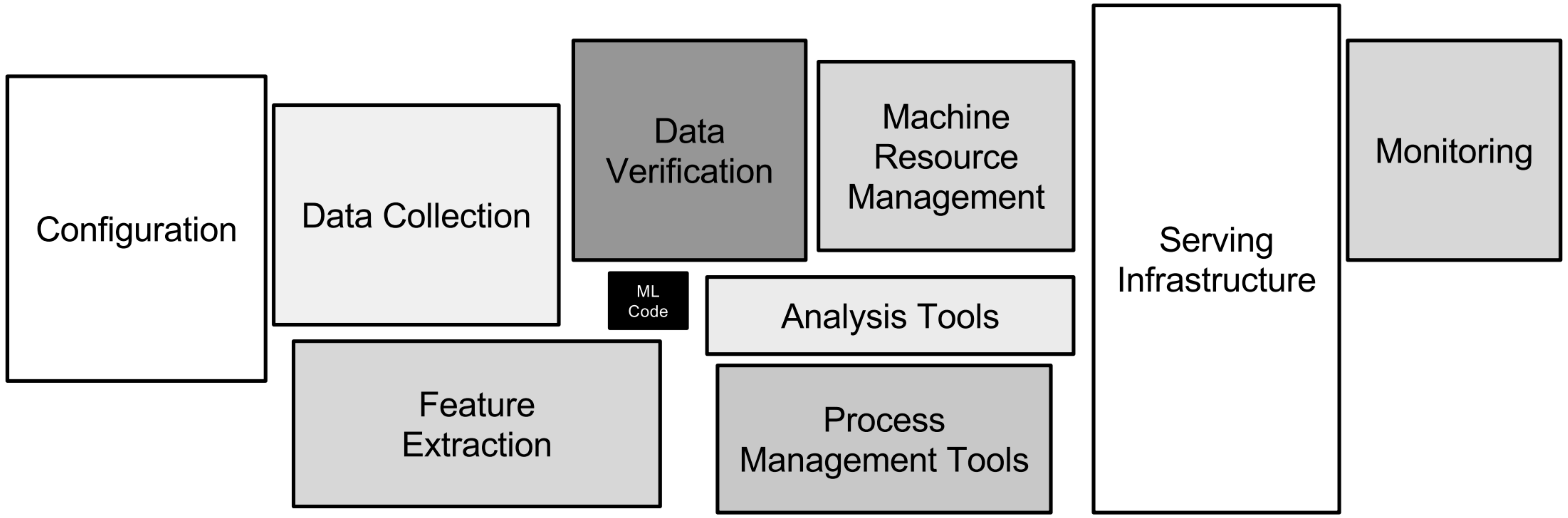
Characteristic of data

- Data is well formatted & aligned by Istio
- Time series data
- Huge amount of data within short time period



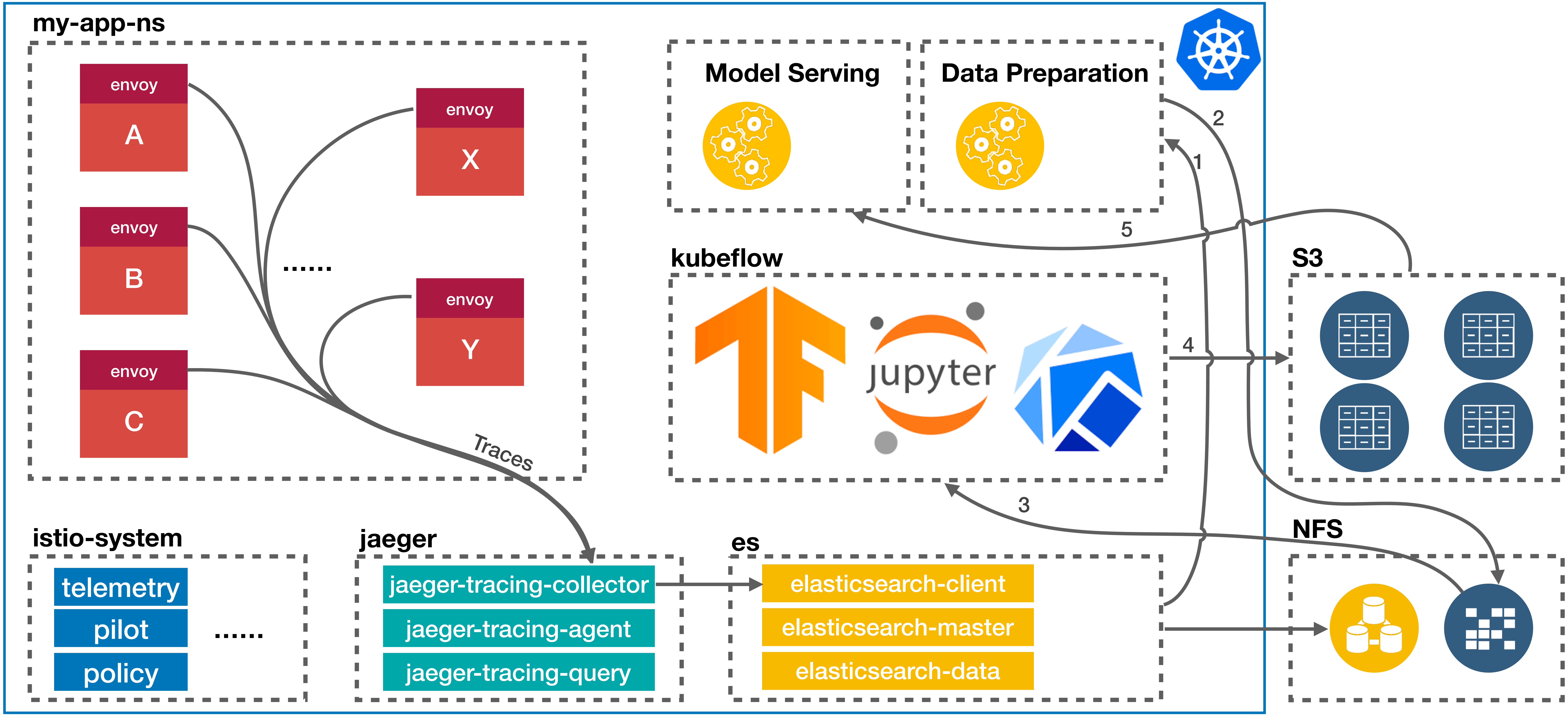
Kubeflow

- Making deployments of machine learning workflows on Kubernetes simple, portable and scalable
- Support multiple ML frameworks
  - TensorFlow
  - Pytorch
  - Caffe
- Distributed training
- Models Serving
- **How we're using it:**
  - Kubernetes cluster of 40 nodes
  - TensorFlow as backend
  - ~700,000 traces collected per day



