

**ZABBIX** '25  
CONFERENCE  
GERMANY

# AI-based Anomaly Detection in Zabbix using IntelliTrend Prisma



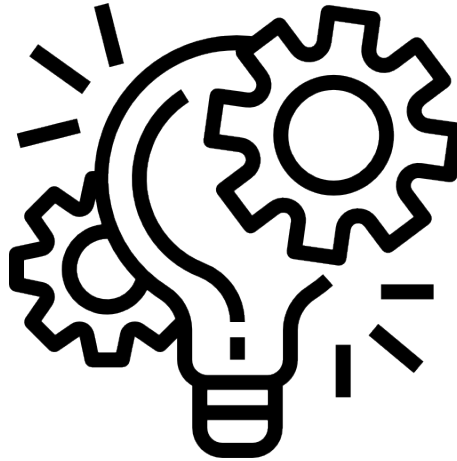
**Wolfgang Alper**

CEO  
IntelliTrend GmbH



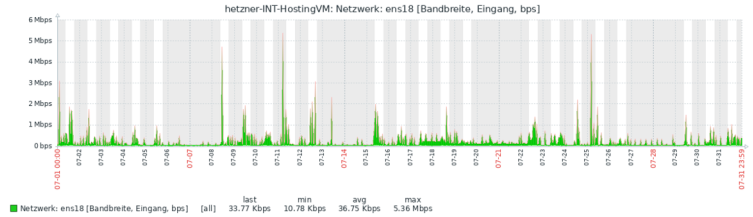
# Zabbix meets AI

## The idea of anomaly detection

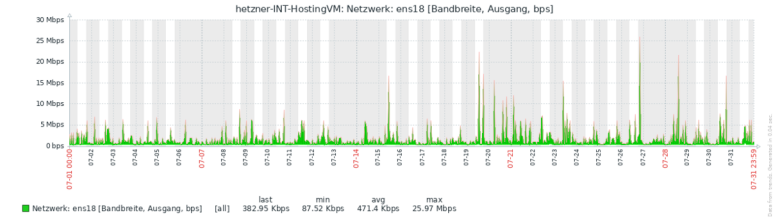


# The idea of anomaly detection

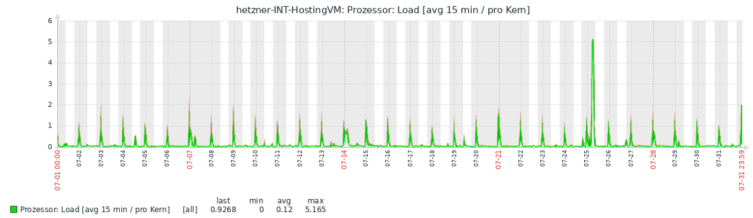
## Network In



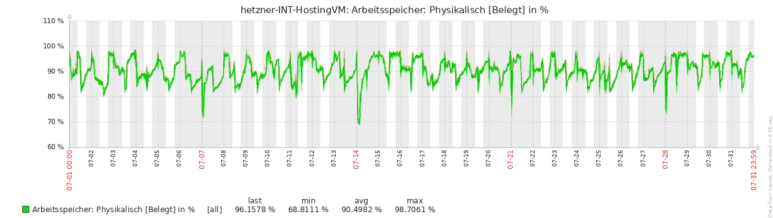
## Network Out



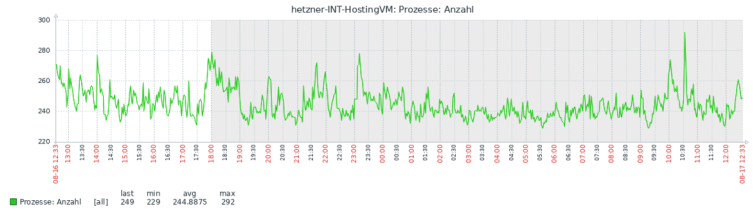
## CPU Load



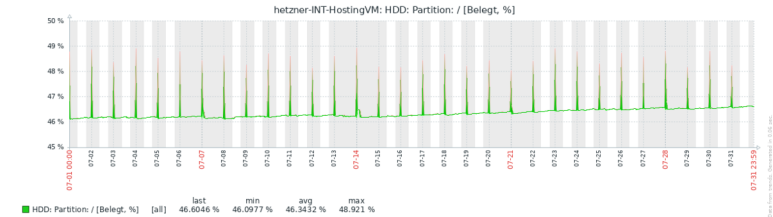
## Memory Used



## Processes Count



## Free Disk Space



## Generic metrics of a system

# The idea of anomaly detection

Place individual metrics in a system-specific **context** to detect **anomalies**

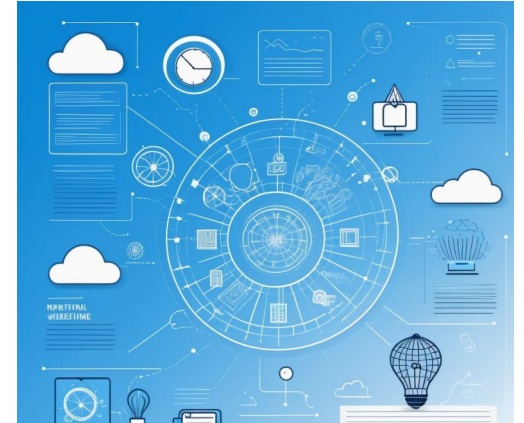
- Instead of looking to **one metric** at a time, have a system that looks at **multiple metrics** at once
- Instead of using **simple** trigger functions, look at the data **as a whole** over a period of time
- Instead of using triggers with **static conditions**, let the system “learn” the specific characteristics over time with **variable conditions**
- Allow this concept to be used across **multiple hosts**



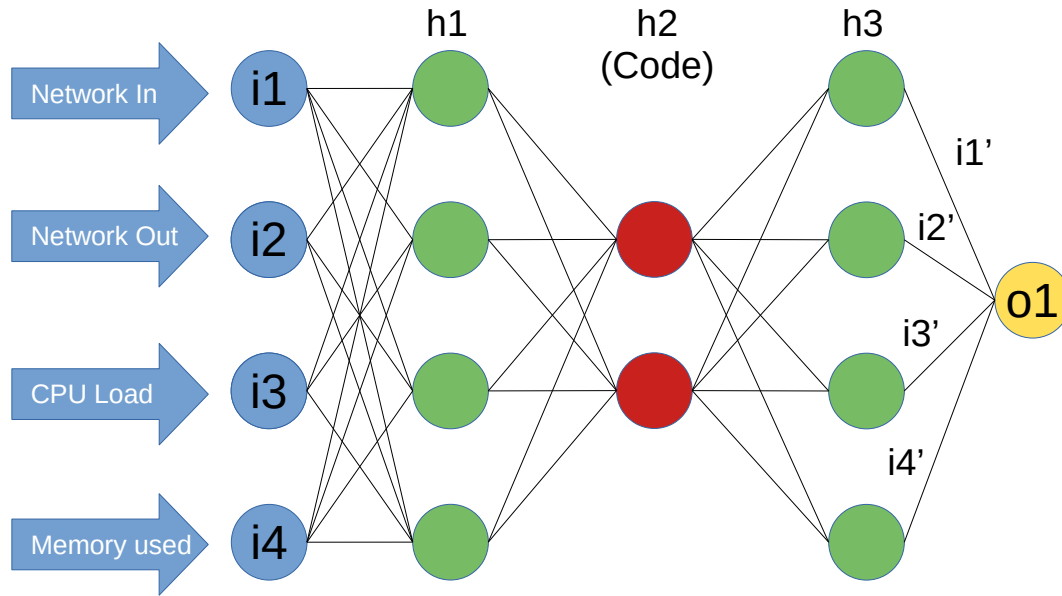
# The idea of anomaly detection

## Examples

- Monitoring server rooms:  
Pay attention not only to the temperature, but also relate it to the power consumption of the systems, air conditioning, etc.
- Server utilization:  
Pay attention not only to CPU utilization, but also to memory usage, the number of users, network traffic, etc.



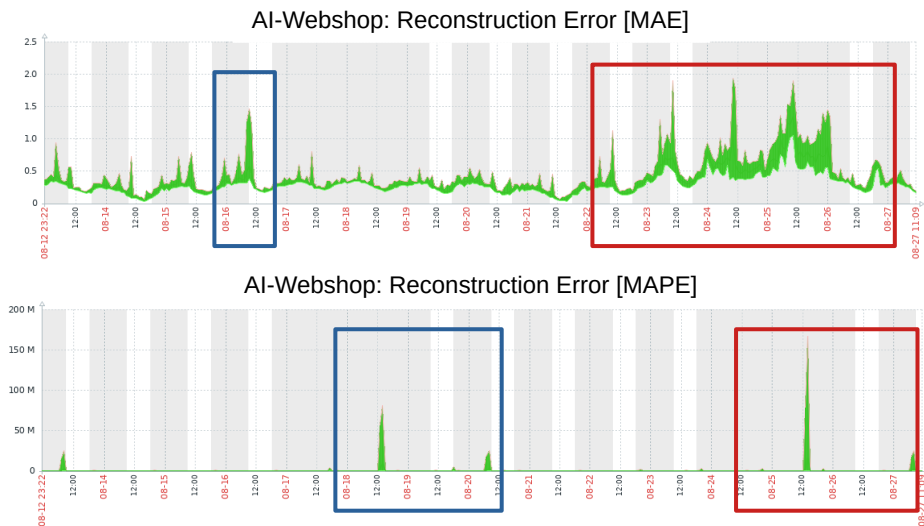
# Long short-term memory based Autoencoder



- Incoming data is connected to i1 to i4 (ix)
- Hidden layer “h1” acts as an **encoder** to extract the features of that data
- Hidden layer “h2” is much smaller and extracts the essence of these features
- Hidden layer “h3” acts as a **decoder** to reconstruct the original data
- If there is a **problem**, hidden layer “h3” will cause a data **reconstruction error**
- This error is calculated as an **anomaly** using a “loss function” in “o1”

# Using a model in production

Reconstruction errors represent the probabilities for anomalies based on their loss function



