DIAGNOSTYKA, 2023, Vol. 24, No. 4

e-ISSN 2449-5220
DOI: 10.29354/diag/176854

APPLICATIONS OF GENERATIVE MODELS WITH A LATENT OBSERVATION SUBSPACE IN VIBRODIAGNOSTICS

Andrzej PUCHALSKI * , **Iwona KOMORSKA**

[1] Casimir Pulaski Radom University, Mechanical Faculty, Stasieckiego 54, 26-600 Radom, Poland
* Corresponding author, e-mail: andrzej.puchalski@uthrad.pl

Abstract

The vibration signal is one of the most essential diagnostic signals, the analysis of which allows for determining the dynamic state of the monitored machine set. In the era of cyber-physical industrial systems, making diagnostic decisions involves the study of large databases from previous registers and data downloaded from machines in real-time. However, the recorded signals mainly concern the operational status of the monitored object. Insufficient training data regarding failure states hinders the operation of classification algorithms. Progress in machine learning has created a new avenue for the advancement of diagnostic methods based on models. These methods now have the capability to produce signals through random sampling from a hidden space or generate fresh instances of input data from noise. The article suggests the use of a Generative Adversarial Network (GAN) model as a tool to create synthetic measurement observations for vibration monitoring. The effectiveness of the synthetic data generation algorithm was verified on the example of the vibration signal recorded during tests of the drive system of a motor vehicle.

Keywords: vibration signal, deep neural network, generative adversarial network, GAN model, synthetic subspace

List of Symbols/Acronyms

AGI – Artificial Generative Intelligence;
D(z) – Discriminator network;
GAN – Generative Adversarial Network;
G(z) – Generator network;
STFT – Short Time Fourier Transform;
VAE – Variational Autoencoders;

1. INTRODUCTION AND RELATED WORKS

In the era of cyber-physical industrial systems, making diagnostic decisions involves the analysis of large databases from previous registers and data downloaded from machines in real time. However, the recorded signals mainly concern the operational status of the monitored object. The inadequate amount of training data related to instances of system failures poses a challenge to the effective functioning of classification algorithms. One step towards improving the effectiveness of automatic anomaly detection systems is to understand the distribution of available data, leading to synthetic data generation [19].

Synthetic data was treated as a supplement to expensive, uncertain, real data limited by available technologies or regulations. Their potential is growing in advanced machine learning algorithms; they constitute almost half of the training data, and analytical and research companies predict that by 2030, they will be the basic data component for models of artificial generative intelligence (AGI).

Augmenting available data increases the applicability and accuracy of machine learning models and creates conditions for introducing artificial intelligence when data for random scenarios is lacking. Synthetic data is used by the automotive, machinery, electronics, pharmaceutical, energy, construction engineering, health care and other industries. The article in reference [17] provides an overview of publications from 2014 onwards, delving into the applications of generative models across diverse domains. It also conducts an analysis of the utilization of deep generative machine learning in the realm of engineering design.

Applications involving AGI applications are widely present in media content (images, texts, audio and video) and the design of parts, materials and medicines [1, 2, 13]. A technique for generating synthetic time series of smart home data based on latent variable generative models was proposed in [16]. Generating artificial voltage collapse in the complex power grid is the subject of research presented in [8]. The article referenced as [9] introduces an examination of feature learning for fault detection in industrial processes, focusing on the application of adversarial autoencoder techniques. The study specifically utilizes the Tennessee Eastman benchmark process as a case

Received 2023-10-17; Accepted 2023-12-12; Available online 2023-12-12

study. A first attempt at applying GANs to unsupervised synthesis of raw-waveform audio was introduced in [4]. Improved version of GAN for synthesising percussive sounds can be found in [15].