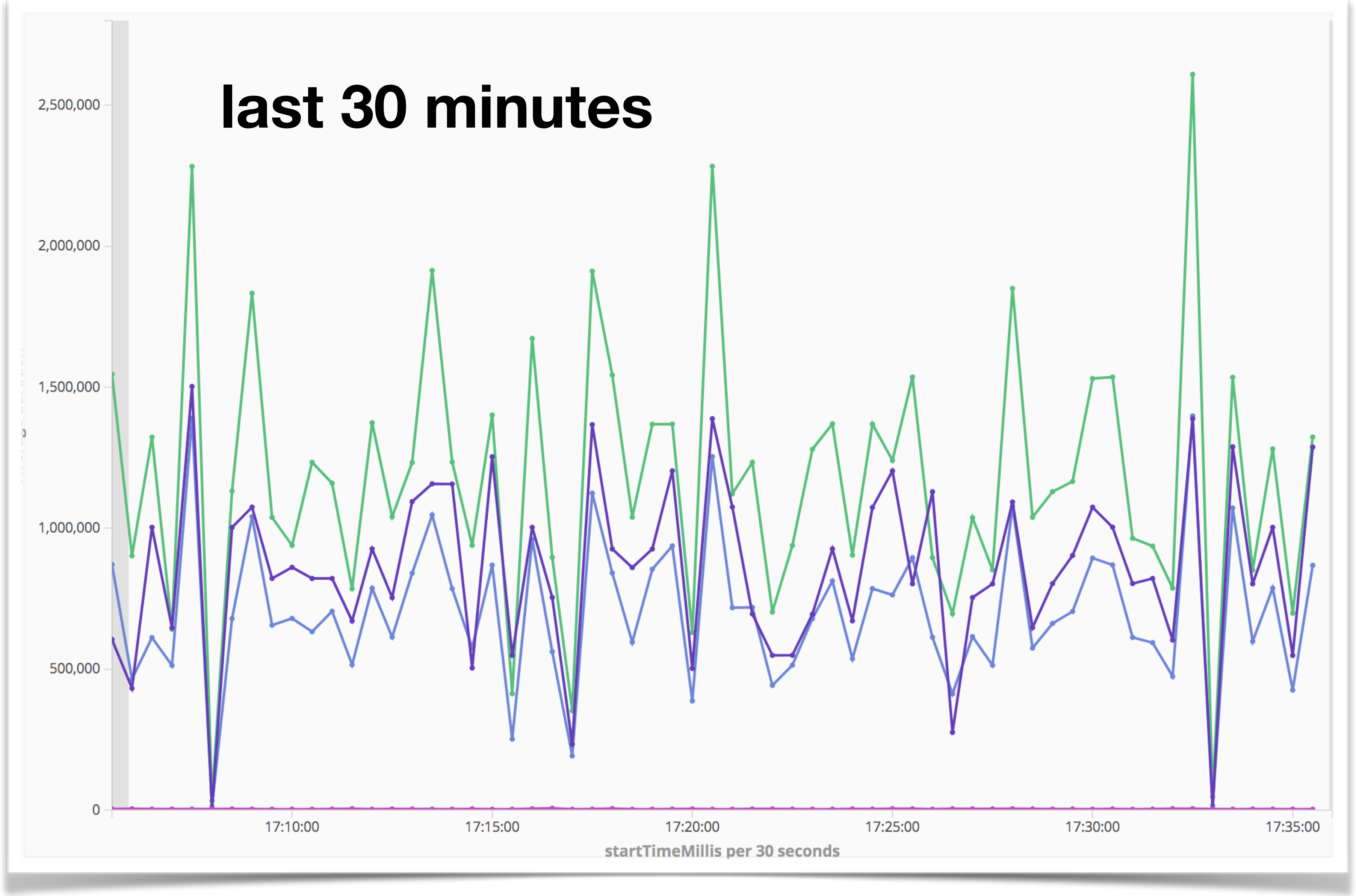
From ***Hidden Technical Debt in Machine Learning Systems***