Example trigger expressions in Zabbix to alert an anomaly:

```
avg(/AI-Webshop/1.loss.mae,600) >={ $MIN_ANOMALY_MAE }  
last(/AI-Webshop/m1.loss.mape) >={ $MIN_ANOMALY_MAPE }  
min(/AI-Webshop/1.loss.mse,300) >={ $MIN_ANOMALY_MSE }  
min(/AI-Webshop/1.loss.msle,300) >={ $MIN_ANOMALY_MSLE }
```

Available loss functions to indicate an anomaly:

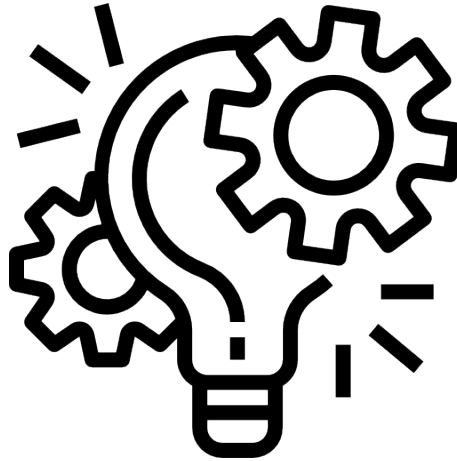
MAE	- Mean Absolute Error
MAPE	- Mean Absolute Percentage Error
MSE	- Mean Squared Error
MSLE	- Mean Squared Logarithmic Error



**ZABBIX**  
PREMIUM PARTNER

# AI-based anomaly detection

## Integration with Zabbix





# Data extraction from Zabbix

Use of **Tags** to identify the **items** that should be used for **training** and later for **inference**

<input type="checkbox"/> Host	Name ▲	Last check	Last value	Change	Tags
<input type="checkbox"/> IMS-Smart IntelliTrend...	Air pressure [Pa] ?	25s	100.81 KPa	+2.6562 Pa	df-ai: df-ai Application: Sensor d...
<input type="checkbox"/> IMS-Smart IntelliTrend...	Humidity [%] ?	25s	39.7236 %	-3.2168 %	df-ai: df-ai Application: Sensor d...
<input type="checkbox"/> IMS-Smart IntelliTrend...	Temperature [C] ?	25s	26.49 C		df-ai: df-ai Application: Sensor d...
<input type="checkbox"/> IMS-Smart IntelliTrend...	Total volatile organic compound TVOC [ppb] ?	25s	39 ppb	+6 ppb	df-ai: df-ai Application: Sensor d...

API  
Request

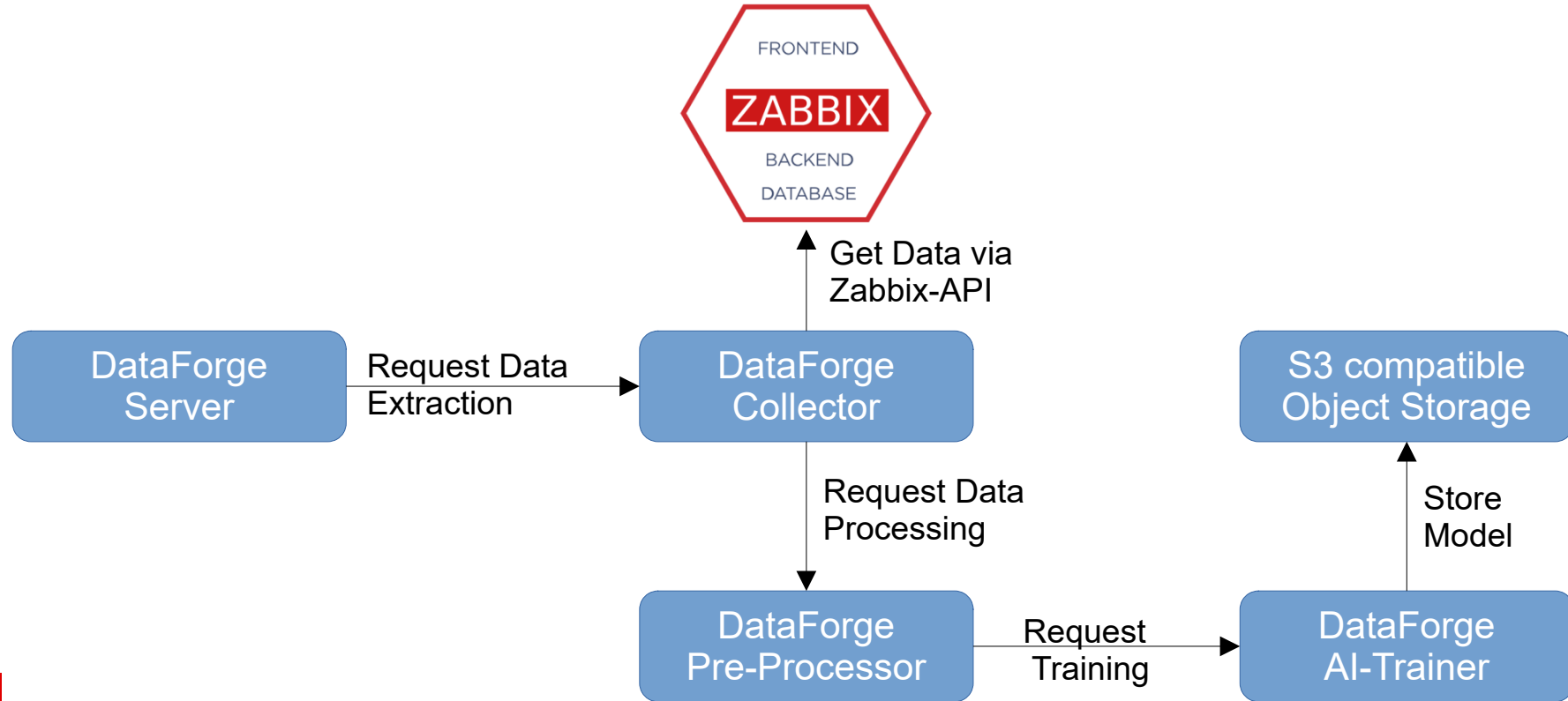
```
{
  "jsonrpc": "2.0",
  "method": "history.get",
  "params": {
    "output": "extend",
    "history": 0,
    "itemids": "912257",
    "sortfield": "clock",
    "sortorder": "DESC",
    "limit": 10
  },
  "id": 1
}
```

