SKIPJAQ

Automating Operations with Machine Intelligence

Rob Harrop

CEO @ **Skipjaq**

Co-founder @ **SpringSource**

Automated performance management

Why automate operations?

Why now?

What does automated operations look like?

How do we build for automation?

Solving a real problem…

Why automate operations?

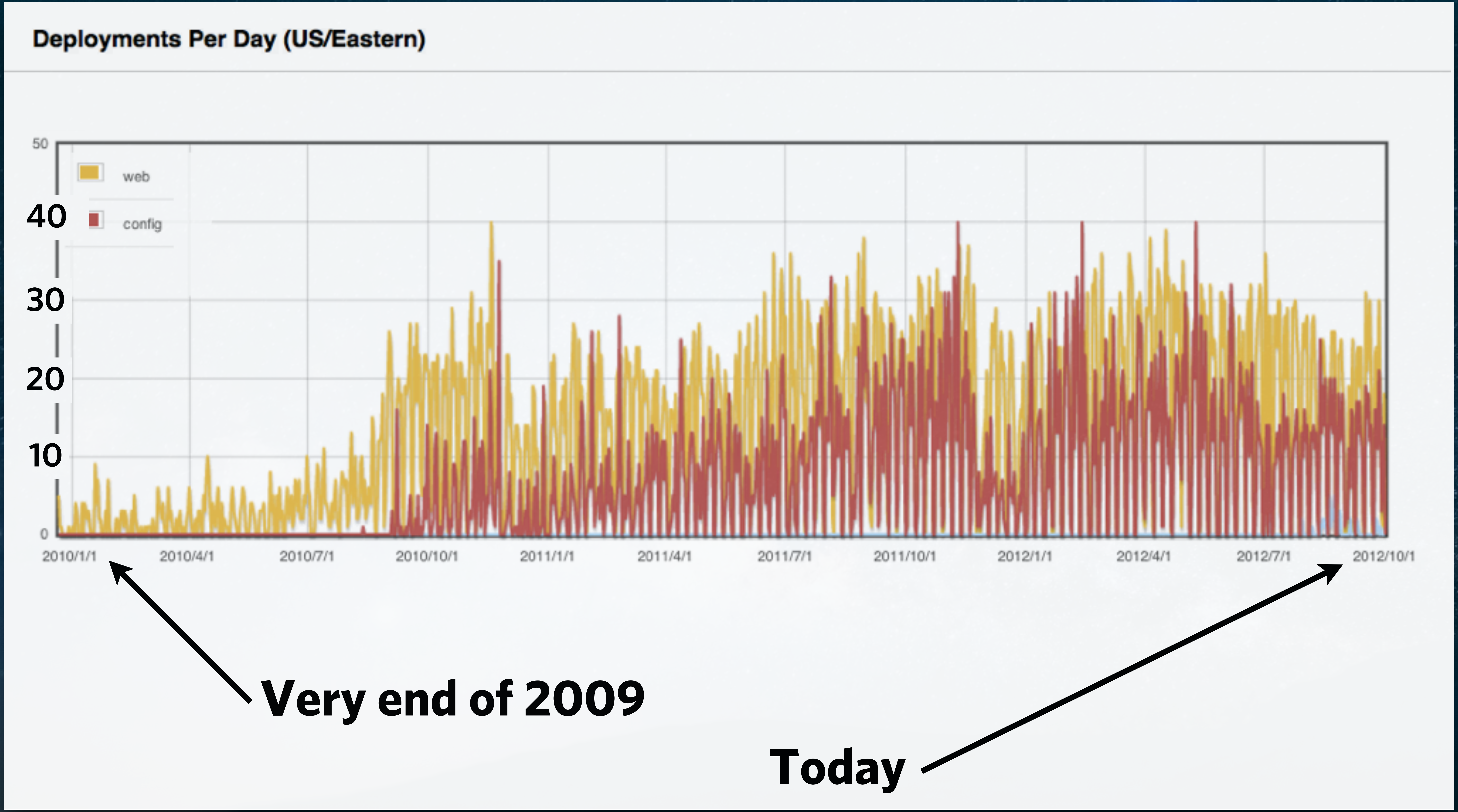
More Complexity

Monolith -> Microservices

Strong -> Eventual Consistency

Assume reliability -> Assume failure

More Deployments



Deployments Per Day (US/Eastern)

Very end of 2009

Today

Credit: Mike Brittain, Engineering Director @ Easy

Less time to identify fixes

Rollbacks more likely

Tiny window for human intervention

Harder

Faster

Why now?

We have to

We can

Trends

- Cloud
- Containers
- Observability
- Microservices
- ML/AI

Current trends provide the impetus and tools for automation by AI

Automated Operations

AlphaGo

Move 37

HA HA!

Move 78 - God’s Touch

AI Human

Types of Operation Actions

Wholly performed by human

Wholly performed by AI

Co-operation between human and AI

Actionable insight

On Metrics

Data **is not** insight

Gathering metrics **is not** automating operations

But, metrics are **critical** to automating operations

Human ≠ Manual

Actions by Human

Testing

Deployment

Provisioning

Cooperative Actions

Anomaly alerting

Rollback broken builds

Dependency upgrade

Actions by AI

Predictive auto scaling

Workload placement

Automatic rollback

Performance optimisation?

Security?

