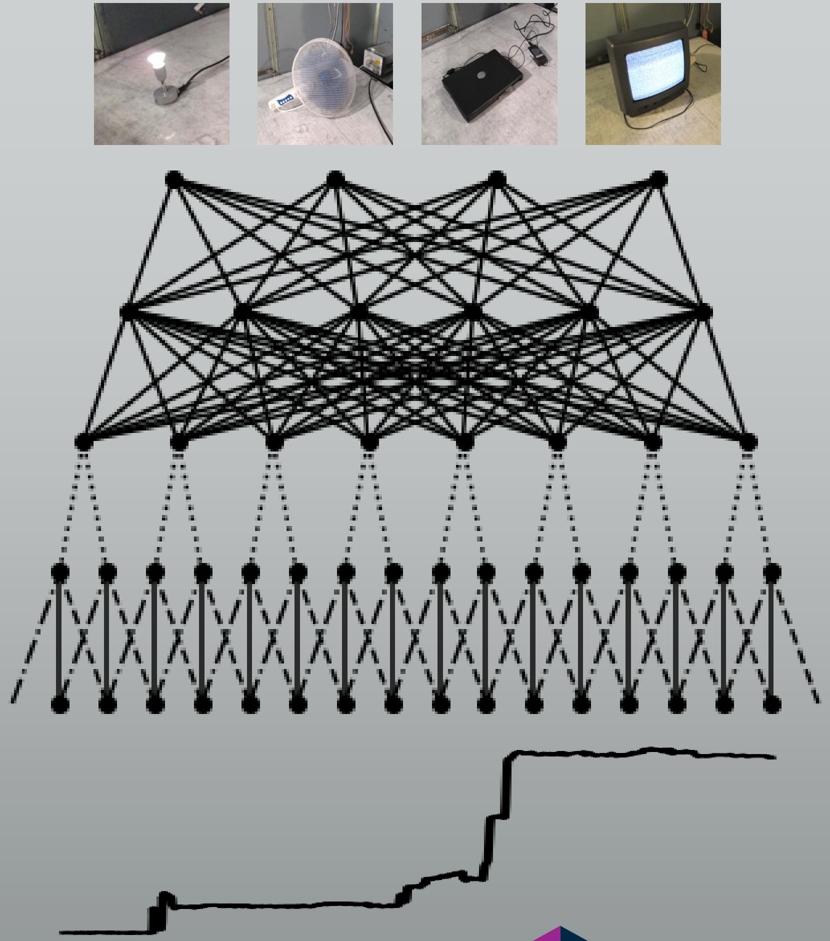
Electrical device classification using deep learning



Dr Richard Lane

Sensor Signal Processing for Defence Conference (SSPD)

15-16 September 2020

Contents

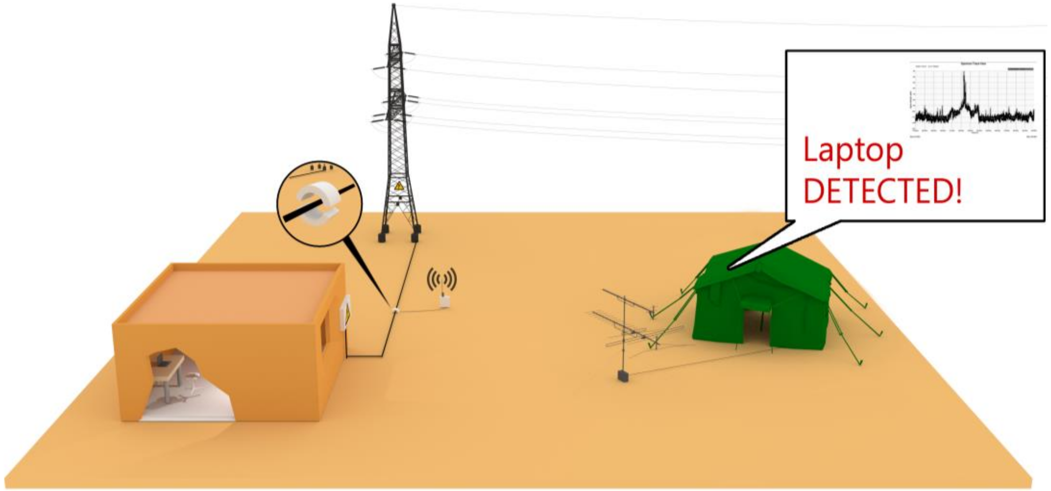
Introduction

Introduction

- Electrical devices unintentionally inject signals onto the mains power supply
- Can measure current using commercially available equipment designed for EMC testing
- Time series produced by various devices are visually distinct when switched on or off
- Allows identification of devices
- Deduce the “pattern of life” of equipment
- Applications:
  - Smart metering
  - Intelligence gathering

https://nl.aliexpress.com/item/Broadlink-SP2-UK-Standard-Smart-Home-Wireless-WiFi-Remote-Control-Electrical-Smart-Plugs-Sockets-by-app/32796081850.html

