

Article citation info:

Puchalski A. A., Komorska I., Generative modelling of vibration signals in machine maintenance, *Eksploracja i Niezawodność – Maintenance and Reliability* 2023: 25(4) <http://doi.org/10.17531/ein/173488>

Generative modelling of vibration signals in machine maintenance

Indexed by:



Andrzej Adam Puchalski^{a,*}, Iwona Komorska^a

^a University of Technology and Humanities in Radom, Poland

Highlights

- Generative models gaining increasing use in various fields, including machinery operation, are presented.
- A VAE variational autoencoder was proposed as a tool for generating measurement observations for vibration monitoring of rotating machinery to complement unbalanced databases.
- The algorithm was optimised and verified in a practical solution to the task of generating data for intermediate states of a demonstration gearbox.

Abstract

The exponential development of technologies for the acquisition, collection, and processing of data from real-world objects is creating new perspectives in the field of machine maintenance. The Industrial Internet of Things is the source of a huge collection of measurement data. The performance of classification or regression algorithms needs to take into account the random nature of the process being modelled and any incomplete observability, especially in terms of failure states. The article highlights the practical possibilities of using generative artificial intelligence and deep machine learning systems to create synthetic measurement observations in monitoring the vibrations of rotating machinery to improve unbalanced databases. Variational Autoencoder VAE generative models with latent variables in the form of high-level input features of time-frequency spectra were studied. The mapping and generation algorithm was optimised and its effectiveness was tested in the practical solution of the task of diagnosing the three operating states of a demonstration gearbox.

Keywords

time-frequency analysis, condition monitoring, anomalies detection, deep generative models, variational autoencoders, data distribution.

This is an open access article under the CC BY license (<https://creativecommons.org/licenses/by/4.0/>)

1. Introduction

The most common way of searching for vibrodiagnostic models is based on time-frequency analysis and the spectral properties of the observed vibration signals. Traditionally, methods such as the short-time Fourier transform, Wigner-Will transform, wavelet transform, envelope analysis, or multifractal analysis are used. In the implementation of the diagnostic process on the basis of signal observations, methods for the identification of dynamic models using a subspace of observations with a reduced dimension relative to the input variables such as the principal component analysis (PCA) method, or its probabilistic equivalent (pPCA), work well [25].

The 21st century has seen a surge in successful artificial intelligence (AI) and machine learning (ML). A large number of machine learning techniques have been developed such as regression, classification, Kalman filters, neural networks, decision trees, support vector machines, and others. A higher degree of knowledge about the object to be diagnosed increases the efficiency of detection, localisation, and identification. The Industrial Internet of Things (IIoT) is a source of huge measurement data sets. However, the recorded signals relate primarily to the fault-free states of the monitored objects [15]. Insufficient training data relating to failure states makes it

(*) Corresponding author.
E-mail addresses:

A. A. Puchalski, (ORCID: 0000-0002-5202-3410) andrzej.puchalski@uthrad.pl,
I. Komorska (ORCID: 0000-0002-6911-0297) iwona.komorska@uthrad.pl,

difficult for regression or classification algorithms to work. The complexity of industrial systems in the era of Industry 4.0 requires the use of both mathematical and empirical models as well as modelling and identification methods developed from control theory, automation, and artificial intelligence techniques.

Although artificial intelligence emerged as a distinct field in the mid-20th century and is now a widely used term, there is no clear understanding of this concept [14,21]. From the early days of AI, methods and tools for decision-making were sought such as rule-based expert systems that used pre-developed rules to generate responses or data sets. AI methods still do not reflect the way humans process data, but only replace them in making quick and accurate decisions.

Most activities require inference under conditions of uncertainty due to the random disruption of the modelled process, incomplete observability, or incomplete modelling since the model is always a simplification of the actual process. A probabilistic modelling approach allows uncertainty to be expressed under conditions of limited or poor data quality. The quantitative level of uncertainty of a random variable in an event with a probability distribution $p(x)$ is determined by the Shannon entropy, i.e., the expected value of the self-information in a random event from this distribution [5,22,29]:

$$H(x) = \mathbb{E}_{x \sim p}[I(x)] = -\mathbb{E}_{x \sim p}[\log_2 p(x)] \quad (1)$$

where $I(x)$ is the event's self-information.

The difference between the two probability distributions

Table 1. Comparison and examples of discriminative and generative models.

Discriminative models	Generative models with latent variables
Discriminative models are based on a conditional probability distribution $p(y x)$, that an observation \mathbf{x} belongs to a certain category y	Generative models learn the joint distribution: $p(x, y) = p(x y)p(y) = p(y x)p(x)$, in order to estimate $p(y x)$ they use Bayes' theorem: $p(y x) = p(x, y)/p(x)$.
They mark the boundary between classes	They model the dispersion of individual classes
They may misclassify outliers	They are sensitive to outliers that can significantly affect the distribution.
Linear regression, logistic regression, NN classification, SVM, decision tree	variation autoencoders VAE, generative adversarial networks GAN

A discriminative model built using mean values of individual features in the observation space does not allow the training set samples to be completed. We assume that the parameters of the distribution of the trait under study in the population, such as the mean or variance, are definite but

$p(x)$ and $q(x)$ with respect to the same random variable is defined by the Kullback-Leibler divergence (KL):

$$D_{KL}(p||q) = \mathbb{E}_{x \sim p} \left[\log_2 \frac{p(x)}{q(x)} \right] \quad (2)$$

The use of probabilistic methods in modelling has been reflected in generative algorithms based on Monte Carlo simulations and Markov processes for the synthesis of time series, the prediction of indicators of the condition of objects, or the estimation of readiness and reliability of vehicles and machines [2,17,23]. The leading AI and machine learning paradigm in data processing and generation is currently deep neural networks [3,5].

The combination of these two paradigms - probability theory and deep learning - has resulted in the construction of generative AI systems [27]. Learning models can be divided into discriminative and generative (Table 1), where discriminative modelling is a supervised learning method using a labelled data set, while generative modelling is usually performed using an unlabelled dataset, i.e., as a form of unsupervised or semi-supervised learning when labels are missing for many or even most of the training examples. Generative artificial intelligence started in the 1960s in chatbots, but it was only in the second decade of the 21st century, with the introduction of variational autoencoders and adversarial networks, that it gained extraordinary popularity in the scientific and industrial communities.

unknown quantities.

A generative model is a multilayer neural network that allows approximations of multivariate probability distributions that we can use to obtain synthetic data. The problem of generating synthetic data is equivalent to the problem of

generating a new vector according to the 'true probability distribution' in a multidimensional vector space, or - in other words - generating a random variable with respect to a specific probability distribution.

