An AI-driven Malfunction Detection Concept for NFV Instances in 5G*

Julian Ahrens Mathias Strufe
Lia Ahrens Hans D. Schotten

May 17, 2018

*This work was supported by the European Union's Horizon 2020 Programme under the 5G-PPP project: Framework for Self-Organized Network Management in Virtualized and Software Defined Networks (SELFNET) with Grant no. H2020-ICT-2014-2/671672.

Introduction

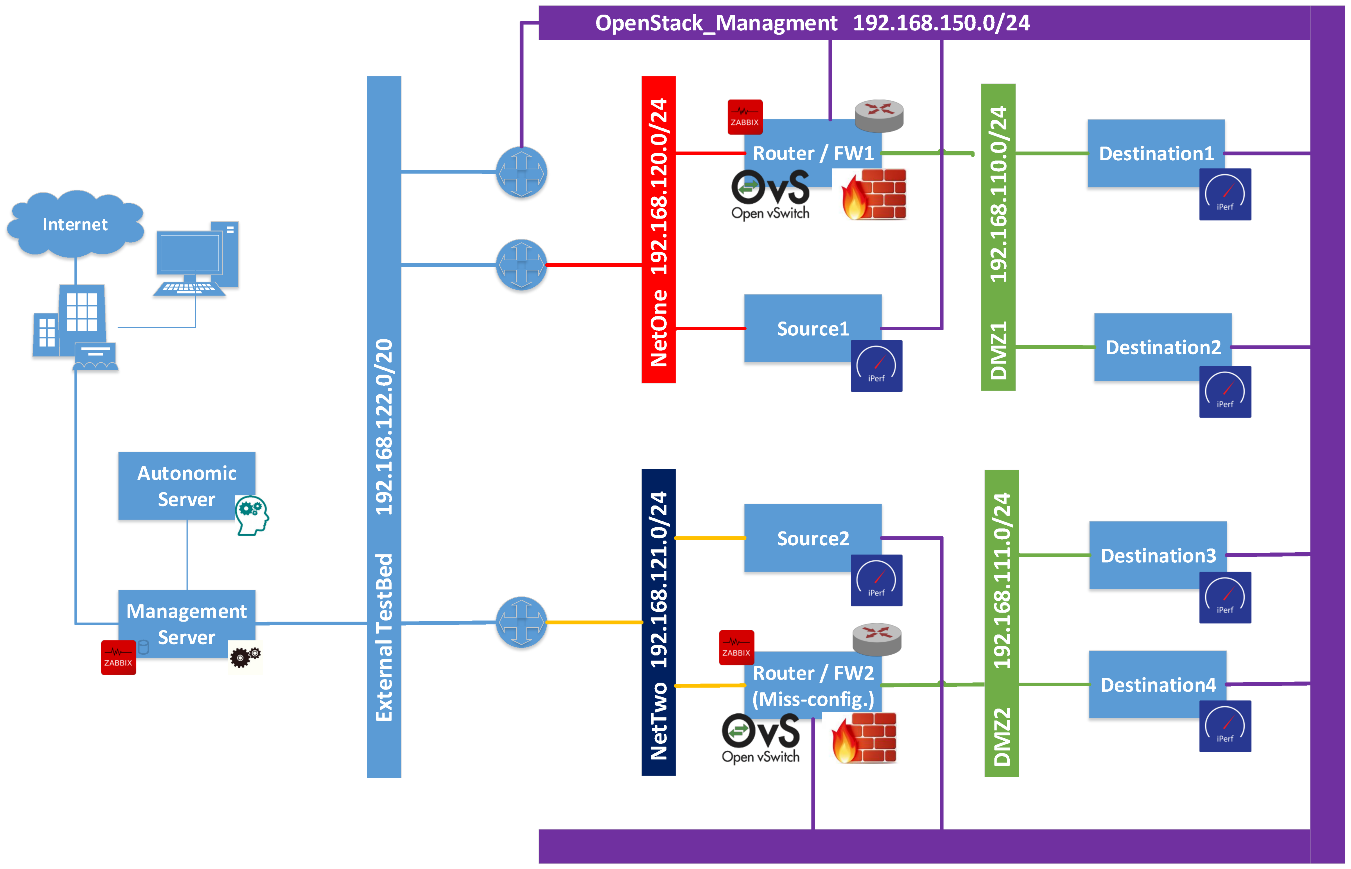
Main Problem

Main Problem

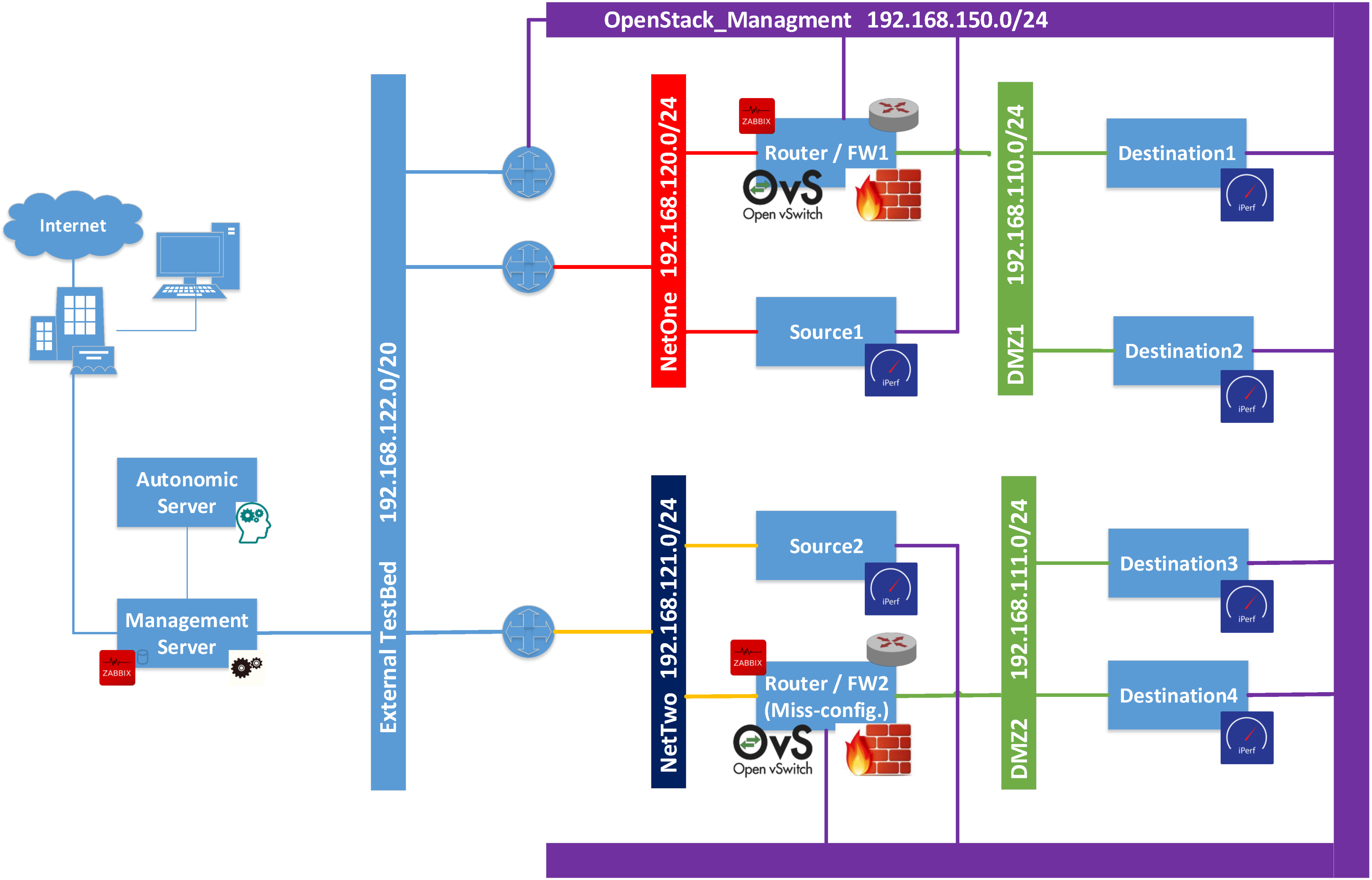
Challenge: detect previously unencountered problems in VNFs using network and performance metrics