https://freepricecompare.com/tag/uk-smart-metering/

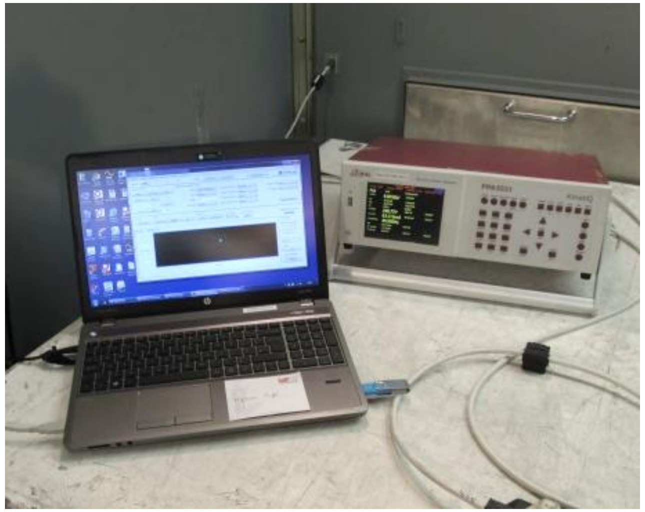


Electrical Device Data

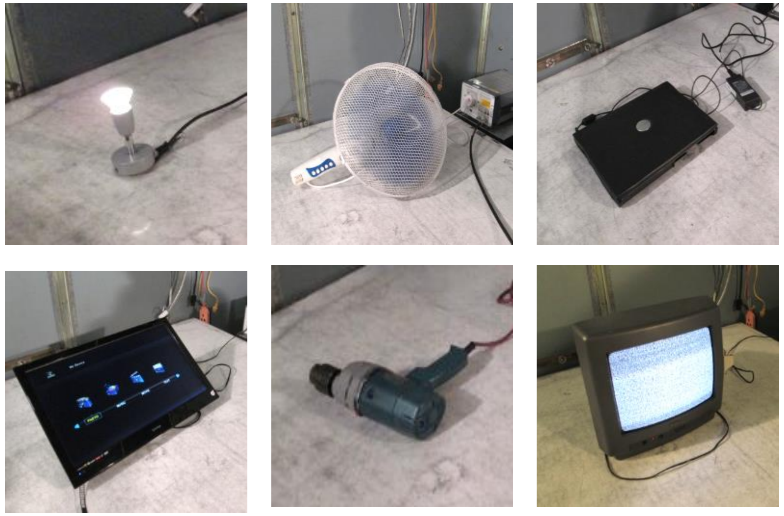
Measurement Set Up

- Measurements of several devices were made:
  - Compact fluorescent lamp (CFL)
  - Cathode ray tube (CRT) television
  - Laptop and charger (2 different models of charger)
  - Power drill
  - Desk fan
  - Filament lamp
  - Flat screen television
  - Halogen lamp
  - Light emitting diode (LED) lamp
  - Nokia and Samsung phone chargers.

Newtons 4th Power Analyzer & Controller Laptop



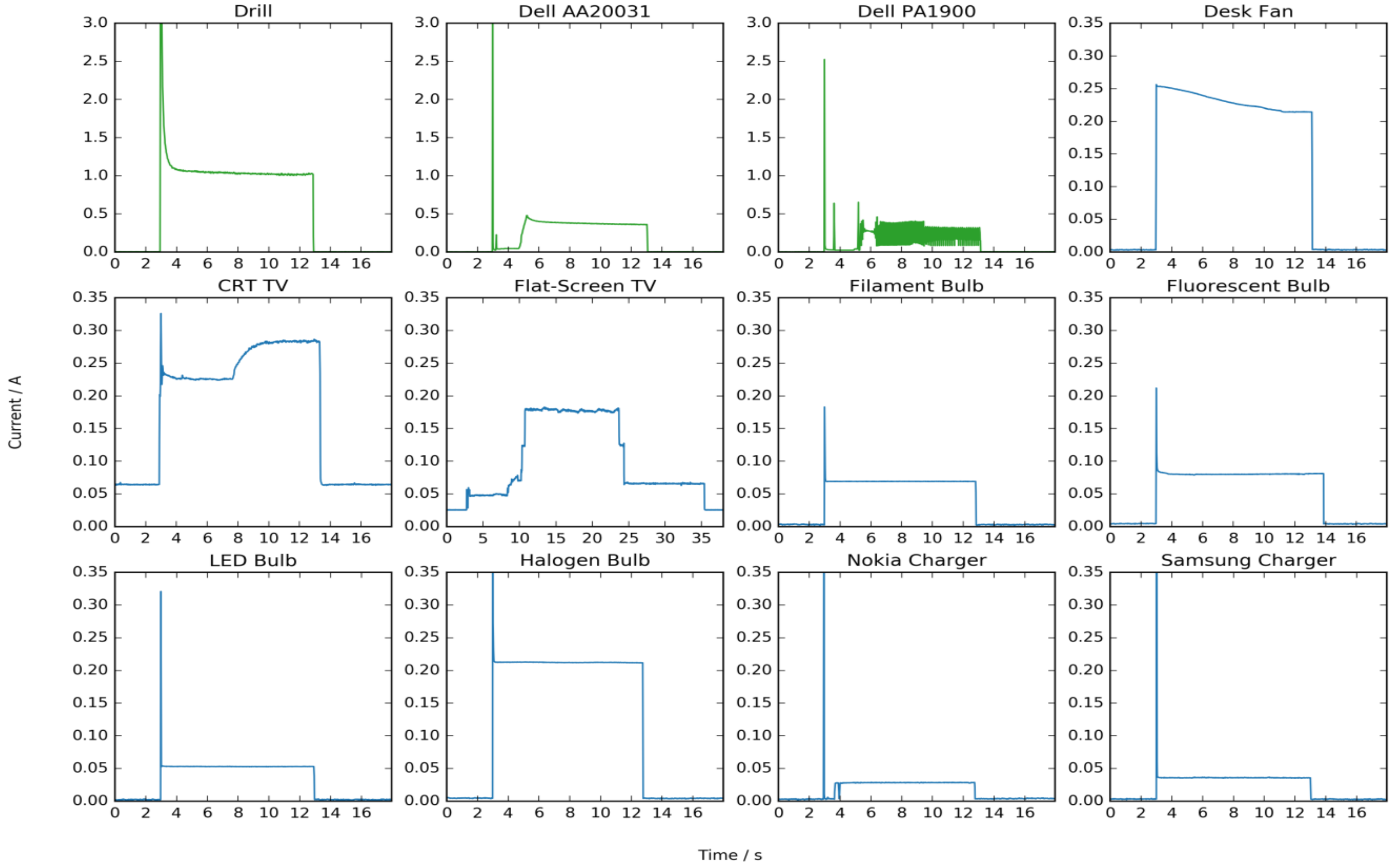
Selection of Devices



- Devices cover broad range of characteristics and have significant variability in the amount of power consumed
- Root mean-square (RMS) current signals measured at sample rate of 50 Hz
- Data recorded for the cycle: power off, power on, power off
- Signals recorded with a consumer unit typically used with building electrical supplies and a 15 m extension lead
- Measurements representative of what could be obtained in the field with a passive measurement device