Actions

and

Actionable Insights

Building for Automation

Requirements for Operations

Visible metrics and logs

Ability to start/stop/restart/move workload

Ability to change configuration

Ability to modify dependencies

Ability to wire/rewire external services

Self-contained package

Disposable processes

Externally-configurable

Externally-observable

Externalised dependencies

Externalised service wiring

12+1 Factor

13th Factor - Observability

Metrics as event streams

Standard metrics

- CPU usage, memory usage, …

Service-specific metrics

- Leads received, items sold, …

Case Study

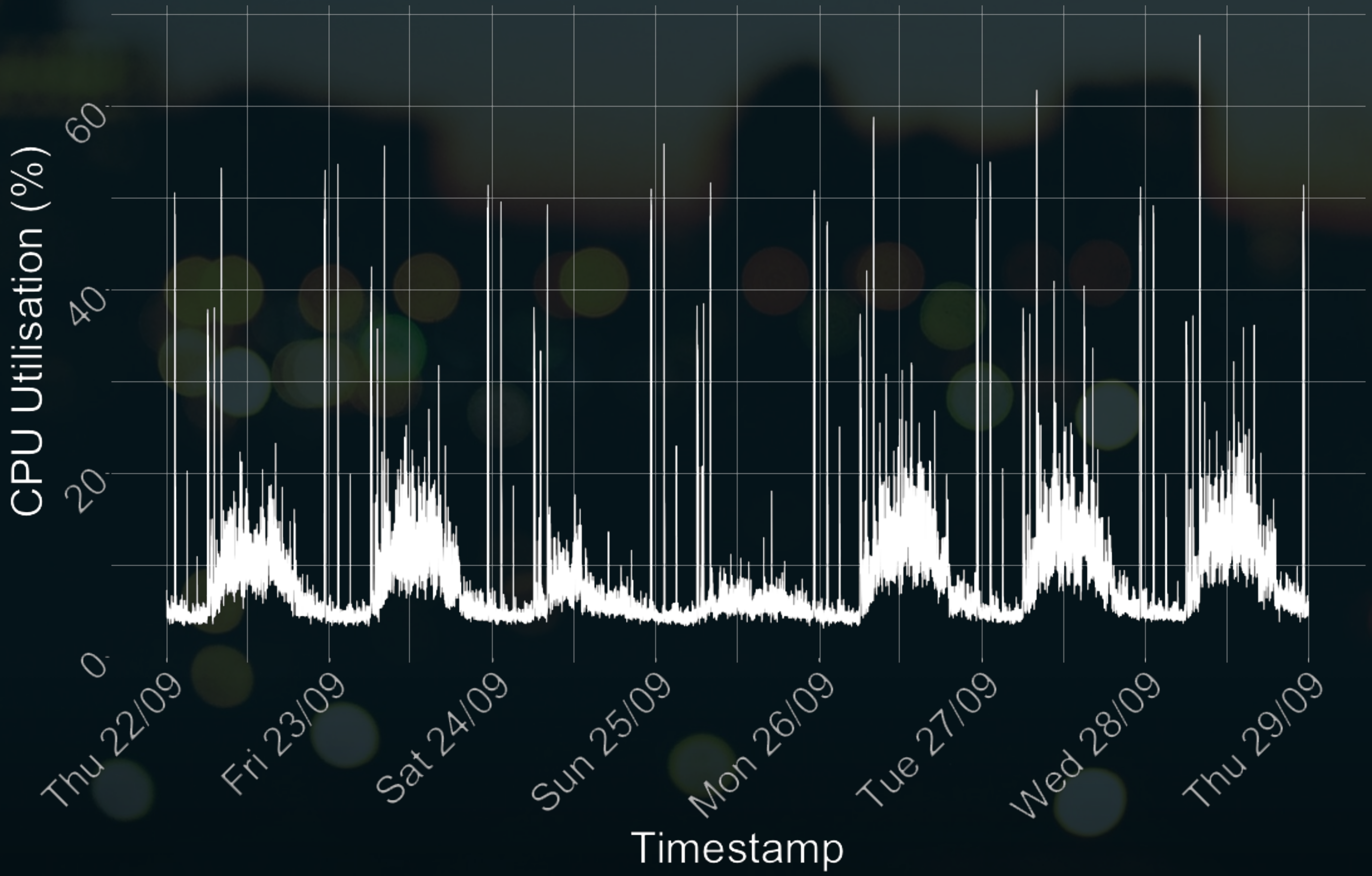
Detecting Anomalous DB CPU

Background

Consumer-facing web application running Rails against PostgreSQL on AWS RDS

Mix of transactional and batch workloads running against the same database

Question: when is the DB unusually overloaded?


CPU Utilisation (%)
60
40
20
0
Thu 22/09
Fri 23/09
Sat 24/09
Sun 25/09
Mon 26/09
Tue 27/09
Wed 28/09
Thu 29/09
Timestamp

Detecting Anomalies

Policy-based

Statistical model

Predictive model

Classification model

Policy Based

Fixed threshold alerting

How well does this work?

