Failure detection and isolation by LSTM autoencoder

Tauno Toikka, Jouko Laitinen, Kari T. Koskinen

Abstract Failure diagnosis on some system is often preferred even the data of the system is not designed for the condition monitoring and does not contain any or contains little example cases of failures. For this kind of system, it is unrealistic to directly observe condition from single feature or neither to build a machine learning system that has been trained to detect known failures. Still if any data describing the system exists, it is possible to provide some level of diagnosis on the system. Here we present an LSTM (Long Short Term Memory) autoencoder approach for detecting and isolating system failures with insufficient data conditions. Here we also illustrate how the failure isolation capability is effected by the choice of input feature space. The approach is tested with the flight data of F-18 aircraft and the applicability is validated against several leading edge flap (LEF) control surface seizure failures. The method shows a potential for not only detecting a potential failure in advance but also to isolate the failure by allocating the anomaly on the data to the features that are related to the operation of LEFs. The approach presented here provides diagnostic value from the data than is not designed for condition monitoring neither contain any example case failures.

1 Introduction

Failure diagnosis is an active area of research and increasing target of application in industry due to increasing interest of condition based maintenance over a scheduled maintenance. Also the evolution of computing power, censoring technology and machine learning algorithms have been boosting the development of failure diagnostics approach in reliability engineering in recent years.

Failure diagnosis can be divided by several subcategories such like: failure detection, failure isolation, failure identification and failure classification [1]. In practice it is desirable to have diagnostics system that is capable of achieving all levels of diagnosis but the goal is cumbersome to achieve when diagnosing a complex system. With an insufficient data conditions, it is typical that the achieved level of diagnosis is only an isolation. One approach to elude the problem is to use autoencoder type neural network. In the following literature autoencoder, or variants of it, are used to perform diagnostics in various conditions.

In [2] a deep learning method for fault classification and degradation assessment was presented. In the study a vibration data of rotating machinery was used and the

Tauno Toikka, Jouko Laitinen, Kari T. Koskinen
Tampere University, FI-33014 Tampere University
e-mail: tauno.toikka@tuni.fi

results were validated by injected failures. The method was compared against a conventional methods and was proven to be superior. In [3] a reconstruction-based auto-associative neural network for fault diagnosis in nonlinear systems was introduced. In the method faults were isolated based on the network reconstruction. An applicability of the method was illustrated on a gas turbine process. The author claimed the method to be robust and not requiring a prior knowledge. In [4] MPL and RBF was used for detecting and isolating faults of the Tennessee Eastman benchmark process. As a novelty they transferred a time domain data to 2D image data. In [5] LSTM network was applied for fault detection and isolation task on electro-magnetical actuators of aircraft. In [6] used vibrational autoencoder (VAE) for failure detection in case of TFT-LCD manufacturing process. In [7] a stacked convolutional sparse denoising auto-encoder (SCSDAE) was used for defect defection in wafer maps in semiconductor manufacturing process. In [8] a deep transfer learning autoencoder was used for predicting remaining useful life of drilling tool. In the method a failure data was used. In [9] a stacked sparse autoencoder was used for steel grinding burn detection in supervised manner. In [10] a stacked long short-term memory autoencoder for anomaly detection in rotary machine was proposed. In [11] sparse autoencoder with PCA and SVM was proposed for power system fault diagnosis. In [12] a stacked denoising autoencoder was proposed for health state identification. In the study the diagnosis method was applied on rolling bearings.

In order to achieve all levels of failure diagnosis by data driven model, the life time data of a set of systems is needed. Life time data of a sets of systems is not available until all systems of some fleet have reached the end of their life and will be discarded. Many times a great diagnostics results have been achieved with the high quality life time data monitored in laboratory by using carefully selected sensors. On the other hand, there is a need for diagnosing systems that are in their early life state and do not yet have any failures in their history. Thus there is a need for tools that can provide failure diagnosis based on the data that is truncated, not life time data, not data from set of similar systems and does not contain necessarily any example cases of failures. This data condition here we call simply as insufficient data conditions. There exists a little study considering the failure diagnosis in insufficient data conditions.

Deep autoencoder is a special type neural network that maps its inputs to its outputs. Here we will demonstrate that carefully constructed LSTM autoencoder neural network can not only provide failure detection but also some low level isolation with insufficient data conditions. This is valuable since failure isolation is important for maintenance decision support since it provides some hint about the location from where the potential failure is developing. The following sections are organized as follows. In section 2 the construction of autoencoder, LSTM neuron type, a nature of the data, training procedure and identification metrics that is a reconstruction error are described. In section 3 are the results of the study followed by discussion on section 4.