API  
Response

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "itemid": "912257",
      "clock": "1724588945",
      "value": "26.31999969",
      "ns": "227934509"
    },
    {
      "itemid": "912257",
      "clock": "1724588885",
      "value": "26.32999992",
      "ns": "87916196"
    }
  ]
}
```

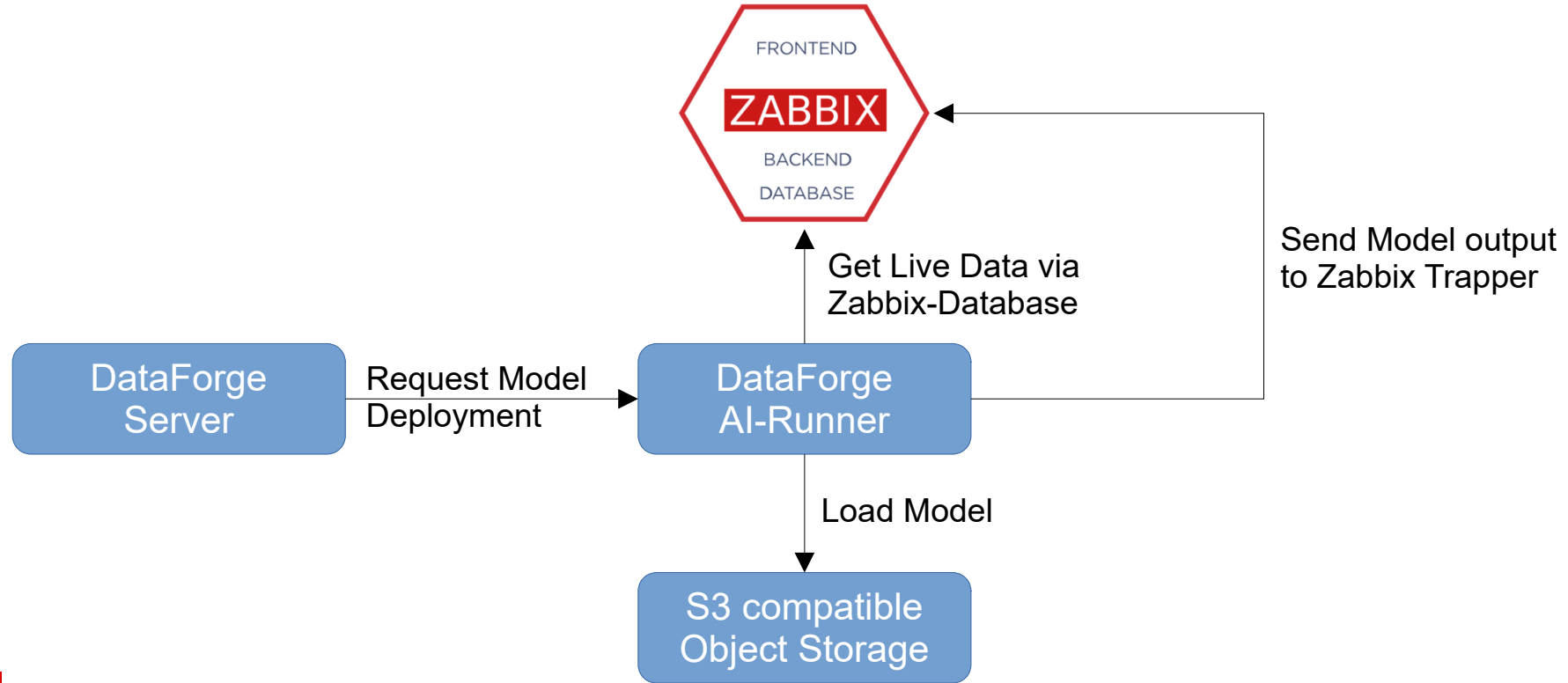
Using the Zabbix-API to get **univariate time-series** data for a given item

# Integration with Zabbix - Training



No changes to Zabbix Source code required

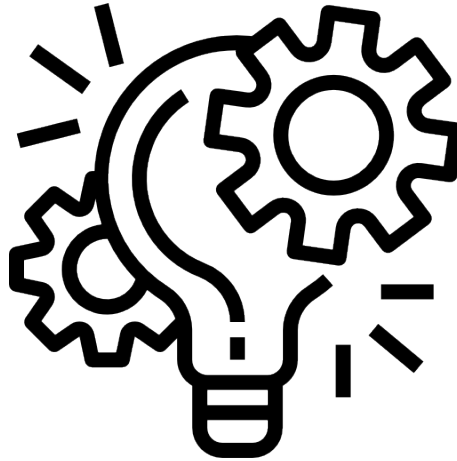
# Integration with Zabbix - Inference



No changes to Zabbix Source code required

# AI-based anomaly detection

## Detecting simple anomalies

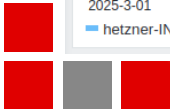
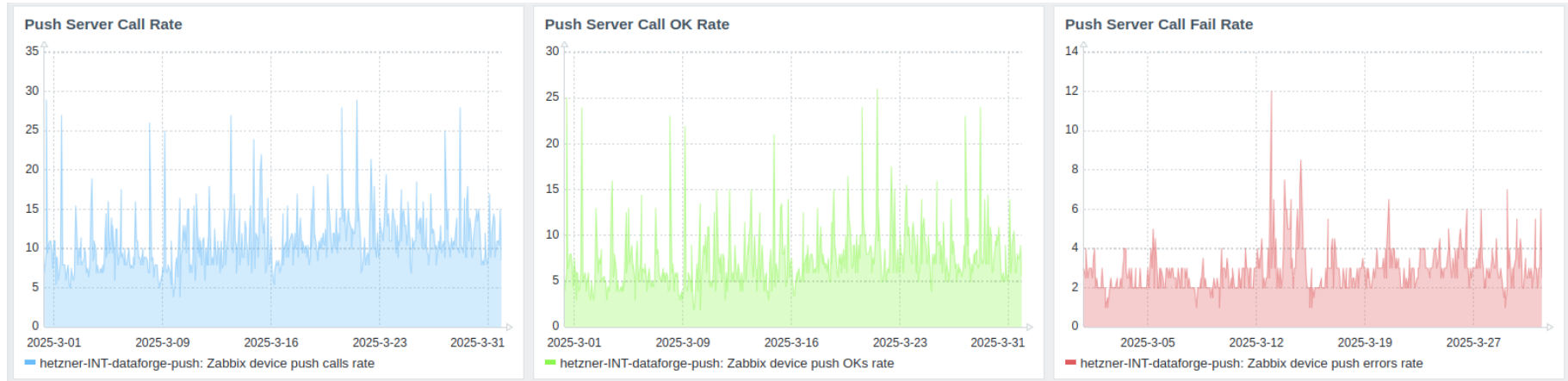


# Detecting simple anomalies

These are the three most important metrics of our Push Notification Server:

- “Call Rate”, “Call OK Rate” and “Call Fail Rate”
- These rates indicate the number of attempted, successful and failed push notification requests over time

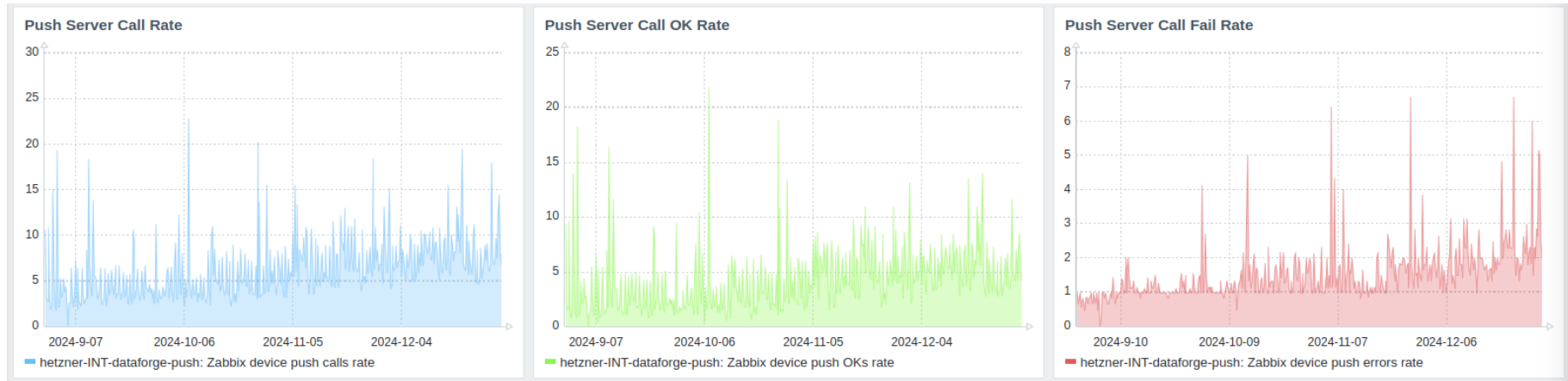
The failure rate of around 20%-30% is based on users who reinstall / uninstall the app, but leave the mediatype configuration as is on their Zabbix user account



# Detecting simple anomalies

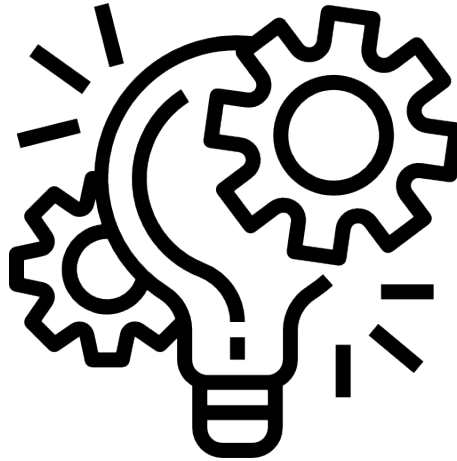
Writing a trigger for these metrics is non-trivial:

- Large spikes can be expected when a customer has a large number of issues that trigger many push notifications
- Looking at a longer time interval all metrics are trending upwards as the application gains users



# AI-based anomaly detection

How can we solve this with AI?



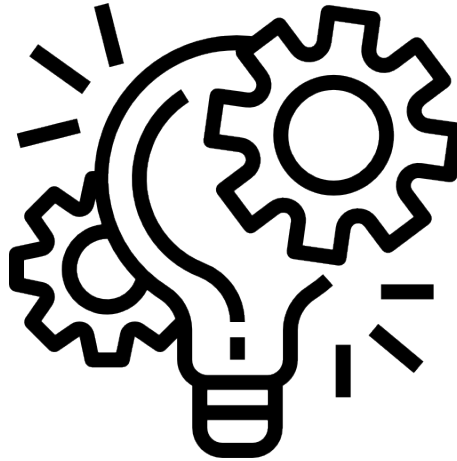
# From data to model

- Create a training dataset with history data from Zabbix
- Create one or more test datasets with history data from Zabbix
- Create a model configuration
- Create / Train a model
- Test the model
- Assign the the model to an AI-Runner



# AI-based anomaly detection

## Datasets



# Tagging Items for Data Extraction

- Mark metrics that AI should use by assigning the tag “df-ai” to relevant items in Zabbix
- These tags will be used to extract the data from Zabbix for:
  - Training the model
  - Testing the model
  - Using the model in production (Inference)

<input type="checkbox"/>	Name ▲	Triggers	Key	Interval	History	Trends	Type	Status	Tags	li
<input type="checkbox"/>	*** DataForge Push Server: Push server raw data: Zabbix device push calls rate	Triggers 2	dfpush.zabbix_device_push_calls_rate		90d	365d	Dependent item	Enabled	df-ai	
<input type="checkbox"/>	*** DataForge Push Server: Push server raw data: Zabbix device push errors/calls absolute ratio	Triggers 2	dfpush.zabbix_device_push_errors_calls_ratio		90d	365d	Dependent item	Enabled	df-ai	
<input type="checkbox"/>	*** DataForge Push Server: Zabbix device push errors/calls recent ratio	Triggers 2	dfpush.zabbix_device_push_errors_calls_recent_ratio	1m	90d	365d	Calculated	Enabled	df-ai	
<input type="checkbox"/>	*** DataForge Push Server: Push server raw data: Zabbix device push errors rate		dfpush.zabbix_device_push_errors_rate		90d	365d	Dependent item	Enabled	df-ai	

# Creating a dataset

- Add host(s) with tagged items
- Choose scheduling mode for automatic update
- Set storage duration for house keeping
- Datasets needs to be created for:
  - Training
  - Testing

< Dataset config details Save

**General**

Name  
IntelliTrend Mobile Push Notification Server Dataset

Description  
Datasets related to the IntelliTrend Mobile Push Notification Server

**Dataset configuration**

hetzner-INT-dataforge-push (5 AI items) ×

Hosts  
Choose the hosts you want to extract data from. Only items associated with the "df-ai" tag will be included in the extraction.

Schedule mode Monthly ▾

**Storage options**

Storage duration Days 90 Hours 0

Specify how long the data will be stored in the S3 bucket. This sets the TTL (time to live) for the data, and S3 will automatically delete it after the specified time has passed. Set to 0 for indefinite storage.

