Real-Time Control AE-TadGAN Model in IoT Edges for Smart Manufacturing

SANGHOON DO, JONGPIL JEONG
Department of Smart Factory Convergence, Sungkyunkwan University
2066 Seobu-ro, Jangan-gu, Suwon 16419
REPUBLIC OF KOREA

Abstract- With the development of the Internet of Things (IoT), real-time processing of data has become an important key as various and many data have been generated at the manufacturing site. The development of IoT has brought Cloud Computing (CC) to attention. However, it has the problem of latency and delay, and traditional centralized data processing can violate real-time processing, drawing attention to distributed processing technology. Edge Compression (EC) technology is a technology that distributes a variety of data at the manufacturing site and enables real-time processing. Distribute the various processes of traditional servers and use a near-field network to compensate for latency and latency problems. In this study, we propose an architecture that allows EC to perform the pre-processing, small-scale analysis, and connection for facility control, which are the processes performed on the server with EC development.

Keywords- IoT, Edge, Smart Manufacturing, MES

Received: April 18, 2021. Revised: June 19, 2022. Accepted: July 21, 2022. Published: September 13, 2022.

1. Introduction

INDUSTRY 4.0's smart industry uses information and communication technology to improve productivity [1]. One of them is the Internet of Things (IoT) system, which allows monitoring and control in various areas due to advances in hardware and low-power communication technology [2]. Industrial IoT (IIoT) is a significant study [3, 4]. IIoT uses a variety of sensors to monitor the performance of the production process [5]. High-performance cloud computing (CC) is attracting attention to processing and analyzing much of IIoT's data. CC's flexibility and scalability of its resources have made its technology a huge success. However, CC's speed and latency do not meet service quality requirements [6], but edge computing (EC) is close to IIoT and can meet the requirements for speed and latency [7]. EC provides computing, storage, and network resources in close proximity to IoT devices. The distance between Edge and IoT is close, so the delay time is reduced and the transmission time is shorter than that of CC [8].

But we can't handle everything with EC in the immediate area. It is true that EC is inferior to CC in performance to CC. And there may be latency and latency because you need to connect to the server to save it in DB. Distributed processing is also difficult in terms of management due to increased management points. Conversely, from a centralized point of view, there are few points of management that make it easier to manage. Therefore, you need to define an advantageous role in Edge to maximize the benefits of distributed processing and ensure smooth production at the manufacturing site. In addition, a study was conducted that as network technology develops through distributed processing through the edge of the network, the use of the edge has advantages due to pretreatment of the network [9]. Through this study, it was designed to reduce the load of the server by pre-processing manufacturing data and parsing and pre-processing IoT data and manufacturing data by adopting the distributed processing method of the edge. There was a study of deep learning (DL) and machine learning (ML) at the edge [10]. There are various characteristics of data in manufacturing. ML is a simple analysis of one layer, so it is sufficiently possible in EC, but DL is an analysis of several layers, so it is necessary to distinguish the areas that can be done. So, we analyzed the small-scale and pre-trained DL and designed it to be able to respond to the facility immediately.

In our study, IoT Edge was designed using the development of EC technology in order to process the large number of data at the manufacturing site in real time. IoT Edge was used to allow distributed processing, away from the existing centralized processing method on the server. There are three types of distributed treatment. First, to prevent the number of network connections from increasing by connecting directly to the facilities from the server, we decided to make a hub-type connection using IoT Edge. Second, data from the facility was distributed across each node of the IoT Edge to reduce the load on the server. Third, Anomaly Detection was designed using

Autoencoder (AE) and Generative Adversarial Networks (GAN). The IoT Edge processes the pre-trained model on the server so that it can respond to the facility in real time. These three objectives are designed to ensure real-time performance, which is the most important factor in the manufacturing field. The final experimental results of Figure 1 show that the average time of the analysis model using AE-TadGAN is 33.1% faster to transmit on IoT Edge than on Cloud.

