Comparative Study in Fault Detection under Data Scarcity for Pharmaceutical Device Manufacturing

Fatemeh Kakavandi[3], Peihua Han[2], Tim Brix Nerenst [3], Roger de Reus[3], Cláudio Gomes [1], Peter Gorm Larsen[1]
[1]*Department of Electrical and Computer Engineering, Aarhus University*, Aarhus, Denmark
[2]*Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology*, Ålesund, Norway
[3]*Finished Product Manufacturing Science and Technology Department, Novo Nordisk A/S*, Hillerød, Denmark
Email: fyka@novonordisk.com

***Abstract*—Developing a data-driven fault detection model for an actual manufacturing process poses significant challenges, particularly during the design phase, when accessing a large volume of data is not feasible. One approach to address data scarcity is the generation of synthetic datasets to aid in the training process. This paper focuses on conducting a comparative study between deep generative models and a simulation model for generating synthetic samples. Various deep generative models such as Variational Autoencoder (VAE) are explored and applied in this study. Additionally, the Finite Element Analysis (FEA) technique is employed to develop the simulation model, leveraging physics-based approaches. Furthermore, the performance of these generative techniques is evaluated and compared using qualitative and quantitative evaluation methods. The results demonstrate that different methods exhibit varying levels of performance based on qualitative and quantitative criteria. Eventually, the impact of synthetic samples generated by different approaches on the fault detection model is evaluated and reported. Synthetic normal and anomalous samples are added to the training set, and the model is tested solely on real data. The results indicate that majority of generative techniques enhance the performance of the fault detection model, requiring fewer training iterations and improving the model's predictive capabilities. Notably, synthetic data generated by VAE significantly improve the fault detection model's F1 score when only synthetic normal samples added to the training set by 0.185 in our case.**

***Index Terms*—Deep Generative Models, Finite Element Analysis, Synthetic Data, Fault Detection.**

I. Introduction

Detecting faults in manufacturing processes is crucial for maintaining product quality and production performance [1]. In pharmaceutical device manufacturing, traditional fault detection methods, such as rule-based approaches, have limitations as they increase process complexity and are only partially reliable due to the challenges posed by batch manufacturing processes [2]. The rise of machine learning (ML) models has offered more advanced solutions [3], but their effectiveness is heavily reliant on the volume and quality of available data [4].

Insufficient data quality and volume significantly impact the performance of complex ML models, compromising their precision and reliability, particularly in fault detection for industrial processes [4]. Addressing data deficiency is a major challenge during the design phase, prompting the exploration of solutions such as introducing synthetic samples to overcome this limitation [5]. Various ML and simulation-based techniques have been proposed to generate precise synthetic samples in response to data shortage challenges.

Deep generative models are powerful tools in ML, capable of understanding complex data distributions and probabilistic features, making them valuable for generating synthetic samples to address data scarcity [6].

Additionally, physics-based models, like FEA, offer distinct advantages in generating synthetic samples, especially in scenarios where accessing a large dataset with real data is impractical. The use of simulation models in the design phase requires less data for calibration and enables the exploration of various scenarios, thus enhancing the fault detection performance when synthetic data, enriched with physics-based knowledge, is used for training [7].

The paper's contribution comprises two interconnected parts. The first part involves a comparative study of deep generative and simulation models for creating synthetic datasets, evaluating the performance of three separate generative models for an industrial use case. Additionally, an FEA-based model is developed for this purpose to leverage physics-based knowledge. The second part investigates the impact of synthetic samples on the fault detection model by evaluating different generated datasets and comparing their effect on fault detection accuracy, ultimately reporting the model that provides the most assistance.

The organization of the paper is structured as follows: Section II presents previous work related to fault detection in pharmaceutical processes and the generation of synthetic samples. Section III describes the adopted methodological steps in generating and evaluating synthetic data and employing the data to assist the fault detection model. Section IV provides the method evaluation, introducing the industrial use case and presenting the related results for comparison. Finally, Section V concludes with the final results and findings.

