# Machine Learning for Infrastructure Monitoring

## Infrastructure Monitoring, a Joint Workshop by the Alan Turing Institute and the ENCORE Network+

23 May 2018

David K. E. Green
dgreen@turing.ac.uk

The Alan Turing Institute

London, United Kingdom
www.turing.ac.uk

erc SINGULARITY

# Prelude

Two major technology trends:

**Monitoring and Infrastructure:**

- Increasing data volume (sensors are cheaper).
- Increased data speed (improved sensor networking).
- Consequences of severe failures increasing (JIT economy, population density).

**Application risk profile:** High failure consequence requires low failure rate.

**Machine learning:**

- Deep Neural Networks for function approximation.
- Data processing capabilities improving.
- Ability to deal with very high dimensional spaces and large data sets.

**Application risk profile:** Low failure consequence tolerates high failure rate.

**Challenge:** bridging the gap while managing risk. Applying new techniques to high consequence fields.

# Overview

**This talk:** machine learning for anomaly detection.

## Machine Learning

Other buzzword names include: data mining, predictive analytics, (narrow) artificial intelligence. What is it really: Fancy statistics and numerical methods for statistical analysis.

## Anomaly detection

Detection of outlier events or observations. Detecting deviations from the expected pattern of a data set.

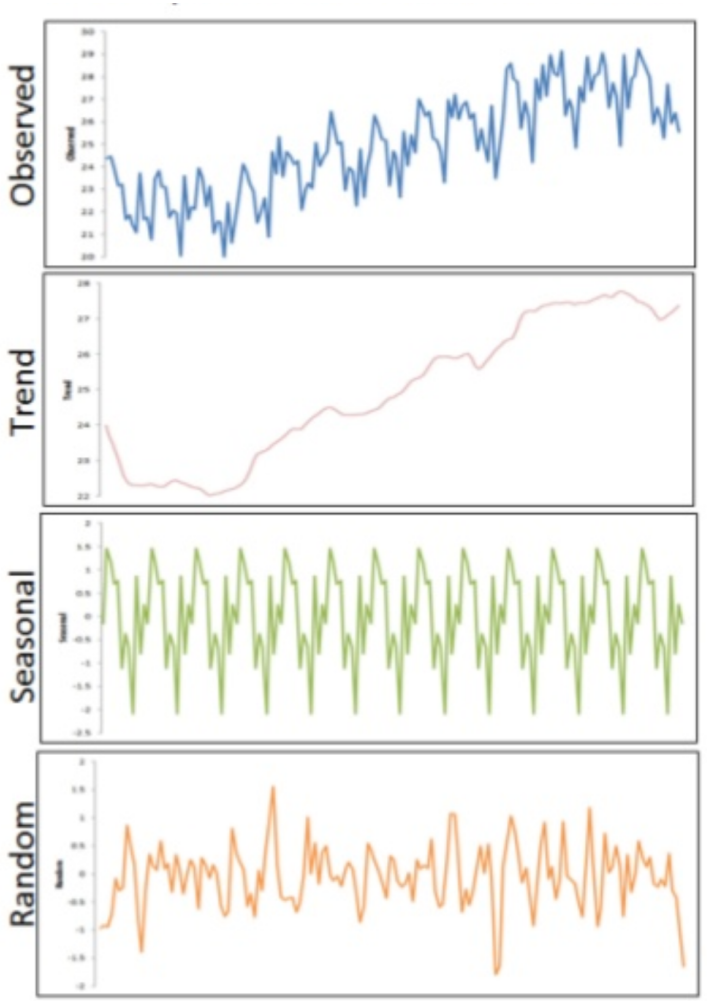
Machine learning for infrastructure anomaly detection:

- Machine learning problem categories for anomaly detection
- Anomaly detection examples
- Briefly: Some more advanced techniques using Deep Neural Networks
- Where to in the future?

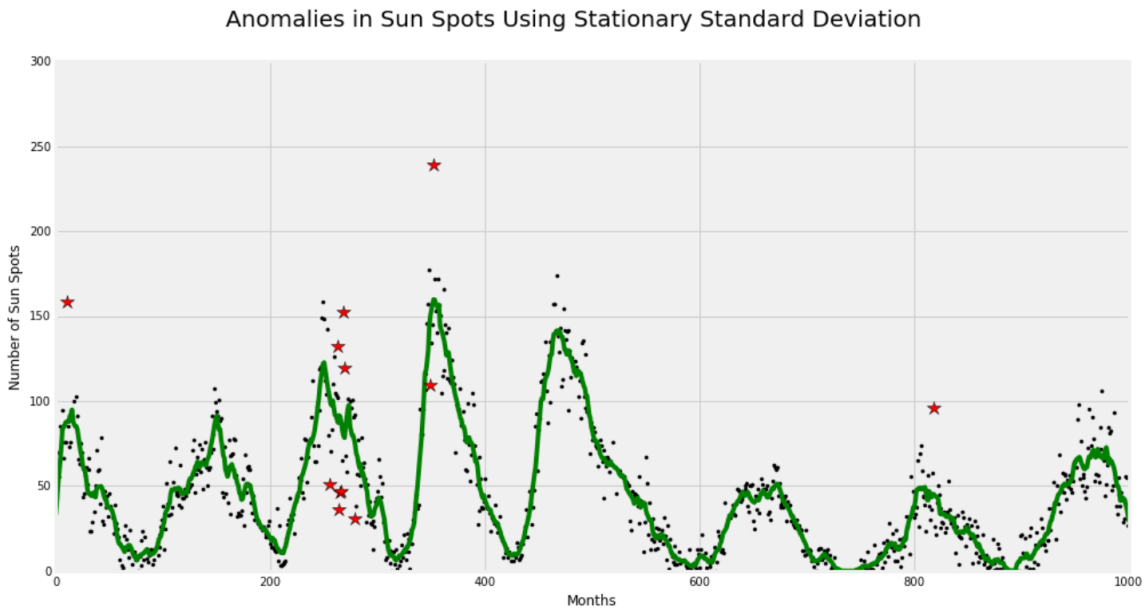
# Components of Time Series

- Need to separate trends from noise.
- Image on the right shows the rough idea.
- Basic decomposition into trend, seasonal and noise components.
- Many more advanced techniques, use depends on data:
  - Multiplicative decomposition
  - Cyclical components