Architecture Overview

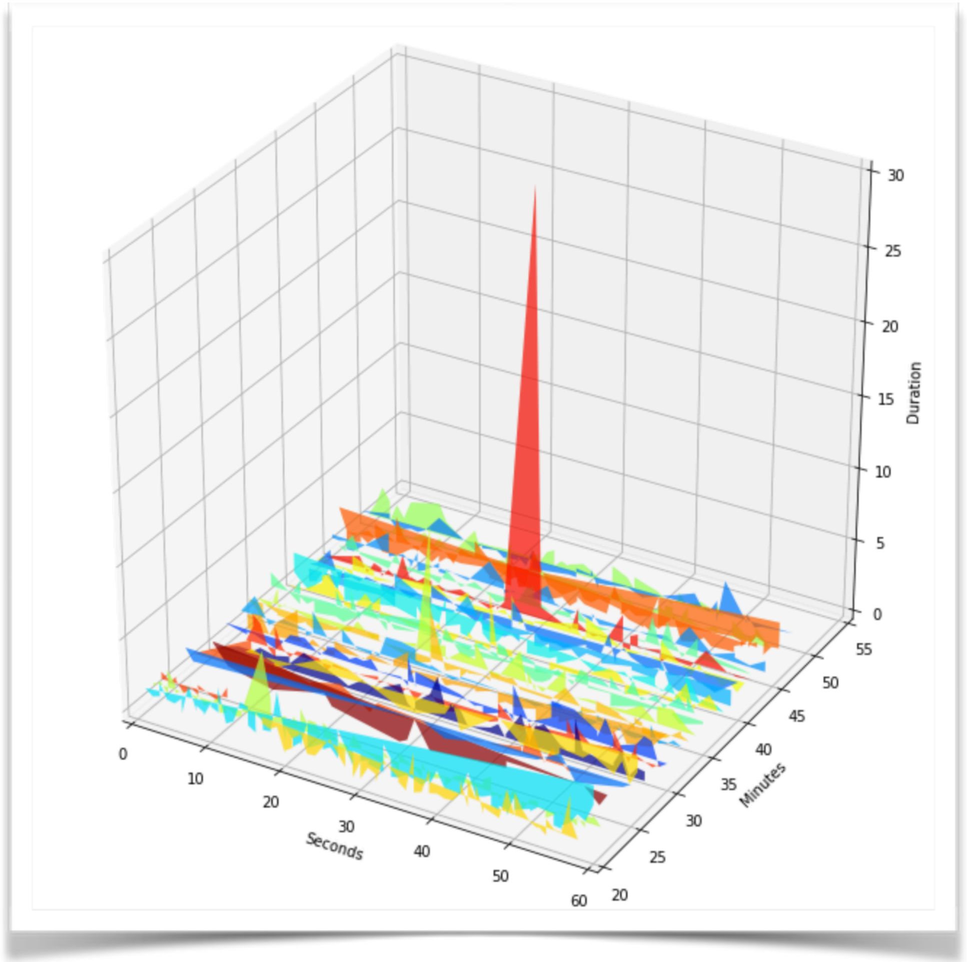


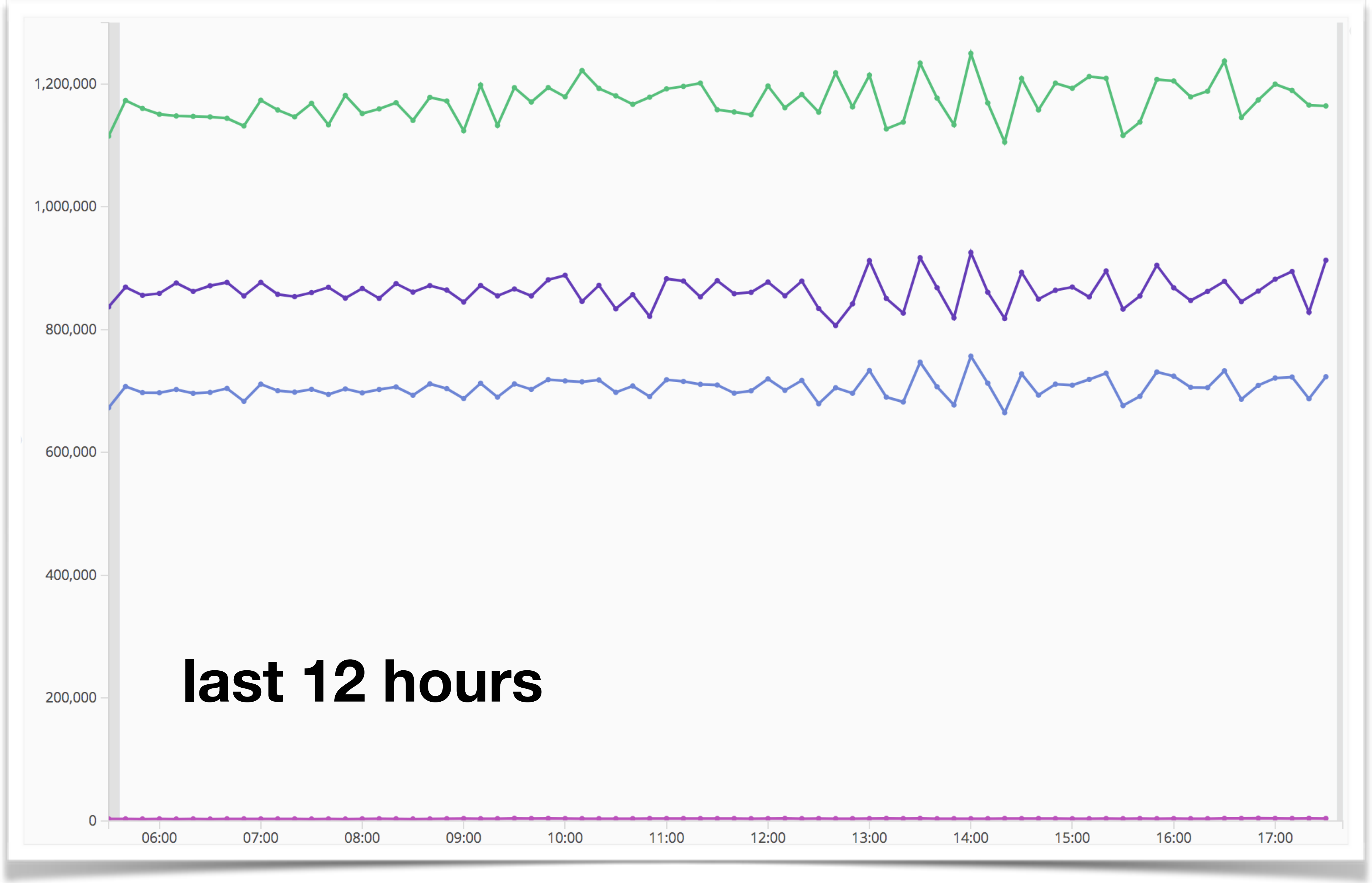
Training & Modeling

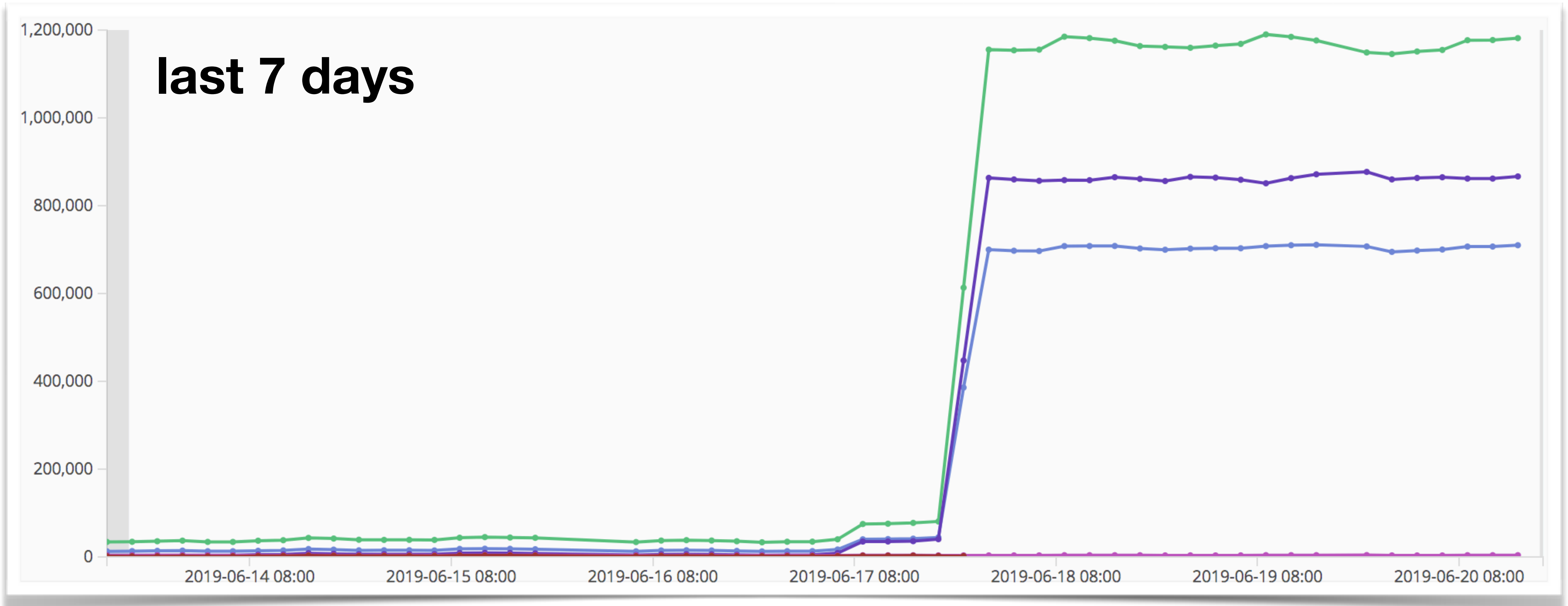
- EDA (Exploratory data analysis)

