Implementation of a Variational Autoencoder for Dimension Reduction of a Hydraulic System

Faras Brumand-Poor*; Faried Makansi; Jiakun Liao; Katharina Schmitz

RWTH Aachen University, Institute for Fluid Power Drives and Systems (ifas), Campus-Boulevard 30, D-52074 Aachen, Germany,*
** Corresponding author: Tel: +49 241 80 47743 E-Mail:*
faras.brumand@ifas.rwth-aachen.de

Abstract

Hydraulic systems are often equipped with a great number of actors and sensors to monitor and enable the control of a system. For continuous monitoring, a vast amount of data needs to be recorded and frequently stored. Prediction based on this big database is often not directly comprehensible by a human and requires preprocessing. With the development of computational power, machine learning becomes a promising approach to handle these huge amounts of data and offers to be a possible solution for the automated processing of raw machine data. In this paper, a deep unsupervised learning (UL) model, the variational autoencoder (VAE), is implemented. The VAE is used for the reduction of high dimensional time series of a simulation of a reference system, a hydraulic press. To handle the time dependency of the dataset, the VAE contains long short-term memory (LSTM) layers. The high dimensional input data matrix is compressed by the first part of the VAE, the encoder, into the latent. This low dimensional representation of the time series is then used by the counterpart of the encoder, the decoder, to reconstruct the original time series. The VAE achieves a compression rate of 99,94 % with a small deviation compared to the original input. This contribution shows the possibilities of compressing big data from a hydraulic system into a low dimensional representation. The compressed data can be used as input for

River Journal, 1–17.

condition monitoring (CM) algorithms, with the benefit that it doesn't contain any redundant information.

Keywords: Data-based condition monitoring, deep unsupervised learning, hydraulic system, VAE.

1 Introduction

In production, safety and efficiency are two essential factors to be ensured. For a well-organized production strategy, the condition of the plant needs to be well monitored. In real life, the plant needs to be examined by an experienced engineer regularly, in case of defective components. These have to be replaced, to avoid potential risks for production. In recent years, data-based condition monitoring has been applied to fluid power systems [1] - [6]. Data-based condition monitoring relies on a constant generation of data by the plant's sensors and actors. Predicting the machine state directly from the unprocessed measurements often exhibits limited performance. This is due to two main reasons: Firstly, the process is fairly complex and needs to be simplified for an algorithm to be able to conclude. Secondly, the dataset is such vast in size, that the algorithm is not able to detect the particular data points for the required prediction. For a specific task in CM, often certain data points of particular sensors/actors are decisive. Therefore a preprocessing of the big data needs to be done, to discover the most crucial information, such that the CM algorithm can achieve the best performance.
One approach to preprocessing is to reduce the size of the original data before further analysis. Hereby, the reduction needs to be done with a minimal information loss and a maximal compression rate. There are several techniques for data reduction, which can be distinguished by the amount of data required to obtain good performance. A good performance means that the algorithm is able to compress the original data in a meaningful representation for further application. Statistical methods achieve reasonable performance with a smaller amount of data [7]. However, the amount depends heavily on the dataset (e.g. complexity, number of dimensions, etc.). Machine Learning algorithms, especially Deep Learning (DL) ones, normally require a greater amount of data compared to classical approaches. For the case, that a plant consistently generates data or a simulation of the reference system is available, machine learning is a good solution for the data reduction task. In order to validate the compression of a DL method, the algorithm is often validated by a counterpart, which aims to reconstruct the

original data with minor error.

The scope of this paper is the dimension reduction of high dimensional time series data of a simulation of a hydraulic press and the investigation of the reconstructed data. A deep unsupervised learning model, the variational autoencoder, is implemented and compresses the input matrix into a reduced vector containing the crucial information of the press, which can be used for further applications like condition monitoring, predictive monitoring or system control.
In section 2, the state of the art of data-based condition monitoring in fluid power systems as well as the VAE are presented. In section 3, the implementation of the model is described. Subsequently, the results of the model are shown. In the last section, the conclusion and the outlook are given.