Advances in machine learning and, in particular, the training of generative models have opened up another opportunity to develop model-based diagnostic methods. In generative modelling, as in the Bayesian approach, we assume that the parameters of interest to us are random variables. There is no doubt about the importance of apriori knowledge of the objects being diagnosed in obtaining information on their technical condition. Therefore, based on expert knowledge (*a priori* information), we can predict their distributions and modify these assumptions on the basis of sample data. Generative models are algorithms that learn the posterior distribution through Bayes' rule. A key factor in generative modelling is, therefore, to know the joint distribution of the data and to be able to predict new observations that could be included in the training set.

Generative artificial intelligence technologies are universal technologies with an undirected purpose that can profoundly affect many areas of life, analogous to steam power, electricity, and computers, associated with the 1st, 2nd, and 3rd industrial revolutions, respectively.

Deep neural network algorithms have evolved over the past few years, achieving the ability to generate signals by random sampling from a latent continuous space (variational autoencoders VAE) or to generate new examples of input data from noise input (generative adversarial networks GAN). VAE networks proposed in 2013 [9,11,12] perform well in cases of signal analysis and anomaly detection, whereas GAN adversarial networks [6,7,20] are typically used for visual data.

Implicit modelling methods using VAE and GAN networks are constantly being developed. In both cases, they can be a solution when rapid access to a sufficiently large amount of high-quality data is impossible or difficult. The latent representation in the VAE model is the multivariate probability distribution that best defines the input data. The distributions returned by the network encoder are forced to be close to a standard normal distribution with parameters (μ , σ). Sampling from this distribution yields a hidden observation subspace, which the network decoder uses to generate a new observation.

The GAN model generator samples the latent variable space, which is assumed to have a Gaussian distribution. The network discriminator attempts to reconstruct the probability distribution using data loss functions that determine the distance between the distribution of the data generated by the GAN and the distribution of the actual observation. Loss functions result from the loss of reconstruction and the regulatory element - KL divergence.

Deep VAE and GAN models are increasingly visible in industrial applications for identifying defective components during industrial production. The data augmentation approach is particularly promising when dealing with limited training data. If we have a small number of observations showing sample faults to teach the network, using generative models, we can generate additional observations for any type of defect.

The authors of the publication [1] demonstrated the advantages of GAN and VAE modelling in the construction of digital twins for monitoring systems using the example of the real and synthetic vibration acceleration signals. In [13], the use of VAE is proposed as an effective method for dimensionality reduction and nonlinear feature extraction that preserves the local structure of training data for monitoring multidimensional nonlinear processes. The article [30] proposes interactive identification of the stability of the power grid by analysing pixel maps using an improved VAE model.

The challenges and future opportunities of biomedical research in relation to VAEs in the fields of data generation and representation learning can be found in [28]. [31] compares the performance of different VAE models in the process monitoring systems of the 4.0 industry. The application of a method that combines the benefits of VAE and GAN has been demonstrated and tested using Tennessee Eastman's benchmark industrial process [8]. Performance measures such as fault detection rate, false alarm rate, and fault detection delay were included in the study. A technique for generating synthetic time series of smart home data based on generative latent variable models has been proposed in [18]. An extensive review of 63 articles from various disciplines published between 2014 and 2021 and an analysis of deep generative machine learning models in engineering design are presented in [19].

To summarise, many applications of convolutional neural networks CNNs in the task of object state classification can be

found in the literature. There are also examples of image generation using VAE variational autoencoders. A novelty presented in this article is the use of VAE variational autoencoders in the task of generating vibrodiagnostic signals.

This paper proposes the VAE variational autoencoder as a tool for generating measurement observations for vibration monitoring of rotating machinery to improve unbalanced databases. Training, validation, and testing of the process of creating a latent subspace of observations using high-level input features of time-frequency spectra and sampling coded variables using a reparameterisation procedure with a variable derived from a standard normal distribution was realised. The loss function minimised during training included reconstruction in the output layer and regularisation in the hidden layer. The effectiveness of the proposed algorithm was tested in a practical solution to the task of diagnosing three operating states of a demonstration gearbox: fault-free, worn, and increased clearance .

2. MATERIAL AND METHODS

2.1. Generative model with latent variable VAE

Generative models with a latent variable are networks with a subspace of observations with dimension reduced relative to the input variables. Generative models of latent variables include VAE and GAN. VAEs are a non-linear extension of pPCA. The VAE training algorithm enables the implementation of a generative process that exploits the properties of the continuous hidden observation subspace. In [26], the algorithm is complemented by a series of Hausholder transformations of latent variables. Other types of generative VAE models, called self-supervised models, with additional variables that result from discrete and deterministic transformations of the data, were proposed in [4]. Over the past decade, many new variants and detailed discussions of the VAE model for different data types have regularly been published [10,27].

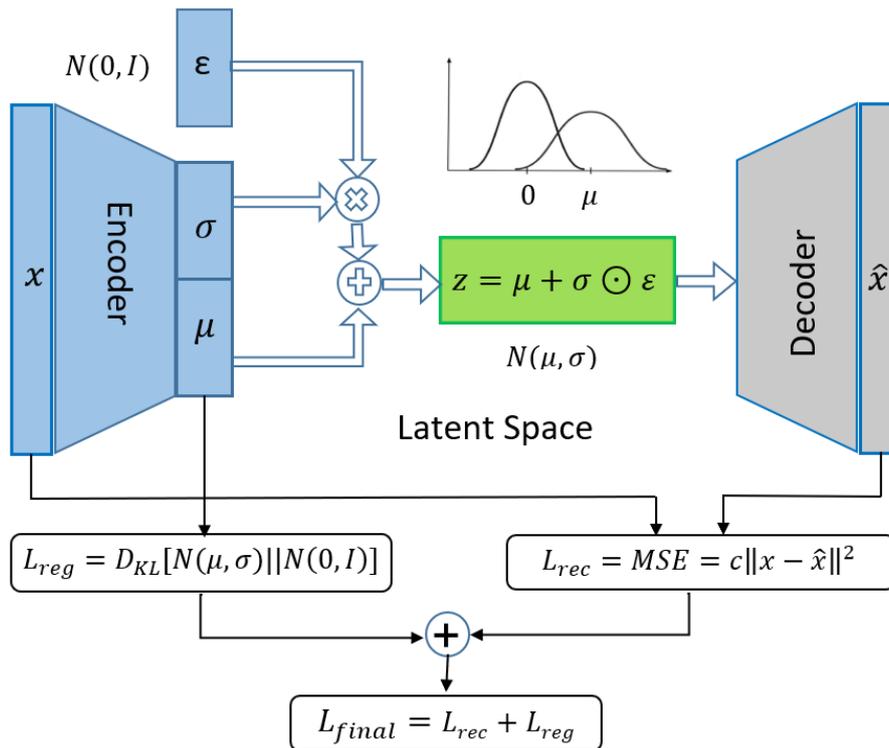


Fig. 1. Simplified schematic of the VAE variational autoencoder

VAE models (Fig. 1) are structured as autoencoders with encoder and decoder layers to learn probabilistic data distributions. The data from the observation space is encoded as a multivariate normal distribution in the latent space using vectors of mean value - μ and standard deviation - σ . The mean

value indicates the mean and the standard deviation VAE delineates the area of the latent variable z representing the reduced output, which is passed to the decoder after sampling.