Not Very

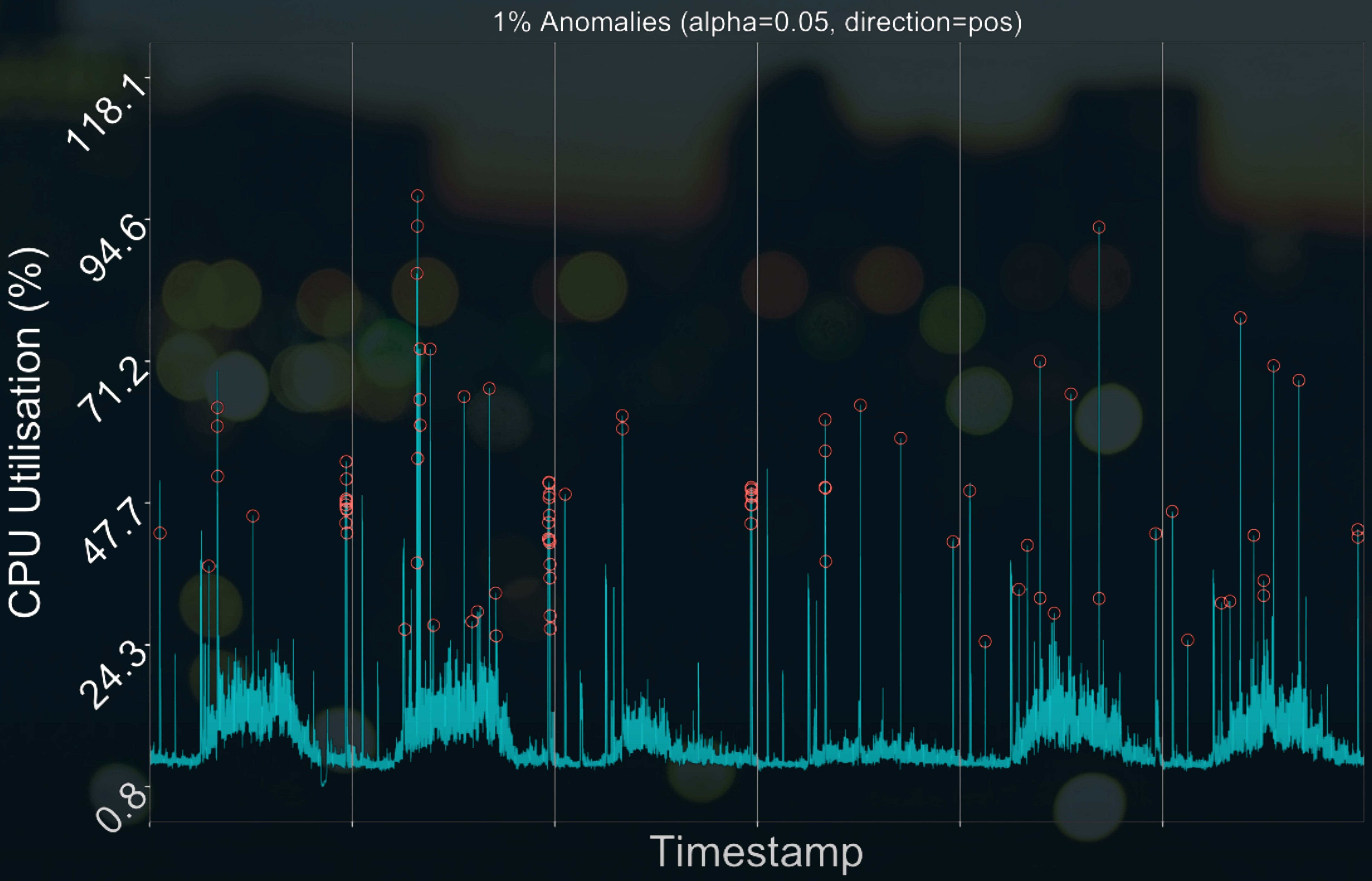
Statistical Model

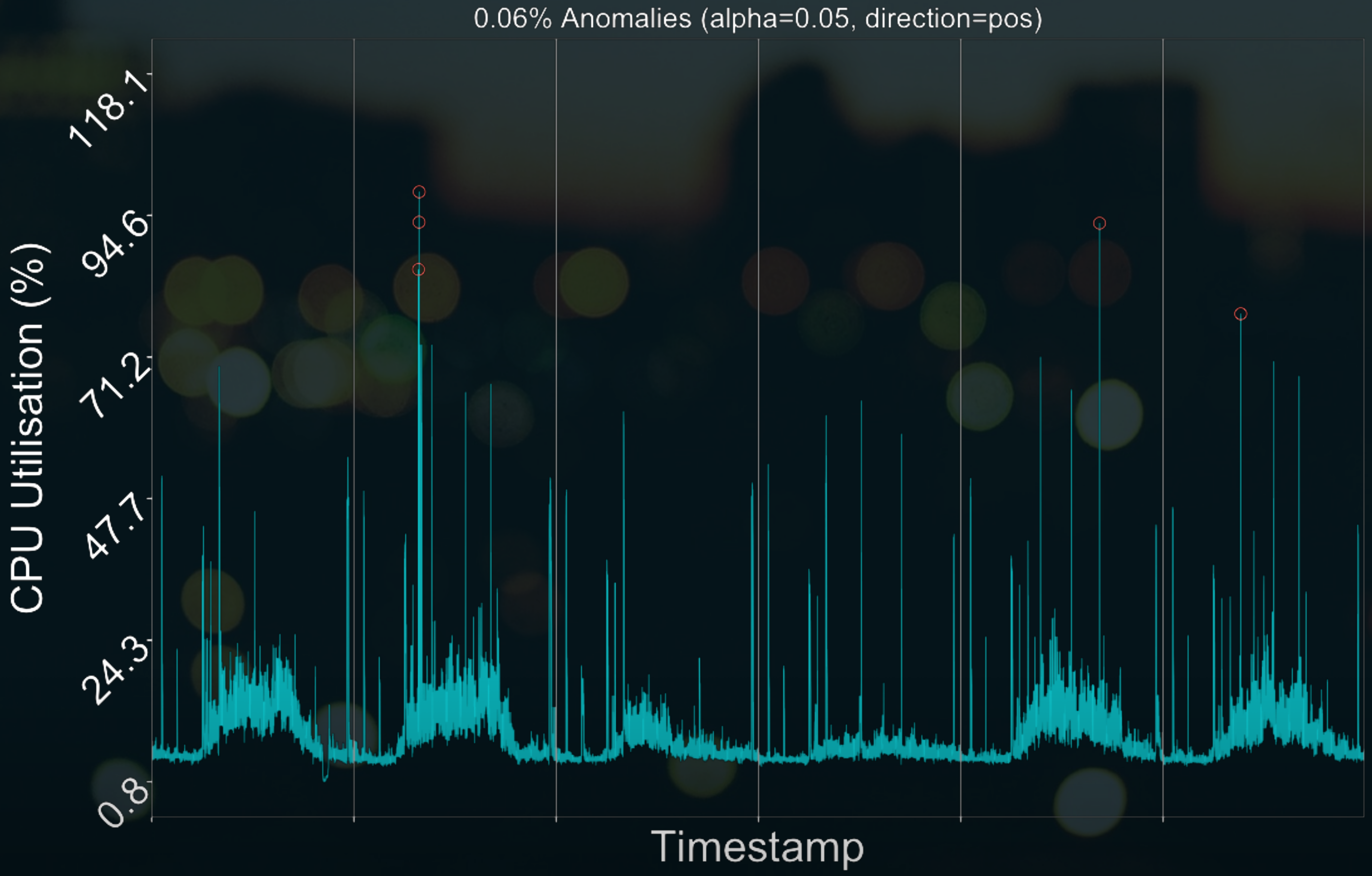
Twitter AnomalyDetection package

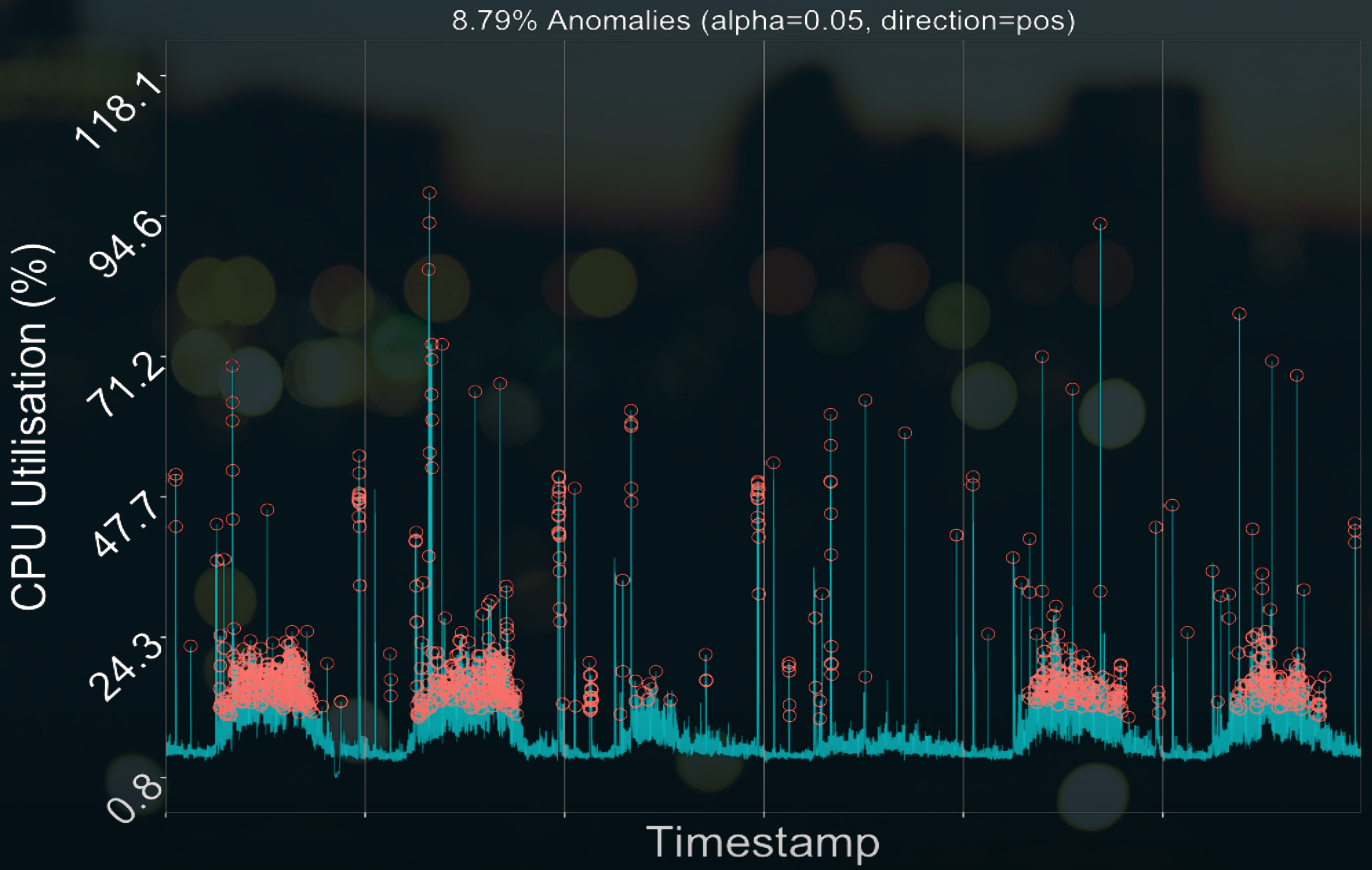