Additive time series decomposition



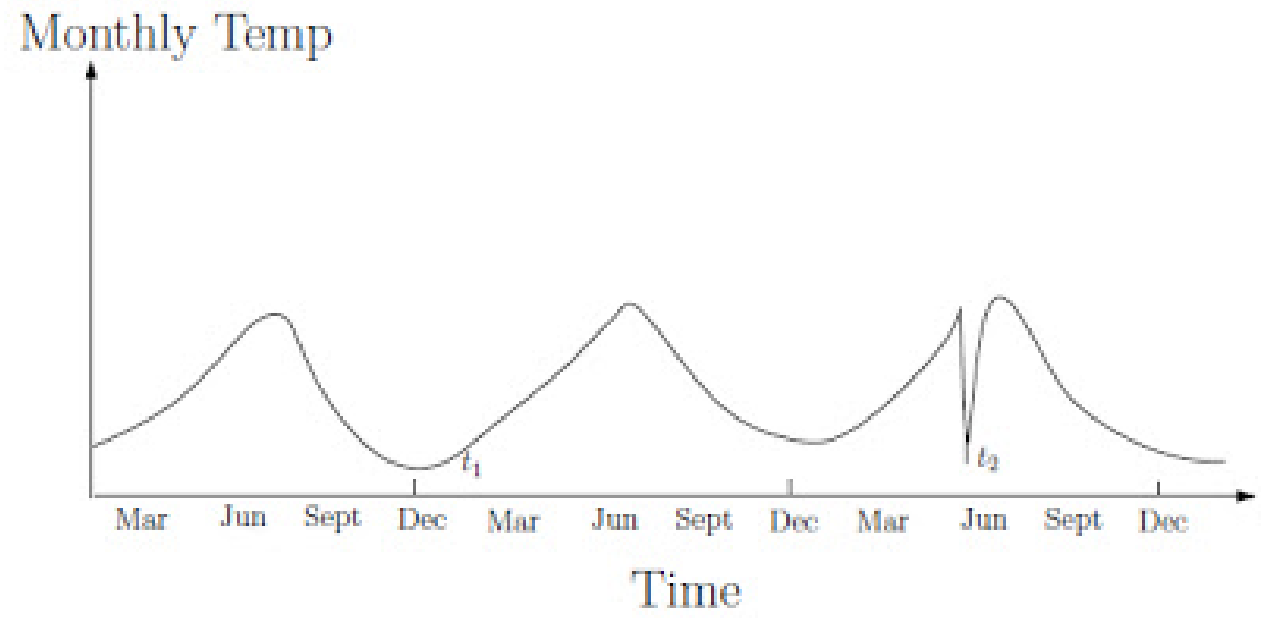
# Anomaly detection




- Standard classifications:
- **Point anomalies:** An observation that deviates from the trend.
- **Contextual anomalies:** Violation of seasonal trend: a 30 degree day is anomalous in Winter, but not so much in Summer.
- **Collective anomalies:** Conditional/contextual anomalies with respect to the entire data set, but not each other (e.g. heart skips a beat).

Image source: `tinyurl.com/ycz6a7jr`

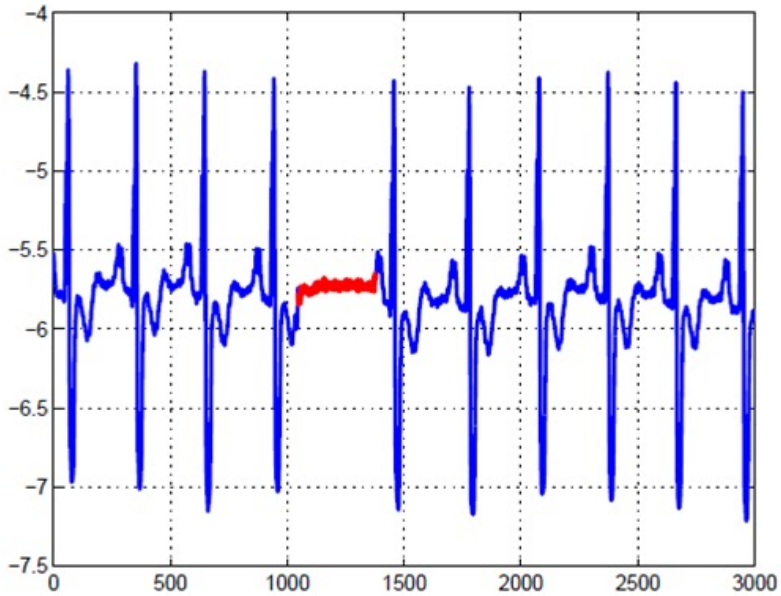
# Anomaly detection




- Standard classifications:
- **Point anomalies:** An observation that deviates from the trend.
- **Contextual anomalies:** Violation of seasonal trend: a 30 degree day is anomalous in Winter, but not so much in Summer.
- **Collective anomalies:** Conditional/contextual anomalies with respect to the entire data set, but not each other (e.g. heart skips a beat).