Example Setup

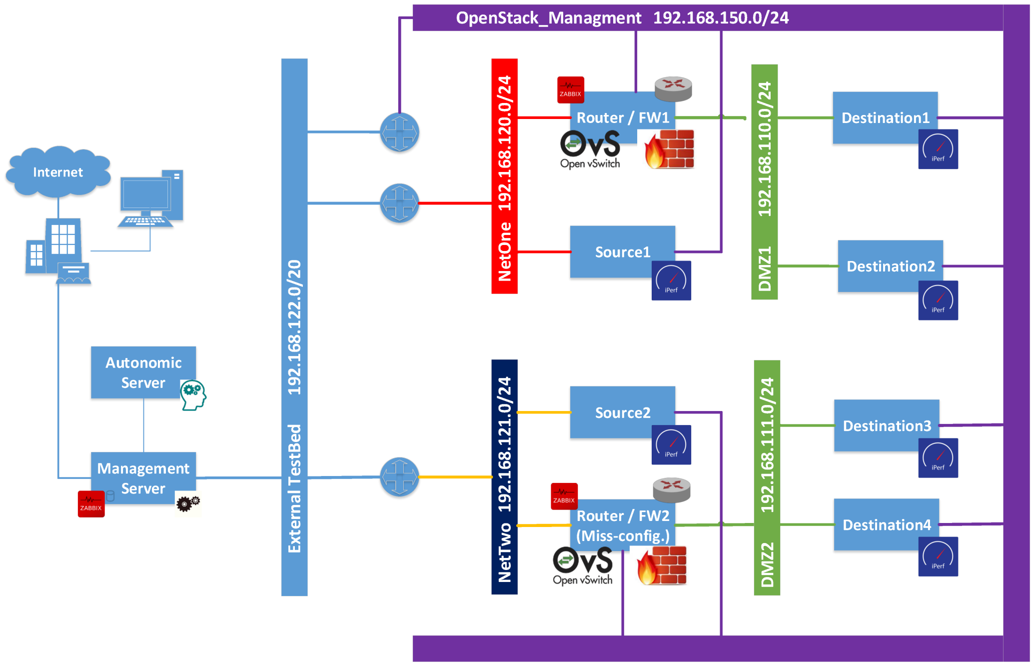
Example Setup

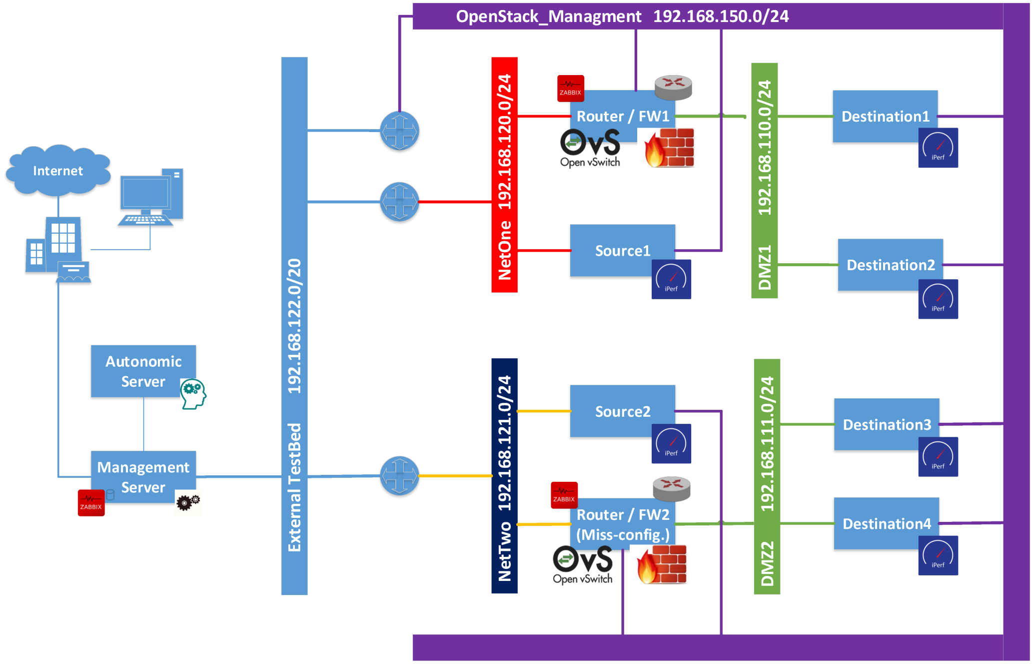


Example Setup

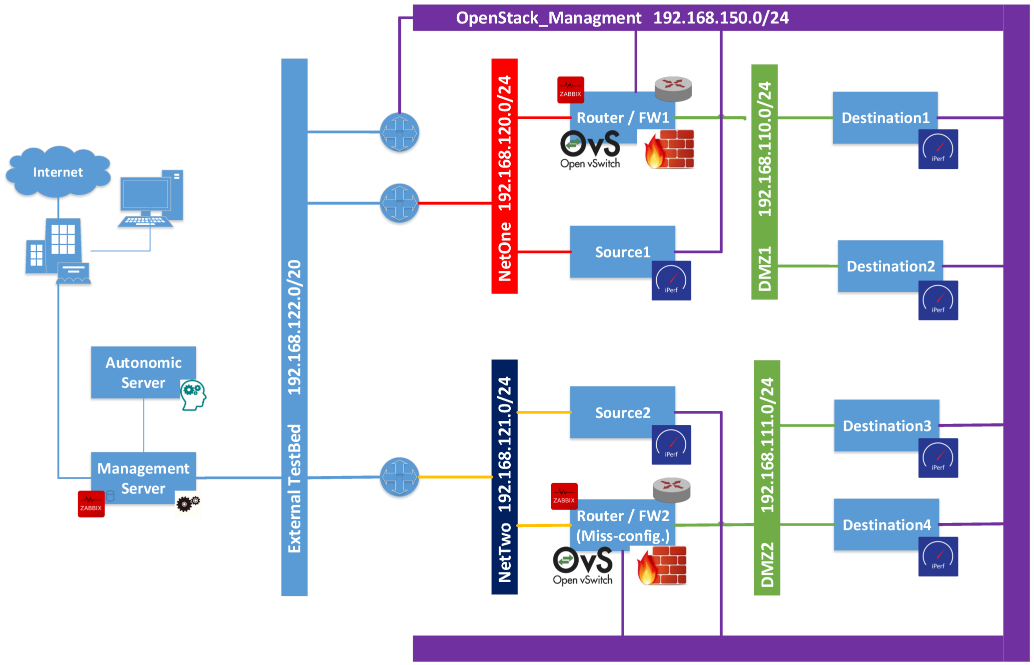


Example Setup



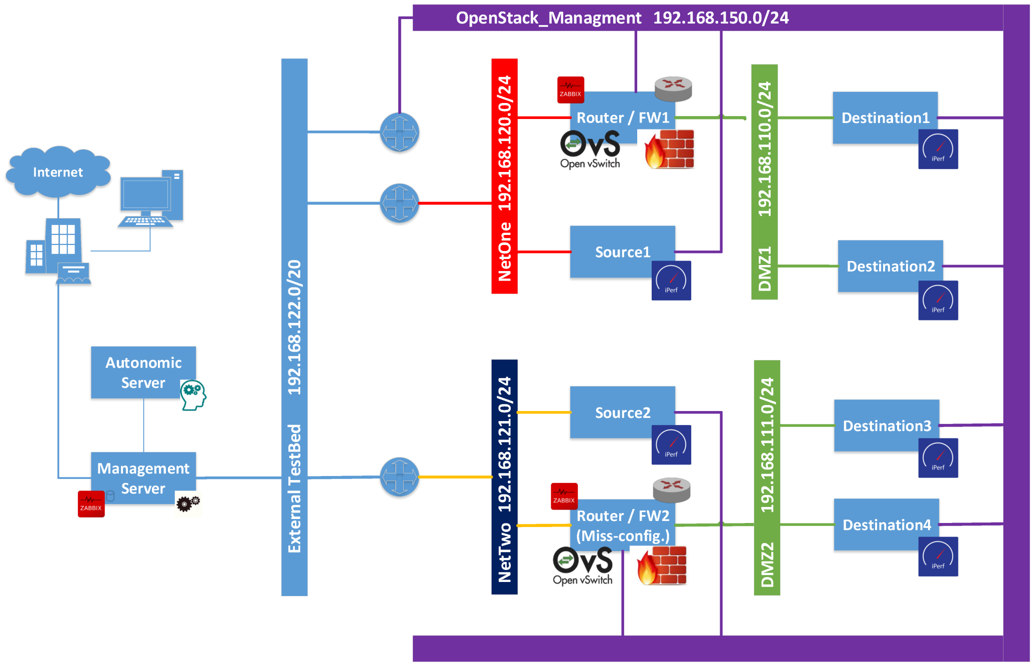


- Example VNF: virtualized firewall



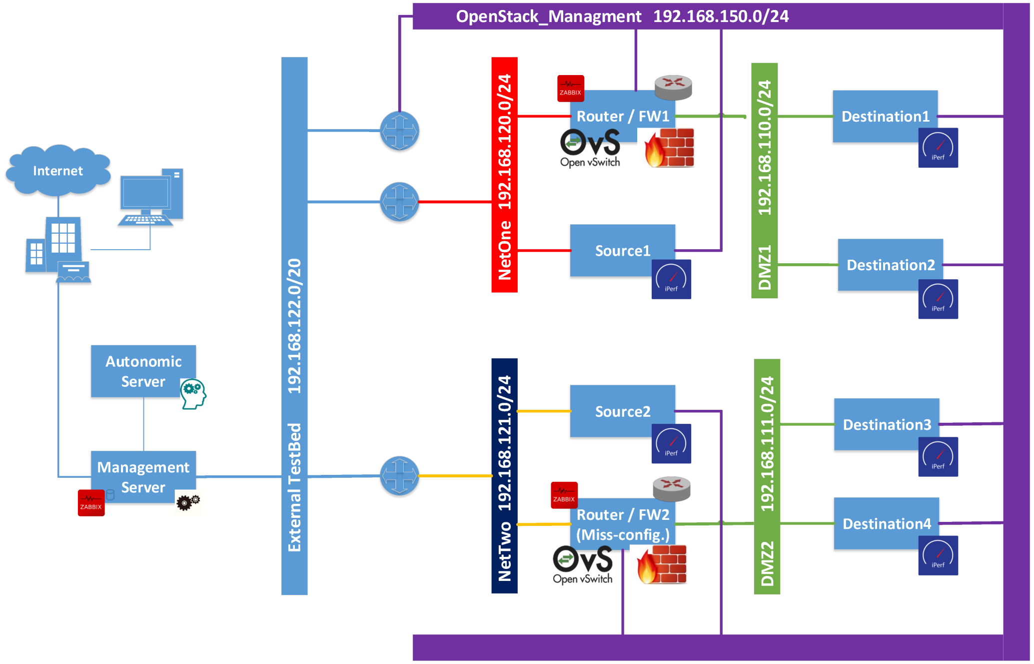