- Seasonal Hybrid ESD

Is this point unexpected in our distribution?

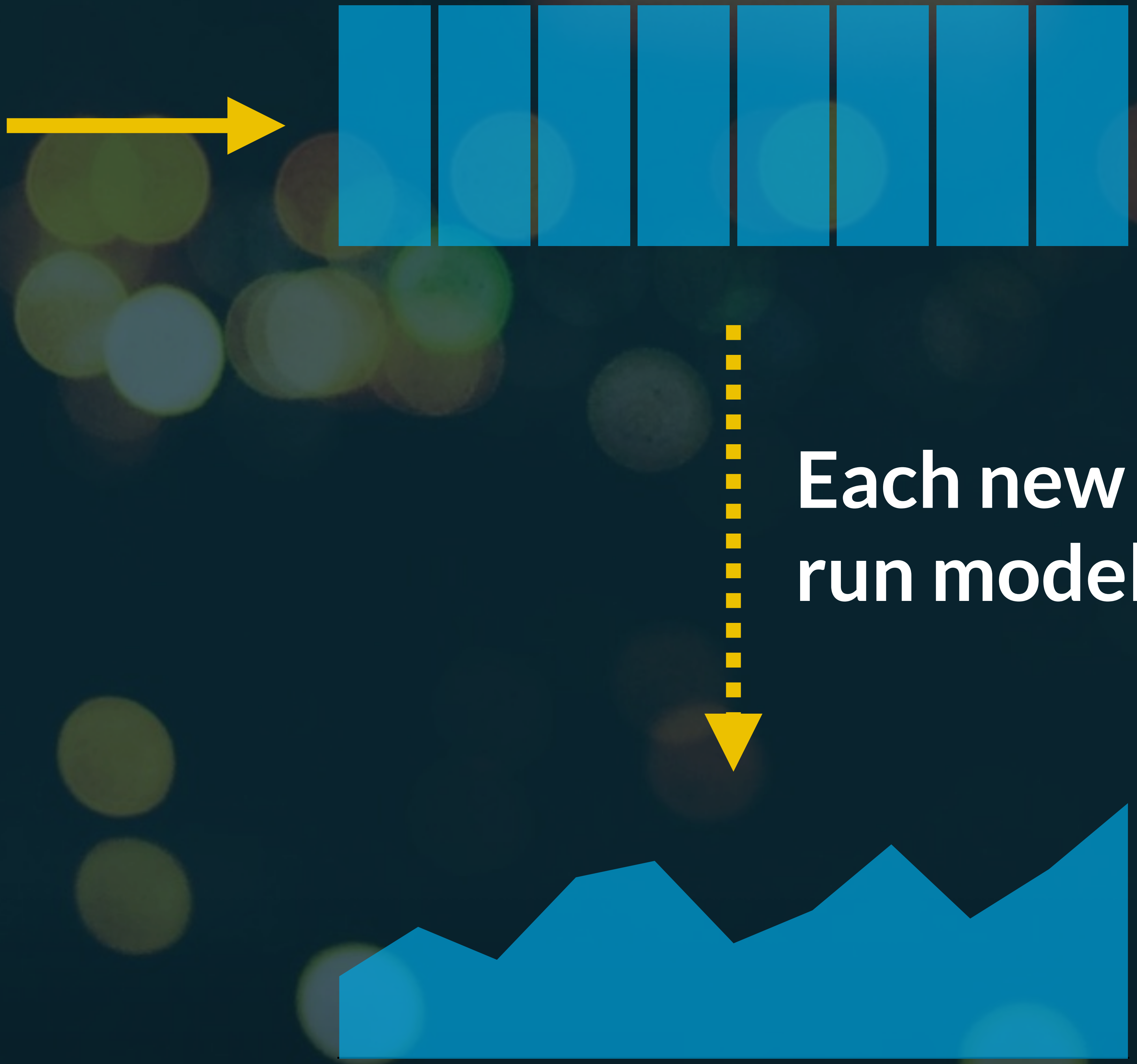
- With seasonal and trend effects removed