Image source: `tinyurl.com/ycz6a7jr`

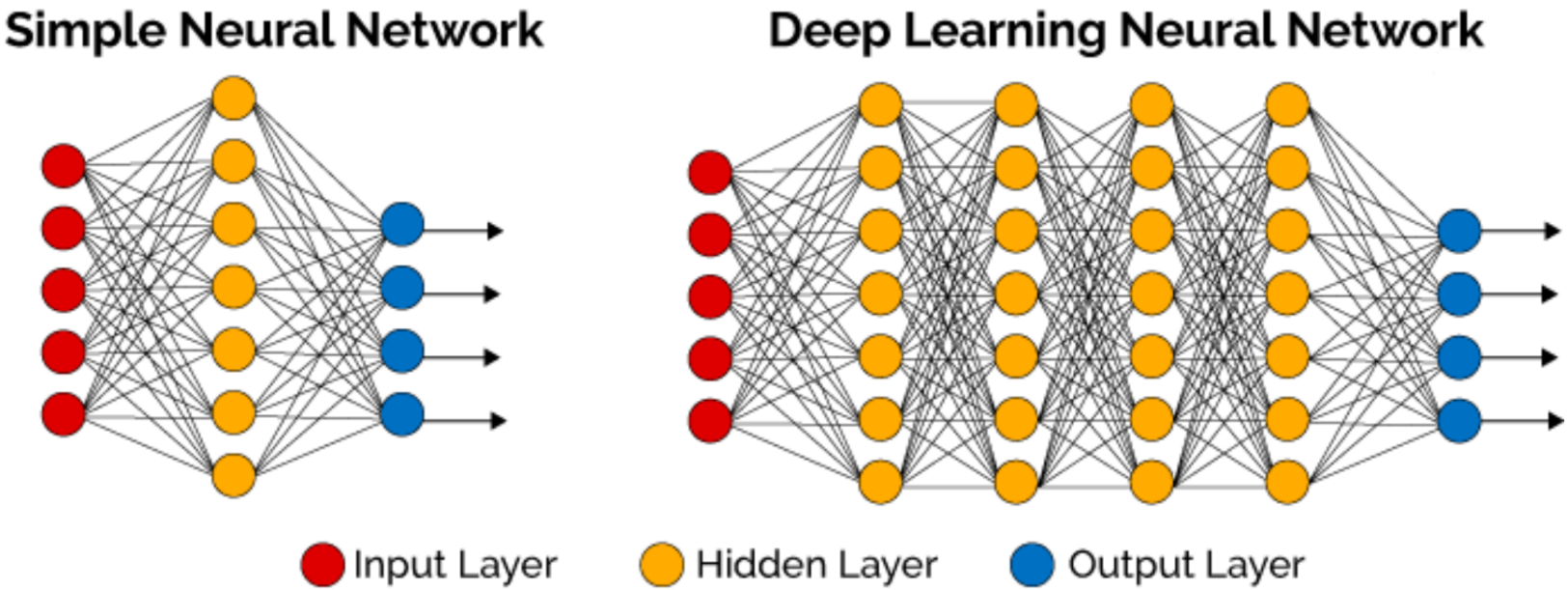
# Anomaly detection




- Standard classifications:
- **Point anomalies:** An observation that deviates from the trend.
- **Contextual anomalies:** Violation of seasonal trend: a 30 degree day is anomalous in Winter, but not so much in Summer.
- **Collective anomalies:** Conditional/contextual anomalies with respect to the entire data set, but not each other (e.g. heart skips a beat).

Image source: `tinyurl.com/ycz6a7jr`

# Neural Networks - Quick background!




- Deep Neural Networks are a function approximation technique.
- Much more flexible than traditional numerical methods.
- Requires the problem at hand to be encoded in an error function.
- Differentiability of the Neural Network components allows for training by Stochastic Gradient Descent.

# Neural Networks - Training

- Get inputs and outputs.
- Define the space of allowable functions.
- Define what is meant by a good function.
- Use calculus to find the function that best fits the data.
- The error function defines a landscape. Walk down to the lowest valley in the error landscape.
- After training, hope that new inputs fed into the network predict the correct outputs.

**Setup:** Have training data samples: $(x, y)$ (input and output). Hypothesis space parameterised by Neural Network weights:

$$f_\theta(x) = y \text{ for } \theta \in \Theta$$

Assign loss (or error) function:

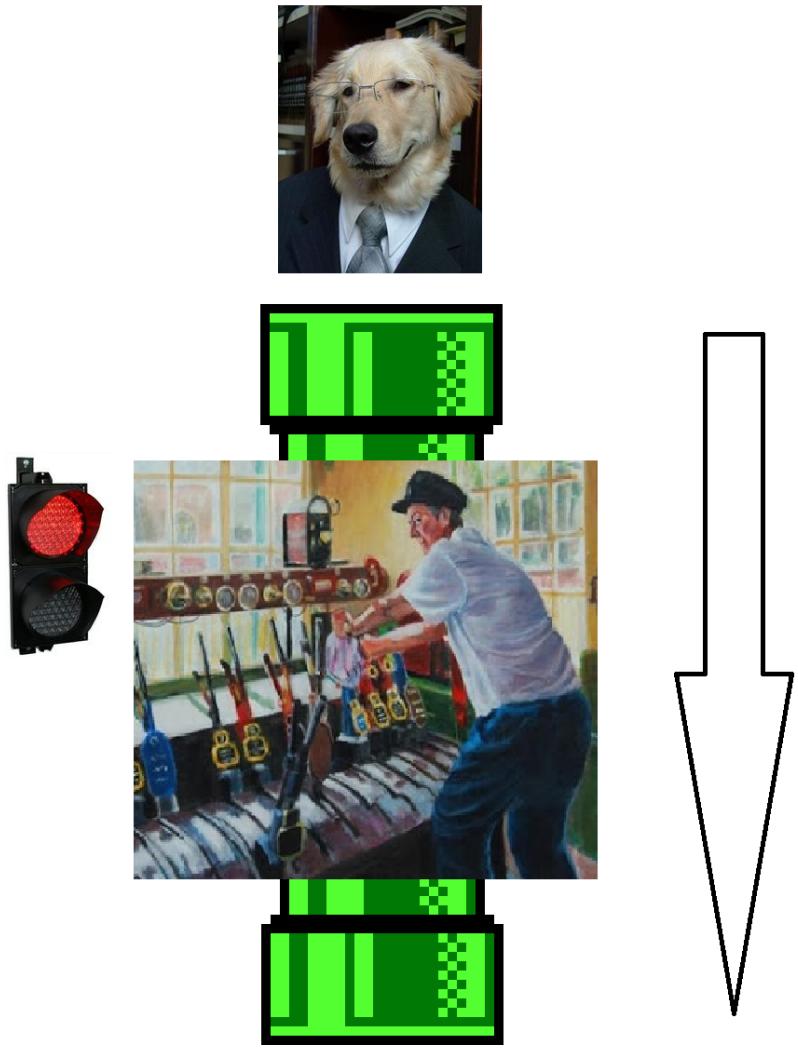
$$J(\theta) = \|y - f_\theta(x)\| \text{ so } f_\theta^*(x) = \underset{\theta \in \Theta}{\text{argmin}}\ J(\theta)$$