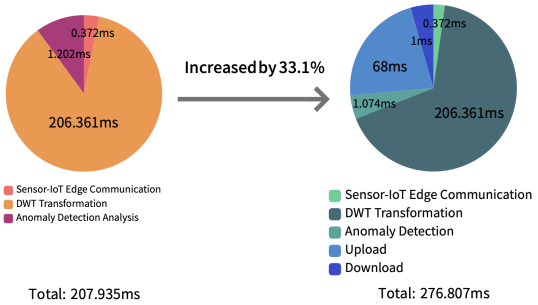

Fig. 1: Comparison of Processing Time vetween IoT Edge (Left) and Cloud (Right)

This study consists of: Section II. describes the relevant operation. Section III. describes an architecture using IoT Edge. Section IV. confirms the results of the analysis on IoT Edge through experiments. Finally, Section V. discusses conclusions and future research plans.

2. Related Work

2.1 Smart Manufacturing

The concept of IoT first appeared in "The Road Ahead," published in 1995. [11, 12] However, it was not noticed due to problems with communication, hardware, and sensor technology. With the recent rapid development of RFID technology, the technology of IoT is drawing attention. This is because IoT is a promising technology because it can build an industrial system by increasing utilization through the development of communication, hardware, and sensors [13]. IoT technology is a technology that allows real-time operation by connecting physical and virtual objects, and it is an application that can be used in various fields other than manufacturing [14, 15]. EC is being studied a lot to handle computational intensive tasks of IIoT. In EC scenarios, dynamic computational offloading techniques are proposed for use with IIoT as a technique that minimizes energy consumption [16]. Supports detachable delay and accuracy-aware video analysis in the cloud edge IoT framework [17], and EC shares pipelines to reduce unnecessary resource costs between edges.

2.2 AE

The anomaly detection method has received a lot of attention over the past few years. Typically, you can classify them into statistical methods, neighborhood-based methods, and dimension-reduction-based methods. AE is an unsupervised critical feedforward artificial neural network architecture (NN) and is a data-dimensional reduction technology consisting of encoders and decoders[18, 19] With the improvement of artificial intelligence, we overcome the shortcomings of intelligent diagnostic methods such as NN and Support Vector Machine (SVM), which require manual design, many pre-analysis, and comparison processes[20]. AE provides an effective way to learn representative features. Sakurada proposed AE-based anomaly detection method [21]. AE is highly detectable because it can capture nonlinear correlation as well as linear correlation. However, the use of AE for image anomaly detection does not always show good results. This is because a single AE does not fully capture the correlation between features in a high-dimensional dataset, resulting in poor detection accuracy. That's how an ensemble-based AE was born [22].

2.3 GAN

Generative Adversarial Networks can successfully perform many image-related tasks, including image generation [23], image translation [24], video prediction [25], and researchers have demonstrated the effectiveness of GAN for anomaly detection in recent images [26, 27]. Previous GAN-related operations contained little time series data. This is because complex time correlations within this type of data pose significant challenges to generative modeling. Three works released in 2019 are drawing attention. First, Li et al. to use GAN for anomaly detection in time series. [28] suggests using the vanilla GAN model to capture the distribution of multivariate time series and to use Critical to detect anomalies. Another approach to this line is BeatGAN [29], an encoder and decoder GAN architecture that can use reconstruction errors to detect abnormalities in heartbeat signals. More recently, Yun et al. [30]We propose a time series GAN that adopts the same idea but introduces time embeddings to support network training. However, their work is designed to learn time series representation instead of anomaly detection.We present TadGAN, a new framework that allows time series reconstruction and effective anomaly detection, to show how GAN can be effectively used for anomaly detection in time series data [31].

3. Real-Time Control AE-Tad GAN Model

In our study, IoT Edge using Edge was designed for real-time data processing in manufacturing sites where a lot of data is generated.

Figure 2 is the architecture we studied. The focus of this architecture is the introduction of IoT Edge on shopfloor to reduce the load on servers in centralized MES. It is designed to allow small-scale AI analysis with analysis models such as AE-TadGAN and distributed processing in IoT Edge with enhanced hardware performance. IoT Edges and Assets, except MES server, have Assets on the shop floor. In addition, the IoT Edge we designed will exist as much as we can process Asset's data in real time on a Node-by-Node basis. Therefore, as the number of Asset increases or the amount of data increases, the number of nodes in IoT Edges increases.