2 State of the Art

2.1 Data-Based Condition Monitoring in Fluid Power Systems

During the production process, the machine state has to be monitored to detect system anomalies. CM can be applied to fluid power systems to automate potential failure detection. One approach for CM is developing a mathematical model of the plant. The fault detection algorithm predicts the machine state by comparing the outputs of the plant and the model [8] - [10]. These model-based approaches require knowledge of the machine and complex modeling, especially if nonlinearities need to be considered.

Data-based CM is another method for CM, which bases its prediction on the data provided by a machine or a simulation of it [1], [7]. Investigations have been carried out on firstly compressing the time series data of the plant using statistical approaches before the application of fault classification machine learning models [11] - [14].

Research has been conducted on unsupervised learning algorithms for the compression of data in hydraulic systems. UL models aim to reduce the size of a given dataset by omitting irrelevant particulars while ensuring a minimal loss of information. Statistical analysis like principal component analysis have been applied to hydraulic systems to reduce a dataset provided by the plant [15] - [17]. In other studies, neural networks are used for the reduction and prove to be efficient for the fault detection of a hydraulic simulation

model [18]. In most cases a classification algorithm is implemented for the actual fault detection, however, Krogerus et. al. directly used an UL method for the working condition of a worn spool of a valve [19].
In recent years, more advanced UL models, the autoencoder (AE) and the stacked autoencoder, have been applied for dimension reduction of data of hydraulic systems [20], [21]. The autoencoder is an UL model especially suitable for dimension reduction [22]. A minimum loss of information is ensured by the AE aiming to reconstruct the original data from the low-dimensional representation. In most recent studies, Mallak et. al. have implemented an AE with long short-term memory, which store information over extended time intervals, to handle time dependencies in their machine data. The data was reconstructed with a small loss of information [21].

2.2 The Variational Autoencoder

The variational autoencoder is an AE that aims to learn a mathematical meaningful encoding [23]. The AE compresses a dataset in an incomprehensible latent representation. The main goal is to achieve high compression. Compared to the AE, the VAE aims to create a meaningful latent space. This is beneficial for CM since the actual machine state prediction is based on the latent space, which is mathematically comprehensible in the case of the VAE. The VAE consists of two parts, the encoder, and the decoder. The first compresses the original input into a low-dimensional representation, which is called the latent. The decoder aims to reconstruct the original input through the latent. The compression quality is normally investigated based on the similarity of original and reconstructed data.

3 Material and Methods

3.1 The Hydraulic Press

The dataset is provided by [1], and the reader is advised to refer to this work for more detailed information. The dataset is obtained by a simulation of a hydraulic reference system, which is displayed in Fig. 1. The simulation is built according to a physical demonstrator machine. The main components are a hydraulic power supply, a proportional directional control valve and a single acting hydraulic cylinder. Additionally, a ram is attached to the cylinder rod and acts against a load force. The load force represents the reaction force of a workpiece. The hydraulic press moves the ram to a defined position while working against a load force. Fig. 2 shows the progressions of the cylinder's

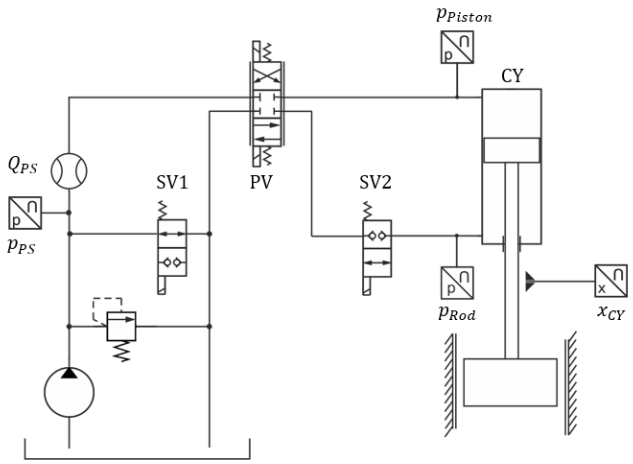

Figure 1: Circuit diagram of the considered hydraulic system

position, piston side pressure and the volume flow of the pressure supply. The cycle starts in idle mode. After 1 second, the ram moves towards the workpiece without external load. During this phase, there is no restriction on the velocity. As soon as the ram reaches a defined distance to the workpiece, the velocity is limited through the control signal and the volume flow to the cylinder, respectively. When the ram is in contact with the workpiece, the piston side pressure increases proportionally to the cylinder stroke. As a fixed position is reached, the backstroke is initiated. After the retraction is finished, the press is in idle mode for the rest of the cycle. Regarding the dataset: There is no empty entry, and it is normalized for better performance in the training process. The dataset contains different working conditions (e.g. internal, external cylinder leakage, no leakage). All features are continuous, except y_{SV1} and y_{SV2}, which are discrete between 0 and 1. The duration of each cycle is 18 seconds, with a sampling frequency of 100 Hz, thus one cycle has the length of 1801. The 9 features are denoted in Table 1 with their abbreviation and physical meaning.