The traditional discriminative model, constructed by considering average values of individual features within the given dataset, lacks the capability to augment the samples within the training set. The challenge associated with deep machine learning is the construction and application of generative models that enable making complex decisions based on synthetic data regarding situations that have never been recorded by the system before. Generative models are multi-layer neural networks that approximate multidimensional probability distributions containing a random element. While the discriminative model focuses on understanding the conditional probability distribution of the target variable given known feature values, the essence of generative modelling lies in understanding the joint distribution of input data and the capacity to forecast new observations that could be integrated into the initial training set [14, 19].

Deep neural network algorithms enable synthetic data generation by randomly sampling from a latent subspace. The most popular models that have been used in industrial solutions are Variational Autoencoders VAE [10,11,12] and Generative Adversarial Networks GAN [7]. Each of these models consists of two deep neural networks. The Variational Autoencoder (VAE) serves as an encoder, compressing the input observation space and transforming it into a multidimensional normal distribution represented by two vectors: the mean value (μ) and standard deviation (σ). Additionally, it functions as a decoder, generating synthetic data by using samples derived from the latent subspace. The role of the decoder in the GAN model is played by a generator network that samples and processes a vector from random noise into new examples of input data. The multidimensional Gaussian distribution is compared with real data by a discriminator, which is a binary classifier. Within the Variational Autoencoder (VAE), the latent vector is produced through the encoding process, whereas in the Generative Adversarial Network (GAN), the latent vector is derived from random noise. To parameterize the encoder, decoder, generator, and discriminator, deep neural networks are employed in both VAE and GAN.

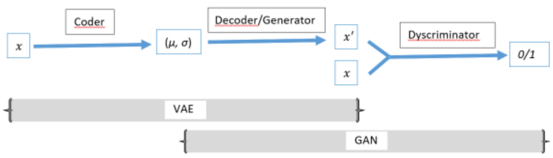


Fig. 1. Structure of VAE and GAN models

The paper suggests utilizing a Generative Adversarial Network (GAN) model as a means of creating synthetic measurement observations to assess the dynamic state of a monitored drive system in a motor vehicle. The article raises the problem of training neural networks fed with data from the analysis of vibration signals. The obtained synthetic observations were verified for their use in supplementing databases representative of emergency states. The efficacy of the proposed algorithm was verified through experiments conducted on both simulated and real-world data.

2. MATERIAL AND METHODS

2.1. Model of generative adversarial networks

Generative adversarial networks GANs proposed in [5,6,18] have become a significant tool in machine learning. GANs, unlike other generative models, do not estimate the probability density explicitly, require a latent spatial variable z and define a stochastic model that can directly generate different data distributions. White noise is the primary observation space of the generator. The generator network $G(z): Z \rightarrow X$ is opposed to the adversary, which is the discriminator network used to determine whether the sample comes from the model distribution or the data distribution. The discriminator $D(x): X \rightarrow < 0,1 >$ is trained to recognize fake data until it can distinguish it from real data. The sigmoid activation function ensures that the output signal is scaled to the range $< 0; 1 >$.

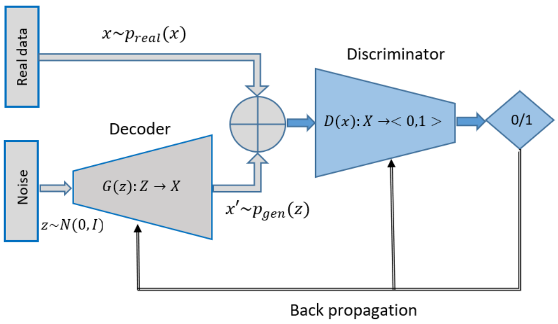


Fig. 2. Simplified diagram of the GAN model

Training is a zero-sum game. Equilibrium occurs when the generator has mastered producing perfect synthetic data, and the discriminator always indicates that the output is true or false with equal probability. During training, the discriminator analyses:

- $x -$ real data, i.e. $x \sim p_{real}(x)$
- $x' -$ synthetic data of the generator, i.e. $x' \sim p_{gen}(z)$