An indicator of reconstruction loss is the mean-square error *MSE*:

$$L_{rec} = MSE = c\|x - \hat{x}\|^2 \quad (3)$$

where:

x is a variable representing the input from the observation space,

\hat{x} is the variable representing the reconstructed data,

c is the weighting factor of the reconstruction.

The normalisation of the latent space to standard normal distributions is guaranteed by the regularisation component of the loss function. The regularisation index is expressed as the Kulback-Leibler divergence between the returned distribution and the standard Gaussian distribution

$$L_{reg} = D_{KL}[N(\mu, \sigma)||N(0, I)] \quad (4)$$

Sampling of the coded latent variables $z \sim N(\mu, \sigma)$ is carried out using a reparameterisation procedure, with the variable ε derived from a standard normal distribution $\varepsilon \sim N(0, 1)$

$$z = \sigma\varepsilon + \mu \quad (5)$$

The VAE algorithm is implemented in the following steps:

- I. The encoder defined by $p(z|x)$ encodes data from the input observation space into a latent subspace;
- II. The decoder $p(x|z)$ samples the latent space using a reparameterization procedure and reconstructs the input data;
- III. The reconstruction error is back-propagated through the network for optimisation. The loss function is the sum of the reconstruction loss and KL divergence.

A trained VAE allows sampling from the latent space to generate synthetic data. It is also possible to seamlessly transform data between the input space of actual observations and the space of new data (morphing). This effect can be achieved by performing simple transformations on selected vectors in the latent space.

Suppose we have two 'points' in latent space, A and B, which represent two recorded observations. The morphing effect is the result of decoding successive points from observation A to observation B according to the relationship:

$$z_{new} = z_A \cdot (1 - \alpha) + z_B \cdot \alpha \quad (6)$$

where z_A and z_B represent extreme observations and α belongs to the interval $\langle 0, 1 \rangle$.

The hidden VAE space is a continuous space and the generation of synthetic data consistent with the input observation space is possible regardless of the number of hidden features.

2.2. Experimental setup

Verification of the presented method was carried out on a demonstration gearbox laboratory bench (Fig.2). The stand consisted of an electric motor, a clutch, and a gearbox with straight teeth (number of teeth: 19). The load was pressure-controlled with a bypass valve up to 5 MPa.

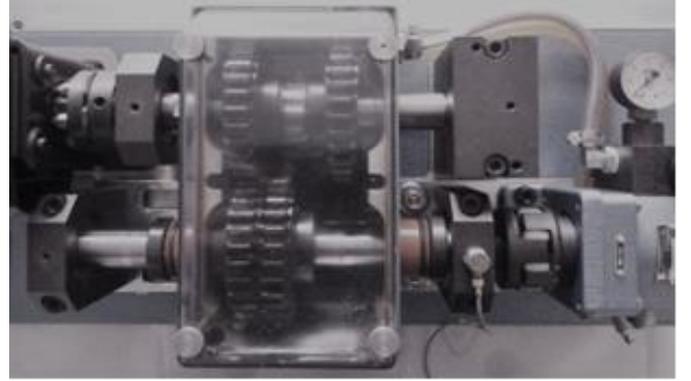


Fig. 2. Station View.

The experiment concerned the following operating states:

- Fault-free (new gears, optimal clearance, parallel position of the shaft);
- Worn teeth;
- Increased valve clearance (+0.3 mm).

The time waveforms of the vibration acceleration signal on the bearing housing and the speed signal were recorded. The measurement path comprised a B&K DeltaTron type 4519-002 acceleration sensor with a frequency range of 1 Hz - 20 kHz, bolted to the bearing housing, and a DBK04 dynamic signal acquisition board equipped with a set of anti-aliasing filters interfacing with an IQtech DaqBook 2005 main data acquisition module, which transmits data via an Ethernet link to a computer. Data was recorded at a sampling rate of 10 kHz.

For a selected rotational speed of 2000 rpm and a load of 1 MPa, 300 time samples were recorded for each condition, of which 70% (210 samples) were used in the training process and 30% (90 samples) in the testing process.

The input of the neural network can be given data in any 1D form (e.g., time series or FFT spectrum), 2D (2-dimensional matrix or greyscale image), 3D (RGB image), etc. In this case, it was decided to use data in the form of a time-frequency spectrum (STFT) because, after generating new data by the network, it is possible to reverse the transformation to the time course [16]. An alternative way is to transfer data to the network

in the form of a spectrum obtained by the wavelet transform method.

Example time waveforms for these three states and their STFT spectra are shown in Fig. 3.