3.2 The Implementation of the Variational Autoencoder

First, the structure of the encoder, which is shown in Fig. 3, is elucidated. The batch size (bs) is set to one for a better demonstration and is the number of

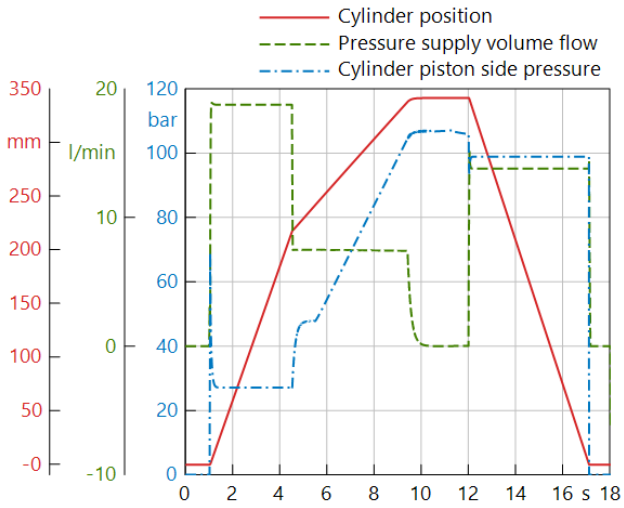

Figure 2: Working Cycle of the hydraulic press

Table 1: The features and their physical meaning

Feature	Physical meaning
x_{CY}	Cylinder position
p_{PS}	Power Supply
p_{Piston}	Piston pressure
p_{Rod}	Rod side pressure
u_{PV}	Control signal propotional valve
y_{PV}	Position propotional valve
Q_{PS}	Volumetric flow rate
$y_{SV1,2}$	Position switching valves SV1,2

bundles of 9 time series utilized in one training iteration, which is denoted as one epoch. As it has been stated before, the input is time series data. The latent size is set to the number of features, 9, thus the time series are completely compressed to one value, respectively. The first two layers consist

of LSTMs in order to consider time dependencies between the data points. The LSTM considers the prior input by storing the information in its hidden state, which is used for computing the output for the subsequent input. In

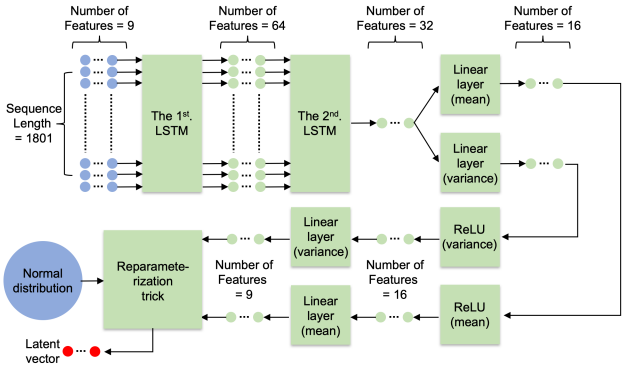

Figure 3: The detailed structure of the encoder

machine learning, a rule of thumb for the layer size is in the power of two. In the first LSTM layer, the input is processed sequentially. For each time step, the measurement features together with the hidden state from the last time step are used to generate the hidden state of the current time step. The size of the generated hidden state is set to 64. So the output of the first LSTM is a sequence of the hidden states in the length of 1801, and the number of features is 64. It is then fed to the second LSTM neural network. In this LSTM, instead of using the whole list of hidden states as output, only the hidden state in the last time step is provided. The size of the generated hidden state is then compressed to 32.