2 Methods

2.1 Autoencoder

In this study we applied autoencoder for detecting and isolating failures. Auto encoder is a special type of artificial neural network that has a capability of capturing the core structure of data without copying the actual data, so called representation learning. It can be also seen as a data compression tool. Autoencoder has a two parts, encoder and decoder. Encoder compresses the data by its internal structure of shrinking number of free parameters of network layers. Decoder decodes the encoded data by its structure of expanding number of free parameters of layers. Final layer of decoder part has an output that has dimensionality same as input dimensionality. When training the autoencoder then the target values are the input values.

2.2 LSTM network

Long Short Term Memory (LSTM) neural network is constructed by using LSTM neurons. LSTM neuron has a capability of remember of its previous output state. The output state is stored until some other neuron gives the activation signal that frees the memory, so called forgot gate. Due to the memory of a LSTM network, the network is useful for modelling the data that has a temporal characteristic, that is the case with the data of this study.

2.3 Network setup

In this study the autoencoder neural network have been build up by using Keras [13] libraries. One example network structure used here is described in figure 1.

```
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 20, 13)            1404
_________________________________________________________________
lstm_2 (LSTM)                (None, 4)                 288
_________________________________________________________________
repeat_vector_1 (RepeatVecto (None, 20, 4)             0
_________________________________________________________________
lstm_3 (LSTM)                (None, 20, 13)            936
_________________________________________________________________
dropout_1 (Dropout)          (None, 20, 13)            0
_________________________________________________________________
lstm_4 (LSTM)                (None, 20, 13)            1404
_________________________________________________________________
time_distributed_1 (TimeDist (None, 20, 13)            182
=================================================================
Total params: 4,214
Trainable params: 4,214
Non-trainable params: 0
_________________________________________________________________
```

Figure 1: Autoencoder structure.

Several other network configurations were also tested. The aspects configured while constructing a variety of networks were:

• Feature space size
• Network depth
• Shrinking layer size between encoder and decoder
• Dropout layers and dropout rate

3 Results

3.1 Data

The dataset used in this study is composed of 43 consequent flight data from single aircraft. Three flights were containing LEF seizure failures. Sample monitoring frequency was 1/10s and total $1.5 * 106$ samples were available.

Several feature space configurations were used and they are described in the table 1.

Table 1: Features (feature space 1 (FS1) feature space 2 (FS2) feature space 3 (FS3))used for testing different autoencoder setups. Marker * shows included features

Flight parameter	FS1	FS2	FS3
Left power leaver angle	*		
Right power leaver angle	*		
Left engine inlet temperature	*		
Right engine inlet temperature	*		
Left compression pressure	*		
Right compression pressure	*		
Left drain air temperature	*		
Right drain air temperature	*		
Left low rotor speed	*		
Right low rotor speed	*		
Left high rotor speed	*		
Right high rotor speed	*		
Dynamic pressure (I)	*	*	
Dynamic pressure (II)	*	*	
Static pressure	*	*	
Ambient temperature	*	*	
Barometer corrected pressure	*	*	
Pressure altitude	*	*	
Air speed	*	*	
Left leading edge flap position command	*	*	
Rigt leading edge flap position command	*	*	
Left inner leading edge flap position	*	*	error position
Right inner leading edge flap position	*	*	error position
Left outer leading edge flap position	*	*	error position
Right outer leading edge flap position	*	*	error position

The feature spaces of the table 1 were constructed by the following intuition:

• FS3 contained only four error positions of the four LEF's of the aircraft thus describing only the behaviour of the LEF's and interrelationship between each other. The motivation of FS3 was to see if the actual source of failure could be isolated among the four LEF's.

• FS2 contained all directly LEF related data plus some features that we assumed to be related indirectly to the LEF behaviour. The aim of the construction of this dataset was to have data that as it maximum amount describes the behaviour of the LEF's, without aiming any obvious accessories.

• FS3 contained the data of SF2 plus some extra data that was assumed to be non-phenomenon related. The aim of this data was to construct a dataset that was hard do classifier since containing non phenomena related information, and this way to test how diagnostic results might be effected if proper feature extraction cannot be done.

The time window for LSTM was selected to be 20 samples (due to computing capacity and especially memory reasons), and thus having the window length of 2 seconds. The window length of 20 samples and feature space FP2 from table 1 having 13 features, can be seen in figure 1 as both as an input and output.