- Example VNF: virtualized firewall
- Data flowing from the sources to the destinations

Example Setup



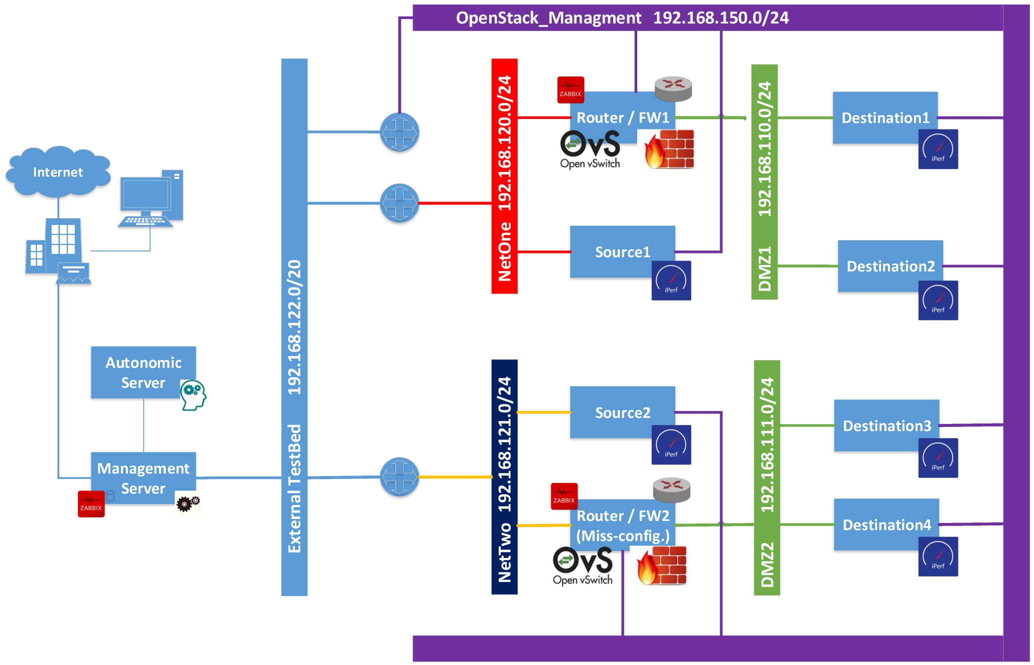
- Example VNF: virtualized firewall
- Data flowing from the sources to the destinations
- Additional traffic between internet and destinations