II. Related work

Fault detection models play a critical role in identifying abnormal changes in a process plant, with methods typically categorized as model-based or data-driven, depending on the involvement of physical models [8].Data-driven methods, such as K-Nearest Neighbor (KNN) and Support Vector Machine (SVM), have been employed to classify batch process data, while Principal Component Analysis (PCA) has been used for fault detection in biopharmaceutical drug product manufacturing [9]. Despite the advancements, modern deep learning models have demonstrated significant progress in fault detection but often require large amounts of high-quality data, which may not always be available [10].

Deep generative models, including variational autoencoders (VAE) and Generative Adversarial Networks (GAN), have shown remarkable potential in generating synthetic data similar to the original dataset, thus increasing the size of the training dataset for various applications, such as image and text generation [11]. These models have been utilized to improve image classification accuracy and generalizability in tasks such as CT segmentation and surface defect detection of steel strips in manufacturing processes, thus enhancing fault detection accuracy with synthetic data [12]. Furthermore, deep generative models have also been explored in time series analysis, where models like TimeGAN have been developed to capture complex patterns and dependencies in time series data, showing promise in anomaly detection and machine fault diagnosis tasks [13].

Synthetic data from simulation models have been widely used for training ML models. Using simulated data has proven beneficial in training neural network models for machine fault diagnosis, especially when labeled fault condition samples were insufficient or unavailable, highlighting the potential of simulation in overcoming difficulties in fault diagnosis tasks [7]. However, challenges arise from the disparities between simulation and real-world scenarios, which requires careful use of simulated data and the application of transfer learning or domain adaptation techniques to bridge the gap between simulation and reality [14].

III. Methodology and Background

The overall methodological steps are depicted in Fig. 1, illustrating the generation of distinct synthetic datasets, their evaluation, and the impact on the fault detection model.

a) Generative models: A deep generative model tries to fit a model of the probability distribution $p(x)$ to a given a dataset $D = \{x_i\}_{i=1}^{N}$. Once fit, the model can be used to generate samples x. Note that here x is time series data. In this paper, we investigate three different deep generative models, VAE, GAN, and TimeGAN, to generate synthetic data for fault detection.

The VAE consists of an encoder $q_\phi(z|x)$ and a decoder $p_\theta(x|z)$ [15]. The encoder $q_\phi(z|x)$ and decoder $p_\theta(x|z)$ are neural network models parameterized by ϕ and θ, respectively. The encoder takes the input x and produces a distribution $p(z)$ where z is the latent vector. The decoder uses the latent vector to sample from $p_\theta(x|z)$ to produce x.

GAN involves two neural network models called generator and discriminator. The generator is able to draw samples from the distribution $p(x)$ by taking a vector z that serves as input to the generator function $f_{\theta_g}(z) \rightarrow x$. The input vector z can be thought of as a source of randomness which is typically represented as a standard Gaussian distribution. The discriminator examines a sample x and returns the estimate $f_{\theta_d}(x) \rightarrow (0, 1)$ of whether x is drawn from the training distribution $p(x)$ or from the generator.

TimeGAN proposed in [13] uses GAN to generate a low-dimensional representation h of data x instead of generating x directly. An autoencoder model with the embedding and recovery functions construct h. The embedding function $f_{\theta_e}(x) \rightarrow h$ maps input x to h while the recovery function $f_{\theta_r}(h) \rightarrow x$ maps h back to input x. The GAN architecture as described earlier is utilized to generate h, where a generator $f_{\theta_g}(z) \rightarrow h$ and a discriminator $f_{\theta_d}(h) \rightarrow (0, 1)$ is included. The latent vector z follows a standard Gaussian distribution.

b) Simulation model: To capture the responses encountered in production, geometrical and assembly variations are introduced, leveraging the Resilient Modeling Strategy proposed in [16] to create robust CAD models that can accommodate dimensional changes, as shown in Fig. 2. The input variation encompasses five geometrical and two assembly variables, selected based on their anticipated impact on the assembly force, and is determined using a Fractional Factorial design.