1% Anomalies (alpha=0.05, direction=pos)
CPU Utilisation (%)
118.1
94.6
71.2
47.7
24.3
0.8
Timestamp


0.06% Anomalies (alpha=0.05, direction=pos)
CPU Utilisation (%)
118.1
94.6
71.2
47.7
24.3
0.8
Timestamp


8.79% Anomalies (alpha=0.05, direction=pos)
CPU Utilisation (%)
118.1
94.6
71.2
47.7
24.3
0.8
Timestamp

Statistical Model



Stream Metrics

Sliding window of observations *(1 month, 1 year?)*

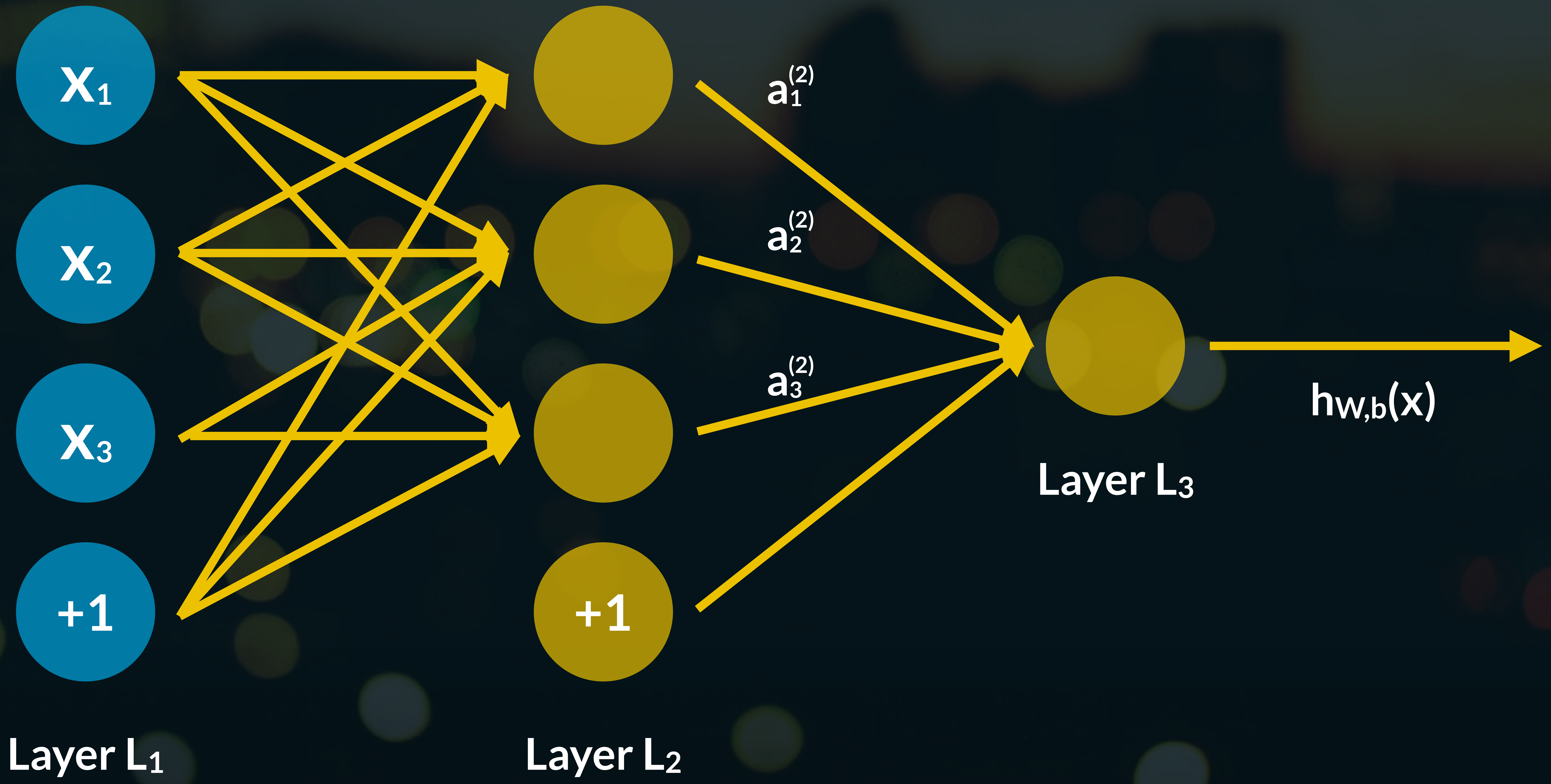
Each new observation run model *(S - H - ESD)*

