

Recurrent Auto-Encoder Model for Large-Scale Industrial Sensor Signal Analysis

Timothy Wong^{1,2} Zhiyuan Luo¹

¹Royal Holloway, University of London, Egham TW20 0EX.

²Centrica plc, Millstream, Maidenhead Road, Windsor SL4 5GD.

19th International Conference on
Engineering Applications of Neural Networks (EANN 2018)
3-5 September 2018 - Bristol, UK

Background

- ▶ Gas compression sub-system at a gas terminal
 - ▶ Powered by aeroderivative gas turbine
 - ▶ Two centrifugal compressors driven on a single shaft
 - ▶ Regulates gas pressure

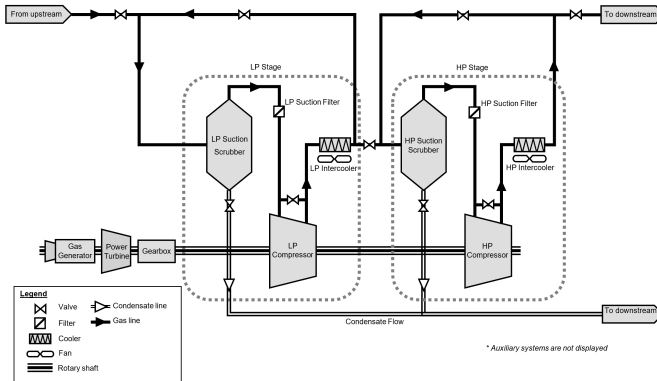


Figure: Simplified process diagram for the compression sub-system

Components

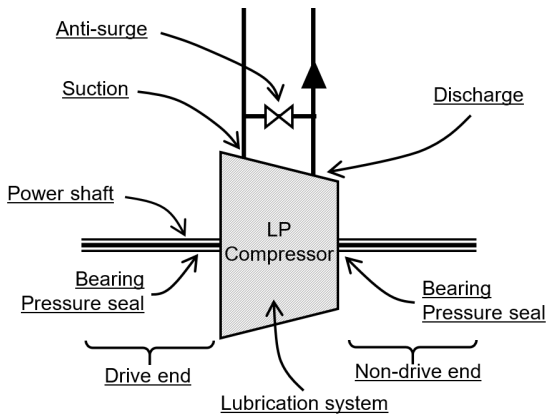


Figure: Centrifugal compressor (LP stage)

Components

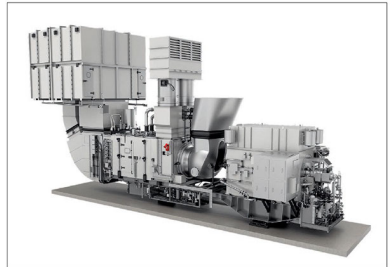
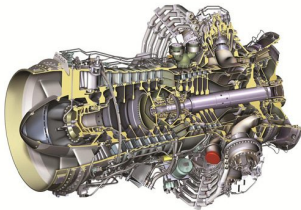


Figure: Jet engine package

Problem

- ▶ Large-scale industrial processes contains many sensors
- ▶ Sensor records continuous stream of real-valued measurements (e.g. temperature, pressure, rotary speed. . . etc.)

Research Question

Can we determine the underlying operating states of the process?

- ▶ Unevenly-spaced time series (inconsistent interval)
- ▶ Can be convert into regularly-spaced time series through downsampling.
- ▶ P sensors can form a P -dimensional multivariate time series.

$$\{\mathbb{R}_t^P : t \in [1, T]\}$$

Recurrent Auto-Encoder Model

- ▶ RNN encoder-decoder model
- ▶ Convert to recurrent auto-encoder by aligning input/output time steps
- ▶ Deep structure learns abstract temporal patterns
- ▶ Performs partial reconstruction on the decoder side
(Decoder dimension smaller than Encoder dimension)

$$\begin{cases} f_{\text{encoder}} : \{\mathbb{R}_t^P : t \in [1, T]\} \rightarrow c \\ f_{\text{decoder}} : c \rightarrow \{\mathbb{R}_t^K : t \in [1, T]\} \end{cases} \quad K \leq P$$

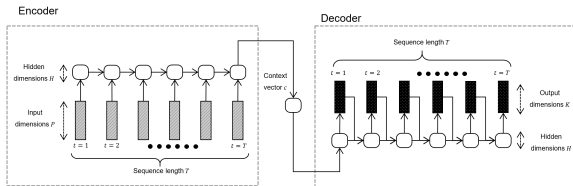


Figure: RNN encoder-decoder model

Long-short term memory

LSTM neuron [HS97]