**Datasets** Extract data

There have been no datasets extracted for this configuration yet. [Click here to start a dataset extraction job.](#)

# Creating a dataset

- After the data extraction process finished, dataset details are updated
- The analysis is helpful to decide whether the dataset will be useful in a training or in a test
- Datasets can be downloaded, decompressed and converted to a variety of formats for further processing in 3th party applications

### < Dataset details

**Extraction completed**  
**Training Dataset**  
Dataset used to train anomaly detection models

**General**

Configured start time	February 1st, 00:00:00
Configured end time	March 1st, 00:00:00
Effective start time ⓘ	February 1st, 00:05:41
Total datapoints available	163.85 K
Missing data points ⓘ	119.00 (0.07%)
Unusable data points ⓘ	19.00 (0.01%)
Dataset size (compressed)	658.67 KiB

**Items**

**Regular time series** • Median update interval: 1m 0s  
**dfpush.zabbix\_device\_push\_calls\_rate**

Total datapoints available	40.29 K
Missing data points ⓘ	32.00 (0.08%)
Unusable data points ⓘ	5.00 (0.01%)

**Regular time series** • Median update interval: 1m 0s  
**dfpush.zabbix\_device\_push\_errors\_calls\_ratio**

Total datapoints available	40.29 K
Missing data points ⓘ	31.00 (0.08%)
Unusable data points ⓘ	5.00 (0.01%)

**Regular time series** • Median update interval: 1m 0s  
**dfpush.zabbix\_device\_push\_errors\_calls\_recent\_ratio**

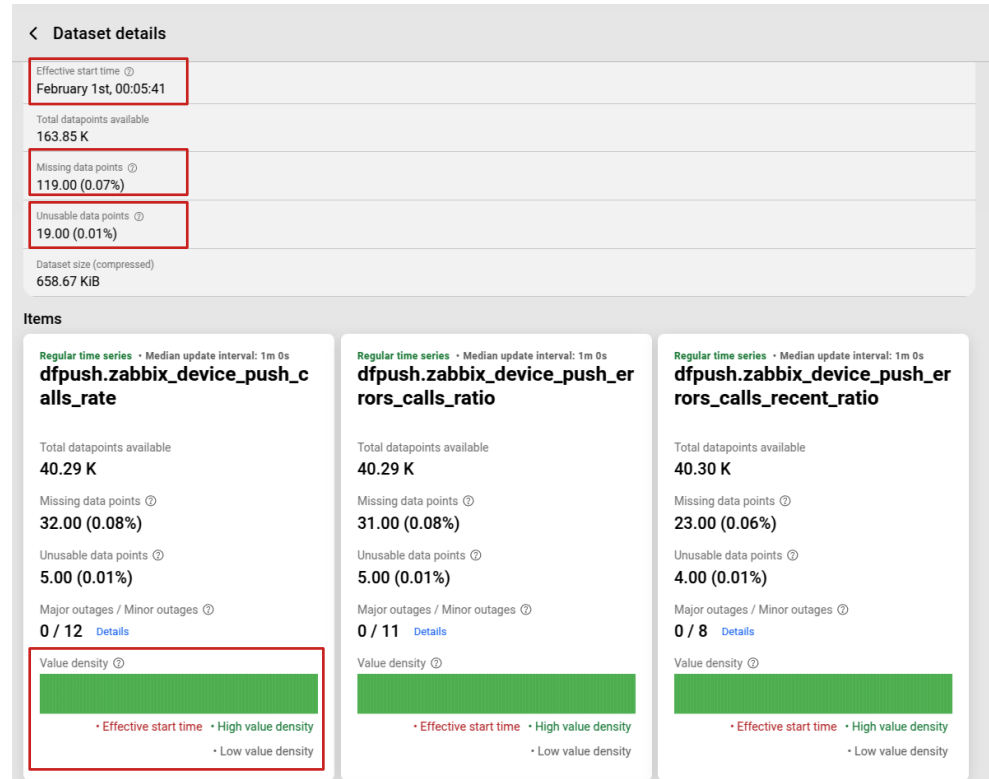
Total datapoints available	40.30 K
Missing data points ⓘ	23.00 (0.06%)
Unusable data points ⓘ	4.00 (0.01%)



# Dataset – Quality of data

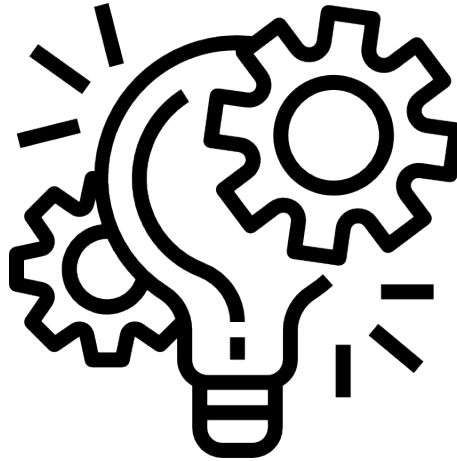
## Analysis of the dataset:

- This dataset has a high value density and looks healthy
- Very few datapoints are missing or unusable
- All of the time series were detected as having a consistent update interval
- There are no major outages and an evenly distributed value density
- The effective start time of the dataset (the time at which each time series had at least one value) is only 5m41s after the configured start time

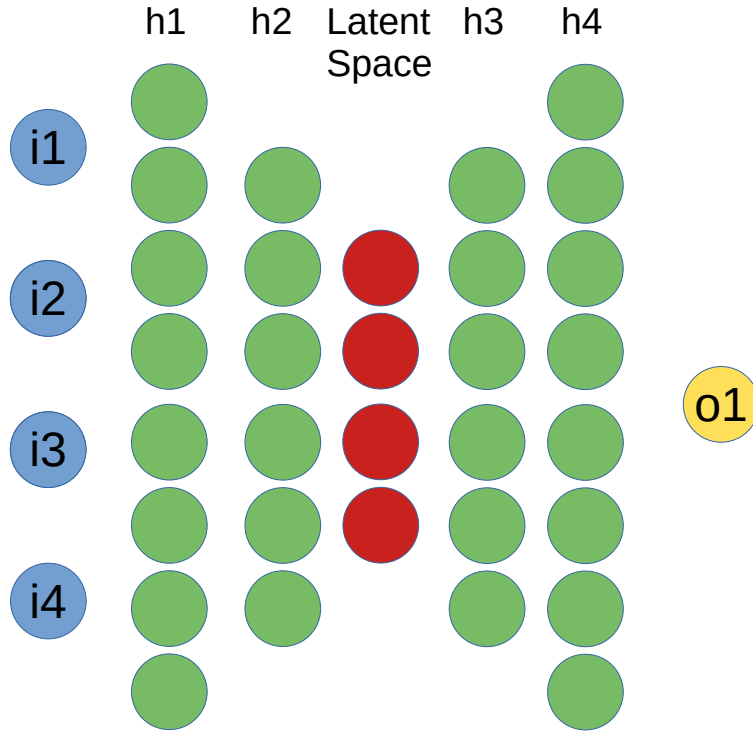


# AI-based anomaly detection

## Creating a Model



# Model configuration



**< Model config details** Save

**General**

Name  
IntelliTrend Mobile Push Notification Server Anomaly Detector

Description  
LSTM-Autoencoder based anomaly detection model for the IntelliTrend Mobile push notification server

Retrain on data update ☐

Dataset  
Training Dataset (March 25th, 15:38) (#27)

AI trainer  
AI-CUDA-24G (Online) ▼

**Architecture**  
Refer to the [documentation](#) for information about model architectures.

Architecture  
LSTM Autoencoder ▼

Latent space size factor  
0.5

Model depth

Model width  
4

Model slope  
0.8

Activation  
ReLU ▼

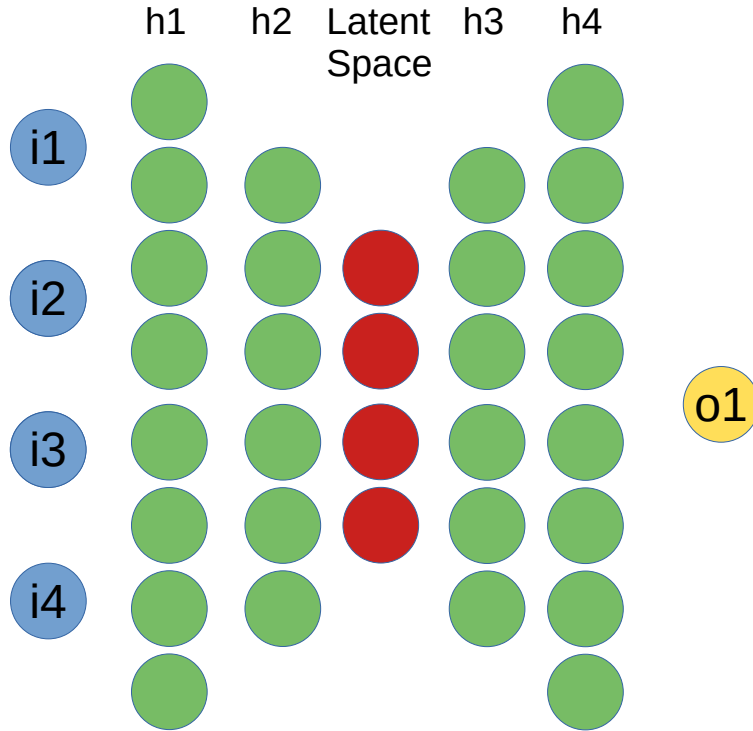
Use dropout ☐

**Training**  
The training options include options that change the training behavior.  
Refer to the [documentation](#) for information about training options.

Epochs  
30

Architecture defines model type, number of layers / neurons, activation function etc.

# Model configuration



< Model config details Save

Refer to the [documentation](#) for information about training options.

Epochs	30
Loss	MSE ▾
Optimizer	Adam ▾

**Preprocessing**

Before passing the data from the dataset to the model training, they are preprocessed to improve model training performance. Refer to the [documentation](#) for information about preprocessing options.

Scaler	MIN_MAX ▾
Use outlier clipping	<input checked="" type="checkbox"/>
Outlier threshold	3
Window size	16
Time step	60
Batch size	64 ▾

🔗 The number of batches per epoch are 629 (total: 18870) with 32 KiB per batch, (total: 589.69 MiB)

**Storage options**

Storage duration	Days 90	Hours 0
------------------	---------	---------

Specify how long the data will be stored in the S3 bucket. This sets the TTL (time to live) for the data, and S3 will automatically delete it after the specified time has passed. Set to 0 for indefinite storage.

**Models** Train now

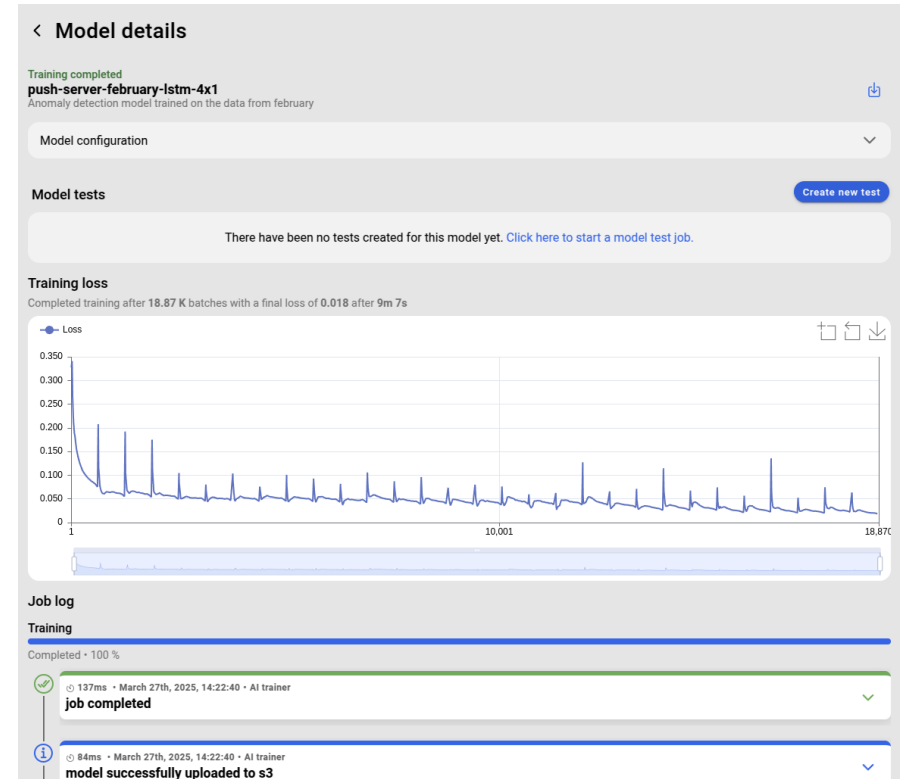