In order to compute a mean (μ) and a variance (Σ), which are eventually used to obtain the latent z, after the LSTM layers, two parallel linear layers are implemented. Each layer has a rectified linear unit (ReLU) as the activation function. Mean (μ) and variance (Σ) vectors are generated after another sequence of linear layers. The data dimension is eventually reduced to 9.
The latent z is randomly sampled by the distribution obtained by μ and Σ, therefore the gradient can not be calculated. This problem is solved by using the reparameterization trick. A random node, which samples a random vector ϵ from a standard normal distribution, is added. The latent space is then calculated by: $z = \mu + \Sigma \odot \epsilon$, with $\odot$ being the Hadamard operator which means a component-wise multiplication. With this function, the gradient can

be calculated by the derivatives of $\frac{\partial z}{\partial \mu}$ and $\frac{\partial z}{\partial \Sigma}$ and used for the optimization of the VAE parameter. The reparametrization trick allows the relocation of the randomness of z to ϵ. However, this is only necessary during the training process. For the validation of the model, the latent vector z is directly assigned with μ. The latent vector generated from the encoder is then fed to the decoder. The structure of the decoder is shown in Fig. 4. The first three layers are reversely arranged compared to the encoder. The latent vector, provided by the encoder is fed to the linear layers and expanded to a size of 32. The repeater loops the input vector 1801 times. In this way, it generates a series of data with a constant trajectory size, which is then fed to the LSTM layers, which eventually computes the reconstructed time series.

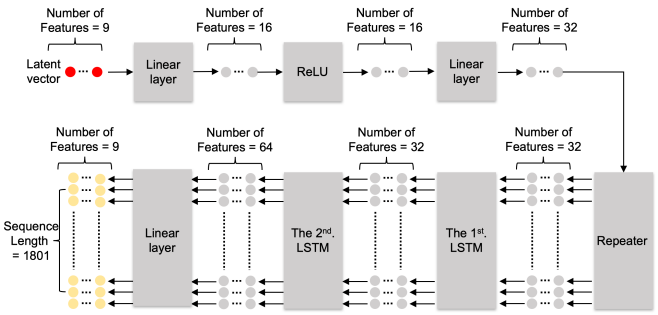

Figure 4: The detailed structure of the decoder

4 Results

This section presents the training and testing results. Firstly, the losses are presented. To obtain a better understanding, the loss is modified. The system parameters are normalized between 0 and 1, except for u_{PV}, y_{PV} and Q_{PS} which are normalized between -1 and 1. Therefore, the loss can be expressed as a percentile number by being computed relatively to these intervals. Secondly, the original and reconstructed time series of Q_{PS} is plotted. Q_{PS} is chosen, since the cylinder velocity is proportional to the volume flow and therefore provides an insight on the dynamic behaviour of the press. Besides, an error plot is given. Since the original dataset contains several entries with a value of 0, the error plots are created by the relative percentage difference (RPD) [24]. The RPD is used to compare two quantities if the original value

is zero. Normally, the error is calculated by dividing the absolute difference of both values with the original value. In the RPD the denominator is the absolute maximum of both values. Lastly, the original and reconstructed time series are quantified by several characteristics, which are given in 2. A detailed description of the features investigated can be found in [25]. Transferring raw data to a new representation is helpful for the interpretability and comprehensibility of a trained ML model, since now the performance of the model can be quantified by more than just the loss. Furthermore, the whole time series are divided into four sections, where each represents a different state of the hydraulic press which were explained in fig. 2. For each section, the characteristics are calculated, weighted, and averaged.

4.1 Results of the variational autoencodder

The VAE is trained with several different sets of hyperparameters in a heuristic approach. The best-performing settings are introduced in the following section.
The best results are obtained with a VAE, which has a batch size of 25 and a learning rate l_r of 10^{-4}. The learning rate is a dimensionless value, since the data is normalized, and is used for the actual tuning of the VAE's parameters during the training process. To detect overfitting, which is that the model learns the training data too well to the extent that it negatively impacts the performance on new data, the VAE is evaluated on a testing dataset. During the testing, the VAE is not optimizing its parameters.
Fig. 5 shows the progress of the mean error for the training and testing data. For a better understanding of the training process, the loss is modified, as described at the beginning of this section, such that a percentile number is obtained. The loss starts at around 216 % and decreases 95.5 % to eventually settle at around 9.5 % after 640 epochs. The testing loss follows the training loss closely, which means that no overfitting occurs during the optimization.
Fig. 6 display the original, reconstructed, and the relative percentage error of the volume flow. The reconstructed time series displays a similar trajectory compared to the original data. As it can be seen, the error is the lowest in the extracting and retracting part of the cycle and increases in the first and last sections. The mean RPD for the whole time serie is 3.8 %. The mean RPD for the other features is displayed in Table. 2 and shows small errors for the remaining ones. Table. 2 shows the characteristics of the 9 features. For each feature, the characteristic is computed for the four sections of the time series, and mean and standard deviation of the percentile error are cal-