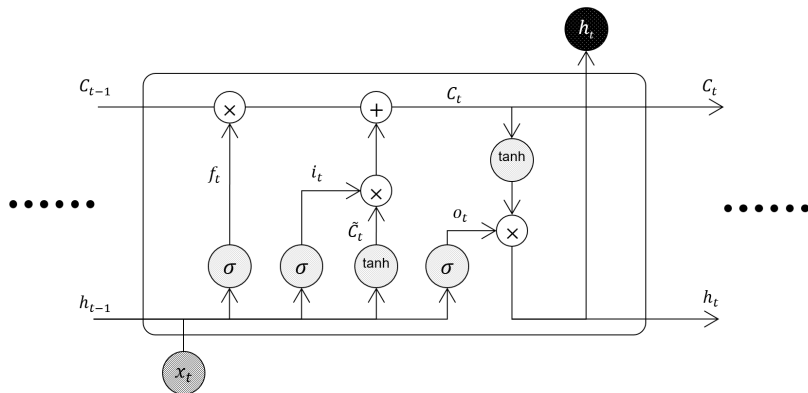


Figure: Internal structure of a long short-term memory block [Ola15].

Long-short term memory

Forget gate

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$$

Input gate

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c)$$

Update hidden state

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t$$

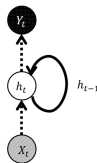
Output gate

$$O_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

$$h_t = O_t \times \tanh C_t$$

Regularisation

- ▶ Prevents network overfitting
- ▶ Dropout wrapper [SHK⁺14]
- ▶ Masks neuron inputs
- ▶ Non-recurrent connections only [ZSV14]



Sampling

- ▶ Recursively generating time series samples with fixed length
- ▶ Samples of length T can be generated from any time series dataset with length $T' \geq T$
- ▶ $T' - T$ samples can be generated
- ▶ Standardising dataset using z-score $z_p = \frac{x_p - \bar{x}_p}{\sigma_p}$

Algorithm 1: Drawing samples consecutively from the original dataset

Input: Dataset length T'

Input: Sample length T

```
1  $i \leftarrow 0$  ;  
2 while  $i \leq i + T$  do  
3   |   Generate sample sequence  $(i, i + T]$  from the dataset;  
4   |    $i \leftarrow i + 1$ ;  
5 end
```

Results

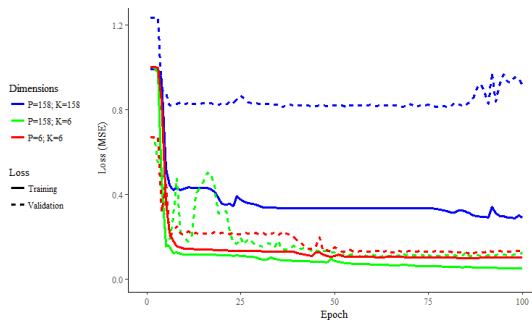


Figure: Effects of relaxing dimensionality of the output sequence on the training and validation MSE losses.

Sequence Reconstruction

- ▶ Specimens were selected randomly for qualitative examination
- ▶ Selected from held-out validation set

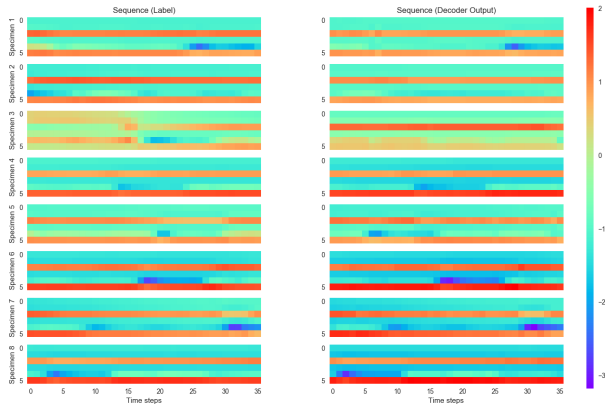
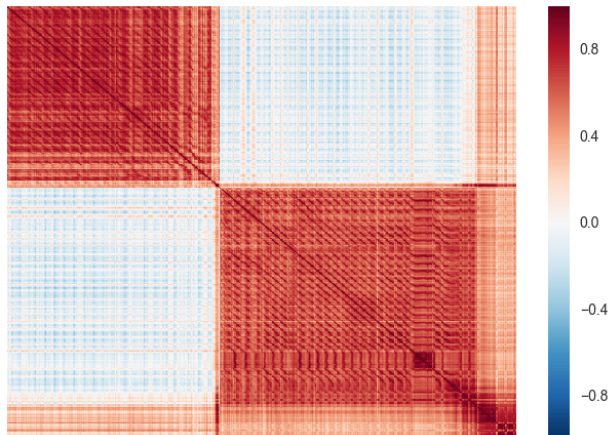


Figure: Randomly selected output sequences in the held-out validation set. Colour represents magnitude of sensor measurements in normalised scale.