Device Signatures

- 12 devices
- 233 time series



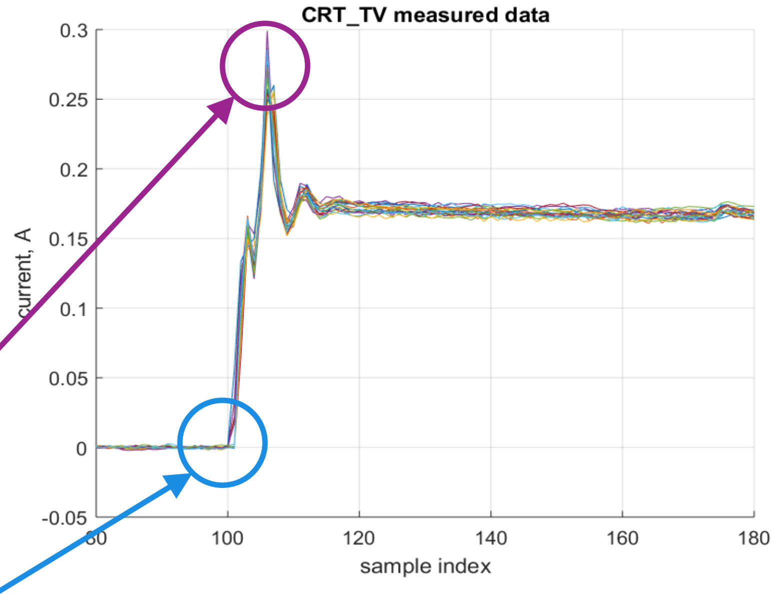
Generating Realistic Simulated Data

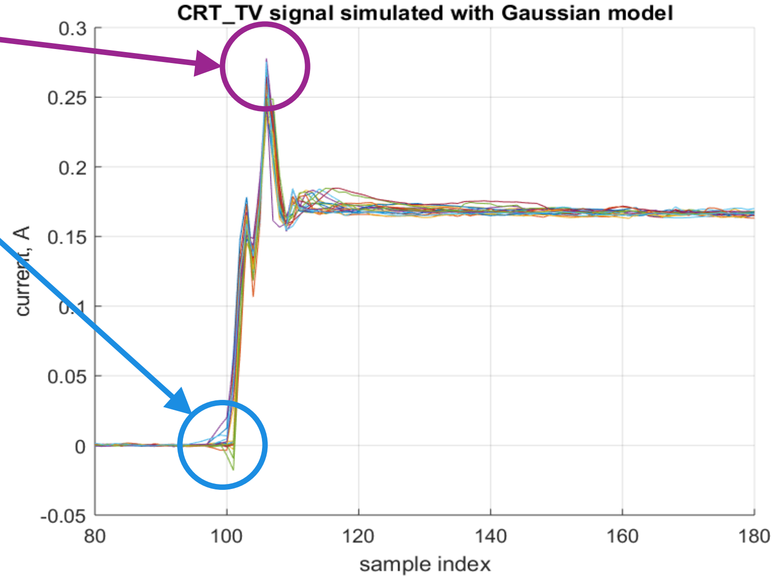
Motivation for simulating data

- Measurements were made for devices operating individually
  - Different days to test the consistency of device signatures over time
  - On/off events manually annotated
- Need to analyze data from devices operating simultaneously
  - Number of device combinations is huge
  - Impractical to annotate large amounts of measured data
- Process for simulating device signatures is needed
- Previous simulation based on measurements was developed
  - Analyzes nonlinear variations in time exhibited by each signature
  - Includes local variations in signal amplitude
  - Constructs a statistical model
  - Can be used to generate signals randomly for each device
- Enables possibility of simulating large amounts of data

Motivation for improved model

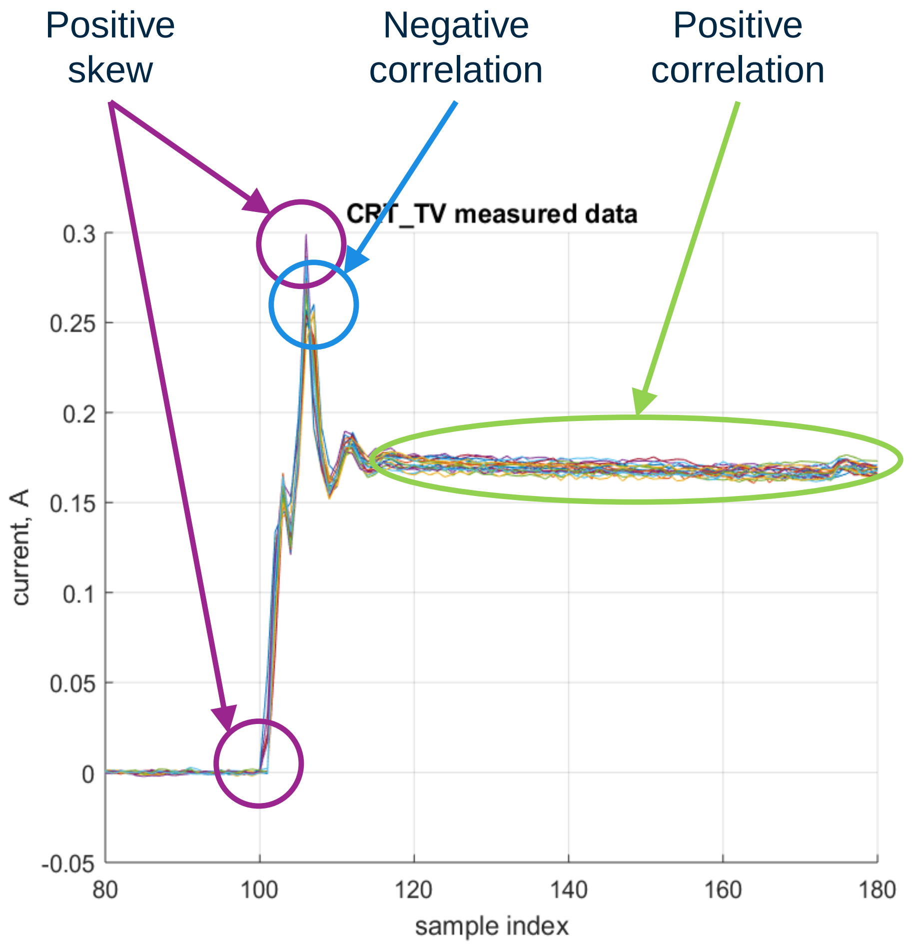
- Previous model
  - Amplitude and time statistics
  - Multi-variate Gaussian probability distribution
  - Reasonable assumption for much of the data
- Gaussian simulated data can't model this properly
  - Some parts of real signals have skewed distributions
  - Real data peak takes on higher values than expected
  - Simulation has significant negative relative current values
- This motivates need for a non-Gaussian distribution