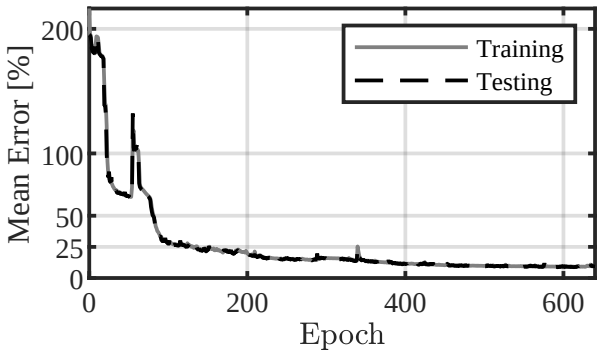

Figure 5: The detailed structure of the vanilla decoder

culated. The empty entries mean, that the error can not be determined, since the characteristic of the original time series is zero, thus the percentile deviation is not defined. Overall, the characteristics show a low error. There are certain outliers; The minimum value of the cylinder position displays a rather high deviation, which is mostly dominated by the second section's deviation. Nevertheless, the other characteristics of x_{CY} display a much smaller error, and besides the mean RPD for the whole cycle is 7.7 %. Furthermore, more extreme values can be observed in the locations of minimum and maximum values of a section. u_{PV}, y_{PV}, and Q_{PS} have high deviation, whereas the other characteristics have low errors with y_{SV1} and y_{SV2} reaching nearly 0. Regarding Q_{PS}, the maximum value also shows a higher deviation, but the other features display good performance.

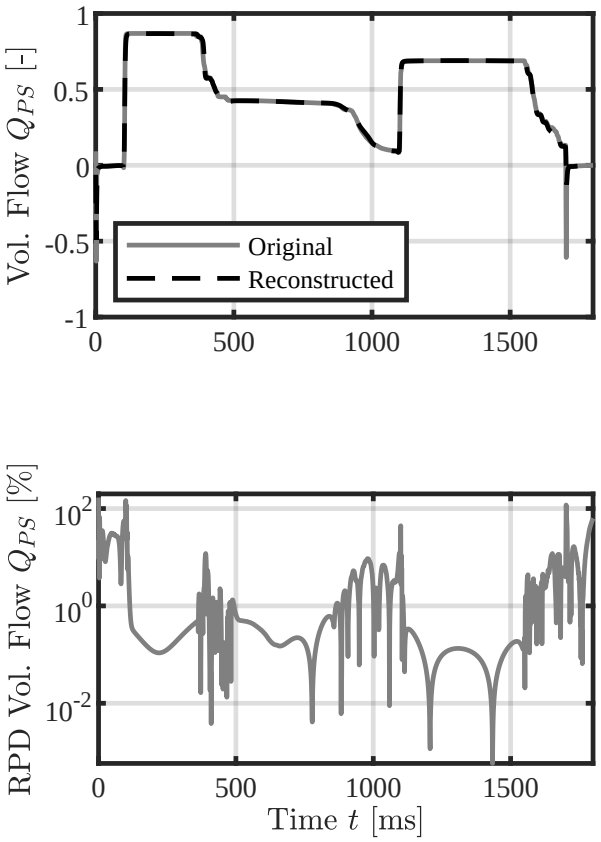

Figure 6: The original (solid grey), the reconstructed (dashed black) volume flow and the RPD plot

The error of most characteristics of y_{SV1}, y_{SV2}, and u_{PV} can not be computed. However, the RPD for all three is smaller than 4 %. Especially, for the control signal of the proportional valve, the RPD is as low as 1.6 % and 1.3 % for the extracting and retracting part, respectively.