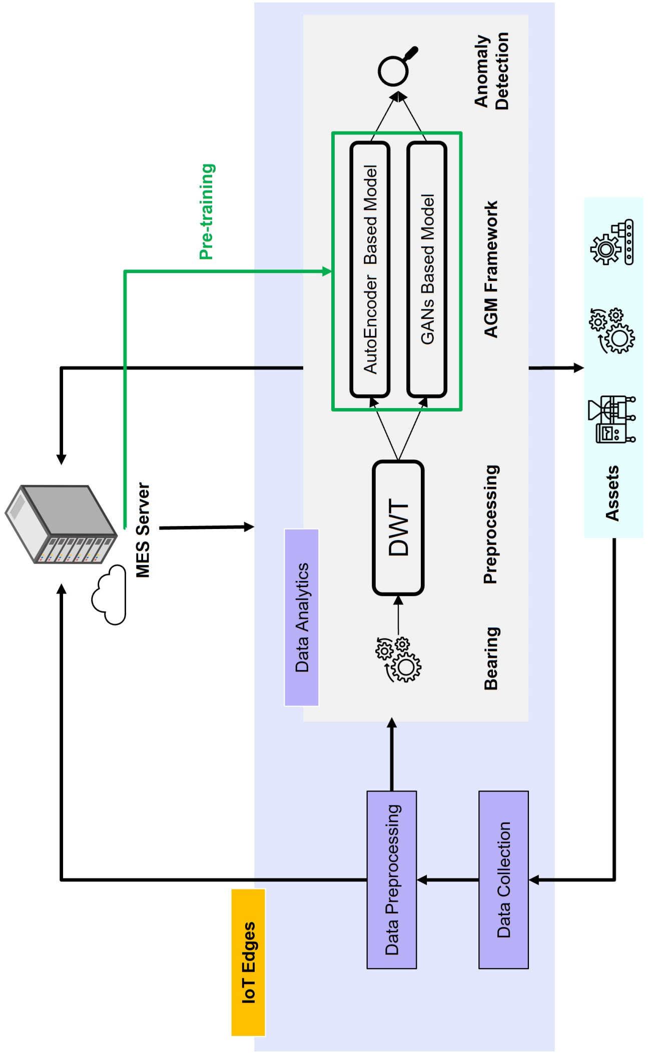

Fig. 2: Small Scale Analysis Model for IoT Edge Using AE-TadGAN

The asset creates data for the shop floor's digitized facilities. Data from Asset will be sent to IoT Edge for collection. The collected data will be preprocessed. The reason for preprocessing is that when the original data is put into the server, the data parsing operation is centralized on the server. In order to reduce the load on the server, it is handled by nodes of each connected IoT Edge as a concept of distributed processing in IoT Edge. In addition, data on facilities that require data analysis can be preprocessed to enable the analysis process to run immediately.

In data analysis, ML or pre-training small-scale AI analysis is possible. The reason why this analysis is possible in real time is that the development of hardware has improved the performance of PCs. In this study, we predict defects with the pre-trained models of AE and TadGAN in MES Server. If raw data measured by a vibration sensor is used as it is, it is difficult to analyze in the form of frequency, and there is a high possibility of false alarm. Therefore, it is important to improve the quality of input data through frequency conversion of raw data. Among the various frequency conversion methods, DWT (discrete wavelet transform) with high time resolution in high frequency areas and high frequency resolution in low frequency areas was used. The above features are important because the fast-changing high frequency has a more important time resolution to determine the position of the point of change, and the frequency resolution to determine the period of change at the slow-changing low frequency. In the case of AGM Framework, among the various types of detection algorithms, models based on AE (Autoencoder) and Generative Adversarial Networks (GANs) were used. In our work, we use the Timeseries Anomaly Detection GAN (TadGAN), a GAN model optimized for time-series data anomaly detection. TadGAN performs better than other anomaly detection models and is recognized in various fields.