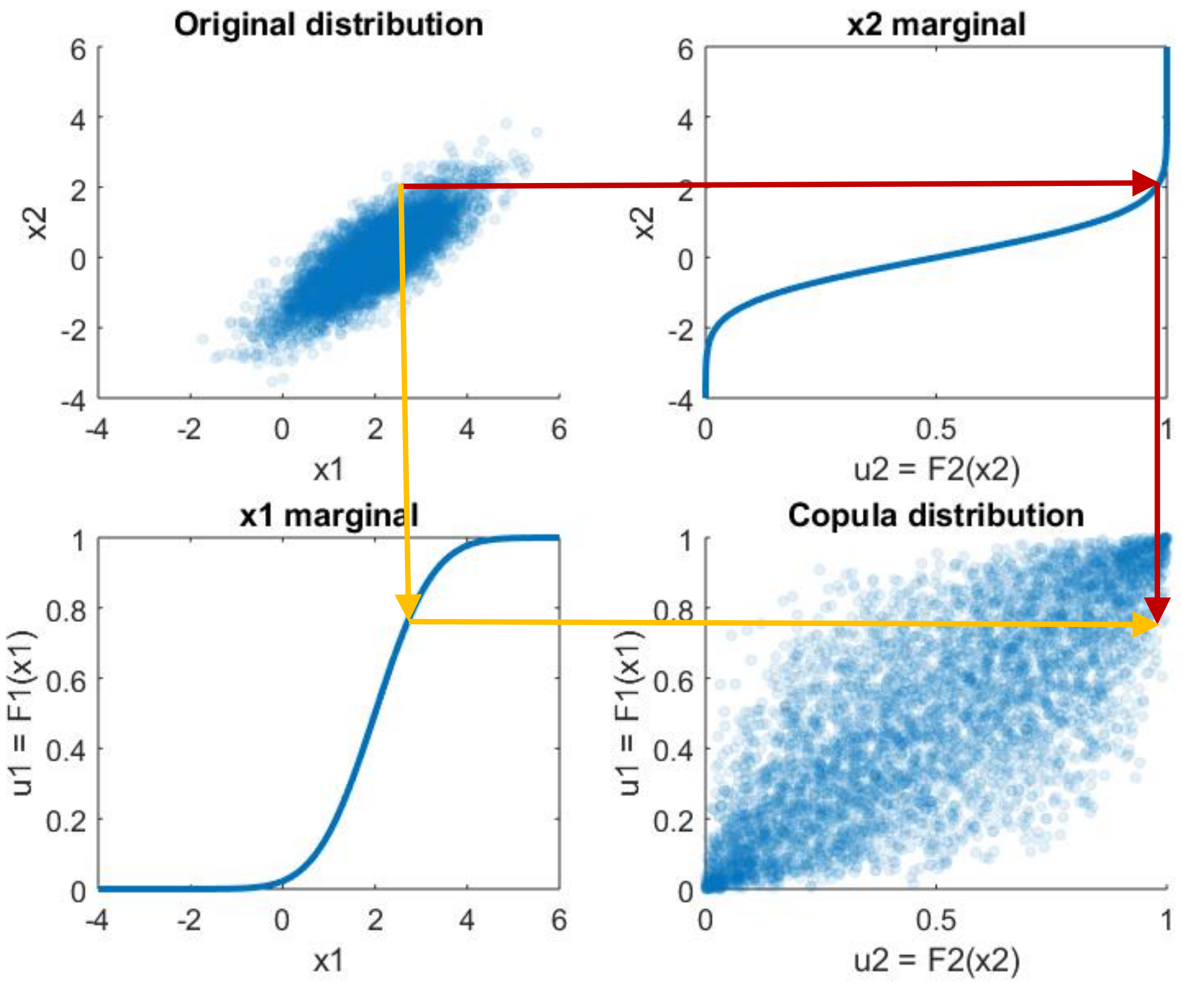
Choosing a suitable model

- Signal characteristics: skewed and correlated
- Skewed data: positive and negative
- Correlated data:
  - Correlations between signal values at different times
- Positive correlations
  - Slowly varying signals likely to be high if previous sample was high
- Negative correlations
  - Sharp spikes in the data often spread over adjacent samples
  - Total constant energy => if one sample is high then adjacent one will be low
- Modelling asymmetric correlated non-Gaussian data is not straightforward
  - Not many suitable probability distribution families exist
  - Possible to model such data using *copulas*