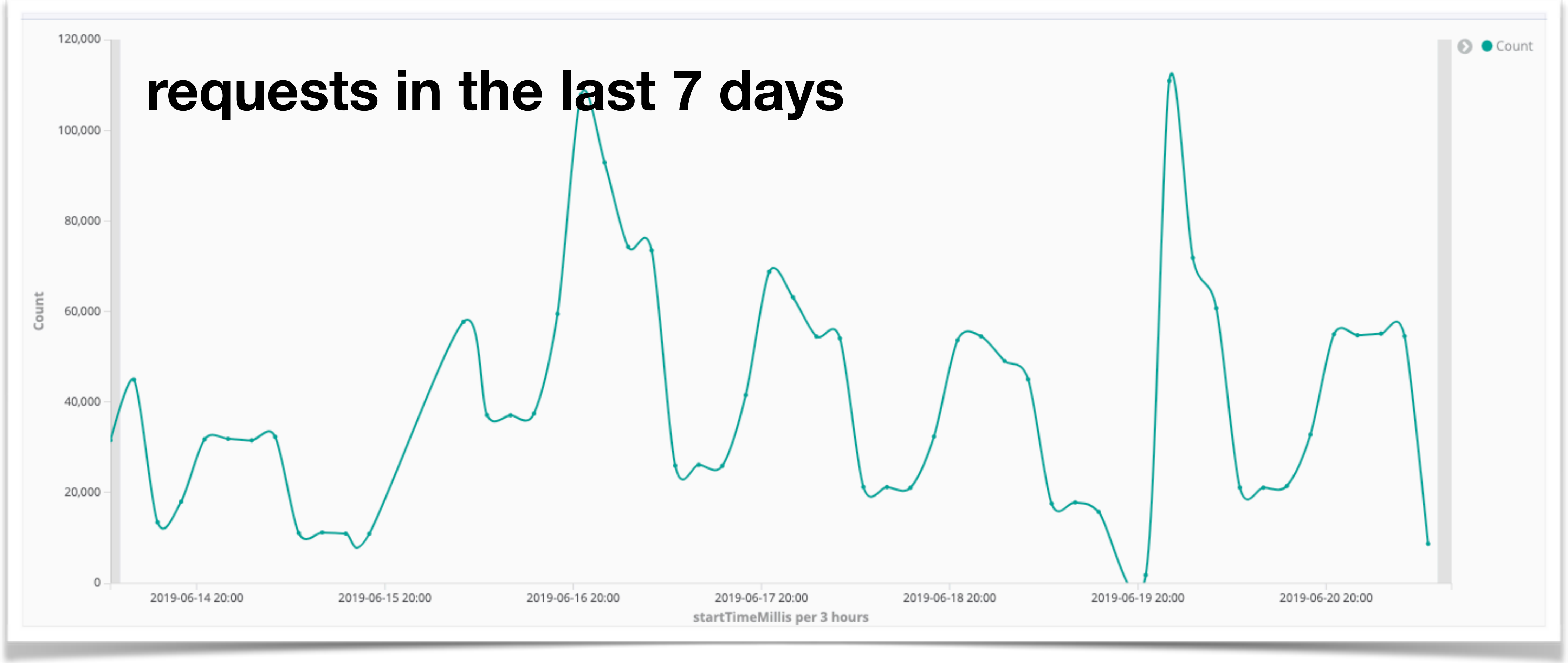

last 30 minutes

slice in minute



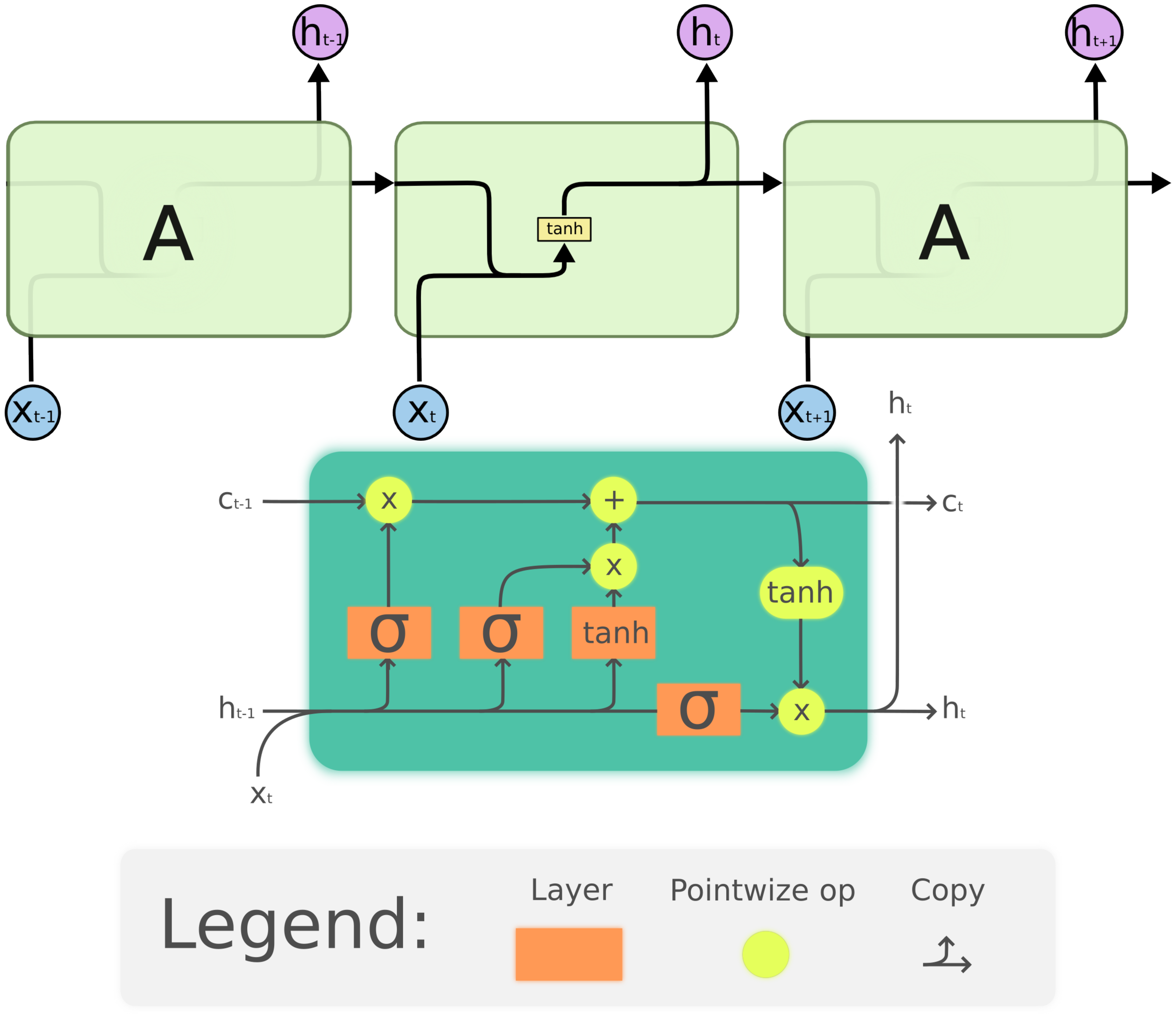

last 12 hours


last 7 days


requests in the last 7 days

Training & Modeling

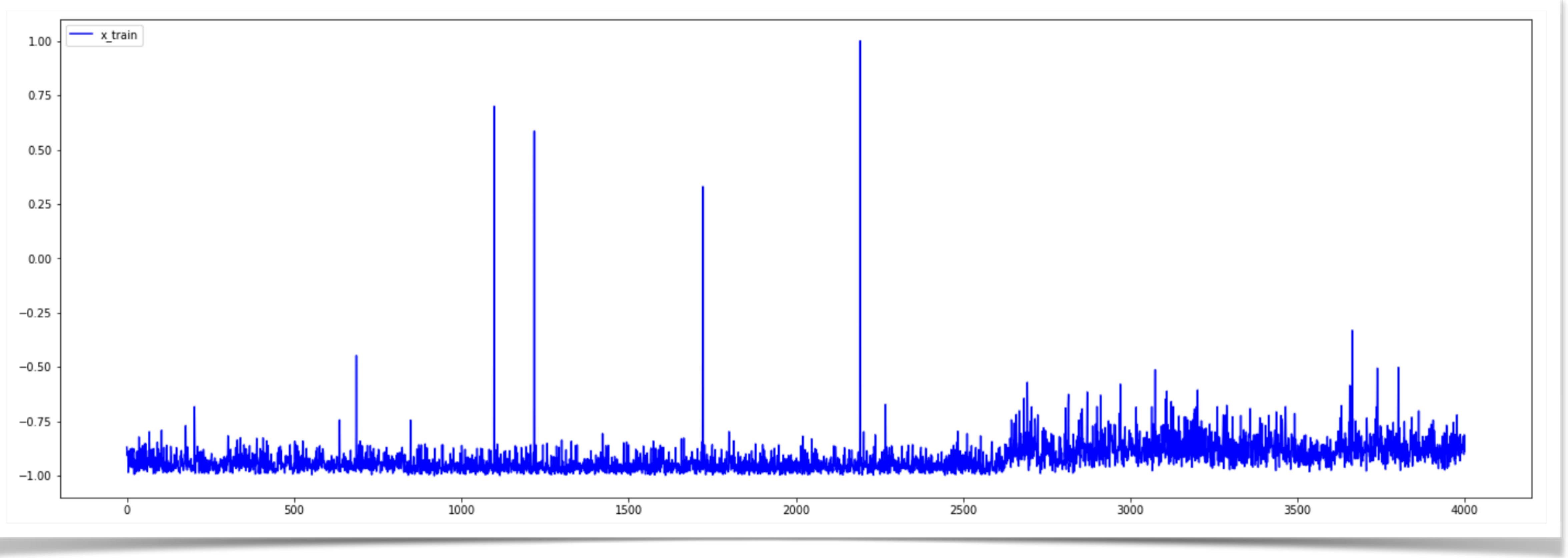
- Duration anomaly detection
- Problem type
  - data is time series
  - there is no label
  - predict time series in the future
- Using LSTM to generate new sequence
- Find anomaly point based on prediction