	CPU	RAM	GPU
IoT Edge	AMD Ryzen 5 3600 6-Core Processor 3.60GHz	16.0 GB	NVIDIA GeForce GTX 1660 SUPER
Cloud	Intel Xeon E5-2686 v4 vCPU 8 2.30GHz	61.0 GB	NVIDIA Tesla V100

Table 1: Test Environment

Data that has been analyzed is controlled by sending a signal to the asset if a defect is predicted. It is possible to simply do a line stop, and it is possible to perform feedback control to change the recipe by checking whether there is a problem with the recipe. It is an architecture that can reduce the load of MES Server and increase real-time performance through distributed processing through IoT Edgs. In addition, the analyzed results are transmitted to the MES server to store the results and processing status on the server. Control through small-scale analysis, facility configuration changes due to operator, and IoT Edge configuration changes are transmitted from MES server to IoT Edge and processed. Asset control is enabled only on IoT Edge and does not communicate directly with MES server. The connection between MES Server and Asset reduces the number of connections by connecting to the IoT Edge, a recruitment group of Asset, because the connection pool increases.

4. Experiment and Results

4.1 Experiment Environments

We experimented with an architecture designed in Figure 2. The experimental environment is the same as Table 1. In order to pre-training analysis models using AE-TadGAN and check the real-time speed difference between IoT Edge and server, AWS Cloud was used as the role of the server, and PC with improved hardware performance was used. The data transfer from the asset to the IoT Edge used commercialized OPC-UA.

The main purpose of this experiment is to measure and compare the time when sensor data is transmitted in real time and Anomaly Detection is performed through analysis on IoT Edge and servers, respectively, to the workplace.

4.2 Data Preprocessing

For anomaly detection, the ”PHM IEEE 2012 Data Challenge” dataset, which is a dataset for vibration values of rotating bearings, was used. The dataset consists of 7172877 vibration values collected from rotating bearings in an environment of 1800 rpm and 4000 N. The sampling frequency is 25.6 kHz, and data is recorded every 1/10 second the bearing rotates.

The results of pretreatment using DWT showing excellent performance at high and low frequencies are shown in Figure 3.

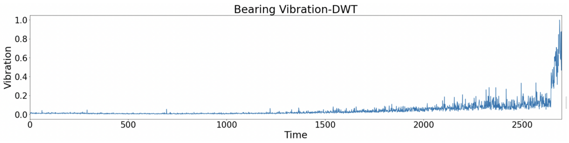

Fig. 3: Bearing Vibration Data after DWT Conversion

4.2 Results

AE-based anomaly detection uses reconstruction errors that occur in the process of compressing and restoring data to perform anomaly detection. Anomaly score is a score that indicates how close each input data point is to an outlier, and reconstruction error is used as anomaly score. After the process of selecting the optimal threshold based on the derived anomaly score, points with scores higher than the threshold are judged as abnormal values. The threshold is selected by considering the reference values of the fault frequency of ISO-10816, abnormalities at the site, and failure experience values. This experiment used USAD (Unsupervised Anomaly Detection on Multivariable Time Series), a model that performs an anonymous detection task with unsupervised setting in multivariate time series. The anomaly score measured in the process of compressing and restoring data converted through DWT using USAD is the same as the top of Figure 4. After setting the threshold based on the measured anomaly score, the section in which the red line rises at the bottom of Figure 4 is determined to be an abnormal value as a result of anomaly detection based on this value.

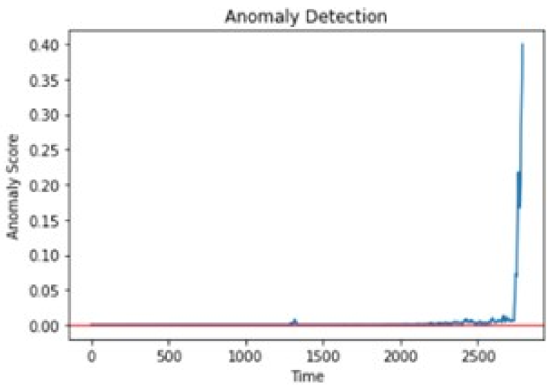

Fig. 4: Anomaly Detection Using USAD

Anomaly detection using GANs is not much different from AE-based anomaly detection, but there is a difference in obtaining an anomaly score by using the output of the dispenser together with the restore error of the generator. The result value of the TadGAN model for the data converted through DWT is the same as the blue color of Figure 5. The result of anomaly detection based on the result value of the TadaGAN model is judged to be an outlier from the point that occurs in the red normal value guideline in Figure 5.