There have been no models trained for this configuration yet. [Click here to start a model training job.](#)

Preprocessing defines operations on the dataset for training including batch size



# Model creation

- Model is created based on model configuration and training dataset
- The number of selected metrics (items) effected the number of neurons
- Model details show the training loss during the training process

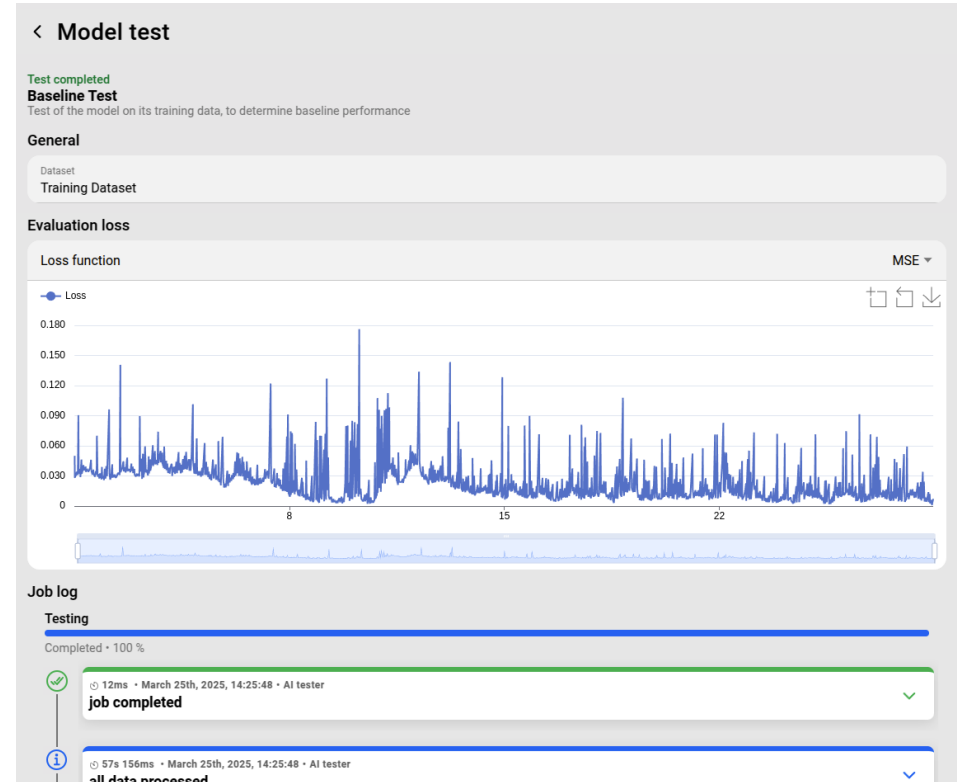


# Model test

- After the model is created, it can be tested using datasets
- Typically there is a baseline test and also validation tests
- Any number of datasets can be created from Zabbix history data
- Different loss functions can be used

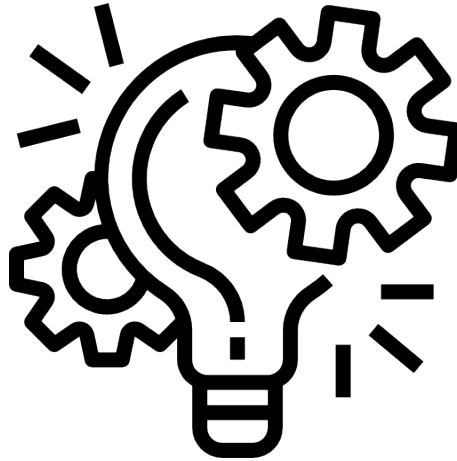
Available loss functions to indicate an anomaly:

MAE	- <b>M</b> ean <b>A</b> bsolute <b>E</b> rror
MAPE	- <b>M</b> ean <b>A</b> bsolute <b>P</b> ercentage <b>E</b> rror
MSE	- <b>M</b> ean <b>S</b> quared <b>E</b> rror
MSLE	- <b>M</b> ean <b>S</b> quared <b>L</b> ogarithmic <b>E</b> rror



# AI-based anomaly detection

## Using the model



# Creating output items

- The result of the model evaluation is sent to Zabbix using items
- Item type must be trapper with type numeric (float)
- The item keys must use a consistent prefix and then be suffixed with the appropriate loss metric like “df.push.msle”

Item

Item Tags Preprocessing

\* Name Reconstruction Error: MSLE

Type Zabbix trapper

\* Key df.push.msle Select

Type of information Numeric (float)

Units

\* History Do not store Store up to 31d

\* Trends Do not store Store up to 365d

Value mapping type here to search Select

Allowed hosts

Populates host inventory field -None-

	Name ▲	Triggers	Key	Interval	History	Trends	Type	Status	Ti
<input type="checkbox"/>	*** Reconstruction Error: MAE		df.push.mae		31d	365d	Zabbix trapper	Enabled	
<input type="checkbox"/>	*** Reconstruction Error: MAPE		df.push.mape		31d	365d	Zabbix trapper	Enabled	
<input type="checkbox"/>	*** Reconstruction Error: MSE		df.push.mse		31d	365d	Zabbix trapper	Enabled	
<input type="checkbox"/>	*** Reconstruction Error: MSLE		df.push.msle		31d	365d	Zabbix trapper	Enabled	



# Deploying the model

- The process of actually using a model is called inference
- The inference configuration defines the model, AI-Runner and the host in Zabbix to receive the model output

The screenshot displays the 'Inferences' section of the Zabbix web interface. A modal dialog titled 'Create an inference configuration' is open, allowing the user to define a new inference setup. The dialog contains the following fields and values:

- Name:** IntelliTrend Mobile Push Notification Server Inference
- Model config:** IntelliTrend Mobile Push Notification Server Anomaly Detector
- Model:** push-server-february-istm-4x1 (March 26th, 12:56) (#55)
- AI runner:** NMS-Runner
- Inference Zabbix host name:** dataforge-push-anomaly-detection
- Zabbix loss item key prefix:** df.push

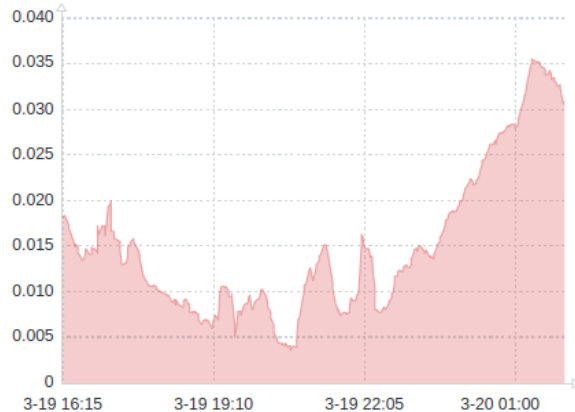
Below the form fields, a note explains the Zabbix item key prefix: 'The Zabbix item key prefix to which the function name is appended. For example, if the item key prefix is example.loss and the selected loss function is MSE, the resulting metric will be sent to example.loss.mse.'

A blue 'Create' button is located at the bottom right of the dialog. The background interface shows a message: 'No inference configs have been created yet.' with a 'Learn more' link. A search bar and a '+' button are also visible.

# Deploying the model

- After deploying the model, the output for each loss function shows up as item in Zabbix
- These items can be used with any trigger function in Zabbix to detect an anomaly

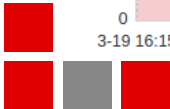
Push Server Reconstruction Error: MSLE



Push Server Reconstruction Error: MSE



Push Server Reconstruction Error: MAE

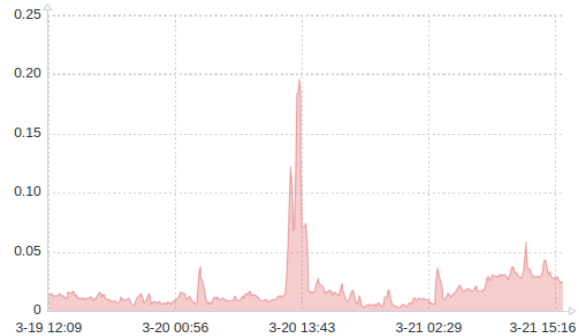


# Detecting an anomaly

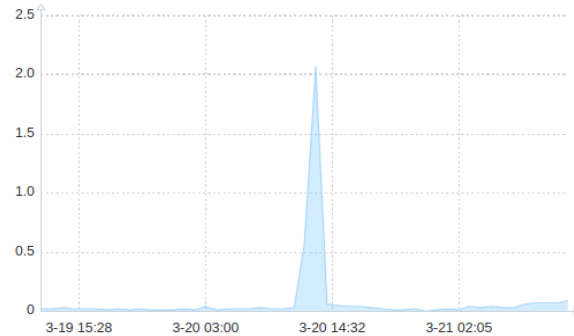
When an anomaly is detected, the values of the loss functions change significantly

For example MSE (Mean Squared Error) is more sensible than MAE (Mean absolute Error)