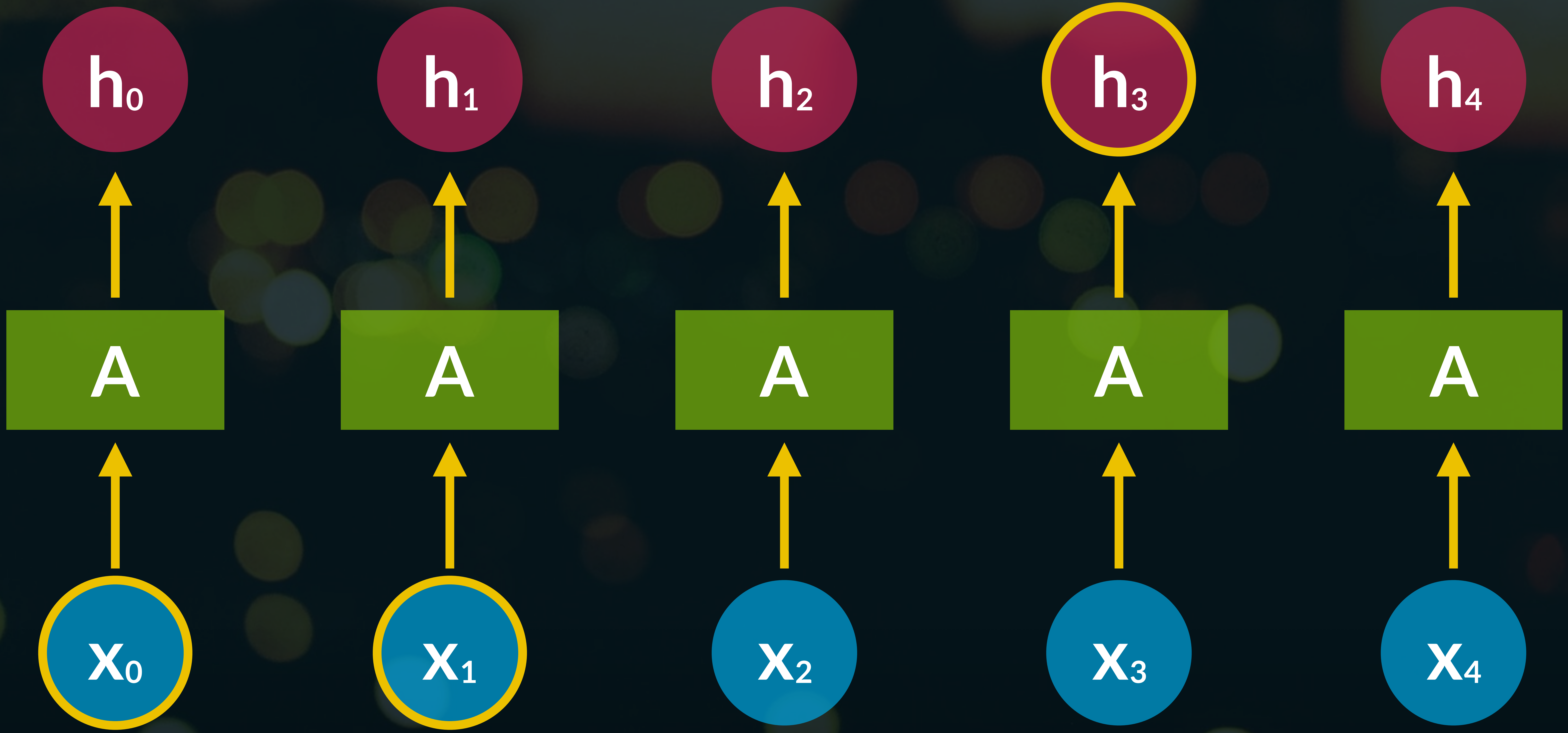
Is the new point an outlier?

Predictive Model

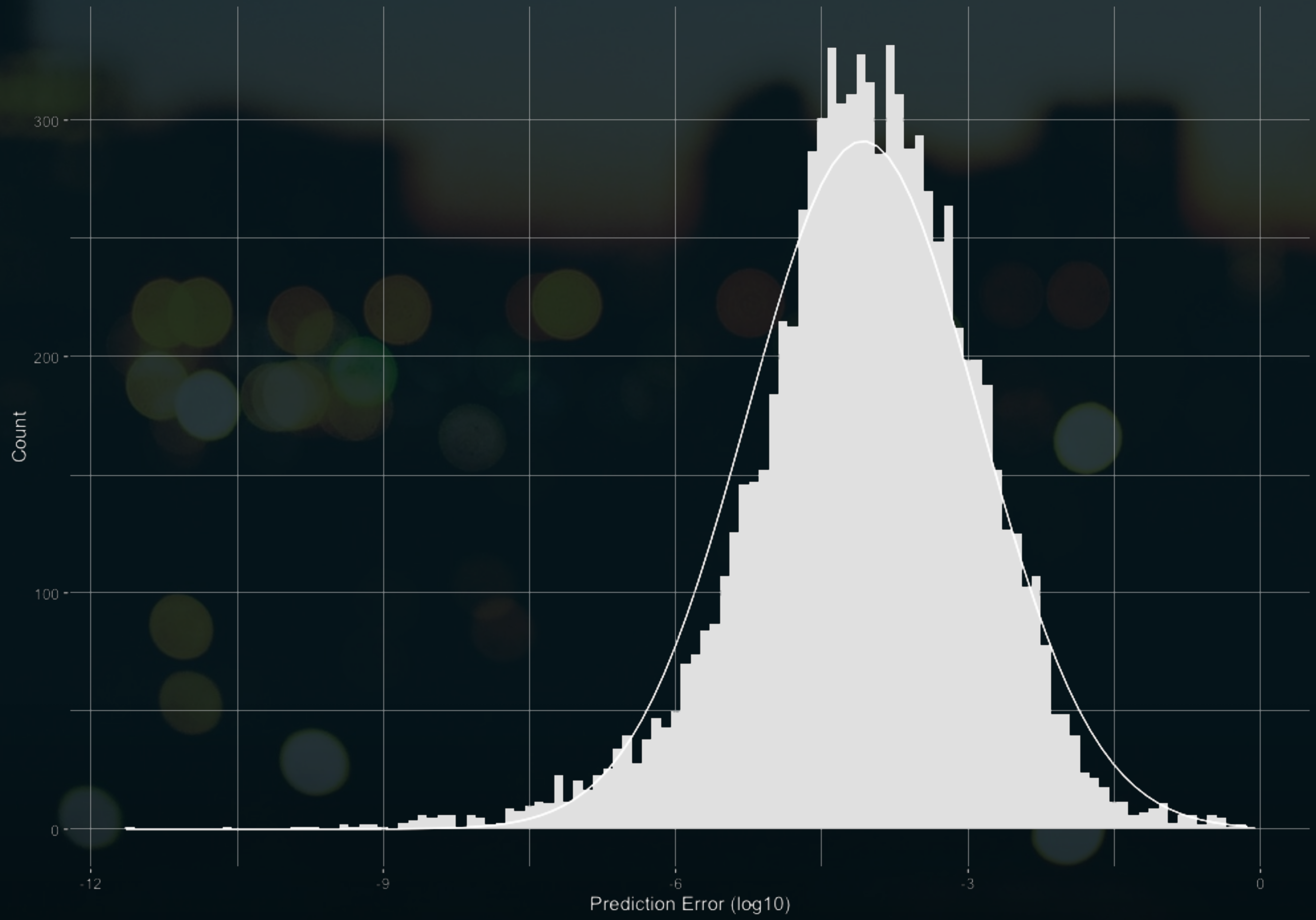
Train a model to predict values in the time series