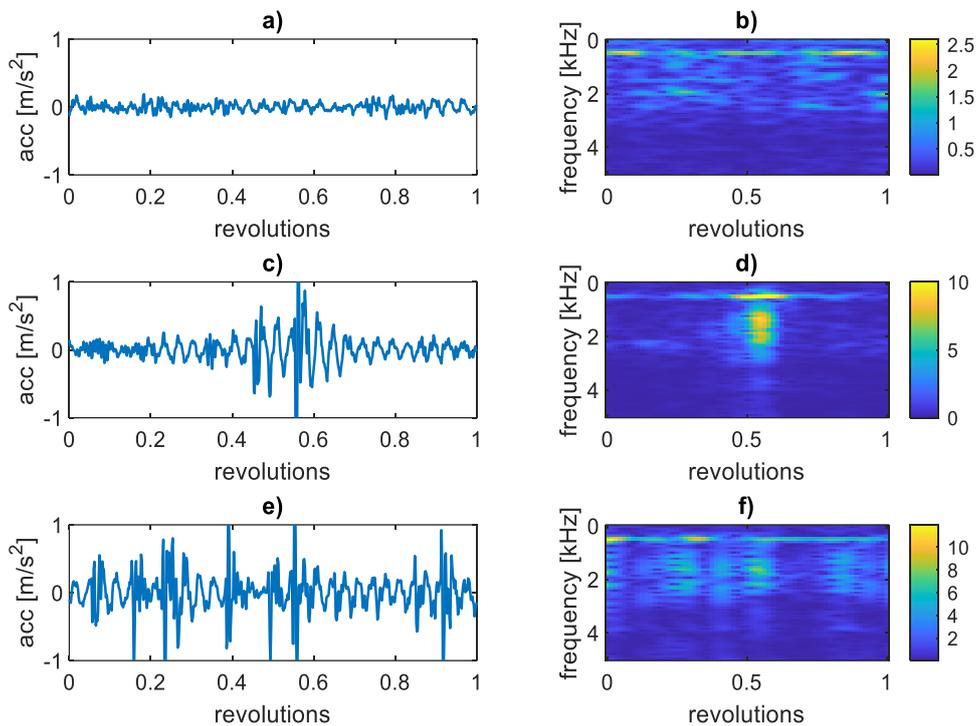


Fig. 3. Examples of time waveforms and spectrograms for a motor shaft speed of 2000 rpm for three gear states: a&b) fault-free, c&d) worn, e&f) increased clearance.

Fig. 4 shows examples of mean frequency spectra.

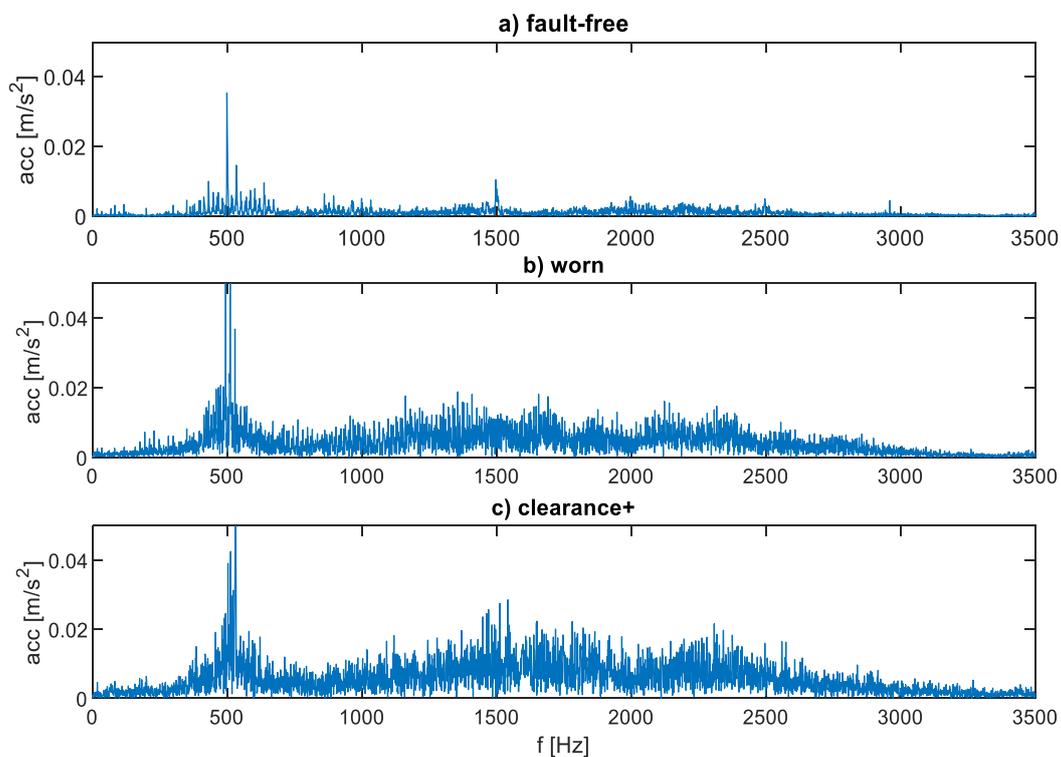
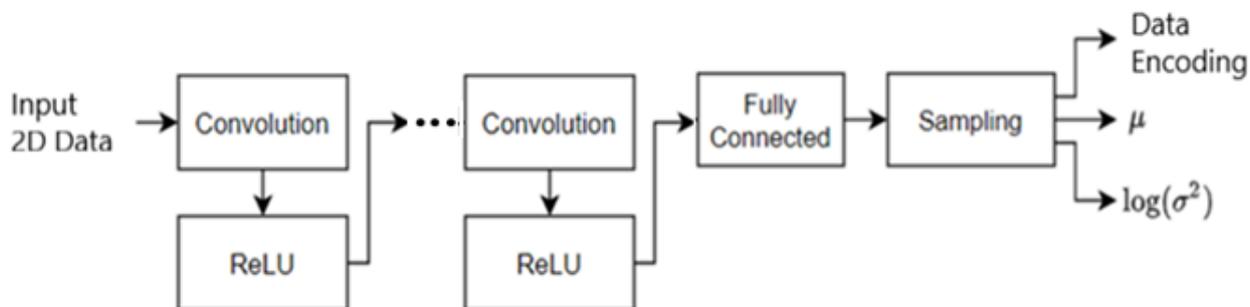


Fig. 4. Frequency spectra for a motor shaft speed of 2000 rpm for three gearbox states: a) fault-free, b) worn, c) increased clearance.

The frequency spectra will be used later in the work to verify the newly generated data. For this purpose, they were divided into four frequency bands: 0-300Hz, 300-750Hz, 750-3000Hz, and 3-5 kHz.

a) Encoder



b) Decoder

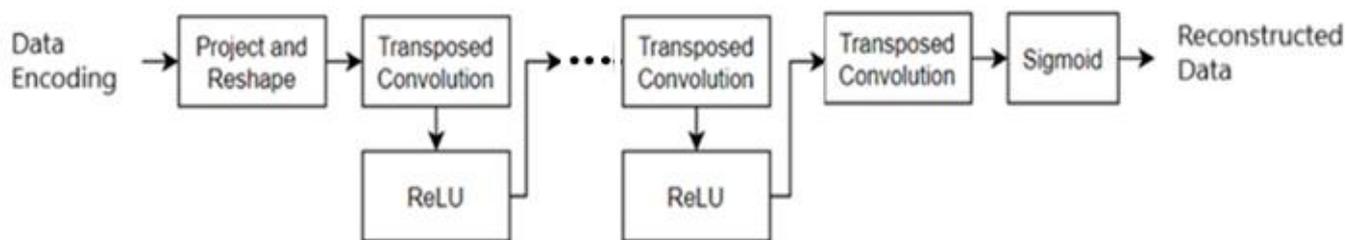


Fig.5. Structure of the VAE network used in the example (a) encoder structure, (b) decoder structure.