Example Setup

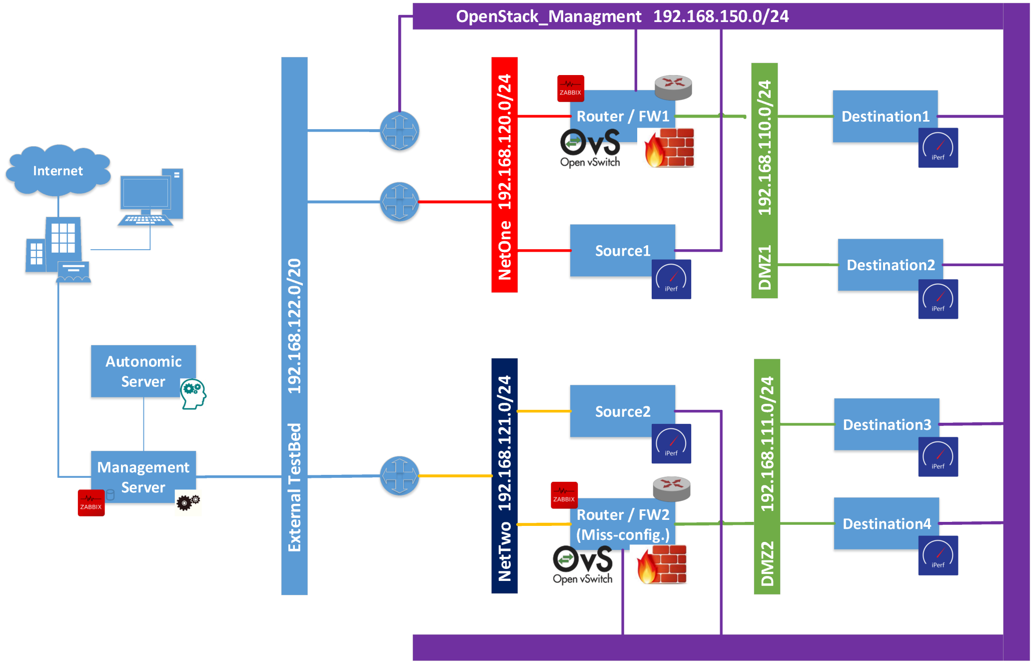


- Example VNF: virtualized firewall
- Data flowing from the sources to the destinations
- Additional traffic between internet and destinations
- Firewalls filtering traffic from source to one of the destinations

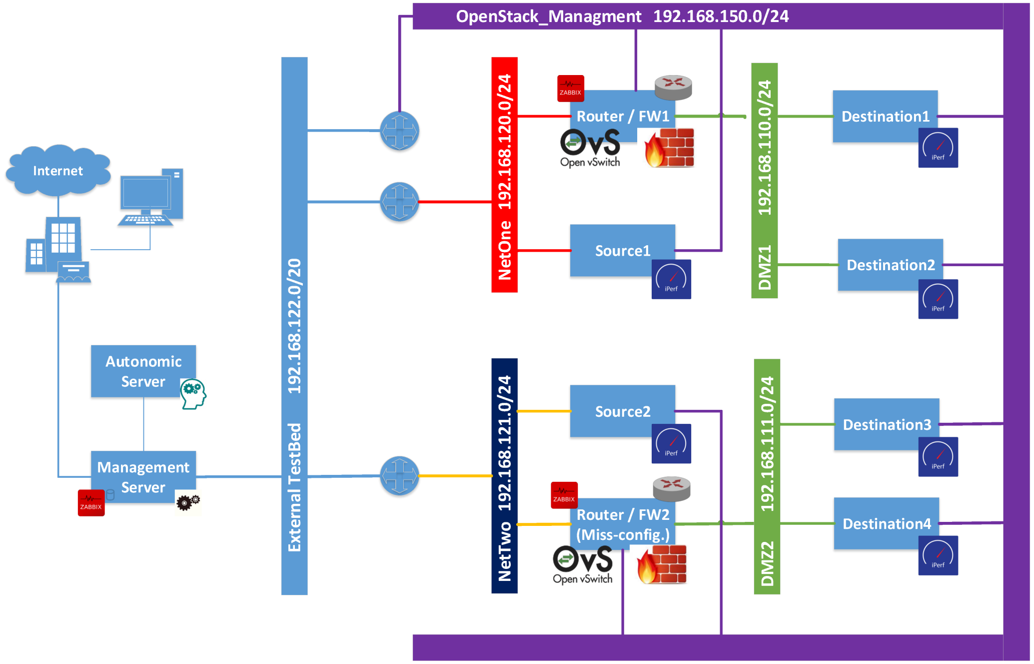
Example Setup



Example Setup

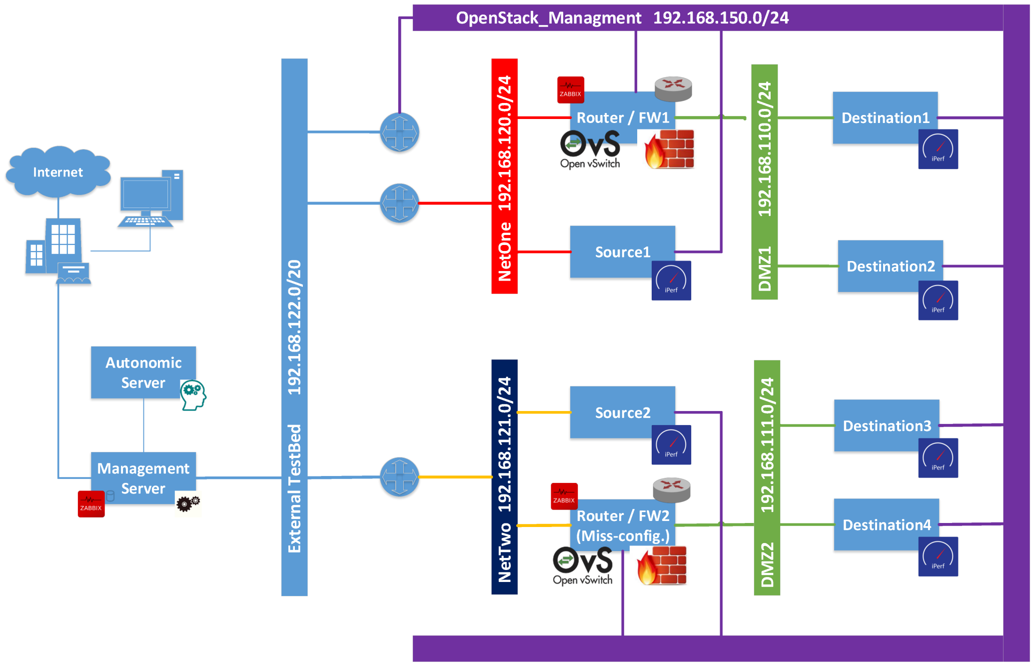


- Features: CPU utilization, memory utilization, HDD utilization, network traffic in and out, each on two interfaces



- Features: CPU utilization, memory utilization, HDD utilization, network traffic in and out, each on two interfaces

  → seven features



- Features: CPU utilization, memory utilization, HDD utilization, network traffic in and out, each on two interfaces