The FEA setup includes an explicit step, where the cap is permitted to translate vertically and is constrained in all other translations and rotations, with general contact applied along with Coulomb friction. The software 3DX (Abaqus solver) is utilized to manage the CAD model, conduct the simulation, and export the Force-time signals for each run. This approach allows for a detailed exploration of the significance of main factors without confounding with 2-level interaction effects, providing valuable insights into the assembly force and its relationship with input variations.

c) Evaluation Techniques: This paper employs qualitative and quantitative techniques to evaluate the introduced models. For qualitative evaluation, Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE) are used to map real and synthetic samples into a two-dimensional space for diversity assessment [17], [18]. In quantitative evaluation, discriminative and predictive scores are considered. The discriminative score gauges the fidelity of synthetic data by training a classifier model on real and synthetic samples. However, the predictive score assesses the usefulness of synthetic samples by ensuring the existence of temporal dependencies.

d) Fault detection model: The real dataset consists of normal and abnormal samples and to detect the abnormal samples, a fault detection model is developed. For this purpose, a deep learning model with convolutional layers is employed. Then, this model is trained on the combination of real and synthetic samples, and the impact of the generative techniques is evaluated.

IV. Industrial Case Study and evaluation

In this section, we evaluate the methodology explained in Section III with an industrial use case, a pharmaceutical device assembly process.

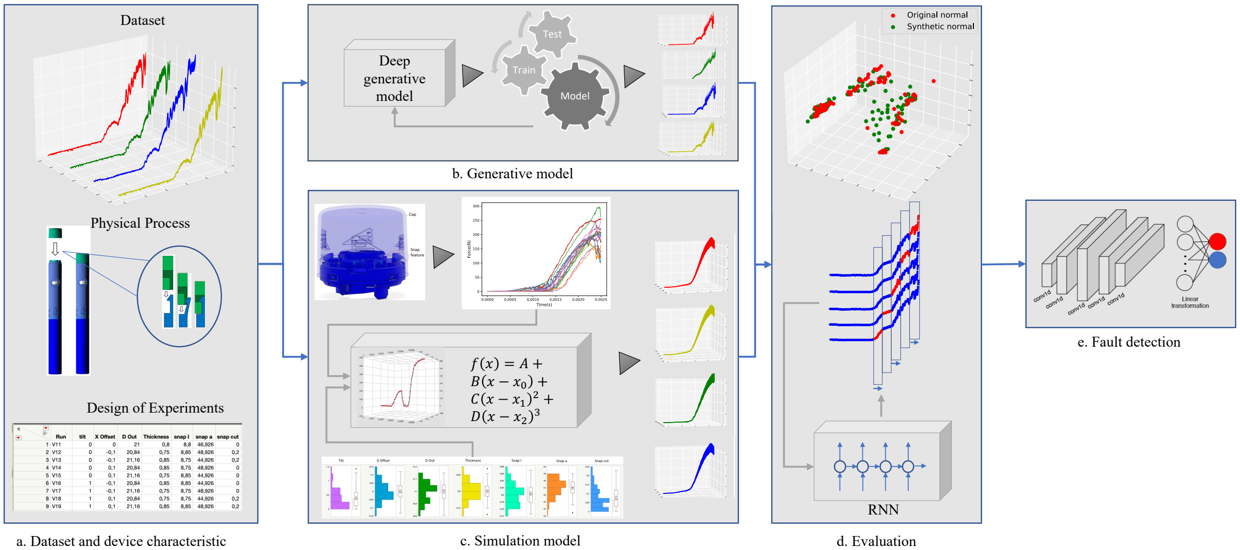


Fig. 1: Illustration of the overall diagram of the applied methodology.

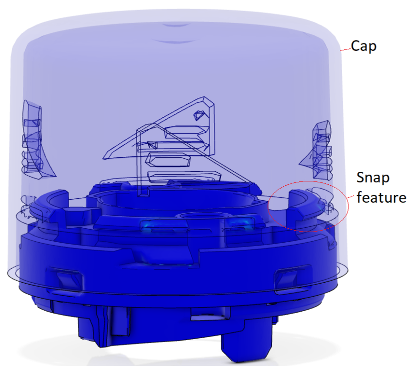


Fig. 2: Simulation geometry, highlighting the snap features which are generating the assembly force.

The process involves a linear motor as depicted in Fig. 3. This process entails mounting a medical device component and sub-assembly module, requiring linear downward movement, prior alignment, and rotational orientation. The assembly involves a segmented ring snap, shown in Fig. 2, where axial force is essential to bend the snap structure and overcome friction during assembly. The linear motor is equipped with a force transducer for continuous process monitoring, allowing for a comprehensive understanding of process quality. The force profiles of normal and abnormal experiments have been collected, where abnormal scenarios were induced separately [19], [20].