During the training phase, the discriminator employs data loss functions to quantify the dissimilarity between the distribution of data produced by the Generative Adversarial Network (GAN) and the distribution of real data. The loss function $H(x)$ is a combination of two components: the loss of the generator fooling the discriminator and the loss of the discriminator classifying true and false data. Given that the discriminator functions as a binary classification model, we employ a binary cross-entropy loss function:

$$H(x) = \mathbb{E}_{x\sim p_{real}}[\log_2 D(x)] + \\ +\mathbb{E}_{z\sim p_{gen}}[\log_2(1 - D(G(z))] \quad (1)$$

The process has double feedback: discriminator - actual observations and discriminator - generator. To minimize the loss function, the discriminator parameters are updated using backpropagation. The loss function is based on the GAN output states, which may result in the discriminator achieving convergence faster than the generator and problems with stable training. Both networks are trained alternately. During joint training, the discriminator weights should be locked to update only the generator weights. Once the GAN converges, the generator can create synthetic data. The GAN training algorithm includes the following:

I. Random noise sampling.
II. Generating synthetic data.
III. Transferring data to the discriminator.
IV. Calculation of binary classification loss.
V. Backpropagation by discriminator and generator.
VI. Model parameterization.

Several iterations of Generative Adversarial Networks (GANs) have been created, such as DCGAN, SRGAN, VAE-GAN, WGAN, cycleGAN, and styleGAN, each featuring distinct adjustments and applications [3].

2.2. Experimental setup

The effectiveness of the synthetic data generation algorithm was verified on the example of the vibration signal recorded during tests of the combustion engine.

Tests were conducted on the four-cylinder spark ignition engine of a Fiat Punto 1.4 with a mileage of 400,000 km during road tests. The examination involved measuring engine vibrations at different speeds and loads. Piezoelectric vibration sensors (B&K Delta Shear type 4393) with a frequency range of 0.1 – 16500 Hz were used, and they were attached to the engine side at cylinder 1. The measurements were taken using a portable data recording device (B&K PULSE type 3560E). Engine block vibrations were recorded vertically and horizontally at a frequency of 65536 Hz. To ensure accurate results, the recorded signals underwent preprocessing, which included applying an anti-aliasing filter to prevent amplification of components within the natural frequency range of the vibration sensor.

Generating vibrations and noise in internal combustion engines is a complicated process. The vibrations result from a mix of periodic waves linked to rotating components and reactions to sudden forces associated with the linear and rotary movements of pistons, as well as excitations induced by the pressure of gas against cylinder walls. Intense and momentary shifts in the vibroacoustic signal arise from the functioning of components like inlet and exhaust valves, injectors, the combustion process, and the interaction between pistons and cylinder linings.

Certain excitations in the engine are regular and repetitive, such as the strokes of the piston against the cylinder lining and the opening and closing of valves (in engines with consistent valve timing). On the other hand, there are excitations that vary with angles, such as during injection and ignition processes. Consequently, when conducting vibration measurements, it is crucial to capture supplementary informative signals that synchronize with the engine's operation, like the position of the crankshaft.

An example of the engine vibration acceleration signal as a function of the crankshaft rotation angle is shown in Fig.3.

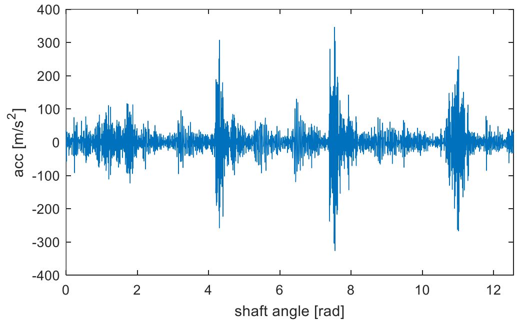


Fig. 3. Example time waveform of engine vibration acceleration for one engine operation cycle at a shaft rotation speed of 2000 rpm

Since the operation of each cylinder causes a different vibration response of the system, each operation cycle has been divided into four parts corresponding to the operation of individual cylinders. The synchronized and superimposed vibration responses for individual cylinders are shown in Fig.4.