  → seven features
- Anomaly used for testing: memory leak

Intelligence

Semi-supervised learning

Semi-supervised learning

- Dataset of known good operation

Semi-supervised learning

- Dataset of known good operation
- No actual anomalies are used during the training process

Semi-supervised learning

- Dataset of known good operation
- No actual anomalies are used during the training process
- Available methods: clustering, dimensionality reduction / sparsity constraints

Semi-supervised learning

- Dataset of known good operation
- No actual anomalies are used during the training process
- Available methods: clustering, dimensionality reduction / sparsity constraints

  → Measure of similarity required

Semi-supervised learning

- Dataset of known good operation
- No actual anomalies are used during the training process
- Available methods: clustering, dimensionality reduction / sparsity constraints
  $\rightarrow$ Measure of similarity required
- Problem: Heterogeneity of data, multiple scales

Semi-supervised learning

- Dataset of known good operation
- No actual anomalies are used during the training process
- Available methods: clustering, dimensionality reduction / sparsity constraints
  - $\rightarrow$ Measure of similarity required
- Problem: Heterogeneity of data, multiple scales
  - $\rightarrow$ Normalization required

Semi-supervised learning

- Dataset of known good operation
- No actual anomalies are used during the training process
- Available methods: clustering, dimensionality reduction / sparsity constraints
  $\rightarrow$ Measure of similarity required
- Problem: Heterogeneity of data, multiple scales
  $\rightarrow$ Normalization required
- Normalization via parametric and nonparametric statistical approaches

Autoencoder

Autoencoder

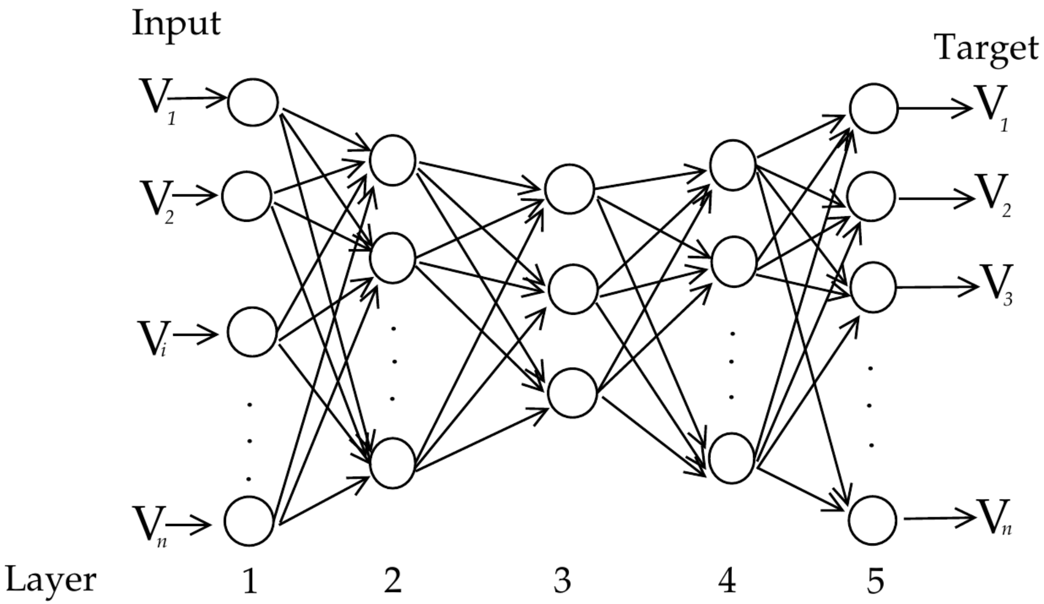
- Dimensionality reduction via Autoencoders

Autoencoder

- Dimensionality reduction via Autoencoders
- Autoencoders are replicating artificial neural networks

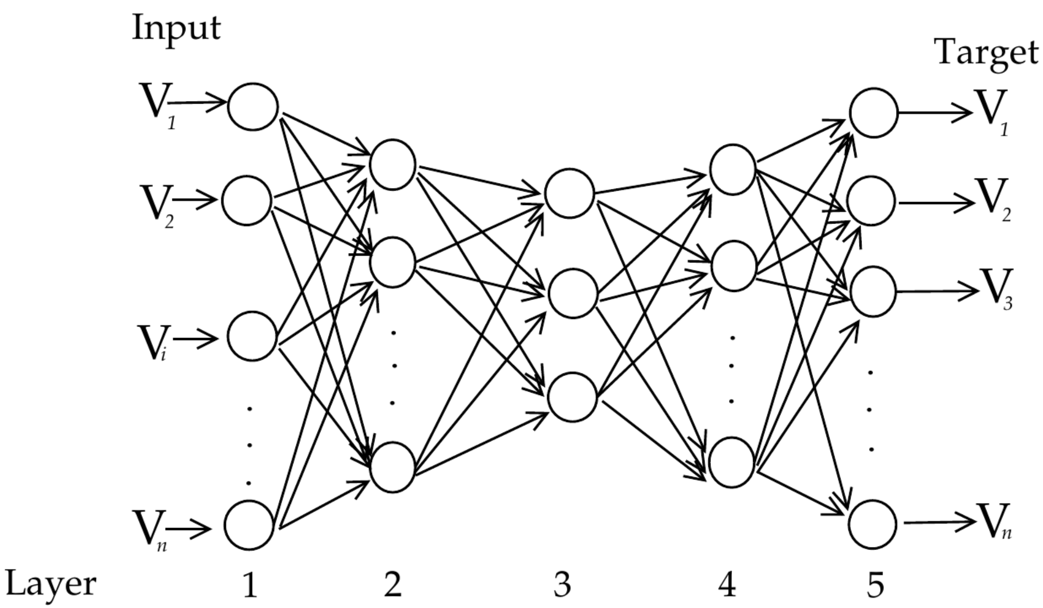
Autoencoder

- Dimensionality reduction via Autoencoders
- Autoencoders are replicating artificial neural networks