5 Conclusion and Outlook

5.1 Conclusion

In this paper, a deep unsupervised learning model with LSTM layers was presented, which reduces the machine data of a hydraulic press into a low dimensional representation and afterwards reconstructs the original input. The quality of the reconstruction is examined by three approaches: Firstly, by the training process. Secondly by plots of the original, reconstructed volume flow and the relative percentage difference error and lastly by several characteristics describing the time series. Regarding the characteristics, the time series are divided into four sections based on the four modes of a working cycle of the hydraulic press. The characteristic for each section is computed and a mean and a standard deviation for the whole time series for each of the 9 features is calculated.

The variational autoencoder has been implemented for the data compression task. In order to handle the time series, the VAE contain long short-term memory layers. The training achieves a 95.5 % decrease and the RPD plot of the volume flow shows good performance, especially in the extracting and retracting phase of the cycle. The characteristics also show rather small deviations, with some outliers. Features like Q_{PS} and x_{CY} have a small number of characteristics with higher deviation but show a good performance regarding the remaining characteristics. Furthermore, the mean RPD of both is also low. Especially for y_{SV1}, y_{SV2} and u_{PV}, where most characteristics can not be computed, the RPD shows good performance, especially in sections 2 and 3 and an overall RPD of less than 4 %. The input matrix has a size of 1801×9 and is reduced to a latent size of 9 with a small information loss. Therefore a compression rate of 99.94 % is successfully obtained.

5.2 Outlook

In future work, the VAE model will be combined with a classification algorithm in order to perform fault detection and diagnosis. The preprocessing of the original data by the VAE is crucial for a good classification. Through the compression into the highly reduced representation, the classification model is provided with the most important information of the machine. Without the data reduction, the classification can't completely focus on the actual fault detection task, since it also needs to filter out redundant data points itself. The combination with a classification model requires an intensive investigation of the latent space. Especially, the trade-off between dimension reduction and

the actual fault detection needs to be examined. With greater latent size, the amount of unique and redundant information for the classification algorithm increases. An optimum size depending on the prediction tasks needs to be found. Furthermore, the VAE creates a mathematical meaningful latent, thus the investigation of the compressed representation in combination with the classification provides a more comprehensive workflow compared to an AE.

Acknowledgement

The authors thank the Research Association for Fluid Power of the German Engineering Federation VDMA for its financial support. Special gratitude is expressed to the participating companies and their representatives in the accompanying industrial committee for their advisory and technical support.

References

[1] Makansi, F.; Schmitz, K., A. Simulation-Based Data Sampling for Condition Monitoring of Fluid Power Drives. IOP Conf. Ser.: Mater. Sci. Eng. 1097 012018, 2021.

[2] Duensing, Y. (Corresponding author); Rodas Rivas, A.; Schmitz, K. Machine Learning for failure mode detection in mobile machinery Drives. In: 11. Kolloquium Mobilhydraulik : Karlsruhe, 10. September 2020.

[3] Zachrison, A. Fluid Power Applications Using Self-Organising Maps in Condition Monitoring., 2008.

[4] Wang, Z.; Lu, C.; Liu, H.; Chen, Z. Performance Assessment of Hydraulic Servo System Based on Radial Basis Function Neural Network and Mahalanobis Distance. Applied Mechanics and Materials. 764-765. 613-618. 10.4028/www.scientific.net/AMM.764-765.613, 2015.

[5] Chawathe, S. S. Condition Monitoring of Hydraulic Systems by Classifying Sensor Data Streams. 0898-0904. 10.1109/CCWC.2019.8666564, 2019.

[6] Siyuan, L.; Linlin, D.; Wanlu, J. Study on application of Principal Component Analysis to fault detection in hydraulic pump. 10.1109/FPM.2011.6045752, 2011.

[7] Helwig, N., Andreas Schütze, Detecting and compensating sensor faults in a hydraulic condition monitoring system. In Proc. Sensor 17th International Conference on Sensors and Measurement Technology, 2015.

[8] Münchhof, M. Model-Based Fault Detection for a Hydraulic Servo Axis. Autom., 2007.

[9] Isermann, R. Fault-Diagnosis Applications: Model-Based Condition Monitoring: Actuators, Drives, Machinery, Plants, Sensors, and Fault-tolerant Systems., 2011.