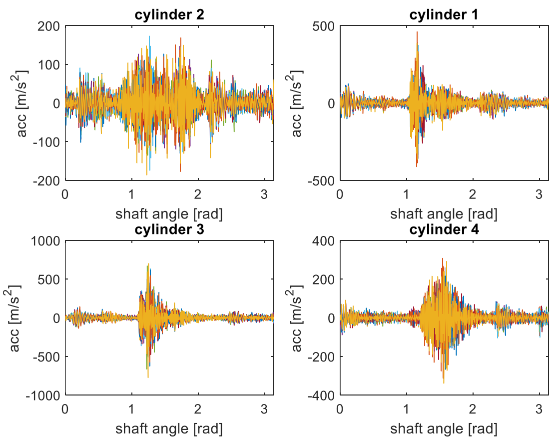


Fig. 4. Time waveforms of the vibration response for each cylinder during 10 engine operation cycles

The vibration responses for each cylinder were subjected to the Short-Time Fourier Transform

STFT and as 128x128 matrices constituted the input database for training the GAN network, consisting of 5920 samples. A visualization of an example sample for each cylinder is shown in Fig.5.

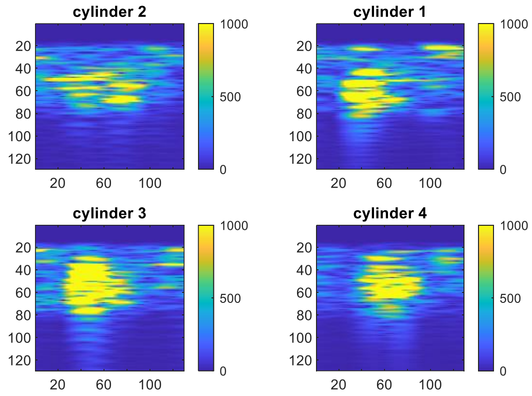


Fig. 5. Time-frequency spectra of the vibration response of each cylinder for an example engine operation cycle

The training data underwent normalization to achieve a zero mean and a standard deviation of one. Subsequently, the mean Short-Time Fourier Transform (STFT) and standard deviation were computed for each frequency bin.

2.3. Structure of GAN

Data in the form of 128 x 128 arrays, visualised as STFT spectrum, were fed into a GAN with 5 hidden layers.

The generator takes a 1x100 vector as input, sampled from a normal distribution, and produces an image with dimensions identical to those in the original training dataset (128x128). Within the Generative Adversarial Network (GAN), the generator transforms the vector within the latent space into a Short-Time Fourier Transform (STFT) image. The specific architecture of the generator is detailed in Table 1.

Table 1. GAN generator architecture

Operation	Output shape
Input z	1 x 1 x 100
Dense (Fully connected)	16384 x 1
Reshape	4 x 4 x 1024
ReLU	4 x 4 x 1024
TransConv (Stride=2)	8 x 8 x 512
ReLU	8 x 8 x 512
TransConv (Stride=2)	16 x 16 x 256
ReLU	16 x 16 x 256
TransConv (Stride=2)	32 x 32 x 128
ReLU	32 x 32 x 128
TransConv (Stride=2)	64 x 64 x 64
ReLU	64 x 64 x 64
TransConv (Stride=2)	128 x 128
Tanh	128 x 128

Doubling the tensor width and height in each layer was achieved using Conv2DTranspose layers with stride=2. ReLU (Rectified Linear Unit) is a type of activation function that introduces non-linearity to the model. It outputs the input directly if it is positive; otherwise, it outputs zero. Tanh, short for hyperbolic tangent, is another type of activation function commonly used in neural networks. Similar to the sigmoid function, tanh squashes its input to be in the range of $(-1,1)$.