Autoencoder

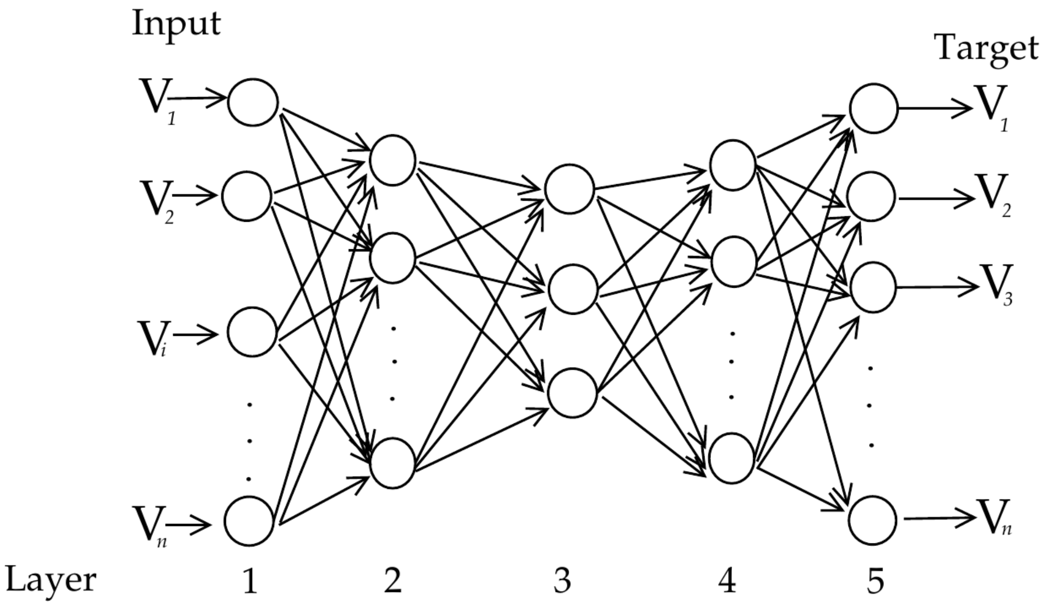
- Dimensionality reduction via Autoencoders
- Autoencoders are replicating artificial neural networks



- Both current value and difference to last value used as features

Autoencoder

- Dimensionality reduction via Autoencoders
- Autoencoders are replicating artificial neural networks

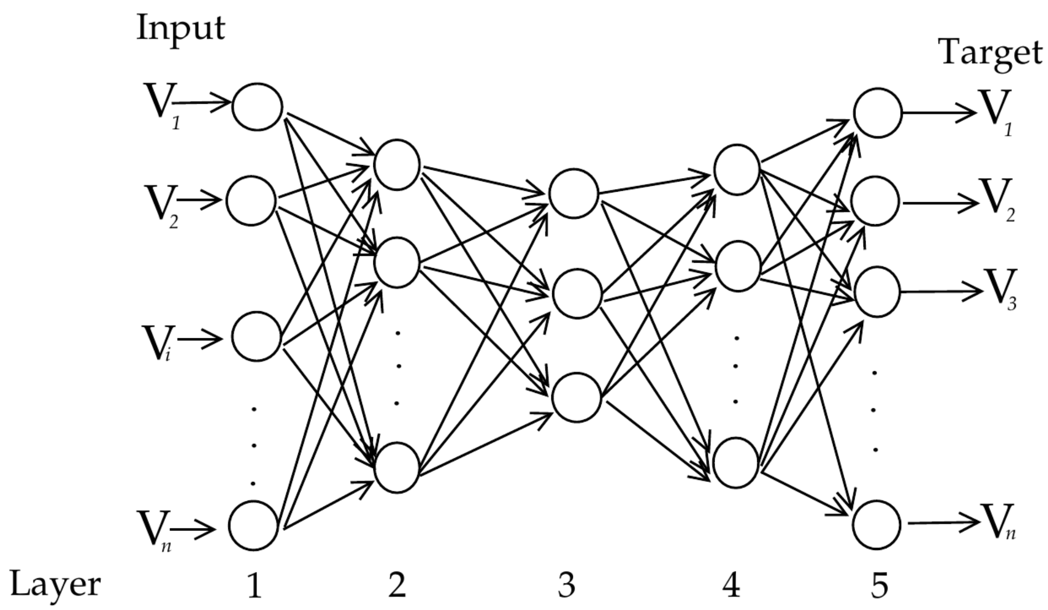


- Both current value and difference to last value used as features

  → 14 features in the example setup

Autoencoder

- Dimensionality reduction via Autoencoders
- Autoencoders are replicating artificial neural networks



- Both current value and difference to last value used as features

  → 14 features in the example setup
- Hyperparameters: number of layers / layer sizes, activation function, learning rate during training

Autoencoder

Autoencoder

- Measure of similarity: Euclidean distance

- Measure of similarity: Euclidean distance
- Autoencoder is trained to best approximate its input by its output in the Euclidean distance over the training data

Autoencoder

- Measure of similarity: Euclidean distance
- Autoencoder is trained to best approximate its input by its output in the Euclidean distance over the training data
- Measure of normality: Euclidean distance of output of autoencoder to its input

Autoencoder

- Measure of similarity: Euclidean distance
- Autoencoder is trained to best approximate its input by its output in the Euclidean distance over the training data
- Measure of normality: Euclidean distance of output of autoencoder to its input
- If this distance is above a certain threshold, a datapoint is considered anomalous by the autoencoder

- Measure of similarity: Euclidean distance
- Autoencoder is trained to best approximate its input by its output in the Euclidean distance over the training data
- Measure of normality: Euclidean distance of output of autoencoder to its input
- If this distance is above a certain threshold, a datapoint is considered anomalous by the autoencoder
- Hyperparameter: anomaly threshold

Autoencoder

- Measure of similarity: Euclidean distance
- Autoencoder is trained to best approximate its input by its output in the Euclidean distance over the training data
- Measure of normality: Euclidean distance of output of autoencoder to its input
- If this distance is above a certain threshold, a datapoint is considered anomalous by the autoencoder
- Hyperparameter: anomaly threshold
- Use an additional validation dataset to adjust the anomaly threshold until the number of false positives reaches a desirable level

Ensembles

Ensembles

- Use more than one autoencoder in order to capture different characteristics of the data

Ensembles

- Use more than one autoencoder in order to capture different characteristics of the data
- Different autoencoders governed by different hyperparameters

Ensembles

- Use more than one autoencoder in order to capture different characteristics of the data
- Different autoencoders governed by different hyperparameters
- Allows for diversity in sparsity, nonlinearity, etc.