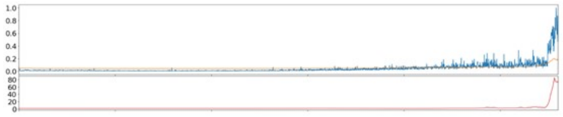

Fig. 5: Anomaly Detection Using TadGAN

To compare processing time on IoT Edge with processing time on server, DWT Transformation and AE and GAN-based anomaly detection were measured on server and processing time on IoT Edge, respectively. The expression of each processing time in time series is as shown in Figure 6.

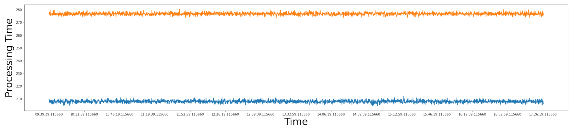

Fig. 6: Required Processing Time to Process Real-time Data in IoT Edge (Below) and Cloud (Above)

At this time, the average processing time required for DWT Transformation and AE and GAN-based anomaly detection was similar in the two environments of IoT Edge and server. This is because the analysis of the small-scale data used in the experiment does not require much computing power, so there is no difference in the processing speed between the PC-class IoT Edge and the server. However, in the case of servers, it takes additional transmission time to upload data from IoT Edge to the server and download analysis results, which increases processing time compared to IoT Edge Computing. From the results of this experiment, it is more efficient in terms of real-time performance to process analysis tasks for small-scale data on IoT Edge. This takes additional time to upload and download data to the cloud, so analysis of small-scale data is recommended for IoT Edge. In addition, when data resources (the physical number of assets and objects) increase, the use of IoT Edge is more important for distributed processing to increase real-time, as network resources and server throughput increase due to upload/download to the server.

5. Conclusions

In this paper, we conducted an experiment that distributed processing using IoT Edge can improve real-time performance rather than centralized processing that adds load to servers due to the development of EC. IoT

Edge designed a simple transmission program that was only sent to the server for distributed processing, and IoT Edge handled the pre-processing process that was loaded on the server to reduce the load on the server, and the small AI analysis model was tested on IoT Edge. Therefore, the server load is increasing day by day due to big data analysis, so we distributed the analysis on the server to IoT Edge in order to fill resources or reduce the impact on other functions of the server. This reduces the load and has less impact on real-time processing, and it can be determined and controlled immediately by IoT Edge near the shop floor to ensure real-time performance as much as the data transmission time. In real life, unlike experiments, there are many assets, so centralized network traffic and centralized delays in processing will be a problem as more IoT data is collected in the future. The solution is to design distributed processing using IoT Edge.

In this study, small-scale AI analysis was selected as AE-TadGAN, but it was not possible to measure how large an analysis would be possible from Edge computing. More experiments are needed and research is needed on how to calculate AI analysis that can be analyzed on IoT Edge. You also need to study how much you can connect to Asset. We will focus on research on the availability of IoT Edge. And to reduce the load on MES, we also need to conduct research on distributed processing on servers.

Acknowledgment

This research was supported by the SungKyunKwan University and the BK21 FOUR (Graduate School Innovation) funded by the Ministry of Education (MOE, Korea) and National Research Foundation of Korea (NRF).

References

[1] G. Aceto, V. Persico and A. Pescapé, ”A Survey on Information and Communication Technologies for Industry 4.0: State-of-the-Art, Taxonomies, Perspectives, and Challenges,” in IEEE Communications Surveys Tutorials, vol. 21, no. 4, pp. 3467-3501, Fourthquarter 2019, Available: https://doi.org/10.1109/COMST.2019.2938259

[2] L. D. Xu, W. He and S. Li, ”Internet of Things in Industries: A Survey,” in IEEE Transactions on Industrial Informatics, vol. 10, no. 4, pp. 2233-2243, Nov. 2014, Available: https://doi.org/10.1109/TII.2014.2300753