3.2 Training

The data for training was further treated in order match to input and output shape of the LSTM autoencoder. By applying a sliding window principle, the temporal length of the data did increase by the multiplication of the window size. For example, with feature space of 13 features and window size of 20 the original data was extended from $1.5 * 10^6$ X 13 to $1.5 * 10^6$ X 20 X 13

As a training data it was selected first 35 consequent flights, thus leaving 4 consequent healthy flight before failure for validating the system, since first failure did occur during flight 40. The training data was further separated to training and test data with radios of 2/3 and 1/3. The test data was used for preventing the overfitting during training. As a validation data it was used 8 last flights from which 4 first were healthy flights and 4 last contained a LEF seizure failure. The data train, test and valid split and some other additional training information are listed in table 2.

Table 2: Training parameters

Lookback	20
Epochs	5
Batch	10000
Learning rate	0.0001
Train test split	0.33
Train samples	883082
Test samples	434951
Valid samples	197226

3.3 Reconstruction error

Turing the training phase, the data that was presenting the normal behaviour of the system, was used for training autoencoder. Turing the training phase of autoencoder, the internal parameters of autoencoder network will be learned so that the network will present the structure of the data that has been used for training. By monitoring also, the loss rate during the training of autoencoder, it was ensured the trained autoencoder is presenting a normal system. During the diagnosis phase, a reconstruction error of the trained autoencoder was calculated. Reconstruction error of autoencoder is a feature vice error between the input and the output of the autoencoder. Error space size is same as the feature space size, and thus there is an error related every feature. A large reconstruction error signals that the data diagnosed does not correspond the data used earlier for training. A large reconstruction error is hinting about an abnormal behaviour and thus providing a potential failure detection. Since there is a reconstruction error related with each feature, an isolation is also provided.

3.4 Failure cases

The target failure for autoencoder to detect and isolate is Leading Edge Flap (LEF) seizure failure demonstrated in the figure 2. In the figure 2 it can be seen that in the middle of the flight actual LEF position does not correspond to the control position and that state is defined here as a failure.

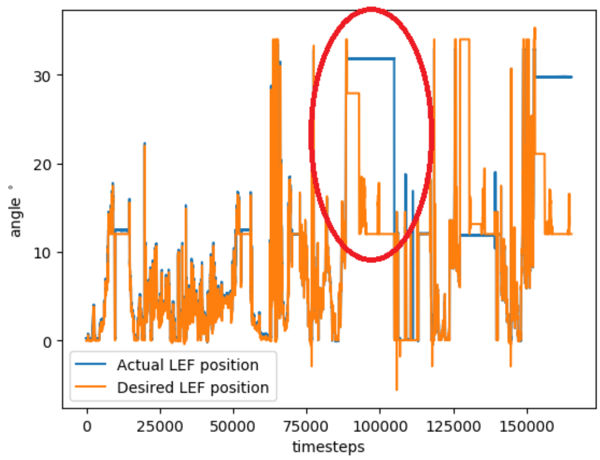

Figure 2: Leading Edge Flap (LEF) seizure failure during the flight no. 40.

The LEF seizure failure occurred during the three flights (fights 40, 41, 43) which all started as a healthy flights (see fig. 3). Before the failure flights there were 39 healthy flights. Healthy flights and the failure flights are presented in figure 3 in terms of the error position of all four LEF's of the aeroplane. The data of the figure 3 is the same data as the data FP3 in table 1 except the sliding window.

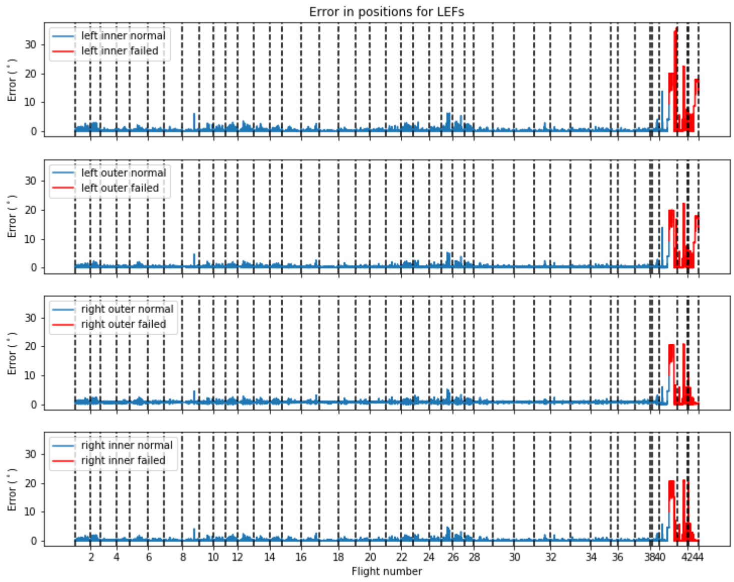

Figure 3: Error positions for all four LEF's of F-18 aircraft during the 43 consequent flights (separated by dashed vertical lines).