[10] Stammen, C. Condition Monitoring für intelligente hydraulische Linearantriebe., Dissertation, 2005.

[11] Jegadeeshwaran, R.; Sugumaran, V. Fault diagnosis of automobile hydraulic brake system using statistical features and support vector machines. Mechanical Systems and Signal Processing. 52. 10.1016/j.ymssp.2014.08.007., 2015.

[12] Helwig, N.; Andreas Schütze, A. Detecting and compensating sensor faults in a hydraulic condition monitoring system. in Proc. SENSOR 2015 - 17th International Conference on Sensors and Measurement Technology, oral presentation D8.1, Nuremberg, Germany., May, 2015.

[13] Schneider, T.; Helwig, N.; Andreas Schütze. Automatic feature extraction and selection for classification of cyclical time series data. pp. 198–206. doi: 10.1515/teme-2016-0072, 2017.

[14] Lurette, C.; Lecoeuche, S. Unsupervised and auto-adaptive neural architecture for on-line monitoring. Application to a hydraulic process. Engineering Applications of Artificial Intelligence. 16. 441-451. 10.1016/S0952-1976(03)00064-2., 2021.

[15] Sudarshan S. Chawathe, "Condition Monitoring of Hydraulic Systems by Classifying Sensor Data Streams",, doi: 978-1-7281-0554-3/19., 2019.

[16] Liu S., Ding linlin, Jiang Wanlu, "Study on Application of Principal Component Analysis to Fault Detection in Hydraulic Pump", doi: 978-1-4244-8452-2/11, 2011.

[17] Schneider, T.; Helwig, N.; Andreas Schütze. "Automatic feature extraction and selection for classification of cyclical time series data."tm - Technisches Messen 2017, 84(3), 198–206. doi: 10.1515/teme-2016-0072, 2017.

[18] Zhenya. W., Chen L., Jian. M., Hang. Y., Zihan. C., "Novel method for performance degradation assessment and prediction of hydraulic servo system", SCI IRAN, 22 1604-1615., 2015.

[19] Korgerus T., Vilenius J., Liimatainen J. and Koskinen K.T., "Self-Organizing Maps with Unsupervised Learning for Condition Monitoring of Fluid Power Systems", 2006.

[20] Huijie, Z.; Ting, R.; Xinqing, W.; You, Z.; Hu-sheng, F. "Fault diagnosis of hydraulic pump based on stacked autoencoders" In Proceedings of the 2015 12th IEEE International Conference on Electronic Measurement & Instruments (ICEMI), Qingdao, China, 16–18 July 2015.

[21] Mallak A., Fathi M., "Sensor and Component Fault Detection and Diagnosis for Hydraulic Machinery Integrating LSTM Autoencoder Detector and Diagnostic Classifiers", doi: https://doi.org/10.3390/s21020433, 2021.

[22] Baldi, P., "Autoencoders, Unsupervised Learning, and Deep Architectures." Proceedings of ICML Workshop on Unsupervised and Transfer Learning, in Proceedings of Machine Learning Research 27:37-49 Available from https://proceedings.mlr.press/v27/baldi12a.html., 2012.

[23] Diederik P. Kingma and Max Welling, "An Introduction to Variational Autoencoders", in Machine Learning., 2019.

[24] Törnqvist L., Vartia P. & Yrjö O., "How Should Relative Changes be Measured?", The American Statistician, 39:1, 43-46, DOI: 10.1080/00031305.1985.10479385., 1985.

[25] M. Christ, A. W. Kempa-Liehr, M. Feindt, "Distributed and parallel time series feature extraction for industrial big data applications", https://arxiv.org/pdf/1610.07717., 2016.

Table 2: Deviation (mean, standard deviation) of characteristics between original and reconstructed data for all 9 features.