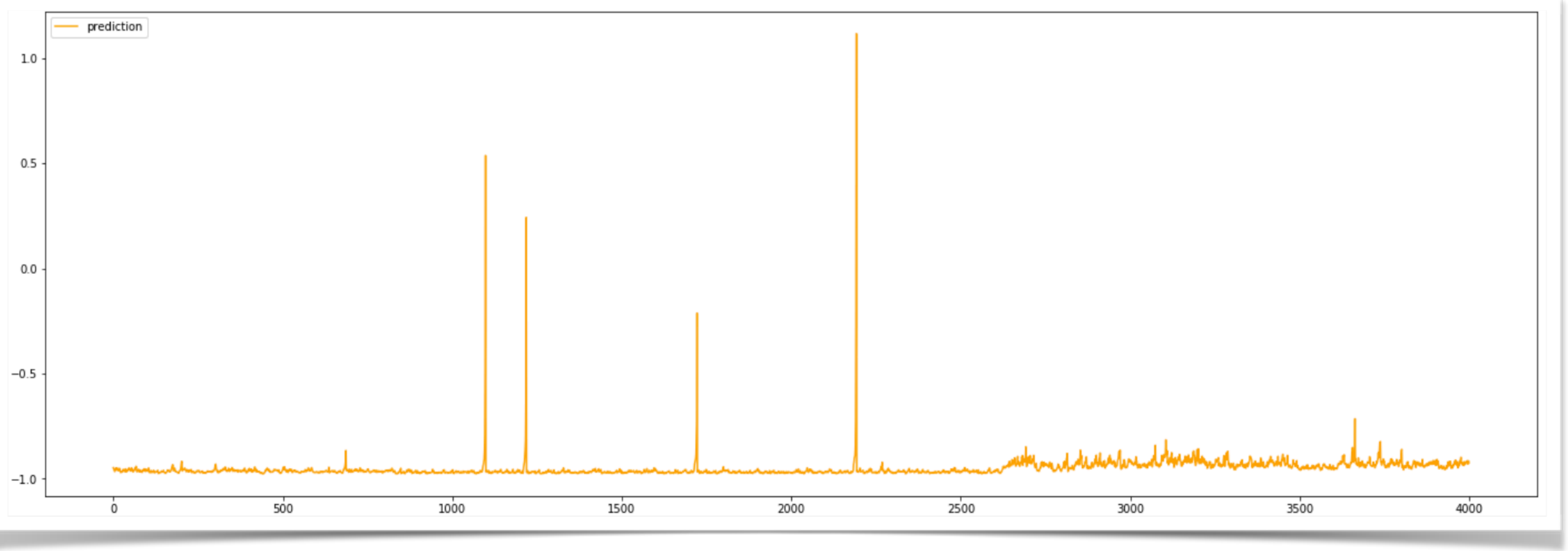


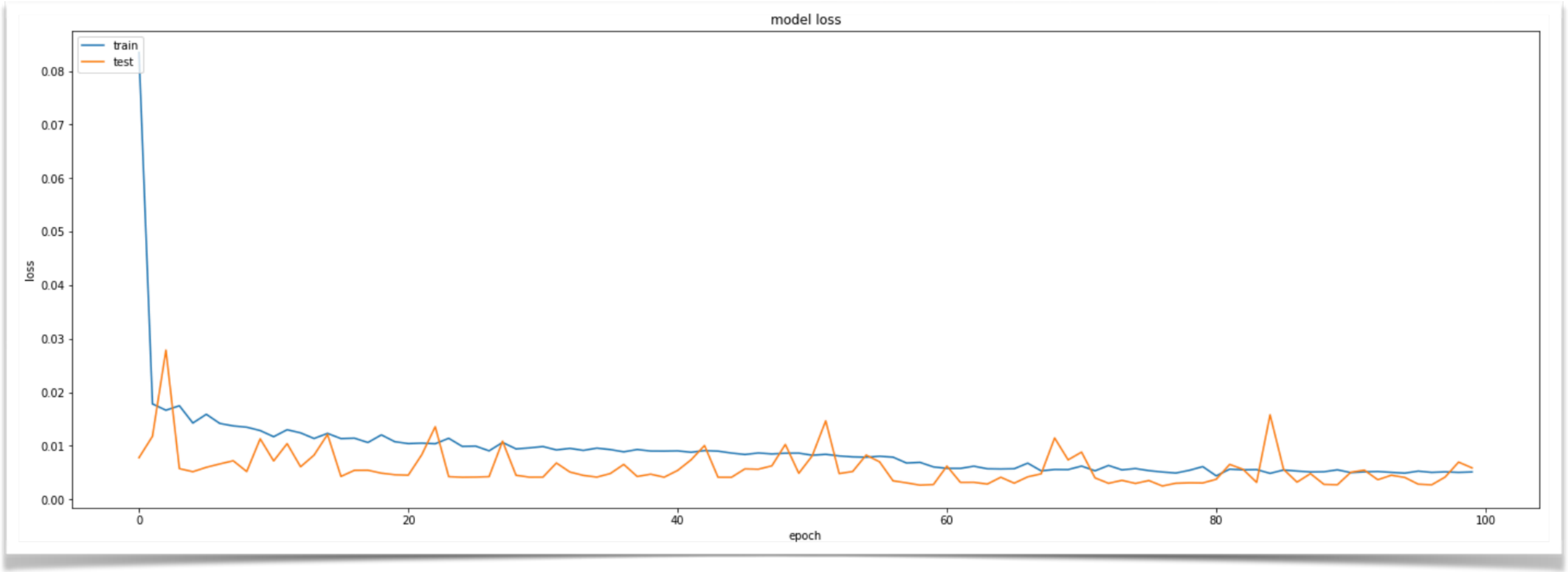
from WikiPedia *Long short-term memory*

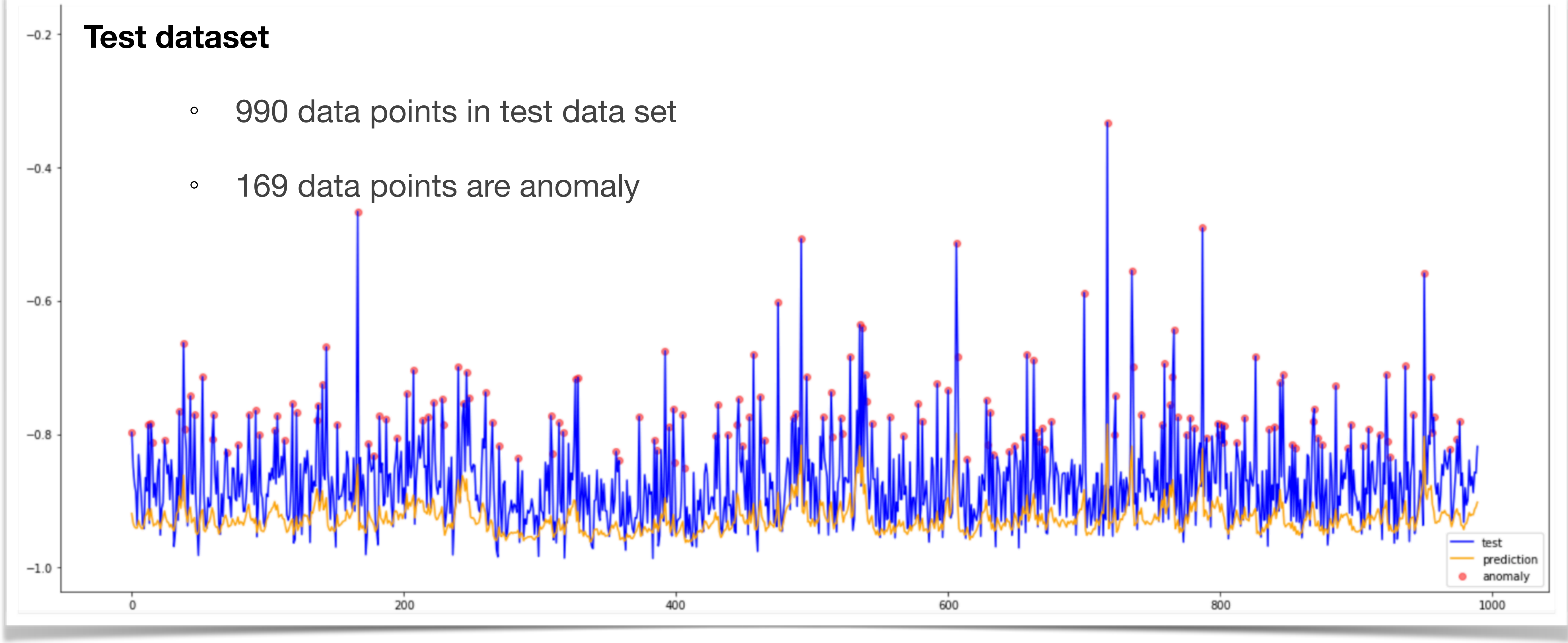
Training & Modeling

Training dataset







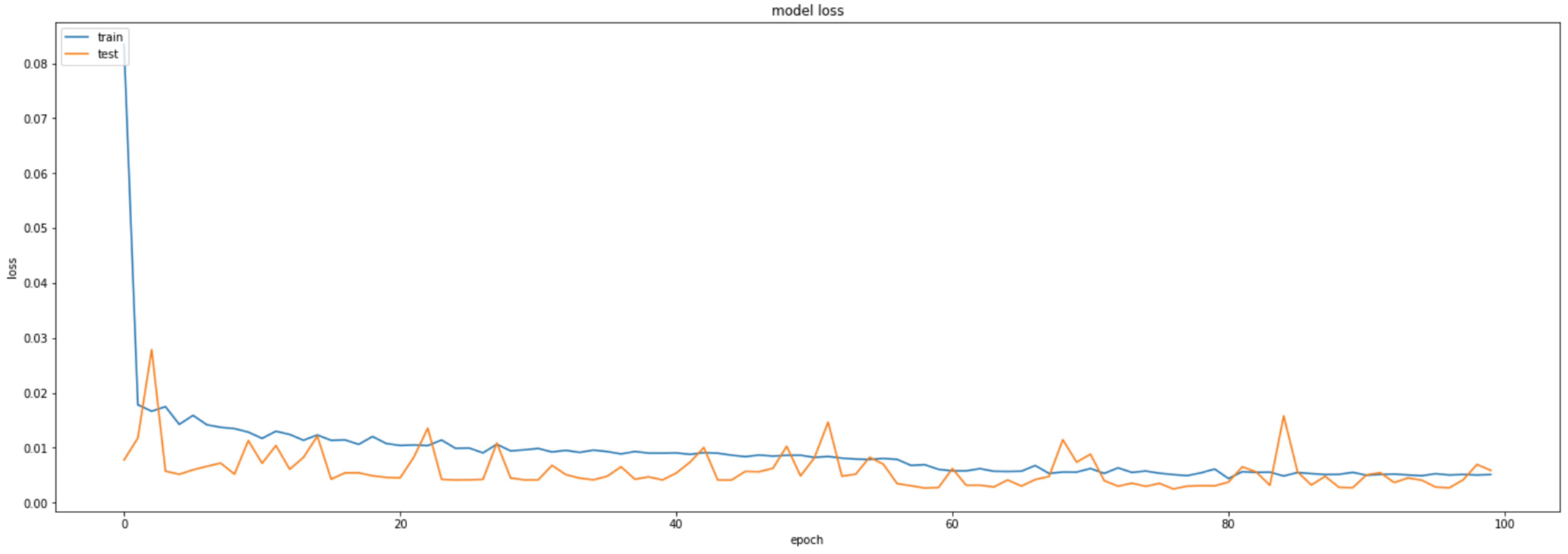


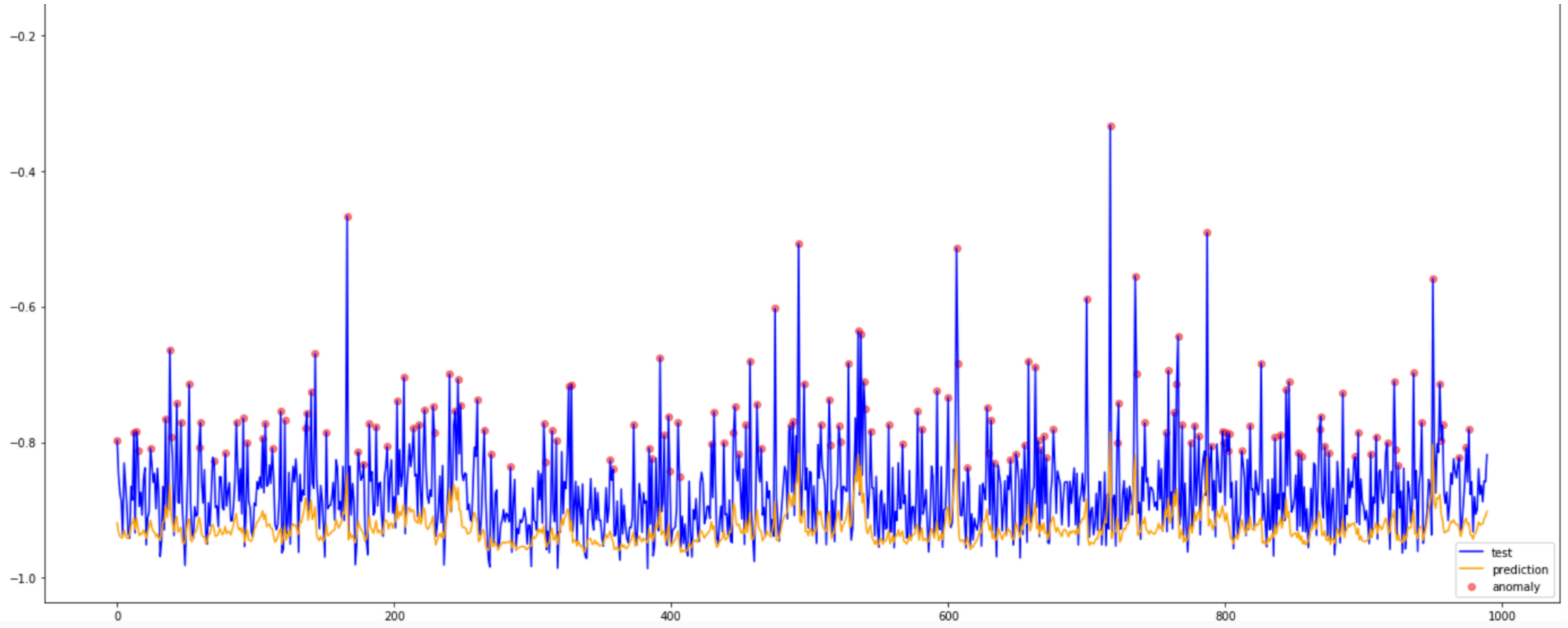
Test dataset

- 990 data points in test data set
- 169 data points are anomaly

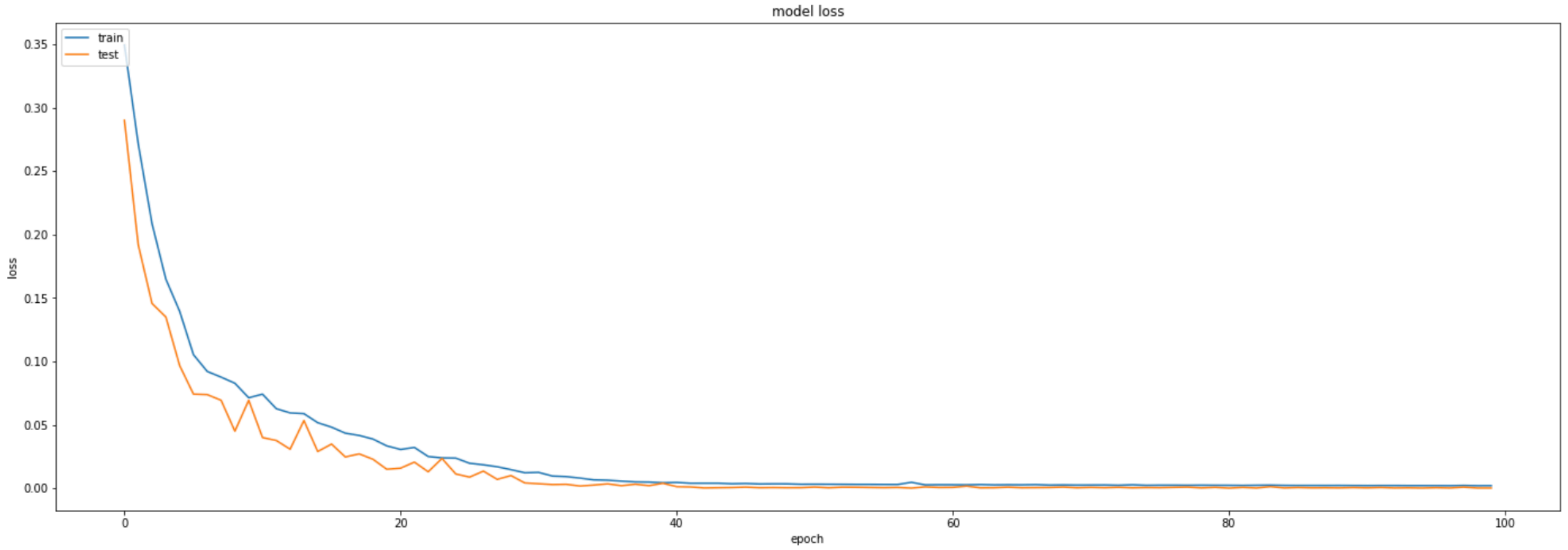
Training & Modeling

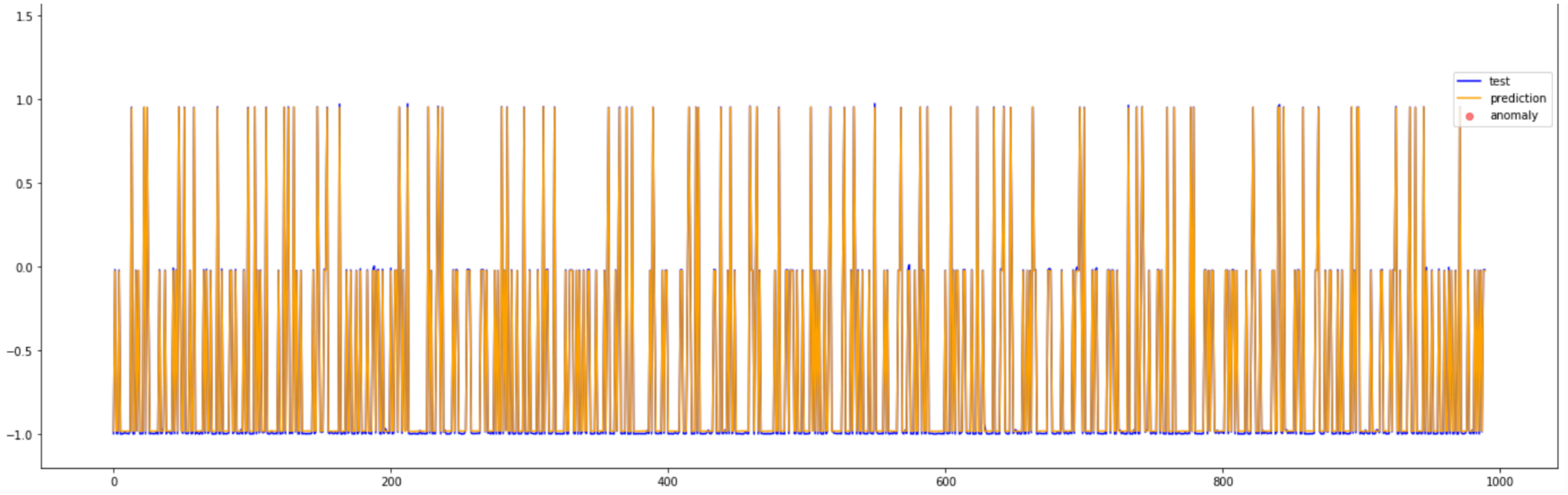
Model with data in 30 minutes



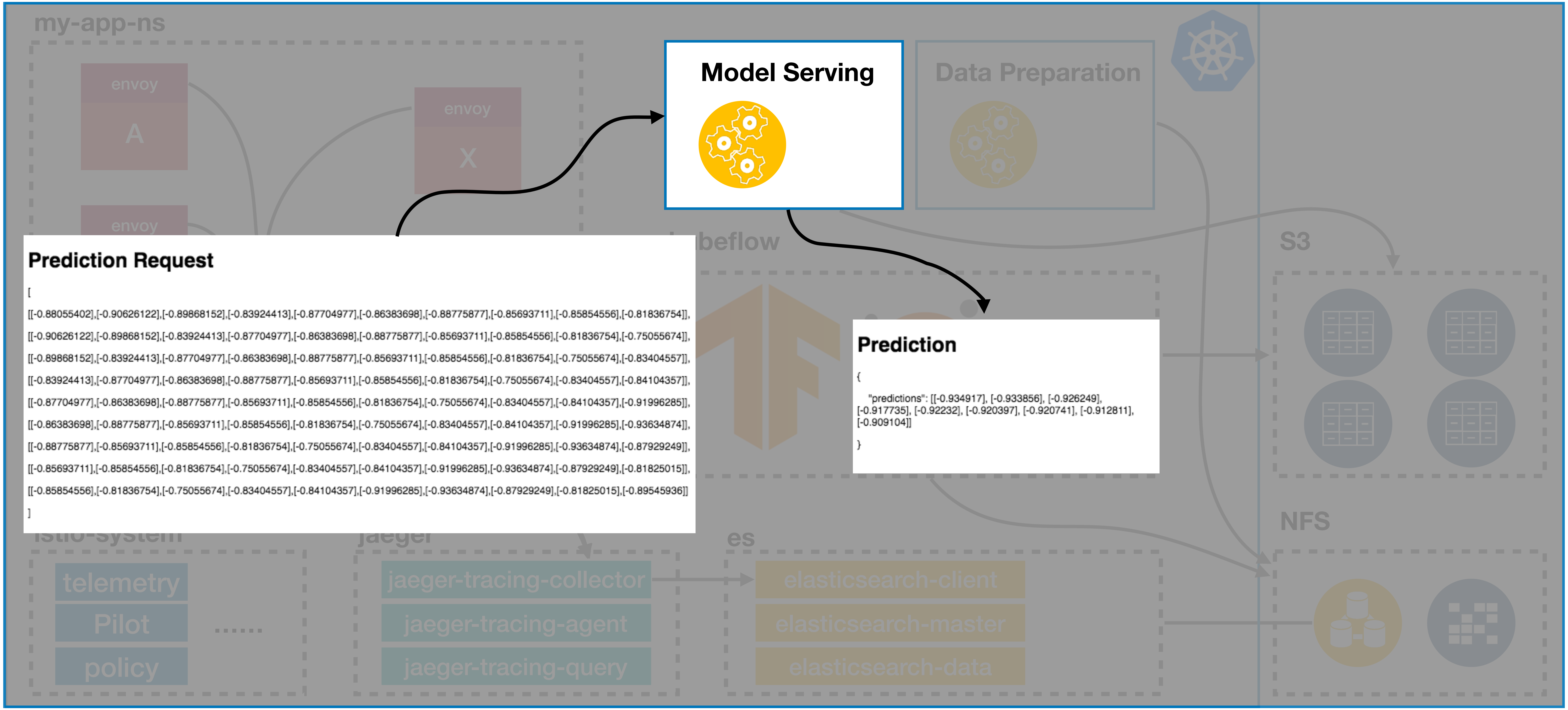