[3] Oztemel, E., Gursev, S. Literature review of Industry 4.0 and related technologies. J Intell Manuf 31, 127–182 (2020). https://doi.org/10.1007/s10845-018-1433-8

[4] W. Z. Khan, M. H. Rehman, H. M. Zangoti, M. K. Afzal, N. Armi and K. Salah, ”Industrial Internet of Things: Recent advances enabling technologies and open challenges”, Comput. Electr. Eng., vol. 81, Jan. 2020. Available: https://doi.org/10.1016/j.compeleceng.2019.106522

[5] J. Pan and J. McElhannon, ”Future Edge Cloud and Edge Computing for Internet of Things Applications,” in IEEE Internet of Things Journal, vol. 5, no. 1, pp. 439-449, Feb. 2018, Available: https://doi.org/10.1109/JIOT.2017.2767608

[6] Duc, Thang Le, et al. ”Machine learning methods for reliable resource provisioning in edge-cloud computing: A survey.” ACM Computing Surveys (CSUR) 52.5 (2019): 1-39. https://doi.org/10.1145/3341145

[7] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta and D. Sabella, ”On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration,” in IEEE Communications Surveys Tutorials, vol. 19, no. 3, pp. 1657-1681, thirdquarter 2017, Available: https://doi.org/10.1109/COMST.2017.2705720

[8] L. Chen et al., ”IoT Microservice Deployment in Edge-Cloud Hybrid Environment Using Reinforcement Learning,” in IEEE Internet of Things Journal, vol. 8, no. 16, pp. 12610-12622, 15 Aug.15, 2021, Available: https://doi.org/10.1109/JIOT.2020.3014970

[9] I. Satoh, ”5G-enabled Edge Computing for MapReduce-based Data Pre-processing,” 2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC), 2020, pp. 210-217, Available: https://doi.org/10.1109/FMEC49853.2020.9144882

[10] P. Subedi, J. Hao, I. K. Kim and L. Ramaswamy, ”AI Multi-Tenancy on Edge: Concurrent Deep Learning Model Executions and Dynamic Model Placements on Edge Devices,” 2021 IEEE 14th International Conference on Cloud Computing (CLOUD), 2021, pp. 31-42, Available: https://doi.org/10.1109/CLOUD53861.2021.00016

[11] Bandyopadhyay, D., Sen, J. Internet of Things: Applications and Challenges in Technology and Standardization. Wireless Pers Commun 58, 49–69 (2011). Available: https://doi.org/10.1007/s11277-011-0288-5

[12] S. Li, L. Xu and X. Wang, ”Compressed sensing signal and data acquisition in wireless sensor networks and Internet of Things”, IEEE Trans. Ind. Informat., vol. 9, no. 4, pp. 2177-2186, Nov. 2013. Available: https://doi.org/10.1016/j.comnet.2010.05.010

[13] F. Tao, Y. Zuo, L. D. Xu and L. Zhang, ”IoT-Based Intelligent Perception and Access of Manufacturing Resource Toward Cloud Manufacturing,” in IEEE Transactions on Industrial Informatics, vol. 10, no. 2, pp. 1547-1557, May 2014, Available: https://doi.org/doi:10.1109/TII.2014.2306397

[14] L. D. Xu, W. He and S. Li, ”Internet of Things in Industries: A Survey,” in IEEE Transactions on Industrial Informatics, vol. 10, no. 4, pp. 2233-2243, Nov. 2014, Available: https://doi.org/10.1109/TII.2014.2300753

[15] Sánchez López, T., Ranasinghe, D.C., Harrison, M. et al. Adding sense to the Internet

of Things. Pers Ubiquit Comput 16, 291–308 (2012). Available: https://doi.org/10.1007/s00779-011-0399-8

[16] S. Chen, Y. Zheng, W. Lu, V. Varadarajan and K. Wang, "Energy-Optimal Dynamic Computation Offloading for Industrial IoT in Fog Computing," in IEEE Transactions on Green Communications and Networking, vol. 4, no. 2, pp. 566-576, June 2020, Available: https://doi.org/10.1109/TGCN.2019.2960767