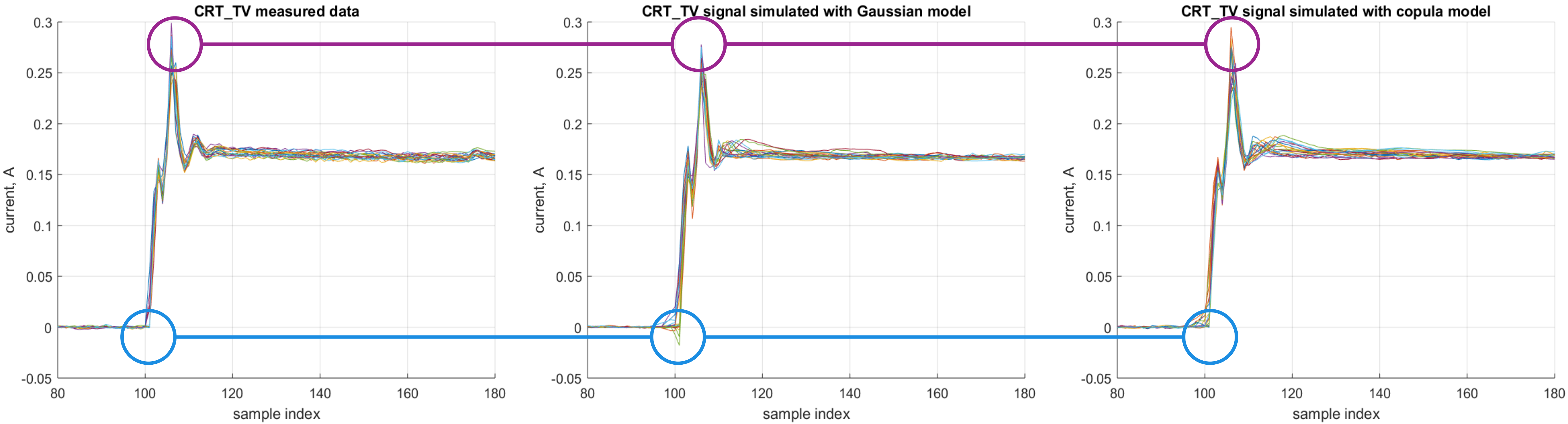


Copulas

- Copulas allow separate modelling of:
  - Distribution of each dimension – “marginal”
  - Distribution of correlations between dimensions – “copula”
- Copula properties:
  - Multivariate distribution function
  - Support on the unit hypercube $[0,1]^d$
  - Uniform marginal distributions for each dimension
- Sklar’s theorem
  - If H is an arbitrary multidimensional distribution function
  - With one-dimensional marginals $F_1, \dots, F_d$
  - $H(x_1, \dots, x_d) = C(F_1(x_1), \dots, F_d(x_d))$
  - For some copula C, which always exists.
- Freedom to choose different distributions for each dimension
  - Including distributions from different families
- Important for signal-modelling task
  - Use non-Gaussian distributions for skewed data (beta distribution)
  - Model correlated data (Gaussian copula)



Simulated data comparison



Measured data

Simulated data (Gaussian)

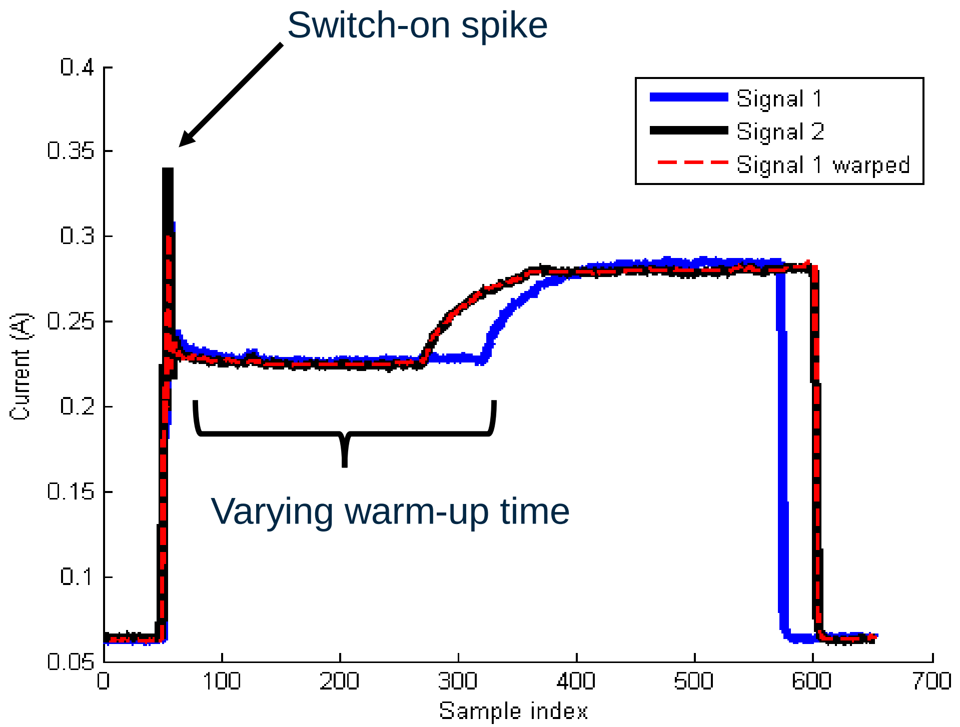
Simulated data (Copula)

- Copula-generated signals:
  - Are realistic compared to measured data
  - Have higher fidelity than those Gaussian-generated signals

Classification Algorithms

Baseline classifiers for comparison

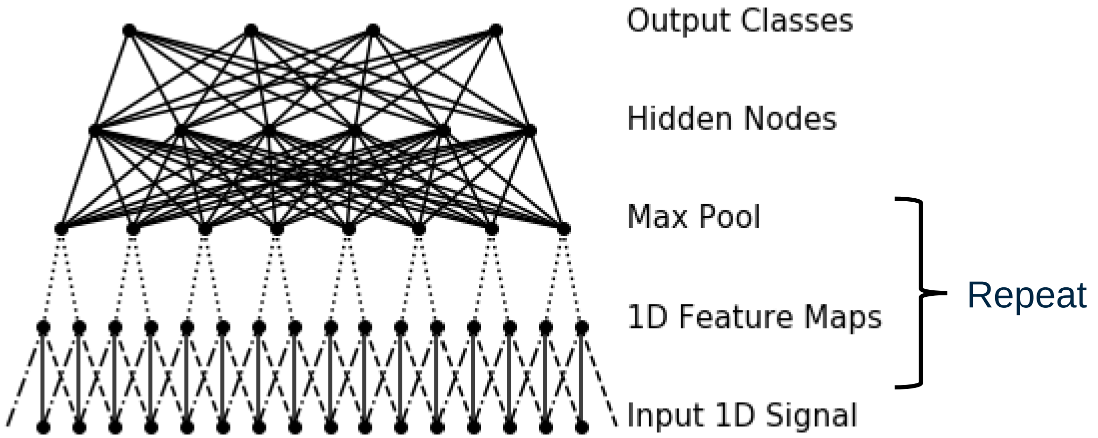
- Correlation classifier
  - Nearest neighbour type of classifier
  - Computes cross-correlation function between pairs of signals
  - “Distance” is inverse of peak of correlation function
  - Deals with linear shifts in time
  - Fast: $O(n_t \log(n_t))$ per comparison*
- Dynamic time warping (DTW)
  - Nearest neighbour type of classifier
  - Computes DTW distance between pairs of signals
  - Deals with linear and non-linear shifts/scales in time
  - Slow: $O(n_t^2)$ per comparison*
- Total computation speed
  - Training: effectively zero (just store the data)
  - *Test: $O(n_{train})$