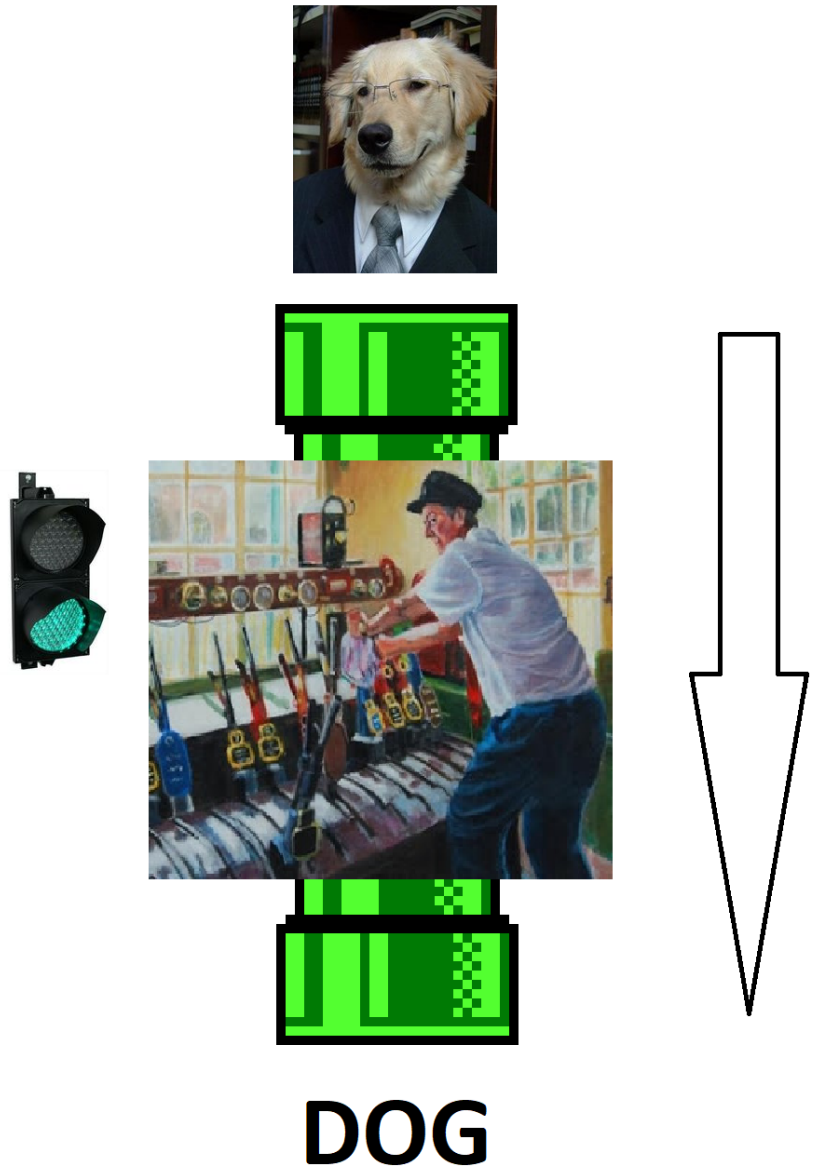
**Train:** Minimise $J(\theta)$ by gradient descent over the training samples:

$$\theta \leftarrow \theta - \alpha \nabla_\theta J(\theta) \text{ for all } (x, y)$$