Despite limited availability, a dataset comprising 100 normal and a few abnormal samples from the pilot line has been collected. Recognizing the need for a larger dataset to train ML-based fault detection models effectively, synthetic samples are crucial to address these limitations and challenges.

a) Generative models results: The synthetic samples generated by generative models, are shown in Fig. 4. The

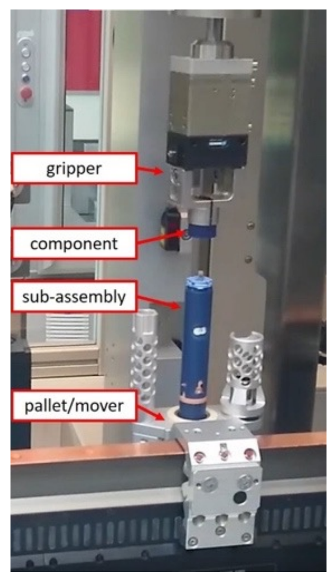
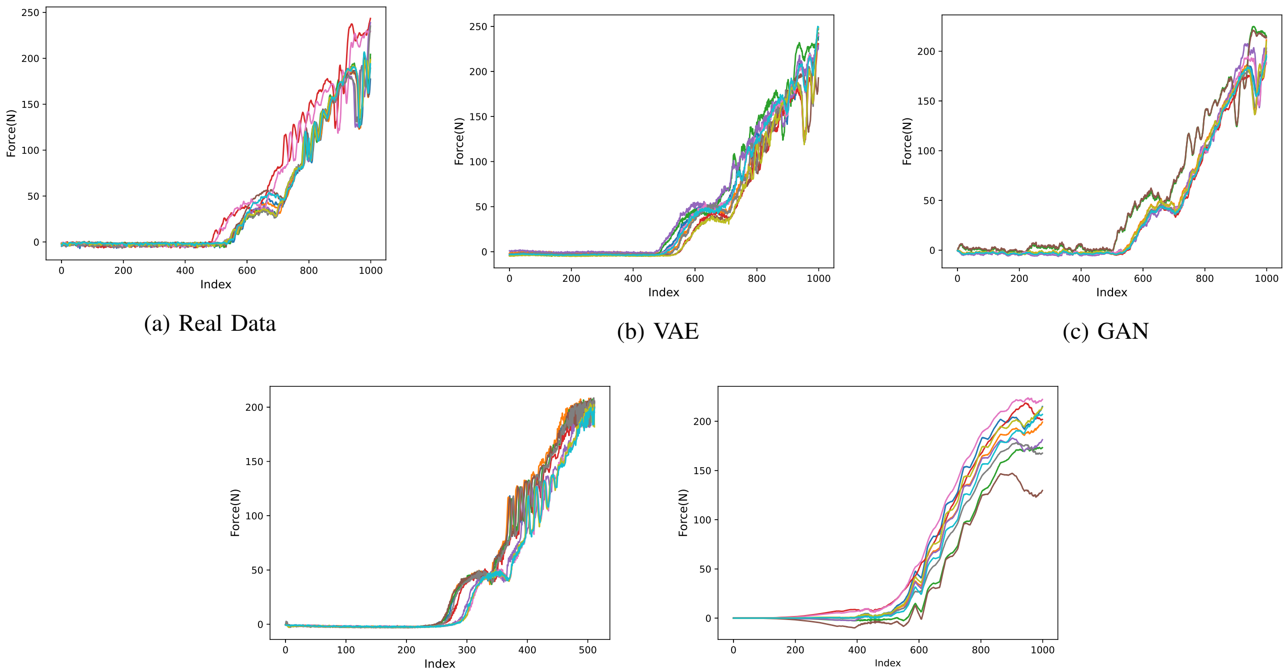


Fig. 3: Illustration of the physical system [19].

VAE model is trained on real samples to generate synthetic ones. It consists of linear transformation layers followed by the Rectified Linear Unit (ReLU) in the encoder and the Tanh transfer function in the decoder. After training for 5000 iterations on normal samples, VAE successfully generates synthetic data resembling real data patterns, as shown in Fig. 4(b).