Model with data in 24 hours





Model Serving



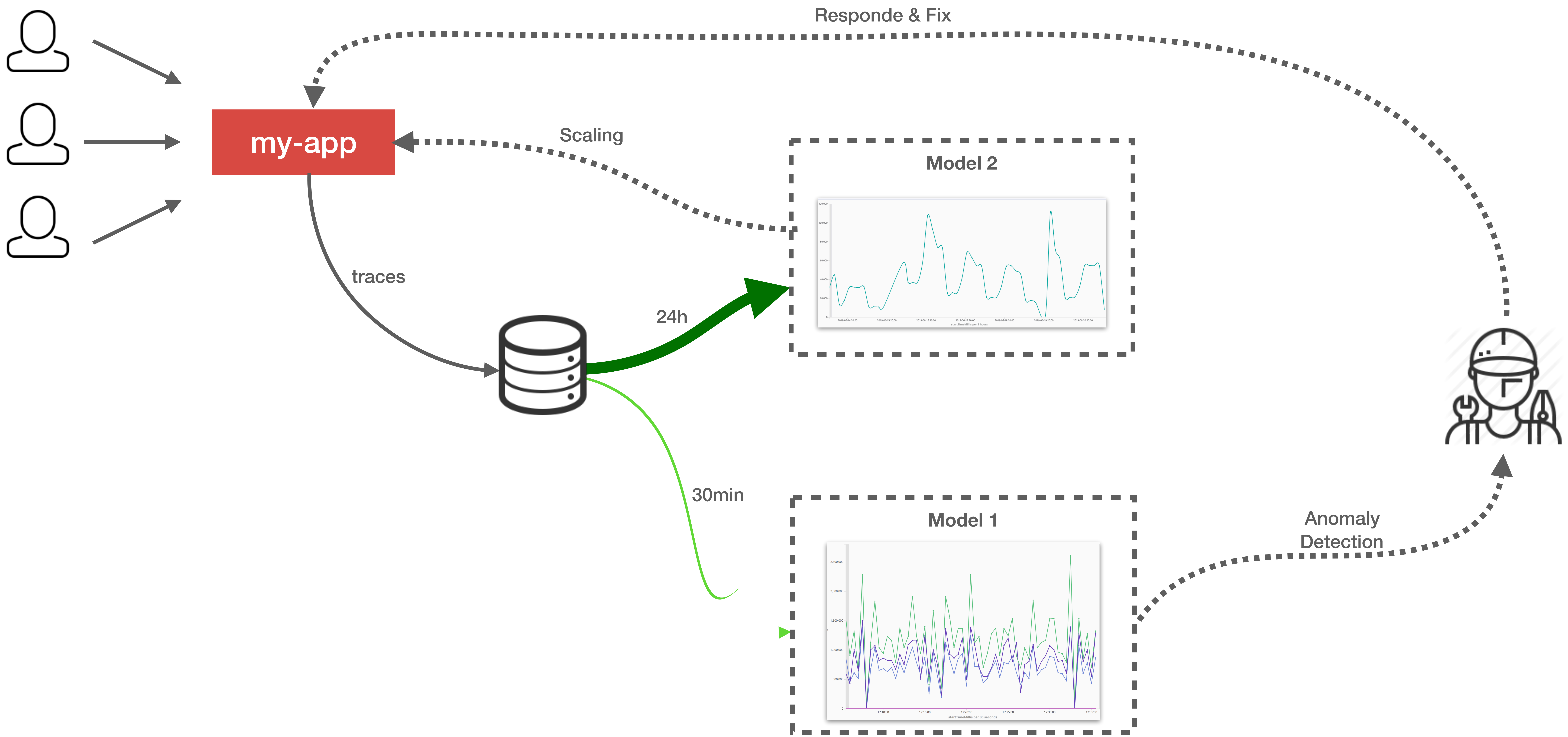
Prediction Request

```
[
[[-0.88055402],[-0.90626122],[-0.89868152],[-0.83924413],[-0.87704977],[-0.86383698],[-0.88775877],[-0.85693711],[-0.85854556],[-0.81836754]],
[[-0.90626122],[-0.89868152],[-0.83924413],[-0.87704977],[-0.86383698],[-0.88775877],[-0.85693711],[-0.85854556],[-0.81836754],[-0.75055674]],
[[-0.89868152],[-0.83924413],[-0.87704977],[-0.86383698],[-0.88775877],[-0.85693711],[-0.85854556],[-0.81836754],[-0.75055674],[-0.83404557]],