**Process:** Hope that $f_\theta^*(\hat{x})$ is ok for $\hat{x}$ without matching $\hat{y}$.

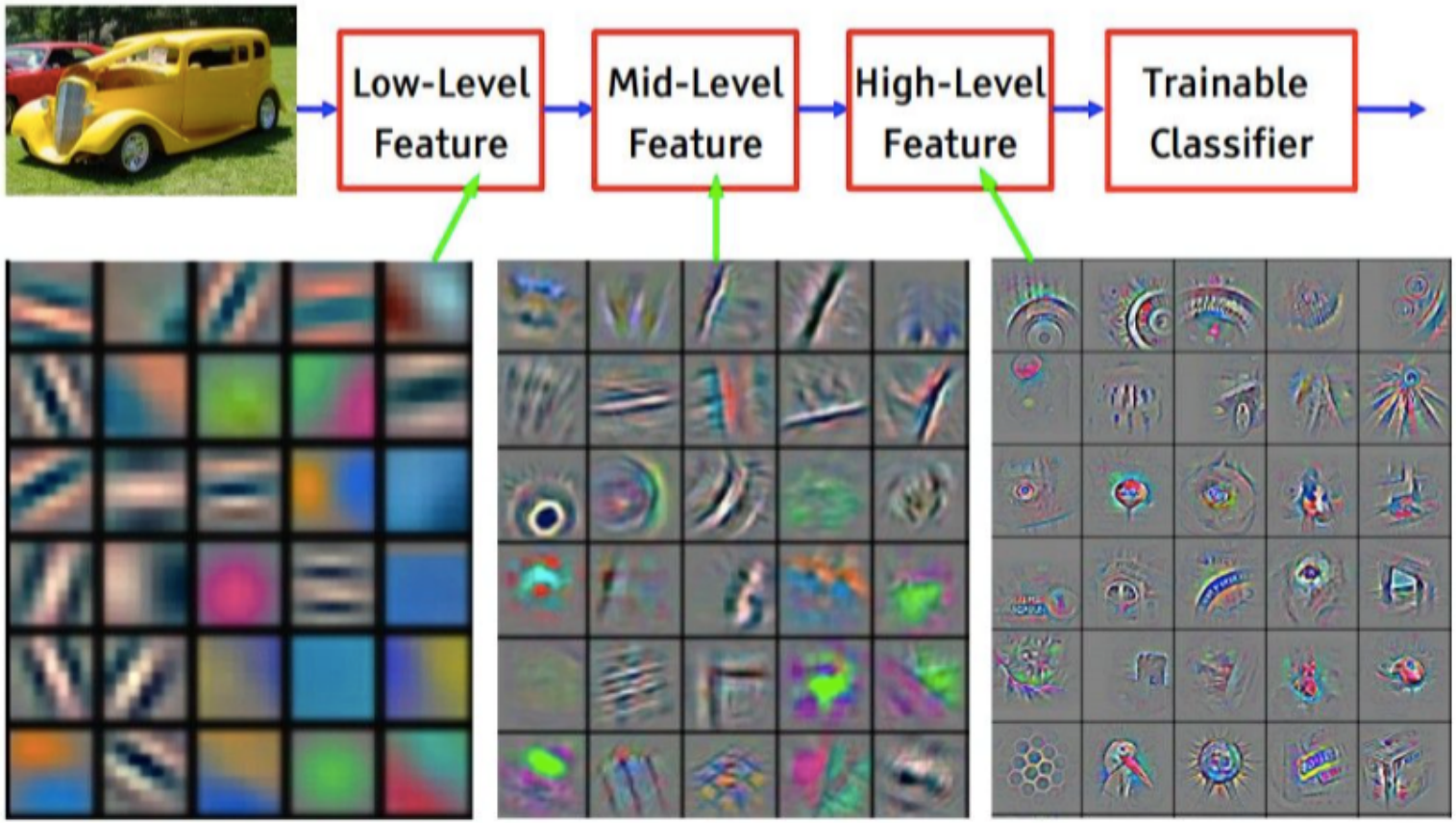
# Neural Networks - Training



# Deep Neural Networks - Quick background!

Deep networks learn hierarchical basis functions - efficient encodings.




Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Image: `yann.lecun.com`

# Machine learning problem categories

Define:

- **Data observations**: $x$.
- **Corresponding data labels**: $y$.

Problem categories:

- **Supervised**: Learn $P(y|x)$ given observations of $y$ and $x$.
- **Unsupervised**: Learn $P(x)$ given observations of $x$.
- **Semi-supervised**: Supervised learning augmented by unsupervised techniques.
- **Reinforcement learning**: decision making (not covered in this talk).

Applications to anomaly detection?

# Supervised learning for anomaly detection

- Given data labelled as either 'ok' or 'anomalous' can detect future anomalies by pattern matching.
- The nominal states of some engineering system may be predicted from data, models or simulations.
- Data or simulated data used as a proxy to train an anomaly detection model.

Procedure:

- Get training data of the form $x$ (observations) and $y$ (labels) that indicate if each $x$ is normal or anomalous.
- **Training phase**: given training data, $(x, y)$, learn function $f(x)$ that best fits the $(x, y)$ pairs.
- **Monitoring phase**: given observations, $\hat{x}$, ask model to output $\hat{y} = f(\hat{x})$. $\hat{y}$ tells you if the observation is ok or anomalous.

# Supervised learning for anomaly detection

Regression techniques:

- Simple methods like linear or logistic regression.
- Deep Neural Networks.
- Gaussian processes.

## Gaussian Process Regression

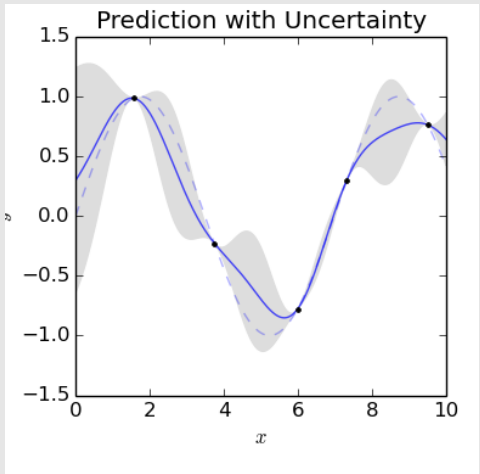



Image source: `tinyurl.com/ybkzvwxj`

## DNN Approximate Gaussian Process Regression

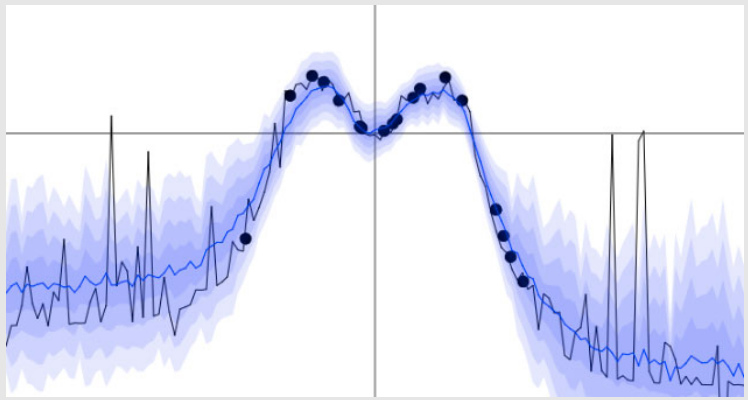


Image source: `tinyurl.com/gqktfpw`

# Traditional model projection techniques

**Problem:** may not have any observations!
If the system behaviour can be predicted, then anomalies can be detected.

## Option 1: Use observations to project forward

- Autoregressive Moving Average (ARMA) family of models: assume that values in the future depend on trends and values from the recent past. Define recency in terms of signal autocorrelation decay.
- More sophisticated - Recurrent Neural Networks

## Option 2: Use a physical model to predict future time behaviour

- Typical approach in engineering
- Partial Differential Equations, dynamical systems
- Circuit and network simulations
- Uncertainty Quantification

# Unsupervised learning for anomaly detection

**Unsupervised learning is all about finding structure in data.**

For anomaly detection:

- Avoid need for labelled anomalies.
- Attempt to find patterns in data, and then classify outliers based on the failure to conform to those patterns.
- In general: build a compressed representation of the data and then classify anomalies based on the error during reconstruction of the data from the compressed model.