Context Vector

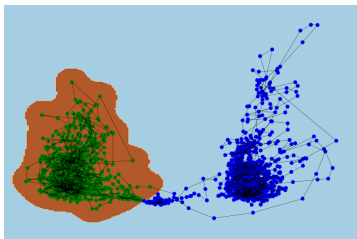
- ▶ Pairwise Pearson correlation
- ▶ Diagonal shows strong correlation among successive context vectors
- ▶ Showing gradual drift of operating state



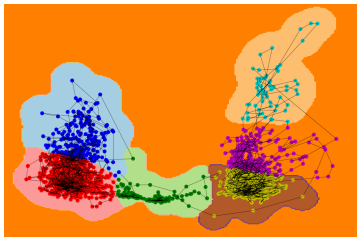
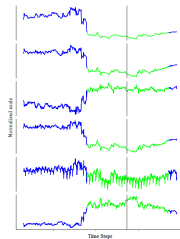
Context Vector

- ▶ Dimensionality reduction (PCA) to 2D
- ▶ Additional clustering algorithms can be applied (e.g. *K*-means)
- ▶ Decision boundary calculated using Support Vector Machine (RBF)

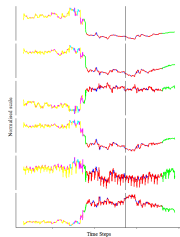
Context Vector: Example 1



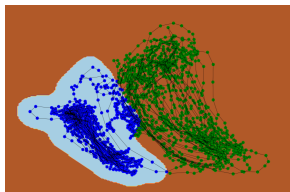
(a) 2 clusters



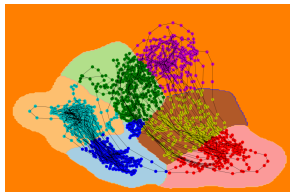
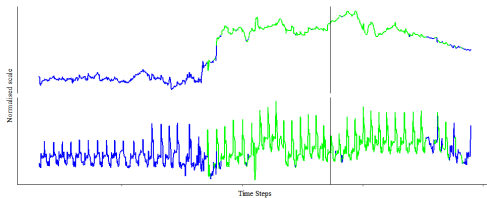
(b) 6 clusters



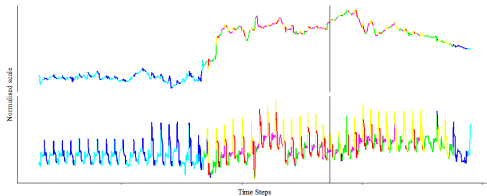
Context Vector: Example 2



(a) 2 clusters

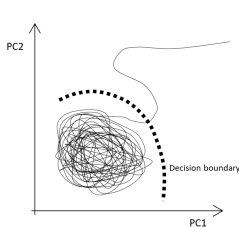


(b) 6 clusters

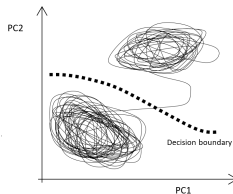


Identify Operating States

1. Drifts around same neighbourhood
2. Drifts across boundary



(a) Process with a single healthy state.



(b) Multi-state process with transition between states.

Figure: Travelling context vector.

Summary

- ▶ Unsupervised clustering method for multidimensional time series
- ▶ Neural network-based time series feature selection
- ▶ Operating states identification

Summary

Application area

- ▶ Works with any real-valued sensor measurement (e.g. temperature, pressure... etc.)
- ▶ Large scale multi-sensor system (Multidimensional time series)
- ▶ Unbounded and unlabelled time series

Further work

- ▶ Categorical sensor measurements (e.g. start vs stop, full vs empty... etc.)
- ▶ Discrete events (e.g. component replacement, unplanned outage,)

Bibliography I

-  Sepp Hochreiter and Jürgen Schmidhuber, *Long short-term memory*, Neural Computation **9** (1997), no. 8, 1735–1780.
-  Christopher Olah, *Understanding lstm networks*, 2015.
-  Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, *Dropout: A simple way to prevent neural networks from overfitting*, Journal of Machine Learning Research **15** (2014), 1929–1958.
-  Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals, *Recurrent neural network regularization*, CoRR **abs/1409.2329** (2014).