The role of the discriminator is to determine whether an image is genuine or generated. Essentially, it tackles a supervised image recognition problem. The structure of the discriminator is listed in Table 2.

Table 2. GAN discriminator architecture

Operation	Output shape
Input x or G(z)	128 x 128 x 1
Conv2D (Stride=2)	64 x 64 x 64
LeakyReLU (α=0.2)	64 x 64 x 64
Conv2D (Stride=2)	32 x 32 x 128
LeakyReLU (α=0.2)	32 x 32 x 128
Conv2D (Stride=2)	16 x 16 x 256
LeakyReLU (α=0.2)	16 x 16 x 256
Conv2D (Stride=2)	8 x 8 x 512
LeakyReLU (α=0.2)	8 x 8 x 512
Conv2D (Stride=2)	4 x 4 x 1024
LeakyReLU (α=0.2)	4 x 4 x 1024
Reshape	16384 x 1
Dense (Fully connected)	1 x 1

The input of the discriminator is a 128x128 image. Then there are 5 convolutional layers sequentially. Finally, the last convolutional layer is flattened into a vector. In the convolutional layers, a stride of 2 was employed to diminish the size of the tensor as it traverses the network. The utilization of the sigmoid activation function in the final layer ensures that the output signal is normalized to a range between 0 and 1. This will predict the probability that the image is authentic.

3. RESULTS AND DISCUSSION

The training set consisted of 2960 samples. The generator and the discriminator were trained simultaneously. The learning rate for both the generator and discriminator was established at 0.0002. Additionally, a gradient decay factor of 0.5 and a squared gradient decay factor of 0.999 were applied to both networks.

The Fig.6 shows the loss function of the generator and the discriminator (Eq.1).

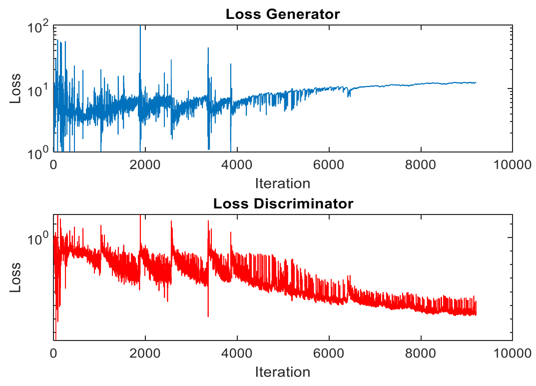


Fig. 6. Changes in the generator and discriminator loss functions while training the GAN

After about 6000 iterations, the discriminator and the generator find equilibrium, the loss function of the generator increases and the loss function of the discriminator decreases. The generator assimilates pertinent information from the discriminator, leading to an enhancement in the quality of the generated images. The comparison for the generated and original STFT after 100 epochs is shown in Fig.7.

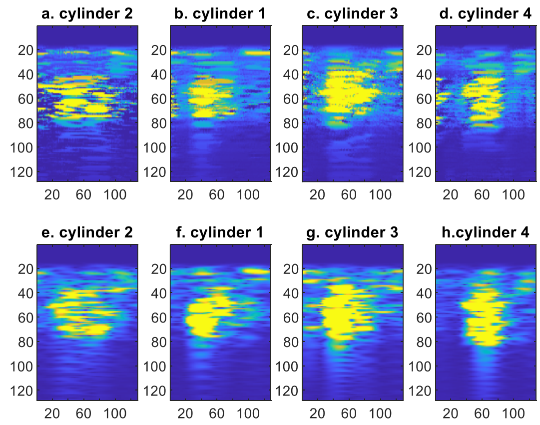


Fig. 7. Generated (a-d) versus original (e-h) STFT

The generated images are intended to be similar, but not identical. A slight checkerboard pattern is noticeable in the generated images (Fig.7 a-d), which is the result of the use of the Conv2DTranspose layer. For comparison, the Upsampling+Conv2D method can be tested in future works.