Features	x_{CY}	p_{PS}	p_{Piston}	p_{Rod}	u_{PV}	y_{PV}	Q_{PS}	y_{SV1}	y_{SV2}
Max	(-11, 21)	(1.5, 1.6)	(-0.1, 3.7)	(-1.2, 2.6)	(-, -)	(-10.1, 211)	(117, 230)	(-, -)	(-, -)
Min	(486, 983)	(4, 6.4)	(10, 15)	(-0.8, 1.3)	(-, -)	(-, -)	(18.5, 18.7)	(-, -)	(-, -)
Mean	(0.8 , 1.8)	(0.5, 1)	(0.1, 1.3)	(0.4, 0.6)	(-, -)	(-0.06, 0.7)	(0.5, 1.5)	(-, -)	(-, -)
Median	(1.2 , 2.3)	(0.1, 0.4)	(-0.02, 1)	(1.4, 2.6)	(-, -)	(-, -)	(-0.5, 1.3)	(-, -)	(-, -)
Standard deviation	(-10.9, 13)	(-2.4, 8)	(0.1, 2.1)	(-0.4, 0.7)	(-, -)	(1.9, 3.8)	(1.1, 1.8)	(-, -)	(-, -)
Variance	(-68.3, 90)	(-7, 18.7)	(0.04, 4.2)	(-0.9, 1.5)	(-, -)	(3.4, 6.8)	(1.6, 1.7)	(-, -)	(-, -)
Skewness	(4.6, 6.1)	(31, 62)	(1.8, 2.1)	(-2.4, 6.1)	(-, -)	(1.4, 1.3)	(0.8, 1.7)	(-, -)	(-, -)
Kurtosis	(-0.2, 3)	(-95, 193)	(0.4, 0.6)	(-0.4, 3)	(-, -)	(1, 2.8)	(2.4, 2)	(-, -)	(-, -)
Abs. energy	(-6.3, 12.2)	(1.7, 2.1)	(0.04, 3)	(0.6, 1.1)	(-, -)	(0.1, 1.1)	(1.5, 2.3)	(-, -)	(-, -)
Mean abs. change	(1.2, 3)	(-18.8, 42)	(2.2, 2.5)	(8.5, 13.2)	(-, -)	(-5.3, 20.4)	(8.4, 6.8)	(-, -)	(-, -)
Abs. sum change	(1.2, 3)	(-18.8, 42)	(2.2, 2.5)	(8.5, 13.2)	(-, -)	(-5.3, 20.4)	(8.4, 6.8)	(-, -)	(-, -)
First minimum	(-26, 32)	(0.4, 0.9)	(5.7, 9)	(0, 0)	(-188, 251)	(-31.1, 38.9)	(-7.6, 13.5)	(-0.01, 0)	(0, 0)
First maximum	(-5.5, 18)	(-1.1, 2.9)	(-0.9, 1.3)	(-1.3, 2.8)	(-48e2, 96e2)	(803, 1.6e3)	(-172, 254)	(0, 0)	(-0.01, 0)
Last minimum	(-26.1, 32)	(0.4, 0.9)	(5.7, 9)	(0, 0)	(-188, 251)	(-31.1, 38.9)	(-7.6, 13.5)	(-0.01, 0)	(0, 0)
Last maximum	(-5.5, 18.6)	(-1.1, 2.9)	(-0.9, 1.3)	(-1.3, 2.8)	(-48e2, 96e2)	(803, 1.6e3)	(-172, 254)	(0, 0)	(-0.01, 0)
RPD	(7.7)	(1.9)	(4.5)	(2.5)	(3.1)	(12.3)	(3.8)	(0.05)	(0.05)

Faras Brumand-Poor received a bachelor's degree in electrical engineering from RWTH Aachen University in 2017, a master's degree in electrical engineering from RWTH Aachen University in 2019, a master's degree in automation engineering from RWTH Aachen University in 2020, respectively. He is currently working as a Research Associate at the Institute for Fluid Power Drives and Systems at RWTH Aachen University. His research areas include deep learning, physics-based learning, fluid transmission lines, and virtual sensory.

Faried Makansi graduated in mechanical engineering at TU Darmstadt with a Master of Science degree. Since 2019, he has been working as a research assistant at the Institute for Fluid Power Drives and Systems (ifas) of the RWTH Aachen University. His general research concerns leveraging digital technologies for fluid power applications, with a special focus on data-driven methods for condition monitoring.

Katharina Schmitz received a graduate's degree in mechanical engineering from RWTH Aachen University in 2010 and an engineering doctorate from RWTH Aachen University in 2015. She is currently the director of the Institute for Fluid Power Drives and Systems (ifas), RWTH Aachen University.