Techniques:

- Clustering (e.g. k-means, spectral clustering)
- Principle Components Analysis
- Support Vector Machines
- Autoencoder Deep Neural Networks
- Filtering e.g. Sequential Bayesian Filtering

# Traditional clustering for anomaly detection

## Outliers: fail to fit any cluster.

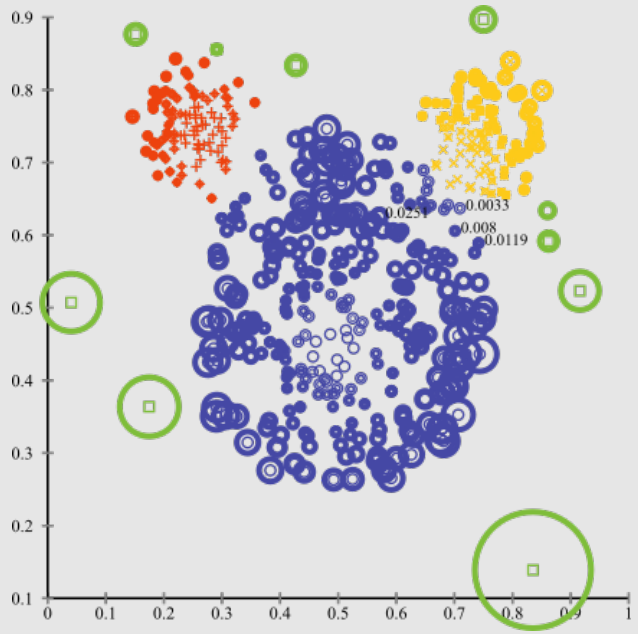



Image: `tinyurl.com/ydfoejvf`

## Clustering is a suite of techniques with different strengths and weaknesses.

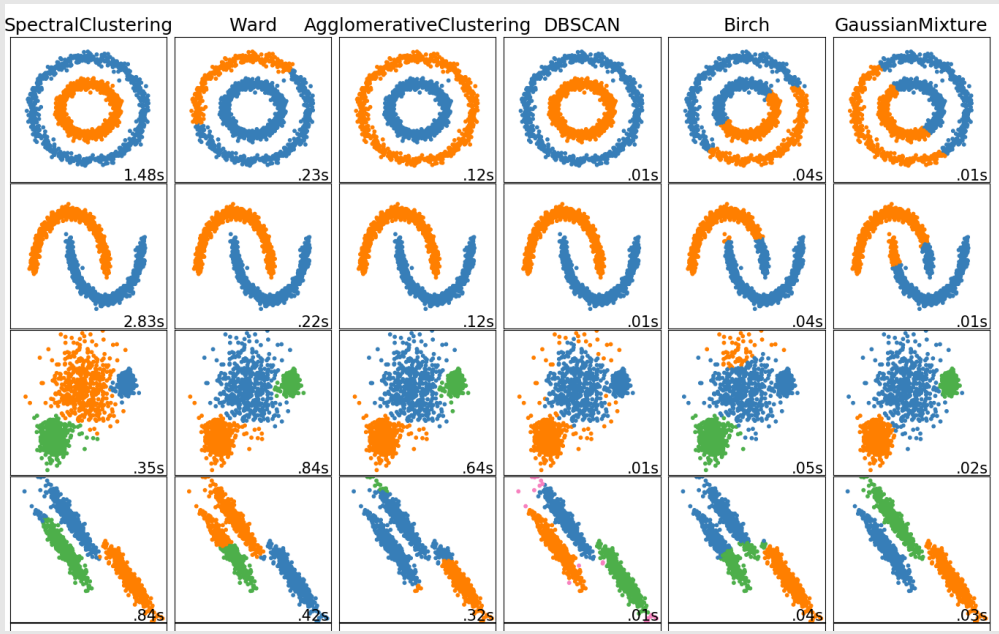


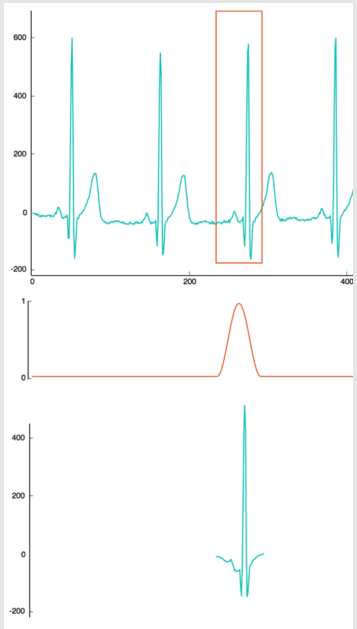

Image: `tinyurl.com/y7xubpnf`

# Time series data clustering: Sliding Window Clustering

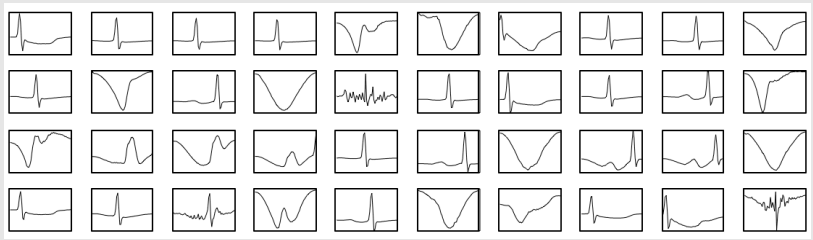
What if the data is cyclical but not seasonal (for example, heartbeat patterns)?