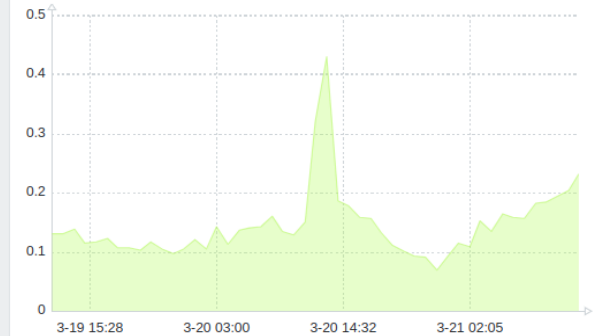
Push Server Reconstruction Error: MSLE



Push Server Reconstruction Error: MSE

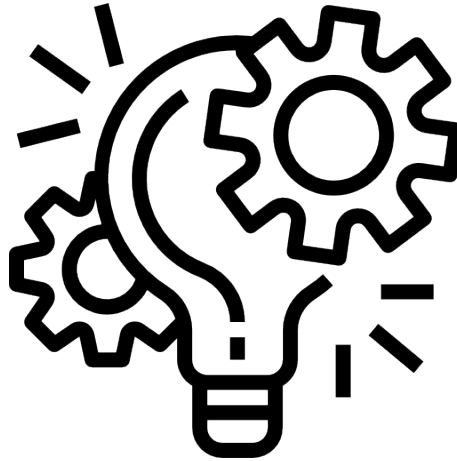


Push Server Reconstruction Error: MAE



# AI-based anomaly detection

Lets look at more examples

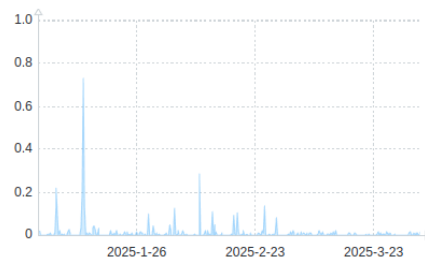




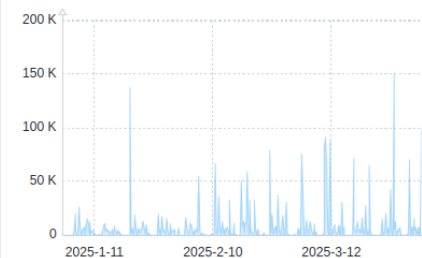
# Example Exchange Server

- Below some KPI's from a Microsoft Exchange Server
- The graphs show the mail queue, average mail size in bytes and %ram and %cpu usage
- There are 17 additional items that will be used for training for a total of 21 items

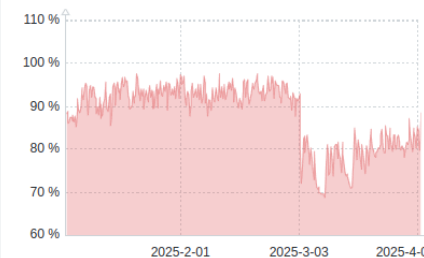
Mail Delivery Queue



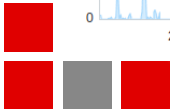
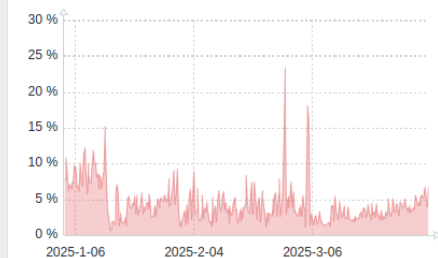
Average Mail Size in Bytes



RAM usage in %



Processor usage in %



# Example Exchange Server

Name ▲	Last check	Last value	Change	Tags
Arbeitsspeicher: Physikalisch [Belegt] in % <sup>2</sup>	1m 47s	87.7376 %	+0.4971 %	df-ai
Arbeitsspeicher: Swap [Belegt] in % <sup>2</sup>	1m 47s	41.9535 %	-0.011 %	df-ai
Exchange: MSEXchange-Datenbank [Information Store / Protokoll: Generierte Bytes/s]	53s	0		df-ai
Exchange: MSEXchange-Datenbank [Information Store / Protokoll: Schreiben Bytes/s]	16s	0		df-ai
Exchange: MSEXchange-Datenbank [Information Store I/O / Datenbanklesevorgänge/s] <sup>2</sup>	1m 30s	0		df-ai
Exchange: MSEXchange-Datenbank [Information Store I/O / Bankschreibvorgänge/s] <sup>2</sup>	1m 39s	0.9874	-0.000143	df-ai
Exchange: MSEXchangeTransport [SMTP-Empfang(_total) / Aktuelle Verbindungen]	53s	2		df-ai
Exchange: MSEXchangeTransport [SMTP-Empfang(_total) / Empfangene Bytes/s]	16s	41373.7277	-116681.7052	df-ai
Exchange: MSEXchangeTransport [SMTP-Empfang(_total) / Empfangene Nachrichten / Delta]	1m 30s	7	-4	df-ai
Exchange: MSEXchangeTransport [SMTP-Empfang(_total) / Empfangene Nachrichten/s] <sup>2</sup>	4s	0		df-ai
Exchange: MSEXchangeTransport [SmtSend(_total) / Aktuelle Verbindungen]	1m 42s	0		df-ai
Exchange: MSEXchangeTransport [SmtSend(_total) / Gesendete Bytes/s]	53s	408635.0496	+378003.4473	df-ai
Exchange: MSEXchangeTransport [SmtSend(_total) / Gesendete Nachrichten / Delta]	4s	14	-2	df-ai
Exchange: MSEXchangeTransport [SmtSend(_total) / Gesendete Nachrichten/s] <sup>2</sup>	11s	1.9698	+0.000057	df-ai
Exchange: MSEXchangeTransport [Zustellungswarteschlange / Warteschlangen extern] <sup>2</sup>	1m 30s	0		df-ai
Exchange: MSEXchangeTransport [Zustellungswarteschlange / Warteschlangen intern] <sup>2</sup>	53s	0		df-ai
Exchange: MSEXchange [Client Type / Messages opened/sec]	1m 42s	2.9536	+2.9536	df-ai
Exchange: MSEXchange [IS Store / Messages opened/sec]	1m 41s	0	-0.9906	df-ai
Exchange: MSEXchange [MapiHttp Emsmdb / aktive Benutzer / Anzahl] <sup>2</sup>	1m 42s	2	+1	df-ai
Netzwerk: Ping [Antwortzeit] <sup>2</sup>	1m 31s	0.53ms	+0.054ms	df-ai
Netzwerk: Ping [Status] <sup>2</sup>	1m 31s	Up (1)		df-ai

Tag items for data extraction

# Example Exchange Server

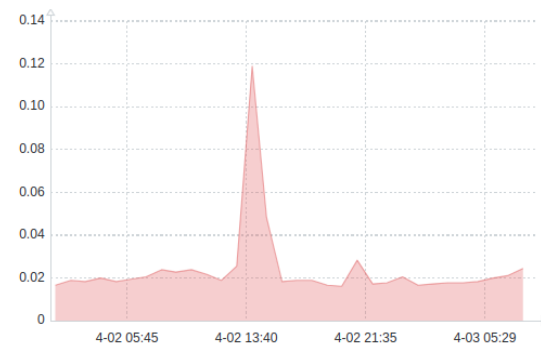
- The training dataset contains 405K datapoints obtained from 21 items over a duration of one month
- The validation dataset contains 464K datapoints obtained from 21 items over a duration of one month
- During the training process with 100 epochs, the dataset will be expanded to 4.8GiB

Extraction completed Validation Dataset	Extraction completed Training Dataset
General	General
Configured start time March 1st, 00:00:00	Configured start time February 1st, 00:00:00
Configured end time April 1st, 00:00:00	Configured end time February 28th, 00:00:00
Effective start time ⓘ March 1st, 00:01:56	Effective start time ⓘ February 1st, 00:01:44
Total datapoints available 464.42 K	Total datapoints available 405.12 K
Missing data points ⓘ 183.00 (0.04%)	Missing data points ⓘ 223.00 (0.06%)
Unusable data points ⓘ 0 (0.00%)	Unusable data points ⓘ 0 (0.00%)
Dataset size (compressed) 1.67 MiB	Dataset size (compressed) 1.44 MiB

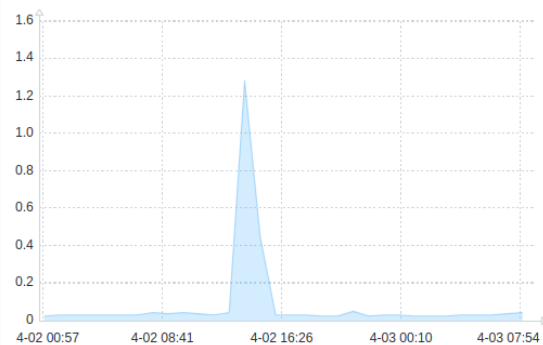
# Example Exchange Server

- We now have created one trigger that monitors a total of 21 related items