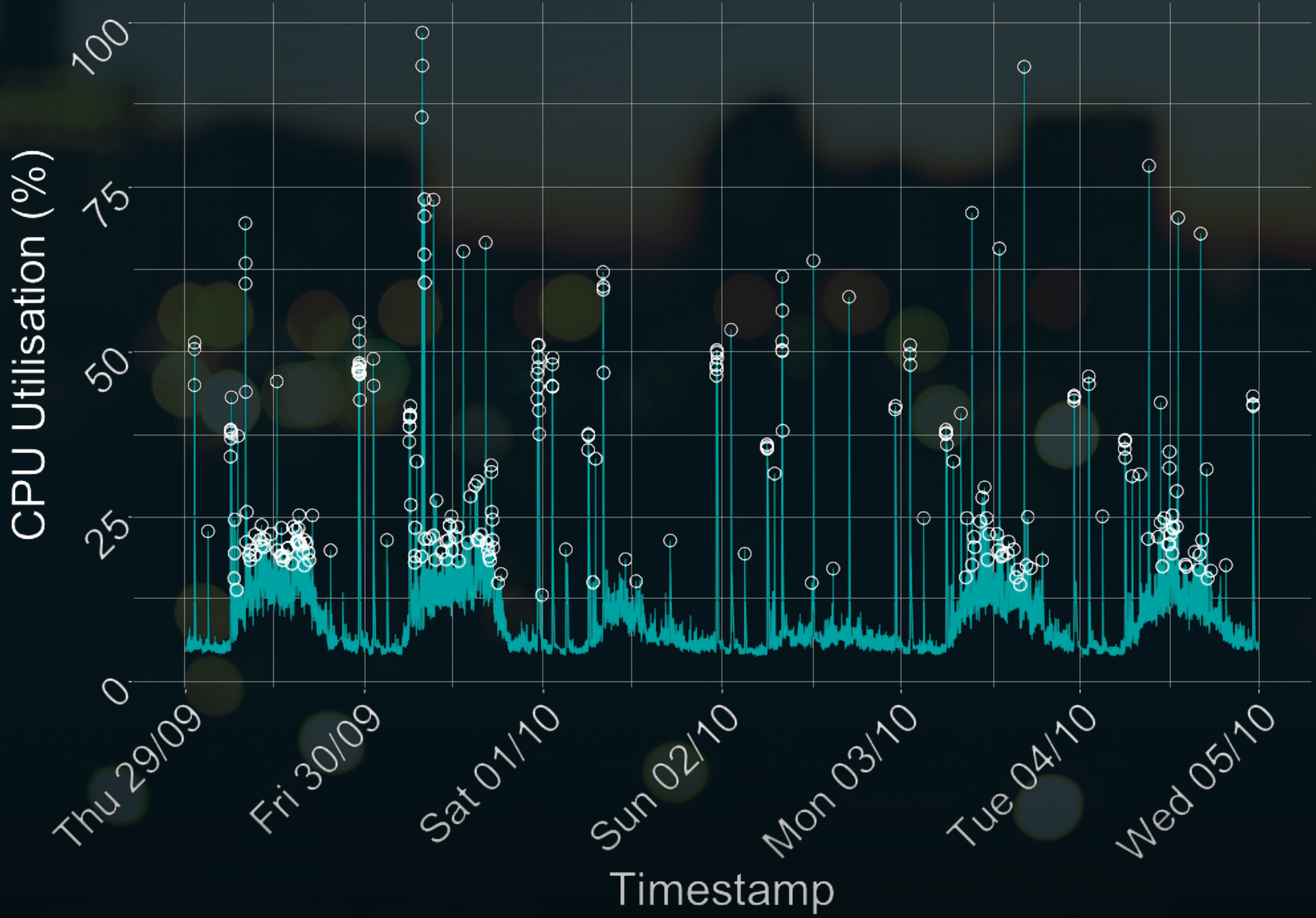
Prediction error > critical value => outlier