Deep learning classification

- Convolutional neural net (CNN) model
  - Robust to linear and non-linear signal variations in time
  - Structure inspired by VGG-16 image recognition architecture
  - (Convolution – Max Pool)$\times\, n_l$ – Dense – Dropout – Softmax
  - Adapted to 1D signals
- CNN model characteristics
  - Architecture search found optimum values
  - Number of feature maps/filters $n_f = 32$
  - Number of max pool layers $n_l = 6$
  - Number of hidden dense nodes $n_h = 32$
  - Resulting network has 28,364 parameters
  - Zero padding at signal edges (signal remains same length)
- Training regime
  - Adam optimizer
  - Categorical cross-entropy loss function
  - 100 training epochs

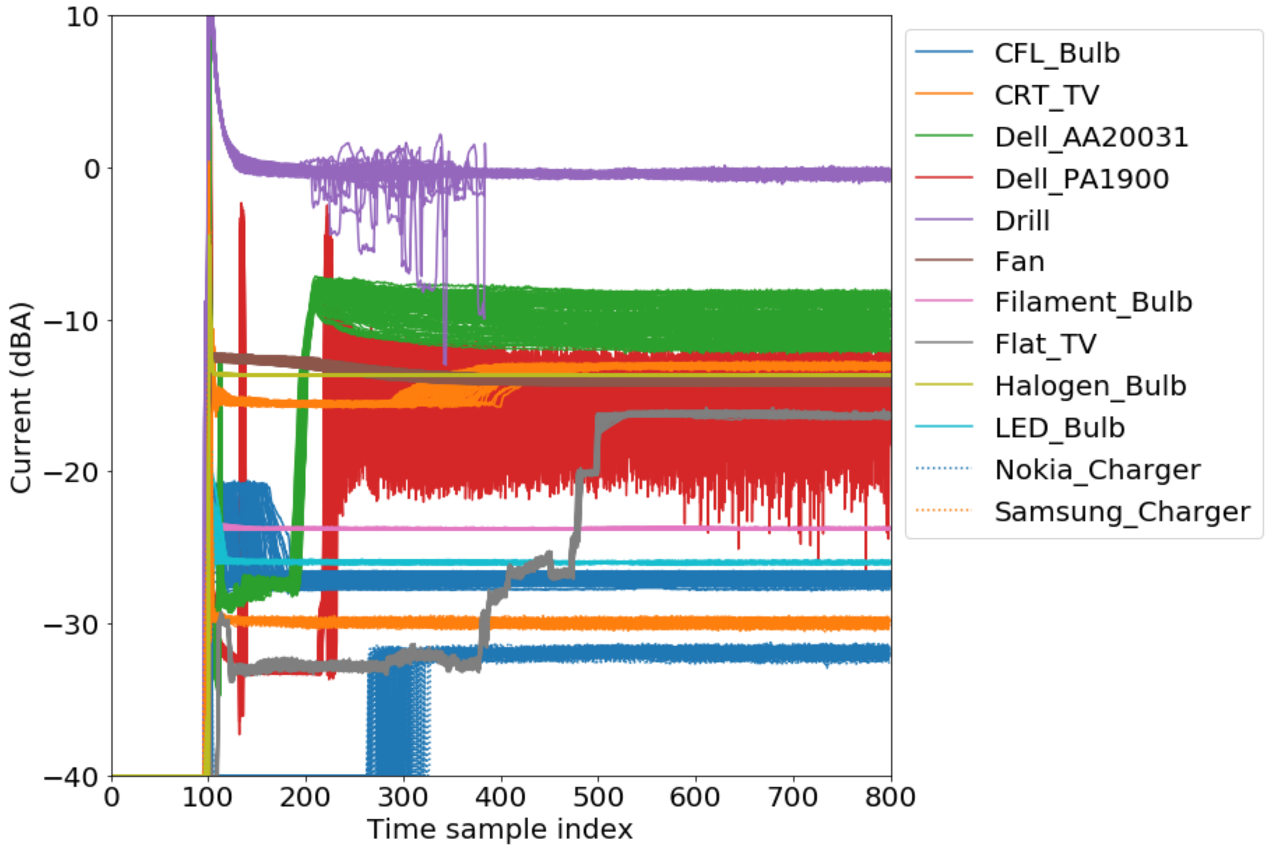
Layer Type	Filter size	#Filters or nodes	Activation function	#Parameters
Conv1D	3	n_f	ReLU	128
MaxPool1D	2	N/A	None	0
Repeat the above two layers $(n_l - 1)$ times				$3104(n_l - 1)$
Dense	N/A	n_h	ReLU	12320
Dropout	N/A	0.5	None	0
Dense	N/A	n_d	Softmax	396