- If any of these items shows an unexpected behavior over time or in relation to other items, this trigger will fire
- To get the root cause, further investigation is needed
- If required, specific triggers can be added

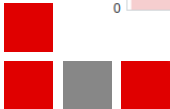
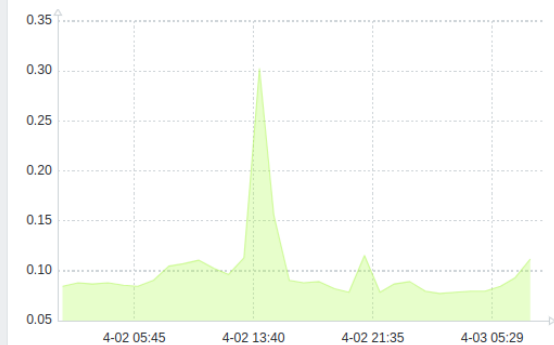
Exchange Server Reconstruction Error: MSLE



Exchange Server Reconstruction Error: MSE



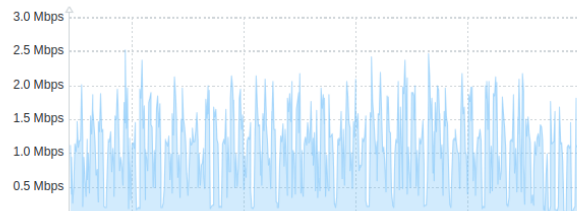
Exchange Server Reconstruction Error: MAE



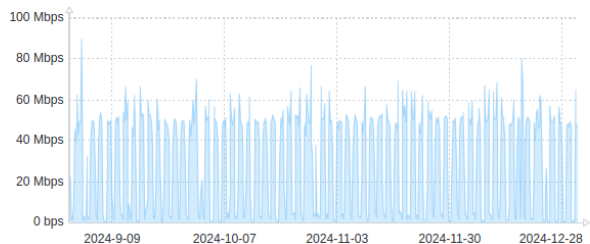
# Example MS-SQL Server

- Below some KPI's from a Microsoft SQL Server
- There are 56 additional items that will be used for training for a total of 62 items

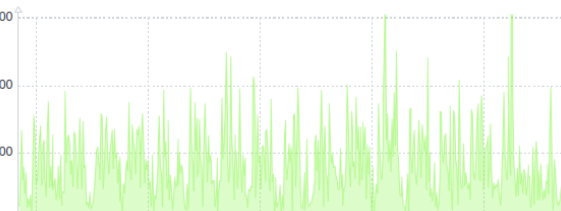
Incoming Traffic in Bytes



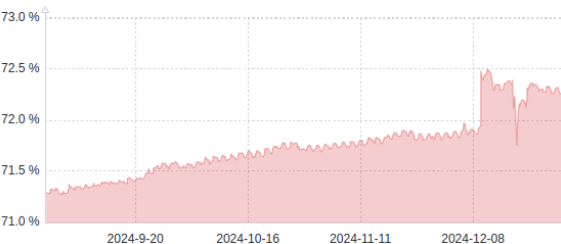
Outgoing Traffic in Bytes



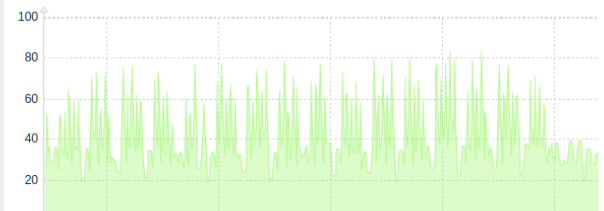
Transactions per second



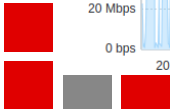
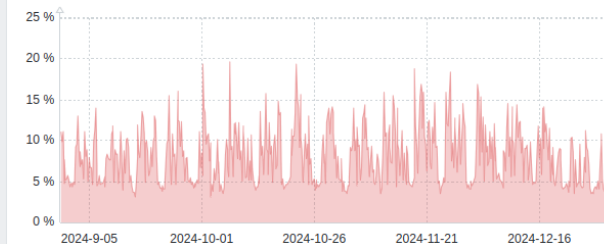
RAM usage in %



Users connected

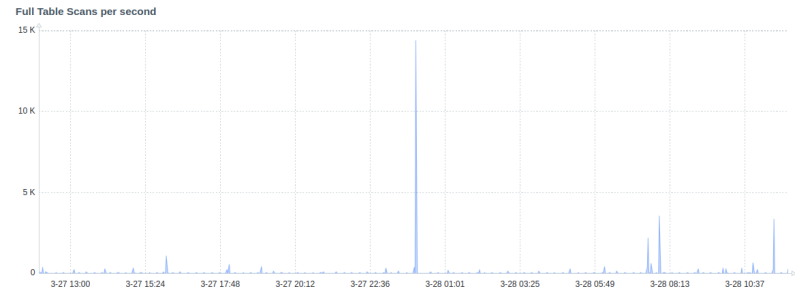
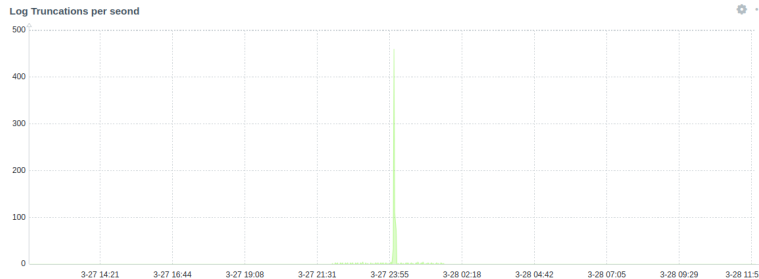
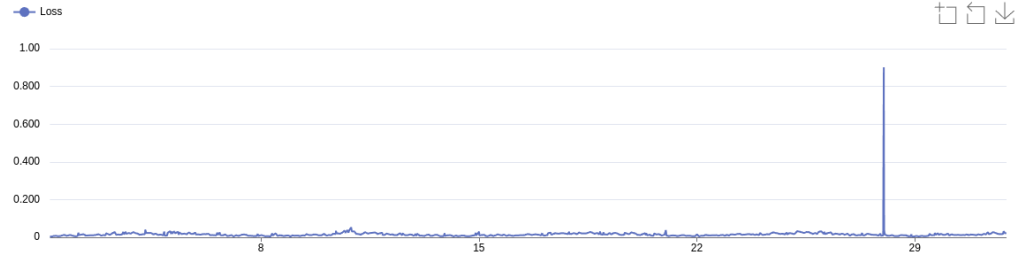


Total processor usage in %



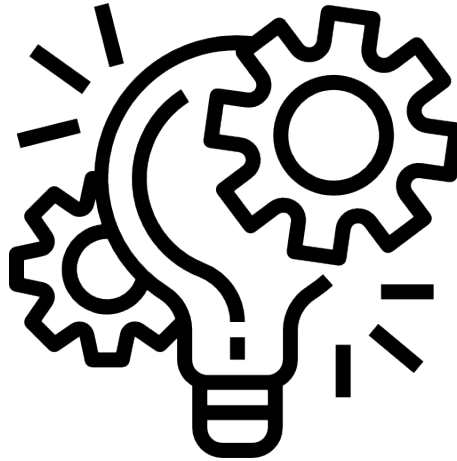
# Example MS-SQL Server

- At 3-27 the model detected an anomaly
- After reviewing the metrics, the culprit was found quickly
- The number of full table scans per second spiked to almost 15.000 while the number of log truncations also spiked at around 480



# AI-based anomaly detection

## Combine AI with regular trigger



# Combine AI with regular trigger

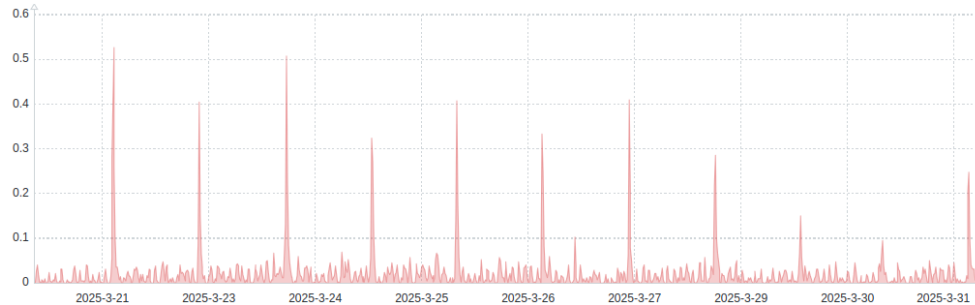
## Challenge

- Every evening at around 21:00, Zabbix sends an alert
- Reason: Processor load gets high

## Possible solutions

- Schedule periodic maintenance
- Use time() function
- ... combine with anomaly detection

15 Minute Processor Load Average in percent



**ZABBIX**  
PREMIUM PARTNER



# Combine AI with regular trigger

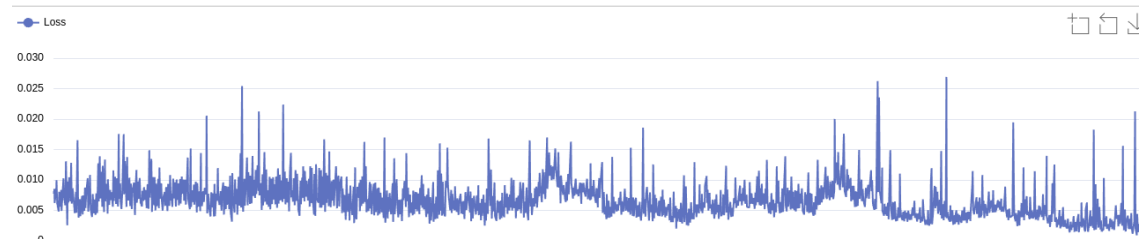
Mark items, create dataset, train model

<input type="checkbox"/>	Name ▲	Triggers	Key	Interval	History	Trends	Type	Status	Tags	Info
<input type="checkbox"/>	*** TPL: OS_Linux [Server, All] [Basic]: Arbeitsspeicher: Physikalisch [Belegt] in %	Triggers 2	vm.memory.size[pused]	2m	30d	365d	Zabbix agent	Enabled	df-ai Application: Arbeitssp...	
<input type="checkbox"/>	*** TPL: OS_Linux [Server, All] [Basic]: Arbeitsspeicher: Physikalisch [FREI] in %		vm.memory.size[pavailable]	2m	30d	365d	Zabbix agent	Enabled	df-ai Application: Arbeitssp...	
<input type="checkbox"/>	*** TPL: OS_Linux [Server, All] [Basic]: Arbeitsspeicher: Swap [Belegt] in %	Triggers 2	system.swap.size[pused]	2m	30d	365d	Zabbix agent	Enabled	df-ai Application: Arbeitssp...	
<input type="checkbox"/>	*** TPL: OS_Linux [Server, All] [Basic]: Arbeitsspeicher: Swap [Frei] in %		system.swap.size[pfree]	2m	30d	365d	Zabbix agent	Enabled	df-ai Application: Arbeitssp...	
<input type="checkbox"/>	*** TPL: Netzwerk Ping [All] [basic]: Netzwerk: Ping [Antwortzeit]		icmppingsec	2m	30d	365d	Simple check	Enabled	df-ai Application: Netzwerk	
<input type="checkbox"/>	*** TPL: Netzwerk Ping [All] [basic]: Netzwerk: Ping [Status]	Triggers 1	icmpping	2m	30d	365d	Simple check	Enabled	df-ai Application: Netzwerk	
<input type="checkbox"/>	*** TPL: OS_Linux [Server, All] [Basic]: Prozesse: Anzahl [aktiv]		proc.num[,run]	2m	30d	365d	Zabbix agent	Enabled	df-ai Application: Prozesse	
<input type="checkbox"/>	*** TPL: OS_Linux [Server, All] [Basic]: Prozessor: Auslastung [Gesamt] in %	Triggers 2	system.cpu.util[,total]	2m	30d	365d	Calculated	Enabled	df-ai Application: Prozessor	
<input type="checkbox"/>	*** TPL: OS_Linux [Server, All] [Basic]: Prozessor: Context switches [/sec]		system.cpu.switches	2m	30d	365d	Zabbix agent	Enabled	df-ai Application: Prozessor	
<input type="checkbox"/>	*** TPL: OS_Linux [Server, All] [Basic]: Prozessor: Interrupts [/sec]		system.cpu.intr	2m	30d	365d	Zabbix agent	Enabled	df-ai Application: Prozessor	

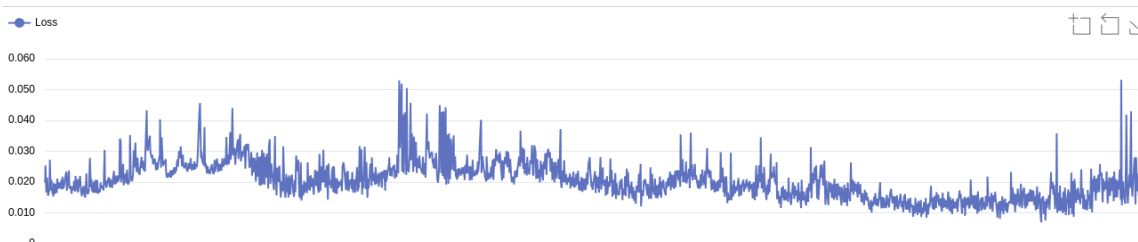


# Combine AI with regular trigger

## Baseline test



## Validation test



Training looks good,  
no anomaly at 21:00  