The encoder and decoder are almost mirror images. The difference lies in the last layer of the encoder (sampling layer), which returns a latent vector reporting the mean values of the input signal features and their standard deviation. An arbitrary vector of 16 features in latent space was adopted. The latent vector contains the compressed form of the input image in a latent space with a reduced number of dimensions.

The data is fed into the encoder as a 128x128 STFT matrix. Hidden layers (convolutional and ReLU) were used. Convolution is carried out by multiplying the filter pixels by the image fragment (matrix) and summing the result. The result is positive when part of the image closely matches the filter, and negative when part of the image is the inverse of the filter. The filter is moved across the image, from left to right and top to bottom, recording the results of the splicing operation along the way. This yields a new array in which a particular feature of the input data is selected, depending on the value of the filter.

The data is then processed by a non-linear activation function before being sent to the next layer. It is crucial to ensure

3. RESULTS

3.1. VAE Network structure

Data in the form of an STFT spectrum was provided on a VAE network which the structure is shown in Figure 5.

the neural network learns complex functions instead of just outputting a linear combination of inputs. There are many types of activation function, but the three most important are ReLU, sigmoid, and softmax. This work uses the rectified linear unit (ReLU) function, which takes a value of zero if the input is negative and otherwise takes the value of the input. To obtain a combined vector of means and variances, a fully connected layer with twice as many output channels as the number of hidden channels was included, i.e., 32. Detailed rules for the selection of network layers can be found, among others, in [24].

3.2. Network training and testing

The encoder and decoder are trained simultaneously. The network is trained to find weights for the encoder and decoder that minimise the loss function between the original input data and the input data reconstructed after passing through the encoder and decoder. When the network is trained, its parameters are monitored, particularly the loss function L_{final} , which is the sum of two components:

$$L_{final} = L_{rec} + L_{reg} \quad (7)$$

The reconstruction error L_{rec} defines the mean squared error MSE of the decoder output (generated data) and the original input data (cf. Eq.3):

$$L_{rec} = \frac{1}{m} \sum_{i=1}^m (x_i - \hat{x}_i)^2 \quad (8)$$

where $m = 128 \cdot 128 = 16384$ is the number of elements of the matrix, the x_i - i -th element of the input matrix, and the \hat{x}_i - i -th element of the reconstructed matrix.

The second component, the Kullback-Leibler divergence, measures the difference between the two probability distributions. Minimising KL loss in this case means ensuring that the learned means and variances are as close as possible to the target (normal) distribution. The loss of KL (cf. Eq.4), L_{reg} is calculated as:

$$L_{reg} = -\frac{1}{2} \sum_{i=1}^K [1 + \log(\sigma_i^2) - \mu_i^2 - \sigma_i^2] \quad (9)$$

where K - is the dimension of the vector in the latent space ($K = 16$), and μ_i and σ_i are the mean value and standard deviation of the i -th feature, respectively.

The course of the loss function is shown in Fig. 6.

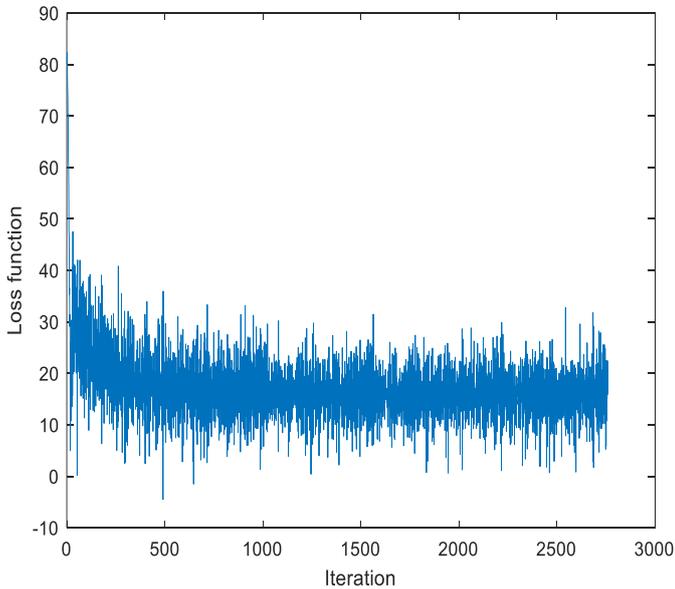


Fig. 6. Changes in the loss function during network training.

The stabilisation of the loss function at its end signifies the end of the learning process for the existing learning set. The mean value of the loss function has stabilised, indicating that, for a given network configuration and data set, the learning process has ended. In contrast, the deviation from the mean value is indicative of significant variation in the learning data.

The resulting latent space during network training was used to generate new time-frequency spectra with features calculated according to equation (Eq.5), similar to the input signals. A random set of such data is shown in Fig. 7.

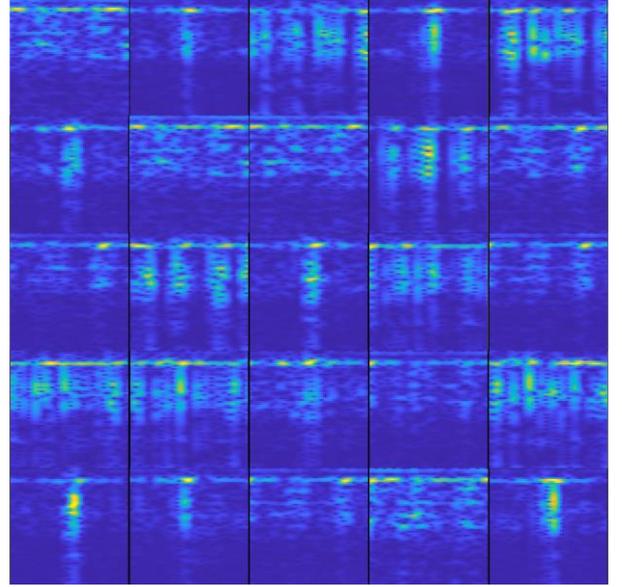


Fig.7. A set of 25 time-frequency spectra randomly generated by the decoder.