3.5 Diagnosis results

From the data of FS3 that was the data with only LEF error positions we were not able to construct a system that would reveal the source of the failure among the four LEF's, neither isolate the individual failure source. When considering a reason for this and explanation may be that in practice the LEF's of the same wing side are physically jointed. On the other hand, the interrelation between left and right side LEF's are computationally corrected by the system automation.

With the data of FS1 a failure detection was achieved since the trained autoencoder did produce large reconstruction during a healthy part of the failure flight and a one flight before. On the other hand, the reconstruction error was large on all channels and thus no meaningful isolation was achieved. The most applicable result was achieved with the data FP2 of the table 1 and with the autoencoder of figure 1. The result is presented in figure 4.

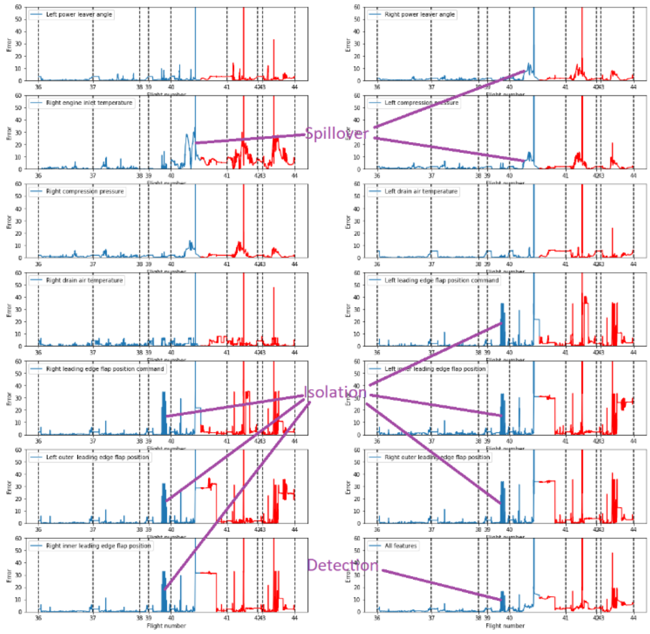

Figure 4: Reconstruction errors for 8 last consequent flights. Blue graph indicates healthy system and red indicates system after first failure.

From the figure it can be seen that the reconstruction error is small during the flights 36, 37 and 38 on all channels. This is correct since those flights have been normal flights also in practice. Also there can be seen a large reconstruction errors related to the LEF seizure (red curvature), but this is not interesting since the failure is obvious and known at that point. What is notable is that during the flight 39 that is the flight before the LEF seizure failure flight, there is large reconstruction error on all channels that are related to LEF operation. Also on the healthy part of the failure flights there can be seen large reconstruction errors.

A large reconstruction error means that the diagnosed data does not correspond the autoencoder model and thus does not correspond the data of previous flights. These large reconstruction errors can be interpreted many ways: potential failure, abnormal flying style, abnormal environmental conditions, use of some rare functionalities of system and so on. Thus system expert is further required to analyse the result. If the system expert does not conclude an ab-normal flying style or environmental conditions one may conclude the potential failure and start further investigations.

Since the reconstruction error is large on specific channels but not on all channels, the isolation is provided among the channels. The isolation is obscured by the spillover effect. The spillover can be seen for example on channel ”Right engine inlet temperature”. The temperature cannot fail, thou the sensor can. Still here the observed large reconstruction error does not present temperature sensor failure, but rather a spillover effect. Due to the internal construction of autoencoder, the autoencoder is forced to compress data. This compression leads to the creation of internal relations that do not necessarily exist in real world. From the result 4 it may be concluded that the ”Right engine inlet temperature” is reconstruction from the LEF operation parameters. In practical diagnosis this incorrect behaviour need to be judged by the system expert.