[[-0.83924413],[-0.87704977],[-0.86383698],[-0.88775877],[-0.85693711],[-0.85854556],[-0.81836754],[-0.75055674],[-0.83404557],[-0.84104357]],
[[-0.87704977],[-0.86383698],[-0.88775877],[-0.85693711],[-0.85854556],[-0.81836754],[-0.75055674],[-0.83404557],[-0.84104357],[-0.91996285]],
[[-0.86383698],[-0.88775877],[-0.85693711],[-0.85854556],[-0.81836754],[-0.75055674],[-0.83404557],[-0.84104357],[-0.91996285],[-0.93634874]],
[[-0.88775877],[-0.85693711],[-0.85854556],[-0.81836754],[-0.75055674],[-0.83404557],[-0.84104357],[-0.91996285],[-0.93634874],[-0.87929249]],
[[-0.85693711],[-0.85854556],[-0.81836754],[-0.75055674],[-0.83404557],[-0.84104357],[-0.91996285],[-0.93634874],[-0.87929249],[-0.81825015]],
[[-0.85854556],[-0.81836754],[-0.75055674],[-0.83404557],[-0.84104357],[-0.91996285],[-0.93634874],[-0.87929249],[-0.81825015],[-0.89545936]]
]
```

Prediction

```
{
    "predictions": [[-0.934917], [-0.933856], [-0.926249],
[-0.917735], [-0.92232], [-0.920397], [-0.920741], [-0.912811],
[-0.909104]]
}
```

Tuning microservices based on result



Thank YOU !