One prevalent issue during GAN training is mode collapse, where the generator identifies a limited set of samples that deceive the discriminator. Consequently, the generator struggles to produce diverse outputs beyond this restricted set. If the discriminator becomes too effective at distinguishing real from generated samples, the generator may decide to adopt a strategy of only producing samples that are more difficult for the discriminator to identify.

Solving the mode collapse problem is an ongoing challenge in GAN training. Scientists and practitioners use various techniques to alleviate this problem. One is to explore alternative loss functions that may help in capturing a broader range of modes in the data. Another approach is introducing mechanisms to explicitly encourage the generator to explore different modes in the data distribution, such as adding diversity-promoting terms to the loss function.

Next, the generated STFTs should be verified. Statistical measures, mean and root mean squared values were calculated for the original and generated STFT according to the formulas

Mean value

$$\bar{X} = \frac{1}{128}\frac{1}{128}\sum_{i=1}^{128}\sum_{j=1}^{128} x_{ij} \tag{2}$$

Rms value

$$X = \sqrt{\frac{1}{128}\frac{1}{128}\sum_{i=1}^{128}\sum_{j=1}^{128} x_{ij}^2} \tag{3}$$

where x is a value in 128 x 128 array, i and j are the row and column numbers respectively.

Mean and RMS values for original and 30 generated arrays were compared and visualised in boxplot Fig.8.

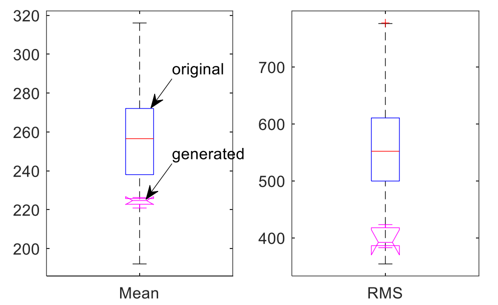


Fig. 8. Comparison of statistical measures calculated for original and 30 generated STFT

The graph shows that the generated matrices (images) are characterized by a small spread of the average and RMS values compared to the original ones. Their median is below the 25th percentile compared to the original ones.

4. CONCLUSIONS AND FUTURE WORK

The paper suggests the utilization of a Generative Adversarial Network (GAN) model as a tool to generate synthetic measurement observations, valuable for machine vibration diagnostics. An example of generating representative time-frequency spectra of the vibration signal recorded on the combustion engine block is presented. The generated spectra are similar, but not identical, to the training set samples. Verification of image mapping using mean and rms values showed that the generated signals were within the acceptable range, although not within the 25-75 percentile range. When training the network, you may encounter difficulties related to the collapse of the GAN mode, which require, for example, the definition of a modified loss function. Further work on the use of the GAN network in vibrodiagnostics will focus on modifying this network in order to obtain even better results.

Deep generative models contribute to scientific progress. Using the example of the tested drive system, it was confirmed that they can provide examples of new situations and forecast input data without the need to conduct long and expensive tests.

Source of funding: *no founding.*

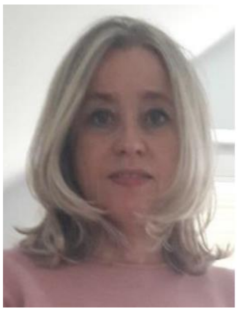
Author contributions: *research concept and design, A.P., I.K.; Collection and/or assembly of data, I.K.; Data analysis and interpretation, I.K.; Writing the article, A.P., I.K.; Critical revision of the article, A.P.; Final approval of the article, A.P.*

Declaration of competing interest: *The author declares no conflict of interest.*