4 Discussion

In the field of condition based maintenance decisions are based on a system diagnosis. A diagnosis can be done in many levels and a data analytic level is a one of them. On the other hand, many times data available is not optimal for system failure diagnosis, but rather designed for other purposes. In this type of so called insufficient data conditions the failure diagnosis requires more careful algorithmic choices.

In this study it was demonstrated that LSTM autoencoder is capable for failure diagnosis even with insufficient data conditions. The level of diagnosis achieved here was failure detection and isolation. Here it was also demonstrated how LSTM autoencoder failure isolation capability is effected by the choice of input feature space. The level of failure isolation is notable achievement when considering the limitations of the data used here. In general, the model build on data cannot present more than the nature of the original data. Since the basis here was that a data does not contain failures, the model cannot present failures. On the other hand, the model can present a normal system and thus anomalous behaviour can be detected against the model of normal system. The significance of the method presented here is that the anomalies will be allocated on features and thus providing isolation.

It is generally known that a knowledge based data pre-processing in an important preliminary step when applying machine learning, which did also apply here. The method proposed here brings up another challenge when interpreting the results, since the system knowledge turns out to be vital also on this site. The method provides only allocated anomalies, so called isolation, but it is the task of system expert to further interpreted if the anomaly actually presents a potential failure or something else.

Benefits and practical application of the methodology are that many real world system providing data have a characteristic of data similar of our data. In practice any system without failure history or with data designed directly for condition monitoring has data conditions similar to this study.

Limitations of this study are that it was done on one system. Reason here was that we had one system with data sufficient to validate our methods. In future more studies with similar methods for different domains would be needed.

References

[1] A. Barua, P. Sinha, and K. Khorasani, “A diagnostic tree approach for fault cause identification in the attitude control subsystem of satellites,” IEEE

Transactions on Aerospace and Electronic Systems, vol. 45, no. 3, pp. 983–1002, 2009.
[2] Z. Li, Y. Wang, and K. Wang, "A deep learning driven method for fault classification and degradation assessment in mechanical equipment," Computers in Industry, vol. 104, pp. 1–10, 2019.
[3] S. Ren, F. Si, J. Zhou, Z. Qiao, and Y. Cheng, "A new reconstruction-based auto-associative neural network for fault diagnosis in nonlinear systems," Chemometrics and Intelligent Laboratory Systems, vol. 172, pp. 118–128, 2018.
[4] P. Hajihosseini, M. M. Anzehaee, and B. Behnam, "Fault detection and isolation in the challenging tennessee eastman process by using image processing techniques," ISA transactions, vol. 79, pp. 137–146, 2018. 12
[5] J. Yang, Y. Guo, and W. Zhao, "Long short-term memory neural network based fault detection and isolation for electro-mechanical actuators," Neurocomputing, 2019.
[6] S. Lee, M. Kwak, K.-L. Tsui, and S. B. Kim, "Process monitoring using variational autoencoder for high-dimensional nonlinear processes," Engineering Applications of Artificial Intelligence, vol. 83, pp. 13–27, 2019.
[7] J. Yu, X. Zheng, and J. Liu, "Stacked convolutional sparse denoising autoencoder for identification of defect patterns in semiconductor wafer map," Computers in Industry, vol. 109, pp. 121–133, 2019.
[8] C. Sun, M. Ma, Z. Zhao, S. Tian, R. Yan, and X. Chen, "Deep transfer learning based on sparse autoencoder for remaining useful life prediction of tool in manufacturing," IEEE Transactions on Industrial Informatics, vol. 15, no. 4, pp. 2416–2425, 2018.
[9] W. Guo, B. Li, S. Shen, and Q. Zhou, "An intelligent grinding burn etection system based on two-stage feature selection and stacked sparse autoencoder," The International Journal of Advanced Manufacturing Technology, pp. 1–11, 2019.
[10] Z. Li, J. Li, Y. Wang, and K. Wang, "A deep learning approah for nomaly detection based on sae and lstm in mechanical equipment," The Interna- tional Journal of Advanced Manufacturing Technology, pp. 1–12, 2019.
[11] Y. Wang, M. Liu, Z. Bao, and S. Zhang, "Stacked sparse autoencoder with pca and svm for data-based line trip fault diagnosis in power systems," Neural Computing and Applications, pp. 1–13, 2018.
[12] C. Lu, Z.-Y. Wang, W.-L. Qin, and J. Ma, "Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification," Signal Processing, vol. 130, pp. 377–388, 2017.
[13] "Keras: The python deep learning library." [Online]. Available: https://keras.io/