Performance Assessment

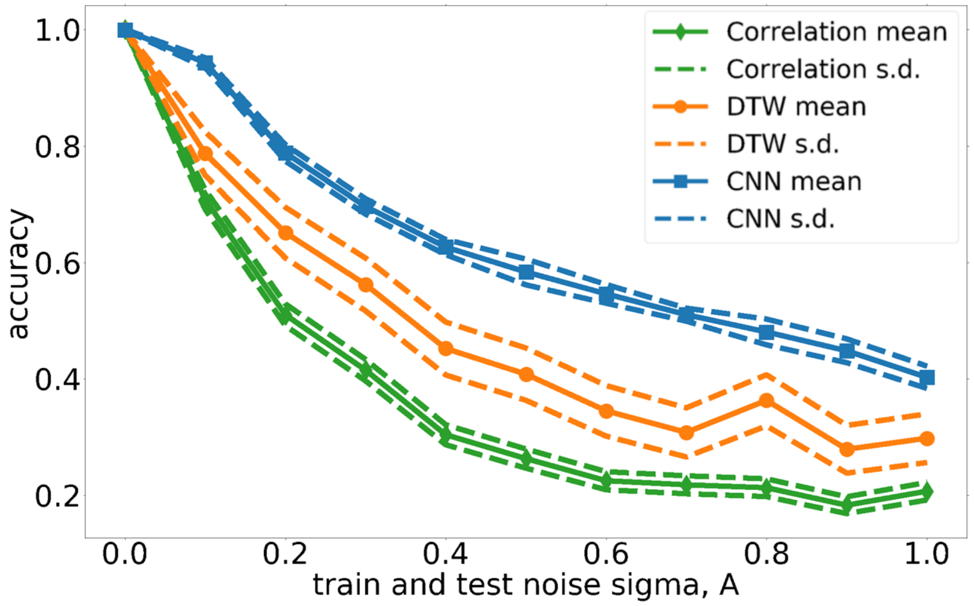
Experiment setup: data and classifiers

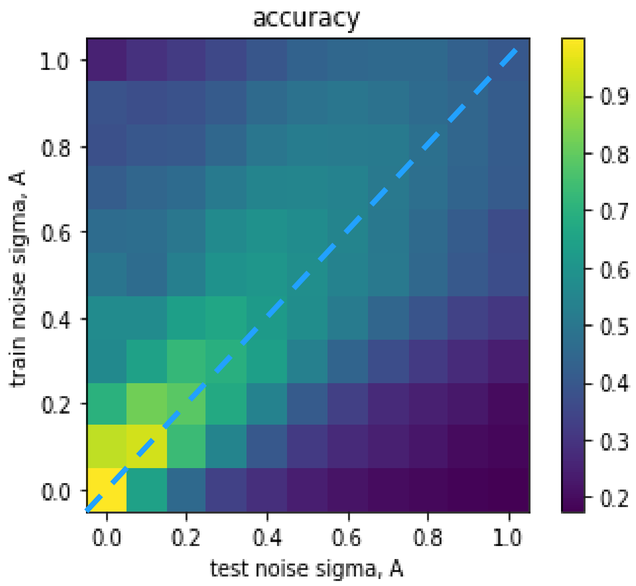
- Signals simulated for each of 12 devices
  - 200 switch-on event signals per device
  - 800 time samples per signal (16 s @ 50 Hz)
- Data split into training (70%) and test (30%) sets
  - 1680 total training examples
  - 720 total test examples
- Three classifiers applied to the data
  - Correlation classifier
  - Dynamic time warping (DTW)
  - Convolutional neural net (CNN)
- Effect of simultaneous devices analyzed
  - Add zero-mean white Gaussian noise
  - Varying standard deviation



Effect of noise on classifier accuracy

- 10 Monte Carlo runs
  - "Random" guess accuracy is 0.08 for 12 devices
- Experiment 1: Same train and test noise power
  - Noise-free data can be perfectly classified
  - CNN has best performance as noise is increased
  - DTW has middling performance
  - Correlation is worst
- Experiment 2: Different train and test noise power
  - Investigated for CNN only
  - Performance best when noise power is same
  - Large drop in accuracy for large mismatch in noise
  - Asymmetry:
    - Better: Trained with noise, tested with noise-free data
    - Worse: Trained noise-free, tested with noisy data
  - Need to train classifier in a variety of noise environments