- Find short time trends with a sliding window.
- Cluster these short time trends into groups.
- Detect anomalies by checking for deviation from known patterns (clusters).

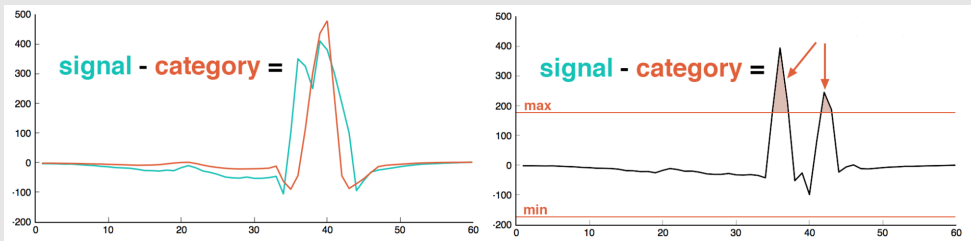
## 1) Moving window feature detection



## 2) Find cluster features



## 3) Anomalies: high variance deviation from closest cluster

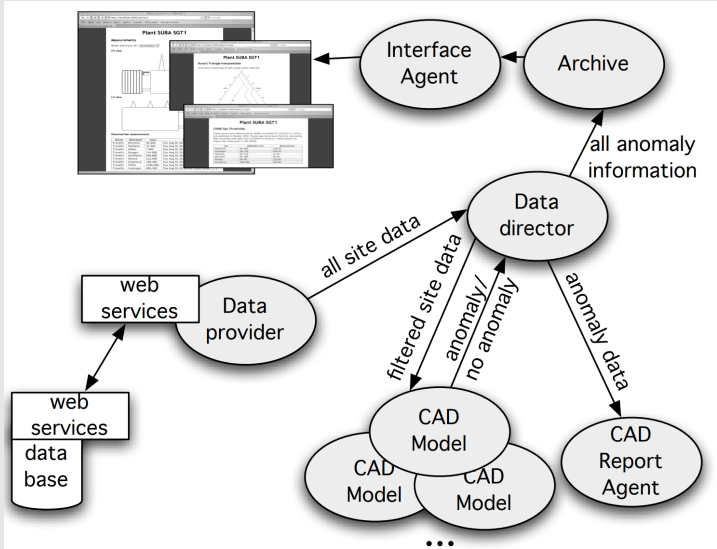



Example from `tinyurl.com/ybtrebcr`

# UK National Grid: Transformer monitoring (circa 2010)

- $>$ 50 sensors installed on two 275/132kV 180 MVA transformers
- Sensors: temperature, vibration, moisture, dissolved gas, weather and load
- Paper: `doi.org/10.1109/pes.2011.6038911`

## Data framework




Communication frameworks matter

## Gaussian Mixture Model Clustering (via EM)

Anomaly detection: Gaussian Mixture Model clustering

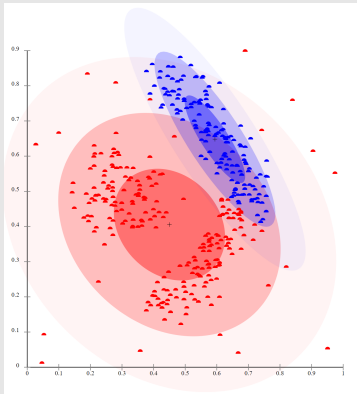


Image: `tinyurl.com/o24jxyq`

# DNN Autoencoder anomaly detection

More modern form of clustering based detrending available using DNNs.