The generated matrices, like the input data, are 128x128 in size. As can be seen, each of the generated spectra can be classified into one of three classes according to their technical condition: fault-free, worn, or increased clearance. This is the first stage of verification of the obtained data.

The next step is to verify the data in the time and frequency domain. For this purpose, it was necessary to obtain the time signals using the inverse short-time Fourier transform iSTFT [16]. In the time domain, basic statistics such as mean value, root mean square (RMS), skewness, and kurtosis were determined. In the frequency domain, meanwhile, mean-square values were calculated in four frequency bands: 0-300Hz, 300-750Hz, 750-3000Hz, and 3-5 kHz. These frequency bands are characteristic of the signal under study and were determined from the frequency spectra (cf. Fig. 4).

Fig.8 presents the distributions of the four statistical features: mean value, root mean square (RMS), skewness, and kurtosis of the newly generated data (shown in red) against the corresponding distributions for the data obtained during the measurements for the three operating states. The verification results for the frequency domain are shown in Fig.9.

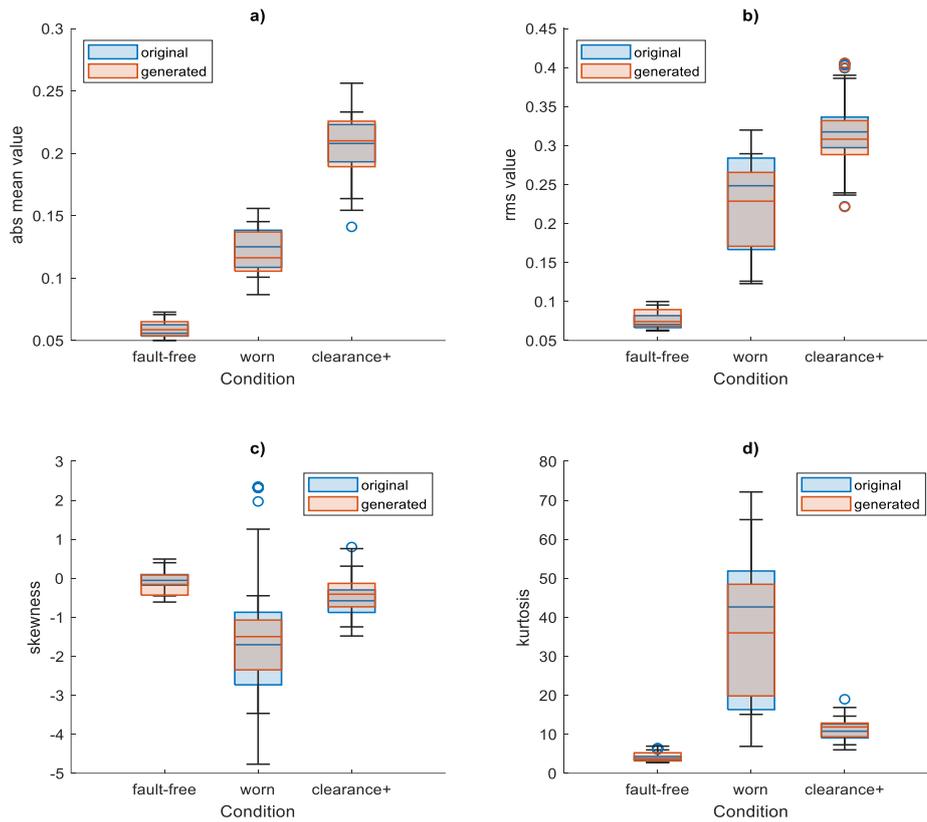


Fig. 8. Verification of the generated data in the time domain. Comparison of the distributions of (a) mean value, (b) mean squared value, (c) skewness, (d) kurtosis, for three operating states: fault-free, worn, or increased clearance.

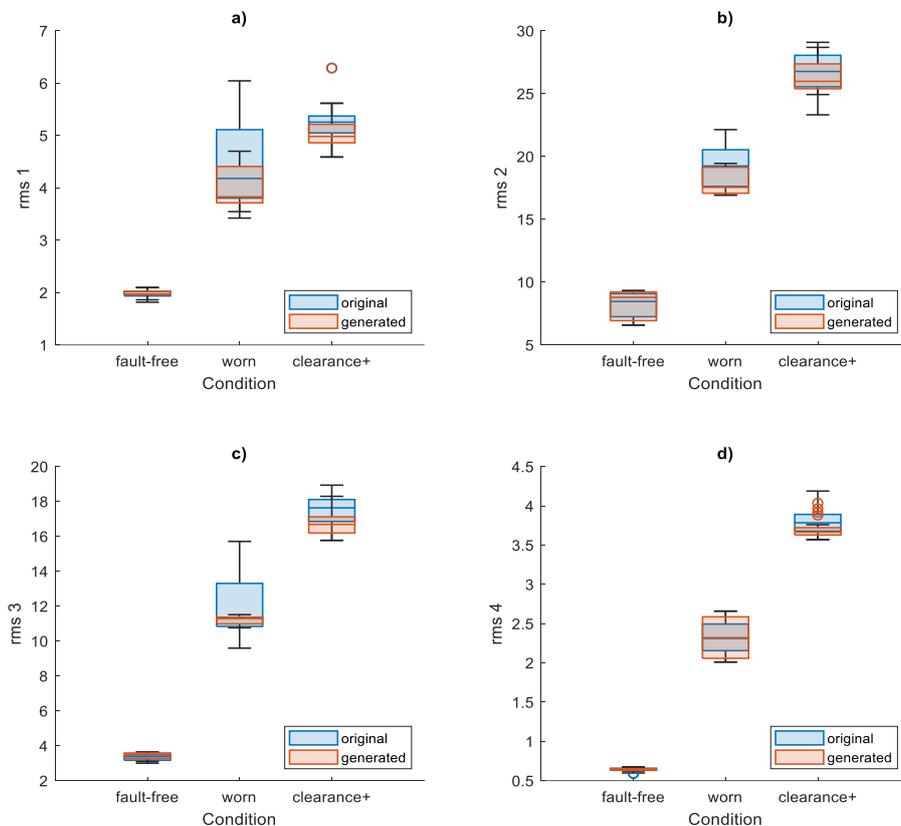


Fig. 9. Verification of the generated data in the frequency domain. Comparison of the mean squared value distributions in four frequency bands: a) 0-300Hz, b) 300-750Hz, c) 750-3000Hz, and d) 3-5 kHz for three operating states: fault-free, worn, or increased clearance.