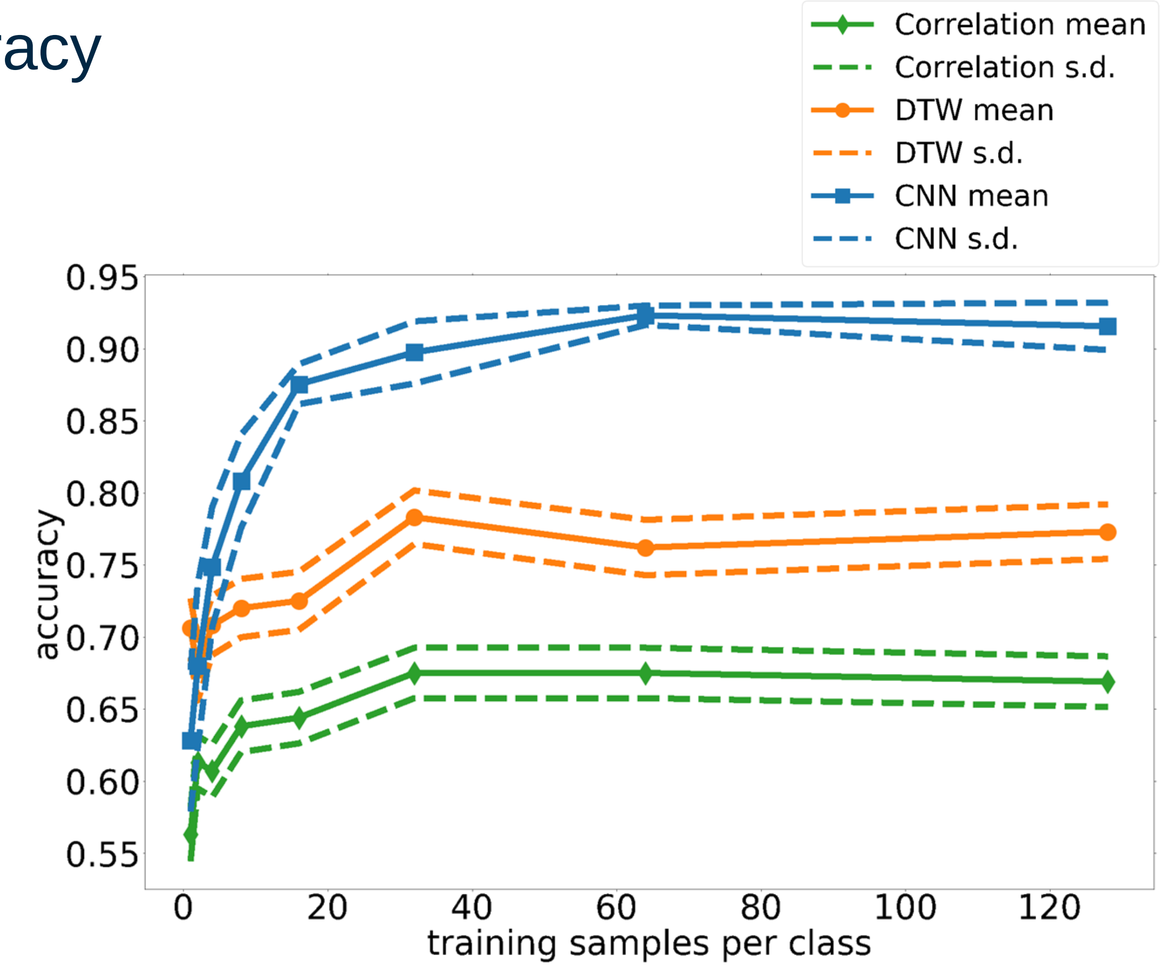




CNN classifier

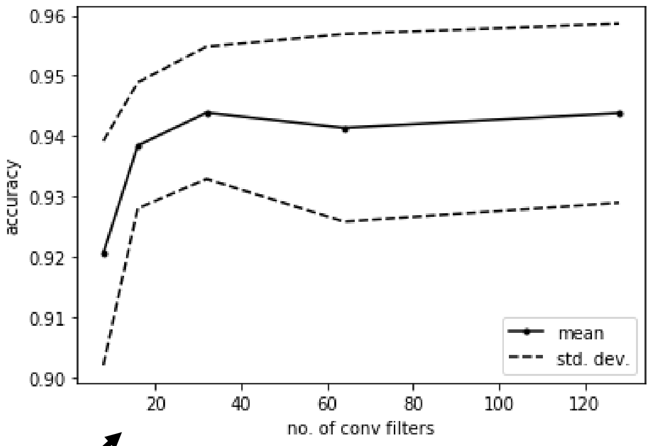
Effect of training data size on classifier accuracy

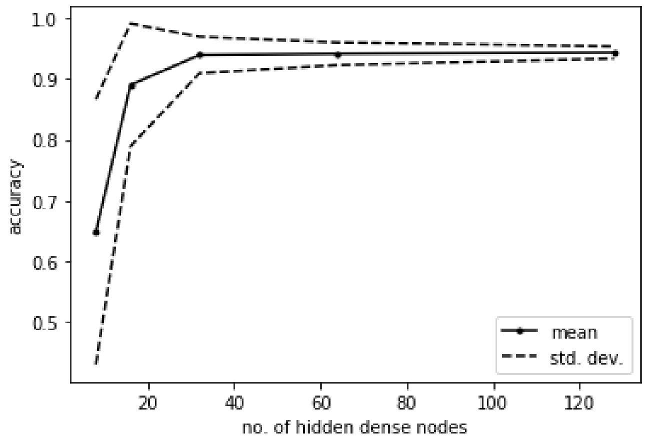
- More data => better model
- Accuracy measured as function of training set size
  - Noise level of 0.1 A used for train and test
  - Three classifiers tested
  - Performance increases rapidly with training set size
  - Plateaus after 32-64 training samples per class
- Small data situation (1-2 samples per class)
  - DTW outperforms the CNN
- CNN gains largest benefit from training data
  - Outperforms other classifiers with four or more training samples per class

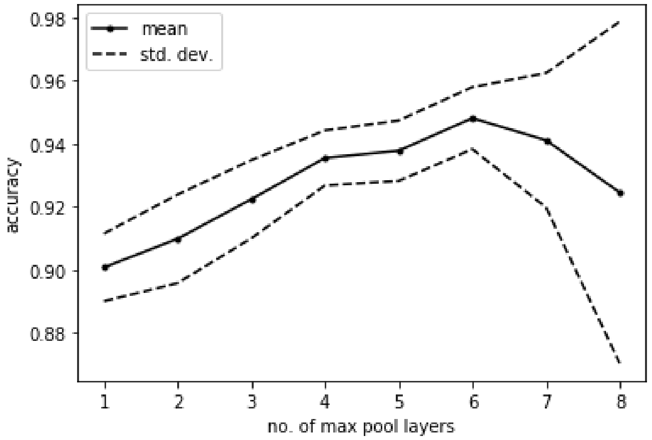


CNN structure sensitivity analysis

- One network structure parameter varied at a time
- Other parameters held at optimum values
  - Number of feature maps/filters $n_f = 32$
  - Number of hidden dense nodes $n_h = 32$
  - Number of pool layers $n_l = 6$
- Gaussian noise added (standard deviation 0.1 A)
  - Added during training and testing
  - Ensures some robustness to noise
- Weak dependence on filters (n_f) and hidden nodes (n_h)
  - As long as parameters are above minimum threshold
- Moderate dependence on layers (n_l)
  - Using 7 or 8 layers starts to lead to overfitting
- Structure should be useful in range of situations
  - Not over-trained to this specific data set







Conclusions

Conclusions and further work

- Sophisticated data simulation process has been designed
  - Can generate unlimited amounts of single-device event data
  - Used to test classifier performance as a function of training data set size or noise level
- CNN classifier developed
  - Maintains fair accuracy even with noise levels of 1.0 A
  - Better than the DTW and correlation classifiers
  - Only 4 examples per class required to outperform other classifiers
- Further work
  - Simulation enables creation of long time series
  - Multiple devices operating simultaneously
  - Annotated switch-on or off events
  - Analyze signal detection algorithms
  - Test using data from a real building