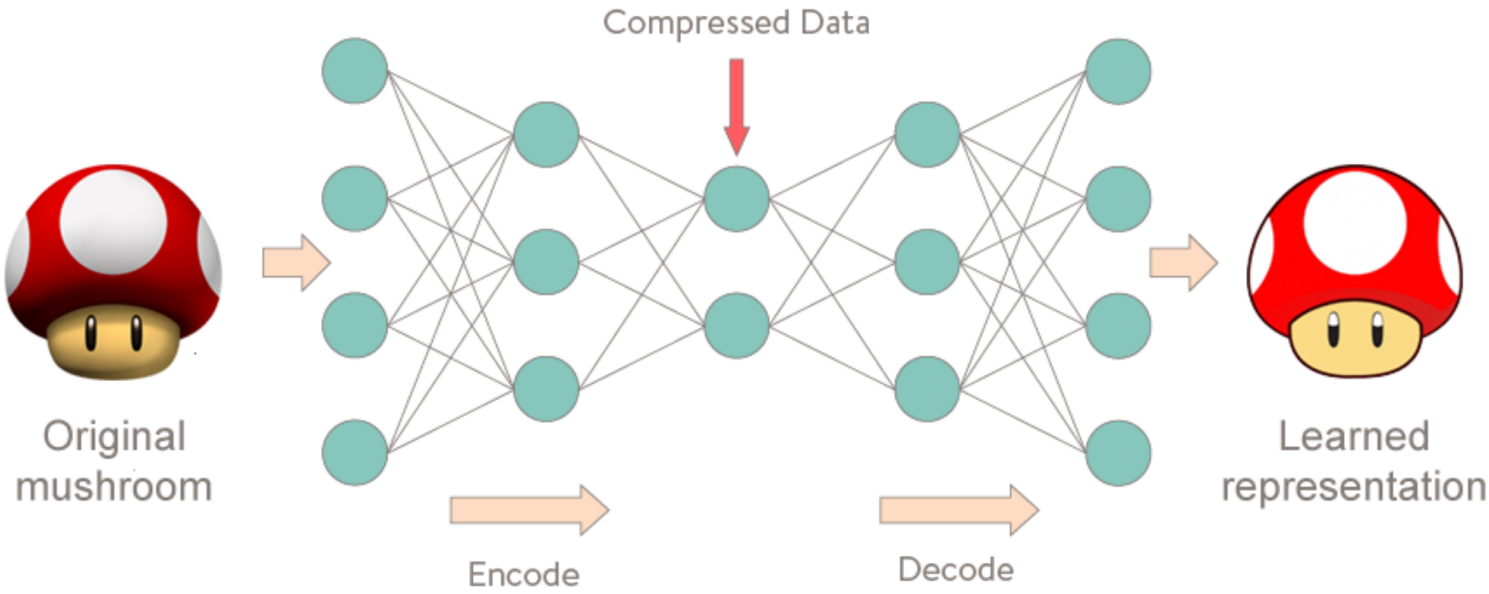



Image source: `tinyurl.com/yckac8x5`

**Training Phase:** Use Deep Neural Network to learn how to reconstruct data from compressed representation.

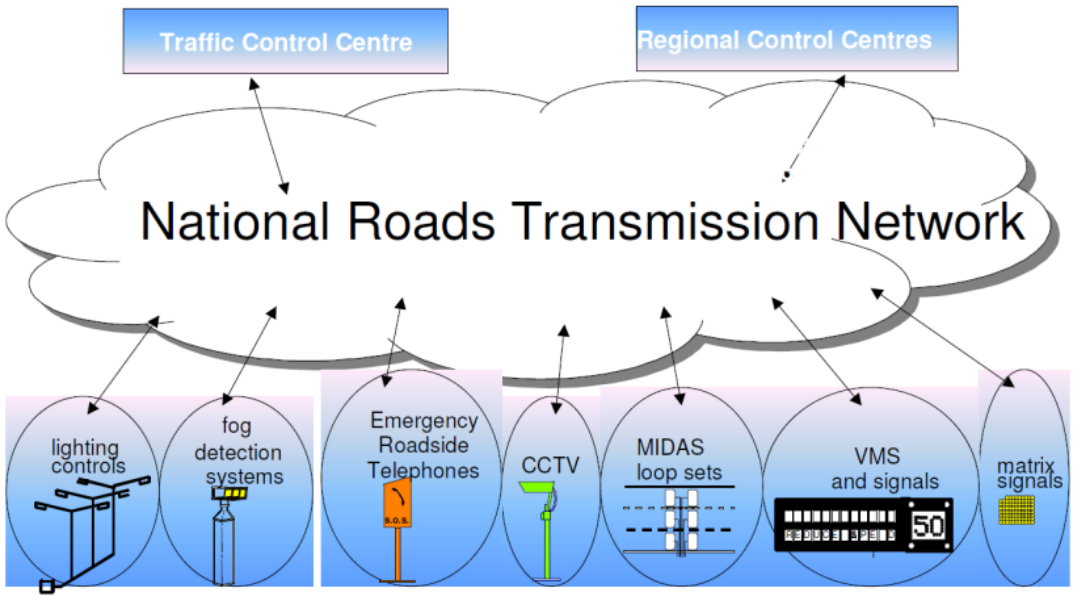
**Monitoring Phase:** Pass observations through the autoencoder. Reconstruction error is a proxy for how different the new observation is from previous observations.

# Data monitoring in more traditional fields

Traditional engineering fields face a number of challenges:

- Heterogeneous data: high dimensional, spatial and/or temporal, highly variable.
- Difficult to propose a general solution, what counts as an anomaly is specific to the intended use of an engineering system.
- Data quality is often poor (non-digital data, one-off observations).
- Non-stationary data sets make differentiation of anomalies from trends difficult.
- Large projects have large computational overhead.

# Data forms: interlinked 'simple' components




Traffic network monitoring

- Interlinked components replicated across project domain.
- **Historically:** Manual data processing by design office engineers. Labour intensive. Reliant on 'engineering judgement' and experience.
- **Future:** Use big data techniques to automate data processing.

Images: `tinyurl.com/ybj8vtwy`

# Data forms: single instance high volume data

Laser point cloud scan of a tunnel

- Project components unlikely to be repeated or mass produced.
- **Historically:** Manual data processing by design office engineers. Labour intensive. Reliant on 'engineering judgement' and experience.
- **Future:** Use big data techniques to automate data processing.

Images: `tinyurl.com/y9vrqcvr`

# Exemplar Application: IT Infrastructure Security

- Information technology infrastructure (e.g. servers, routers, network traffic etc) monitoring with machine learning techniques is a rapidly growing field.
- A number of commercial operators (e.g. DarkTrace, Netflix).
- Deep Neural Network techniques.
- High quality, high speed data available.
- Willingness to adapt to new technologies.
- High speed environment renders humans less useful.
- IT administration skill set transferable to large scale data processing.

# Replicating this success for Infrastructure in general

- Improve data collection capabilities (more and better sensors).
- Improve data processing frameworks and connectivity.
- Humans are too slow: improve algorithms using AI techniques.

Image: `tinyurl.com/y8buusl3`

# DCE at the Turing: System health

At The Alan Turing Institute: Data-Centric Engineering Program:
Project: Estimating System Health from Streaming Sensor Data

**Goals:**

- Quantify engineering system performance from data
- Use sensor data to estimate preventative maintenance schedules
- Characterise anomalies, not just flag them
- AI driven reliability and time to failure analysis

**Interdisciplinary work at the intersection of:**

- Mathematics
- Computer Science
- Engineering knowledge
- Risk analysis
- Uncertainty Quantification
- Data science
- Probability

# Conclusions: Challenges and risks

Our new methods are very powerful and we really want to use them for projects:

But: Data based methods can behave in strange ways. No physical knowledge. Can be dangerous in the real world.

Image: `arxiv.org/pdf/1612.03242v1`

# Conclusion

**Thank you!**