[17] Y. Zhang, J. -H. Liu, C. -Y. Wang and H. -Y. Wei, "Decomposable Intelligence on Cloud-Edge IoT Framework for Live Video Analytics," in IEEE Internet of Things Journal, vol. 7, no. 9, pp. 8860-8873, Sept. 2020, Available: https://doi.org/10.1109/JIOT.2020.2997091

[18] S. Takaki and J. Yamagishi, "A deep auto-encoder based low-dimensional feature extraction from FFT spectral envelopes for statistical parametric speech synthesis," 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016, pp. 5535-5539, Available: https://doi.org/10.1109/ICASSP.2016.7472736.

[19] Sarroff, Andy, and Michael A. Casey. "Musical audio synthesis using autoencoding neural nets." Proceedings of the International Society for Music Information Retrieval Conference (ISMIR2014). International Society for Music Information Retrieval, 2014.

[20] Y. Zhang et al., "Intelligent fault diagnosis of rotating machinery using a new ensemble deep auto-encoder method", Measurement, pp. 151, 2020. Available: https://doi.org/10.1016/j.measurement.2019.107232

[21] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with non-linear dimensionality reduction", Proceedings of the MLSDA 2014 2ndWorkshop on Machine Learning for Sensory Data Analysis, 2014. Available: https://doi.org/10.1145/2689746.2689747

[22] Z. Chen et al., "Evolutionary multi-objective optimization based ensemble autoencoders for image outlier detection", Neurocomputing, vol. 309, pp. 192-200, 2018. Available: https://doi.org/10.1016/j.neucom.2018.05.012

[23] Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets." Advances in neural information processing systems 27 (2014). Available: https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html

[24] J. Zhu, T. Park, P. Isola and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2242-2251, Available: https://doi.org/10.1109/ICCV.2017.244

[25] Vondrick, Carl, Hamed Pirsiavash, and Antonio Torralba. "Generating videos with scene dynamics." Advances in neural information processing systems 29 (2016). Available: https://proceedings.neurips.cc/paper/2016/hash/04025959b191f8f9de3f924f0940515f-Abstract.html

[26] T. Schlegl, P. Seebck, S. M. Waldstein, G. Langs and U. Schmidt-Erfurth, "f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks", Medical Image Analysis, vol. 54, pp. 30-44, 2019. Available: https://doi.org/10.1016/j.media.2019.01.010

[27] L. Deecke, R. Vandermeulen, L. Ruff, S. Mandt and M. Kloft, "Anomaly detection with generative adversarial networks", 2018, [online] Available: https://openreview.net/forum?id=S1EfylZ0Z.

[28] Li, D., Chen, D., Jin, B., Shi, L., Goh, J., Ng, SK. (2019). MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks. In: Tetko, I., Kůrková, V., Karpov, P., Theis, F. (eds) Artificial Neural Networks and Machine Learning – ICANN 2019: Text and Time Series. ICANN 2019. Lecture Notes in Computer Science(), vol 11730. Springer, Cham. Available: https://doi.org/10.1007/978-3-030-30490-4_56

[29] B. Zhou, S. Liu, B. Hooi, X. Cheng and J. Ye, "BeatGAN: Anomalous Rhythm Detection using Adversarially Generated Time Series", Proc. of the 28th Int. Joint Conf. on Artificial Intelligence (IJCAI), pp. 4433-4439, 2019. Available: https://doi.org/10.24963/ijcai.2019/616

[30] Yoon, Jinsung, Daniel Jarrett, and Mihaela Van der Schaar. "Time-series generative adversarial networks." Advances in neural information processing systems 32 (2019). Available: https://proceedings.neurips.cc/paper/2019/hash/c9efe5f26cd17ba6216bbe2a7d26d490-Abstract.html

[31] A. Geiger, D. Liu, S. Alnegheimish, A. Cuesta-Infante and K. Veeramachaneni, "TadGAN: Time Series Anomaly Detection Using Generative Adversarial Networks," 2020 IEEE International Conference on Big Data (Big Data), 2020, pp. 33-43, Available: https://doi.org/10.1109/BigData50022.2020.9378139