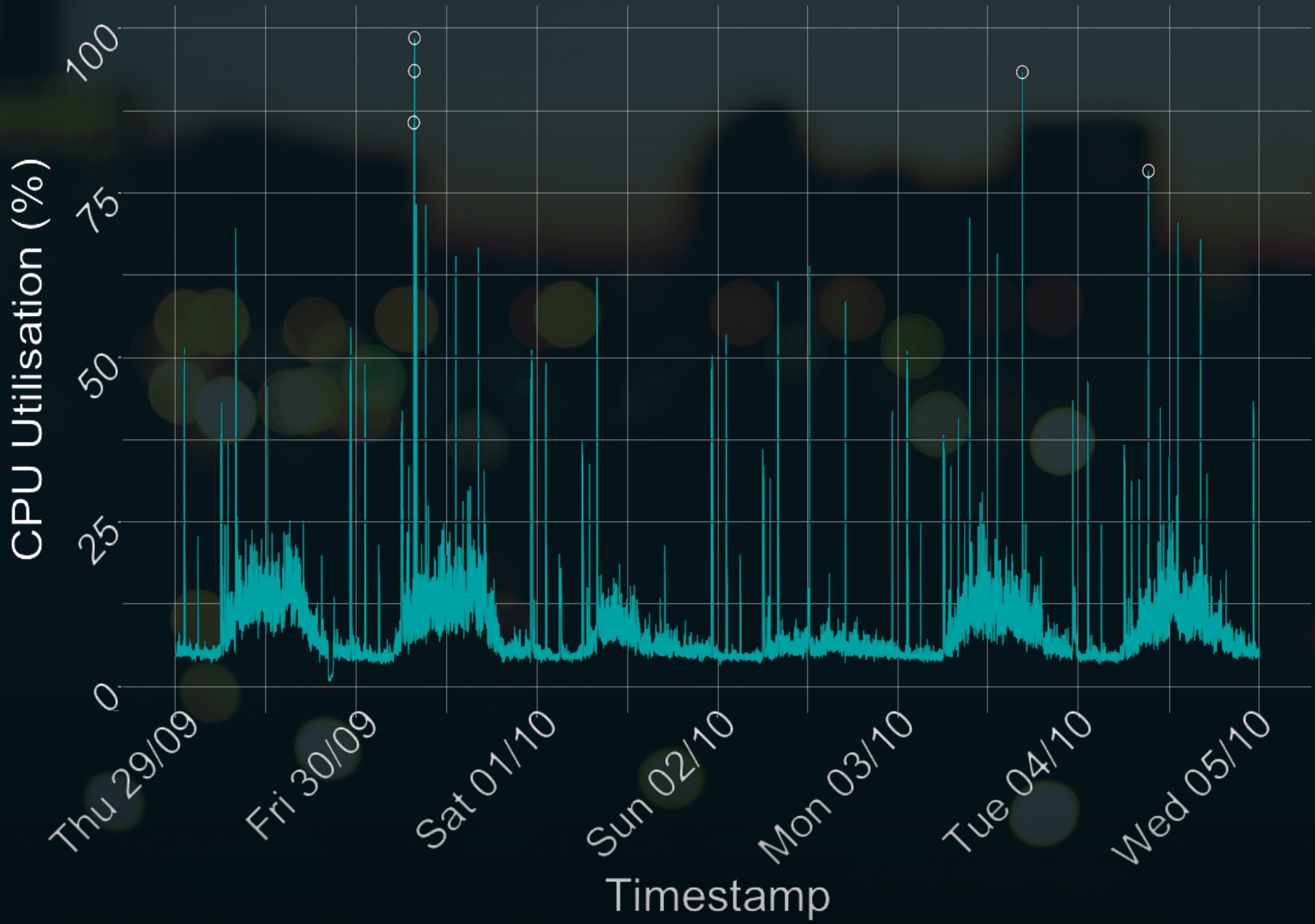

x_1
x_2
x_3
+1
$a_1^{(2)}$
$a_2^{(2)}$
$a_3^{(2)}$
+1
$h_{W,b}(x)$
Layer L_3
Layer L_1
Layer L_2



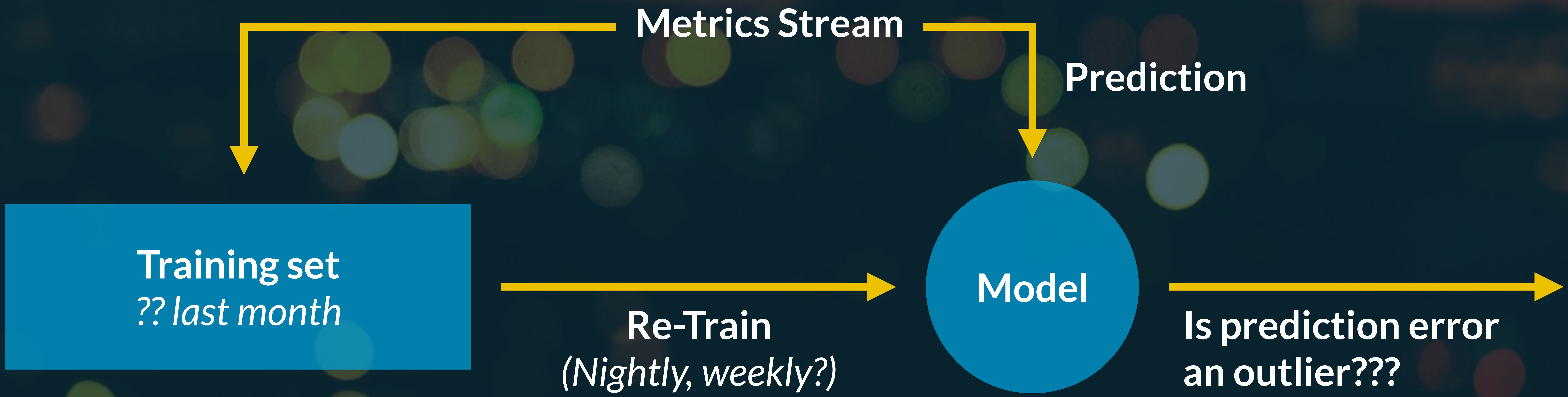
From: http://colah.github.io/posts/2015-08-Understanding-LSTMs/


Count
300
200
100
0
-12
-9
-6
-3
0
Prediction Error (log10)


CPU Utilisation (%)
100
75
50
25
0
Thu 29/09
Fri 30/09
Sat 01/10
Sun 02/10
Mon 03/10
Tue 04/10
Wed 05/10
Timestamp


CPU Utilisation (%)
100
75
50
25
0
Thu 29/09
Fri 30/09
Sat 01/10
Sun 02/10
Mon 03/10
Tue 04/10
Wed 05/10
Timestamp

Predictive Model



Metrics Stream

Prediction

Training set
?? last month

Re-Train
(Nightly, weekly?)

Model

Is prediction error an outlier???

Handling Anomalies

Actionable alerts

- Confidence in predictions

No alerts for pointless things

Handling Anomalies

Taking action

- Rewiring services to read-replica?
- Kill long-running queries?

Handling Anomalies

Confidence in the model leads to confidence in automation

Summary

Increasing complexity and deployment speed **make operational automation a must**

We **must** build services that are **ready for automation**

Simple models can often **beat complex ones**

Cheap compute and storage makes **large-scale ML available to everyone**

Thank You