REFERENCES

1. Abdel-Jaber H, Devassy D, Al Salam A, Hidaytallah L, EL-Amir M. A review of deep learning algorithms and their applications in healthcare. Algorithms 2022; 15(2): 71. https://doi.org/10.3390/a15020071.
2. Alqahtani H, Kavakli-Thorne M, Kumar G. Applications of generative adversarial networks (GANs): an updated review. Archives of Computational Methods in Engineering 2021; 28(2): 525–552. https://doi.org/10.1007/s11831-019-09388-y.
3. Dash A, Ye J, Wang G. A review of generative adversarial networks (GANs) and its applications in a wide variety of disciplines -- from medical to remote sensing. 2021. https://doi.org/10.48550/arXiv.2110.01442.
4. Donahue C, McAuley J, Puckette M. Adversarial audio synthesis. ICLR 2019:1–16. https://doi.org/10.48550/arXiv.1802.04208.
5. Goodfellow I, Bengio Y, Courville A. Deep learning. Massachusetts Institute of Technology Press 2016.
6. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S. Generative adversarial networks. Communications of the ACM 2020; 63(11): 139–44. https://doi.org/10.1145/3422622.
7. Goodfellow I. NIPS 2016 tutorial: generative adversarial networks. 2017. https://doi.org/10.48550/arXiv.1701.00160.
8. Hassani H, Razavi-Far R, Saif M, Palade V. Generative adversarial network-based scheme for diagnosing faults in cyber-physical power systems. Sensors 2021;21(15):5173. https://doi.org/10.3390/s21155173.
9. Jang K, Hong S, Kim M, Na J, Moon I. Adversarial autoencoder based feature learning for fault detection in industrial processes. IEEE Transactions on Industrial Informatics 2022; 18(2): 827–34. https://doi.org/10.1109/TII.2021.3078414.
10. Kingma DP, Rezende DJ, Mohamed S, Welling M. Semi-supervised learning with deep generative models. 2014. https://doi.org/10.48550/arXiv.1406.5298.
11. Kingma DP, Welling M. Auto-encoding variational bayes. 2022. https://doi.org/10.48550/arXiv.1312.6114.
12. Kingma DP. Fast gradient-based inference with continuous latent variable models in auxiliary form. 2013. https://doi.org/10.48550/arXiv.1306.0733.
13. Nikolenko SI. Synthetic data for deep learning. Springer optimization and its applications 2021; 174. https://doi.org/10.1007/978-3-030-75178-4.
14. Puchalski A, Komorska I. Generative modelling of vibration signals in machine maintenance. Eksploatacja i Niezawodnosc–Maintenance and Reliability 2023;25(4). http://doi.org/10.17531/ein/173488.
15. Ramires A, Chandna P, Favory X, G'omez E, Serra X. Neural percussive synthesis parameterised by high-level timbral features. ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2019; 786-790.
16. Razghandi M, Zhou H, Erol-Kantarci M, Turgut D. Variational autoencoder generative adversarial network for synthetic data generation in smart home . 2022. https://doi.org/10.48550/arXiv.2201.07387.
17. Regenwetter L, Nobari AH, Ahmed F. Deep generative models in engineering design: a review. 2022. https://doi.org/10.48550/arXiv.2110.10863.
18. Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, Chen X. Improved techniques for training GANs. 2016. https://doi.org/10.48550/arXiv.1606.03498.
19. Tomczak JM. Deep generative modeling. Deep Generative Modeling. Cham: Springer International Publishing 2022; 1–197.

Andrzej PUCHALSKI received Ph.D. degree in Automatic Control and D.Sc. in Construction and Maintenance of Machines from Technical University of Warsaw. He works as Associate Professor at Radom University. Currently, he is the Dean of the Faculty of Mechanical Engineering. His main field of research include technical diagnostics and mechatronics.



Iwona KOMORSKA, DSc., DEng. – Associate Professor at Radom University. Her current research interests include technical diagnostics, data mining, machine learning, signal processing, modelling & simulation, mechatronics.