during backup



**ZABBIX**  
PREMIUM PARTNER

# Combine AI with regular trigger

Reconfigure the trigger – only fire if there is also an anomaly

\* Name

Event name

Operational data

Severity ☐ Not classified ☐ Information ☐ Warning ☒ Average ☐ High ☐ Disaster

\* Expression 

Edit

Insert expression

And Or Replace

A and ((B and C) or (D and E))

Target	Expression	Action	Info
<input checked="" type="checkbox"/>	And		<a href="#">Remove</a>
<input type="checkbox"/>	└ A last(/hetzner-INT-dataforge/df.hetzner.mse)>0.06		<a href="#">Remove</a>
<input type="checkbox"/>	└ Or		<a href="#">Remove</a>
<input type="checkbox"/>	└ And		<a href="#">Remove</a>
<input type="checkbox"/>	└ B {TRIGGER.VALUE}=0		<a href="#">Remove</a>
<input type="checkbox"/>	└ C min(/hetzner-INT-dataforge/system.cpu.util[.total],{CPU_USED_INTERVAL})>{CPU_USED_WARNING_MAX}		<a href="#">Remove</a>
<input type="checkbox"/>	└ And		<a href="#">Remove</a>
<input type="checkbox"/>	└ D {TRIGGER.VALUE}=1		<a href="#">Remove</a>
<input type="checkbox"/>	└ E min(/hetzner-INT-dataforge/system.cpu.util[.total],{CPU_USED_INTERVAL})>{CPU_USED_WARNING_OK}		<a href="#">Remove</a>

[Test](#)

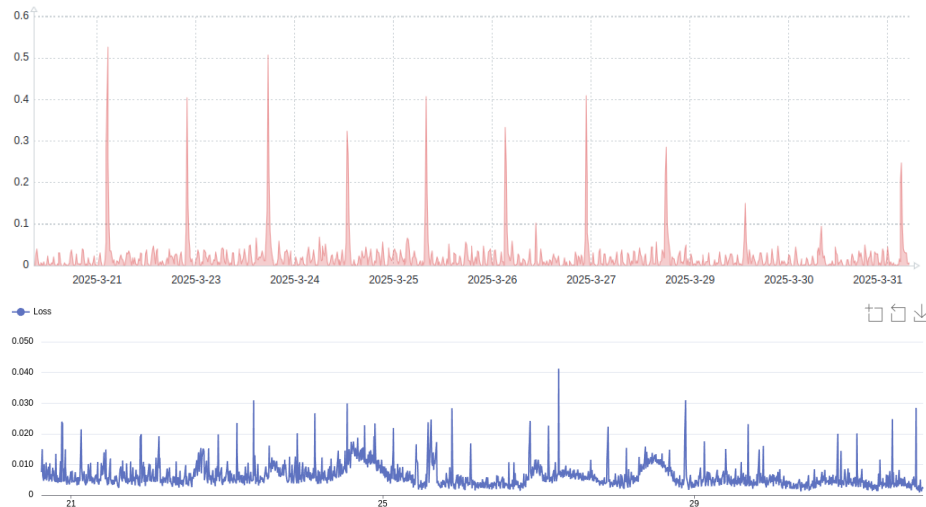


# Combine AI with regular trigger

It works – no alarm sent during the backup at 21.00

Average	OK	AI: {HOST.NAME} - Prozessor: Auslastung > {CPU_USED_WARNING_MAX} % [Über {CPU_USED_INTERVAL} Sekunden]	<code>last(/hetzner-INT-dataforge/df.hetzner.mse)&gt;0.1 and ((TRIGGER.VALUE)=0 and min(/hetzner-INT-dataforge/system.cpu.util[,total], {CPU_USED_INTERVAL})&gt;{CPU_USED_WARNING_MAX}) or ((TRIGGER.VALUE)=1 and min(/hetzner-INT-dataforge/system.cpu.util[,total], {CPU_USED_INTERVAL})&gt;{CPU_USED_WARNING_OK}))</code>	Ena
Average	PROBLEM	{HOST.NAME} - Prozessor: Auslastung > {CPU_USED_WARNING_MAX} % [Über {CPU_USED_INTERVAL} Sekunden]	<code>((TRIGGER.VALUE)=0 and min(/hetzner-INT-dataforge/system.cpu.util[,total],{CPU_USED_INTERVAL})&gt;{CPU_USED_WARNING_MAX}) or ((TRIGGER.VALUE)=1 and min(/hetzner-INT-dataforge/system.cpu.util[,total],{CPU_USED_INTERVAL})&gt;{CPU_USED_WARNING_OK}))</code>	Ena

15 Minute Processor Load Average in percent



**ZABBIX**  
PREMIUM PARTNER

# Combine AI with regular trigger

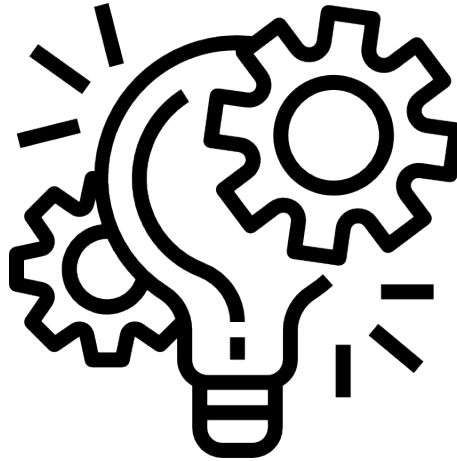
But – It sends alarms when the load is at an unusual time like here at 11:18



**ZABBIX**  
PREMIUM PARTNER

# AI-based anomaly detection

## Hardware requirements



# Hardware requirements

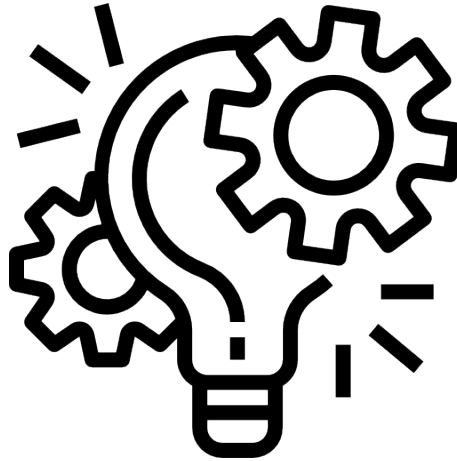
Hardware requirements for these autoencoder models are quite moderate. As an example we will use the 100 epoch MS-SQL server model with 62 items and one month of data:

- NVIDIA RTX-4090: Training process used 1.3GiB of DRAM and 458MiB of VRAM with the GPU hovering around 60% utilization and the CPU loading 2 cores for the training process. Training took **6m 32s** or 3.92s per epoch using this setup.
- AMD R9 7950X3D CPU: Training process used 533MiB of DRAM with the CPU loading 8 cores for the training process. Training took **20m 42s** or 12.42s per epoch using this setup.
- AMD R7 3700U CPU: Training process used 510MiB of DRAM with the CPU loading 4 cores for the training process. Training took **8h 53m** or 5m 32s per epoch using this setup.
- Inference does not require a GPU. For example an AMD R9 7950X3D CPU can handle upto 560 NVPS. The MS-SQL server model with 62 items equals to 0.51 NVPS.



# AI-based anomaly detection

## Summary





# Summary

AI-based anomaly detection systems can:

- ... learn simple and complex threshold values
- ... learn temporal dependencies (Certain values are expected at certain times)
- ... learn value dependencies (Certain values must maintain a mathematical relationship between each other)
- ... learn dependencies across multiple systems and services
- ... improve monitoring setups by looking into anomalies across many items at once
- ... express the magnitude of an anomaly using the value of reconstruction error instead of just true/false

**ZABBIX '25**  
CONFERENCE  
GERMANY

# Thank you



**Wolfgang Alper**

CEO  
IntelliTrend GmbH