Ensembles

- Use more than one autoencoder in order to capture different characteristics of the data
- Different autoencoders governed by different hyperparameters
- Allows for diversity in sparsity, nonlinearity, etc.
- In the test scenario: three to seven layers, central layer sizes from two to eight neurons, tanh and ReLU activation functions, total of 216 autoencoders

Ensembles

- Use more than one autoencoder in order to capture different characteristics of the data
- Different autoencoders governed by different hyperparameters
- Allows for diversity in sparsity, nonlinearity, etc.
- In the test scenario: three to seven layers, central layer sizes from two to eight neurons, tanh and ReLU activation functions, total of 216 autoencoders
- Vote among autoencoders: a datapoint is considered normal if enough of the autoencoders consider the datapoint normal; otherwise, it is considered anomalous

Evaluation

Results

Results

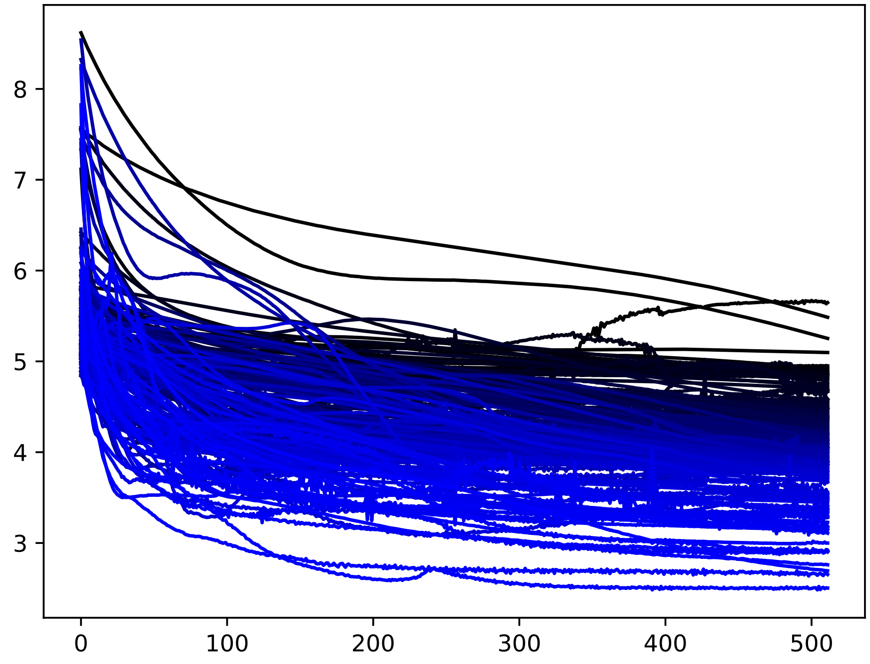
- Significant generalization performance

Results

- Significant generalization performance
- Most of the autoencoders converge very well on the validation dataset

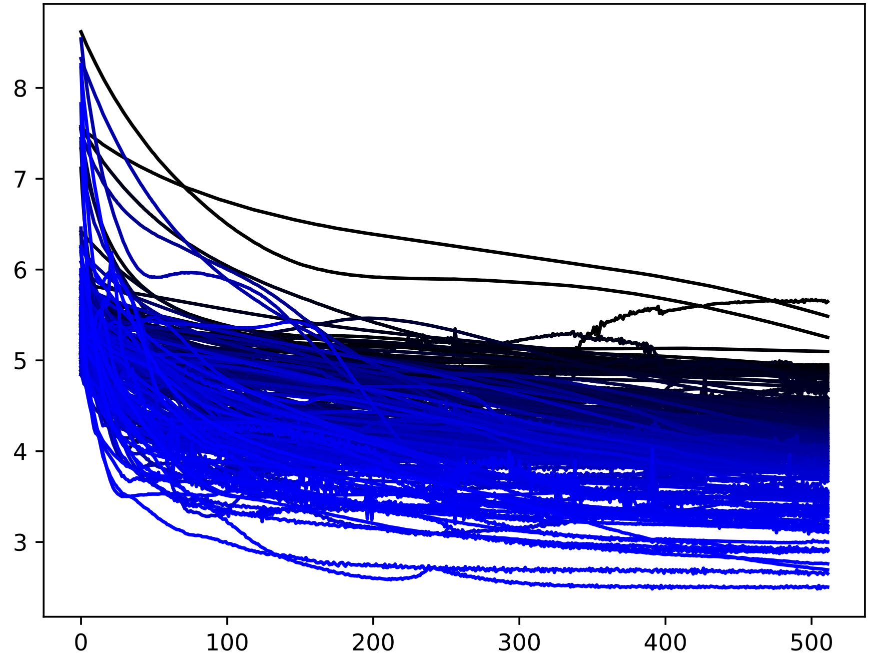
Results

- Significant generalization performance
- Most of the autoencoders converge very well on the validation dataset



Results

- Significant generalization performance
- Most of the autoencoders converge very well on the validation dataset



- 80.4% detection rate vs. 0.7% false positives

Questions?

Thank you!