Based on the analysis of Fig.8 and Fig.9, the data generated was considered to fall within or slightly beyond the benchmark data distributions. Any deviations may be related to STFT-iSTFT transformation errors. It was assumed that the verification was successful.

3.3. Generation of data examples of intermediate operating states

After labelling the images according to class membership, feature values were visualised - a vector in the latent space with class division (Fig.10).

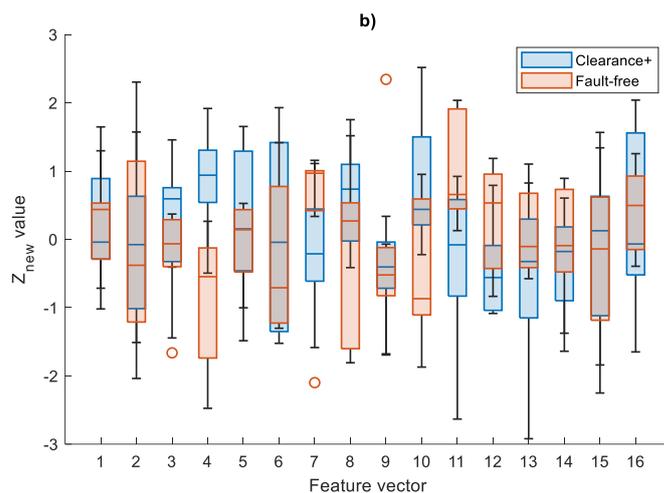
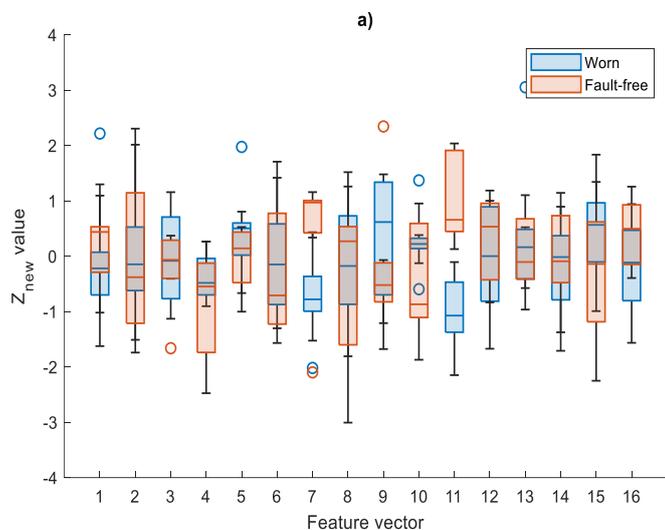


Fig.10 Feature vector of the 100 newly generated time-frequency spectra (a) comparison for the fault-free and worn states, (b) comparison for the efficient and increased clearance states.

A feature was sought in the latent space that corresponds to class membership. For 'fault-free' and 'worn' states, this is feature No. 11, and for efficient and increased clearance states, this is feature No. 4. Fig. 11a shows the transformation of the spectrum from the fault-free class to the worn class by gradually changing the value of feature no. 11 in the range $\langle -2, -1.5 \rangle$, and Fig. 11b shows an analogous change from the fault-free class to the increased clearance class by changing the value of characteristic No. 4 in the range of approx. $\langle -2, 1.5 \rangle$, according to the formula (Eq.6).

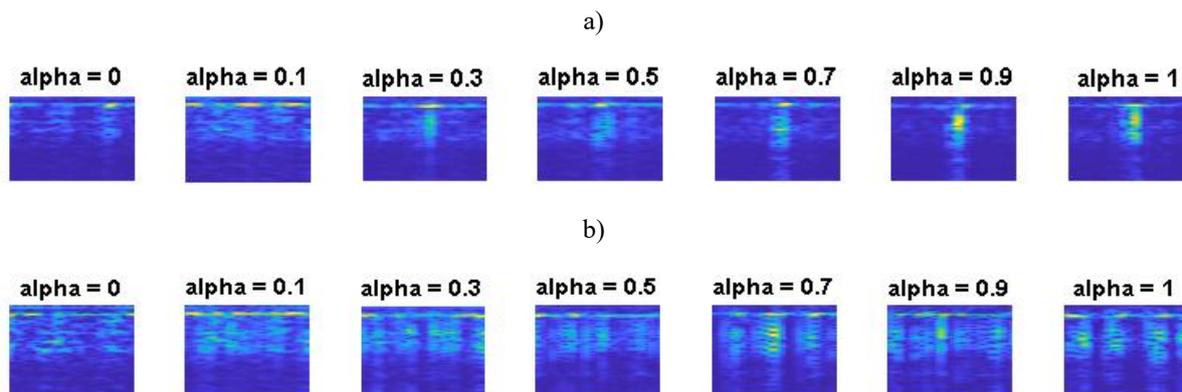


Fig.11. Transformation of signals from the fault-free class to the class: a) worn, b) increased clearance.

Note that small changes in the coefficient α do not always lead to analogous changes in the observed spectrum, since the random parameter ε (cf. Eq.5) means that a pattern can be taken from any point in the latent space distribution to generate a new spectrum $N(\mu, \sigma)$.

4. CONCLUSIONS

This article presents a method for generating diagnostic signals of missing intermediate states of wear or damage on machinery using deep neural networks. The use of generative models with latent variables Variational Autoencoder VAE in a diagnostic

task was proposed. A procedure based on solving a classification task while analysing the estimated distributions of the recorded training data was used to implement the forecasting and decision-support process. Using a probabilistic model (probability density distribution function) of the input data, datasets can be generated to complement the real data. This opens up the possibility of a new perspective on the problem of running machine operations. Using the demonstration gear damage example, it is shown how the VAE network describes

the time-frequency spectrum with its own feature vector. A verification of the generated data was carried out.

It has been demonstrated how changing the value of one of the latent space features allows a 'transition' from one operational state to another. Further work will aim to select and optimise the model structure for selected industrial applications. The subject of the research will be the development of algorithms for the diagnosis and ongoing estimation of the machine's life expectancy.

References

1. Booyse W, Wilke DN, Heyns S. Deep digital twins for detection, diagnostics and prognostics. *Mech Syst Signal Process.* 2020 Jun;140:106612. <https://doi.org/10.1016/j.ymsp.2019.106612>
2. Borucka A, Niewczas A, Hasilova K. Forecasting the readiness of special vehicles using the semi-Markov model. *Eksplot i Niezawodn.* 2019 Dec 31;21(4):662–9. <https://doi.org/10.17531/ein.2019.4.16>
3. Foster D. *Generative deep learning: Teaching Machines to Paint, Write, Compose, and Play.* O'Reilly Media Inc.; 2019. 330 p. https://books.google.com/books/about/Generative_Deep_Learning.html?hl=de&id=RqegDwAAQBAJ
4. Gatopoulos I, Tomczak JM. Self-supervised variational auto-encoders. *Entropy.* 2021 Jun 14;23(6):747. <https://doi.org/10.3390/e23060747>
5. Goodfellow I, Bengio Y, Courville A. *Deep Learning.* MIT Press; 2016. <http://www.deeplearningbook.org>
6. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial networks. *Commun ACM.* 2020 Jun 10;63(11):139–44. <https://doi.org/10.1145/3422622>
7. Goodfellow I. NIPS 2016 Tutorial: Generative Adversarial Networks. 2016 Dec 31; <https://doi.org/10.48550/arXiv.1701.00160>
8. Jang K, Hong S, Kim M, Na J, Moon I. Adversarial Autoencoder Based Feature Learning for Fault Detection in Industrial Processes. *IEEE Trans Ind Informatics.* 2022 Feb;18(2):827–34. <https://doi.org/10.1109/TII.2021.3078414>
9. Kingma DP, Rezende DJ, Mohamed S, Welling M. Semi-supervised learning with deep generative models. In: *Advances in Neural Information Processing Systems.* 2014. p. 3581–9. <https://arxiv.org/pdf/1406.5298v1.pdf>
10. Kingma DP, Welling M. An introduction to variational autoencoders. Vol. 12, *Foundations and Trends in Machine Learning.* 2019. p. 307–92. <https://doi.org/10.1561/22000000056>
11. Kingma DP, Welling M. Auto-encoding variational bayes. In: *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings.* 2014. <https://doi.org/10.48550/arXiv.1312.6114>
12. Kingma DP. Fast Gradient-Based Inference with Continuous Latent Variable Models in Auxiliary Form. 2013 Jun 4. <https://doi.org/10.48550/arXiv.1306.0733>
13. Lee S, Kwak M, Tsui KL, Kim SB. Process monitoring using variational autoencoder for high-dimensional nonlinear processes. *Eng Appl Artif Intell.* 2019 Aug;83:13–27. <https://doi.org/10.1016/j.engappai.2019.04.013>
14. McCarthy J, Minsky ML, Rochester N, Shannon CE. A proposal for the Dartmouth summer research project on artificial intelligence. *AI Mag.* 2006;27(4):12–4 <https://doi.org/10.1609/aimag.v27i4.1904>
15. Michnej M, Młynarski S, Pilch R, Sikora W, Smolnik M, Drożyner P. Physical and reliability aspects of high-pressure ammonia water pipeline failures. *Eksplotacja i Niezawodność – Maintenance and Reliability* 2022; 24 (4): 728–737, <http://doi.org/10.17531/ein.2022.4.13>
16. Portnoff MR. Time-Frequency Representation of Digital Signals and Systems Based on Short-time Fourier Analysis. *IEEE Trans Acoust.* 1980 Feb;28(1):55–69. <https://doi.org/10.1109/TASSP.1980.1163359>
17. Puchalski A, Komorska I, Ślęzak M, Niewczas A. Synthesis of naturalistic vehicle driving cycles using the markov chain monte carlo method. *Eksplot i Niezawodn.* 2020 Jun 30;22(2):316–22. <https://doi.org/10.17531/ein.2020.2.14>
18. Razghandi M, Zhou H, Erol-Kantarci M, Turgut D. Variational Autoencoder Generative Adversarial Network for Synthetic Data Generation in Smart Home. In: *IEEE International Conference on Communications.* 2022. p. 4781–6.

<https://doi.org/10.1109/ICC45855.2022.9839249>

19. Regenwetter L, Nobari AH, Ahmed F. Deep Generative Models in Engineering Design: A Review. Vol. 144, Journal of Mechanical Design, Transactions of the ASME. 2022. <https://doi.org/10.1115/1.4053859>
20. Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, Chen X. Improved techniques for training GANs. In: Advances in Neural Information Processing Systems. 2016. p. 2234–42. <https://doi.org/10.48550/arXiv.1606.03498>
21. Shannon BC, Tollman SG. A neuropsychological examination of multiple sclerosis and its impact upon higher mental functions. South African J Psychol. 1994;24(3):152–62. <https://doi.org/10.1177/008124639402400307>
22. Shannon CE. A Mathematical Theory of Communication. Bell Syst Tech J. 1948 Jul;27(3):379–423. <https://doi.org/10.1002/j.1538-7305.1948.tb01338.x>
23. Świderski A, Borucka A, Grzelak M, Gil L. Evaluation of Machinery Readiness Using Semi-Markov Processes. Appl Sci. 2020 Feb 24;10(4):1541. <https://doi.org/10.3390/app10041541>
24. Taye MM. Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions. Vol. 11, Computation. 2023. p. 52. <https://doi.org/10.3390/computation11030052>
25. Tipping ME, Bishop CM. Probabilistic Principal Component Analysis. J R Stat Soc Ser B Stat Methodol. 1999 Sep 1;61(3):611–22. <https://doi.org/10.1111/1467-9868.00196>
26. Tomczak JM, Welling M. Improving Variational Auto-Encoders using Householder Flow. 2016 Nov 29; <https://doi.org/10.48550/arXiv.1611.09630>
27. Tomczak JM. Deep Generative Modeling. Deep Generative Modeling. Cham: Springer International Publishing; 2022. 1–197 p. <https://doi.org/10.1007/978-3-030-93158-2>
28. Wei R, Mahmood A. Recent Advances in Variational Autoencoders with Representation Learning for Biomedical Informatics: A Survey. IEEE Access. 2021;9:4939–56. <https://doi.org/10.1109/ACCESS.2020.3048309>
29. Witten E. A mini-introduction to information theory. Vol. 43, Rivista del Nuovo Cimento. 2020. p. 187–227. <https://doi.org/10.1007/s40766-020-00004-5>
30. Zhang T, Feng H, Chen Z, Wang C, Huang Y, Tang Y, et al. An Interactive Insight Identification and Annotation Framework for Power Grid Pixel Maps using DenseU-Hierarchical VAE. 2019 May 22. <https://doi.org/10.48550/arXiv.1905.12164>
31. Zhu J, Jiang M, Liu Z. Fault detection and diagnosis in industrial processes with variational autoencoder: A comprehensive study. Sensors. 2022 Dec 29;22(1):227. <https://doi.org/10.3390/s22010227>