The generator of the GAN shares the same structure as the VAE encoder; however, in this case, the input is the random sample, and the output is generated synthetic data. Then, the synthetic sample is fed to the discriminative model, where the structure comprises linear transformation layers followed by Leaky ReLU transfer functions. Trained with two different



(a) Real Data (b) VAE (c) GAN

(d) TimeGAN (e) FEA

Fig. 4: Synthetic data samples generated by different models from normal real data samples.

learning rates for 10000 iterations, GAN's synthetic samples after applying noise cancelling filters exhibit similarity with the actual data, as depicted in Fig. 4(c).

TimeGAN uses Gated Recurrent Unit (GRU) hidden layers and is trained with a learning rate of 10^{-3} for 7000 iterations. Together with a low-pass filter resulting synthetic data in Fig. 4(d) reveal resemblance to existing temporal patterns, with exceptions in capturing the force profile's decrease towards the process end.

b) Simulation model results: The FEA for the snap process is carried out via Abaqus software. With the Design of Experiment (DoE), the effect of different geometrical and assembly variations has been studied, resulting in different recorded force responses. The Force signals were then used to define a model where the DoE variables are considered input and the Force signal as output. Furthermore, using JMP Pro software version 16.0.0, a Spline function is fitted to the simulation data.

Afterwards, we generate input variables by changing the DoE variables within the accepted boundary. New sets of input samples are generated randomly with the normal distribution where the mean value for each variable is the nominal value which are then given as inputs to the snap-process formula (a Spline formula), thereby generating simulation based synthetic data. The synthetic data from the snap process formula is shown Fig. 4(e).

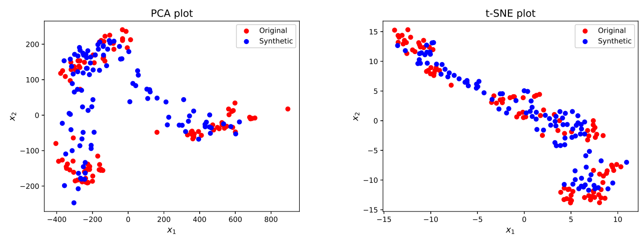
c) Evaluation of synthetic data: To evaluate the quality of synthetic data, we measure three different criteria: *diversity, fidelity, usefulness*. The evaluation process comprises assessing diversity to ensure the synthetic samples distribution encompasses real data. Additionally, fidelity evaluation aims to make synthetic samples indistinguishable from real ones. Finally, usefulness evaluation ensures that synthetic samples are as beneficial as real ones for predictive purposes.

Qualitative evaluation: The synthetic and real data distributions are visualized in a two-dimensional space using PCA and t-SNE, as shown in Fig. 5. The synthetic data generated by VAE, with both PCA and t-SNE methods, shows overlap with the actual data distribution according to Fig. 5(a). The highest overlap can ensure that no outlier data is introduced while using VAE. GAN-based synthetic data is adjacent to the actual data in some areas, and in some points, it has some distance from the real data points as shown in Fig. 5(b). The distance from the real data can stretch the actual data distribution boundary, leading to more generalization for future classification purposes. Comparatively, TimeGAN attempts to generate synthetic samples similar to the real data; however, it generates samples that can be considered outliers since the samples are far from the actual data, as shown in Fig. 5(c).

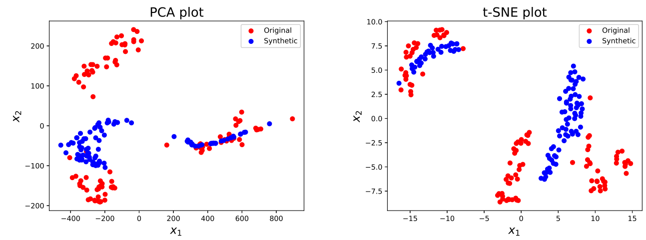
The simulation-based synthetic samples, however, only cover part of the data distribution, as expected, since the simulation model was conducted with the predefined setting and cannot replicate all the possible assembly scenarios as shown in Fig. 5(d).

Quantitative evaluation:

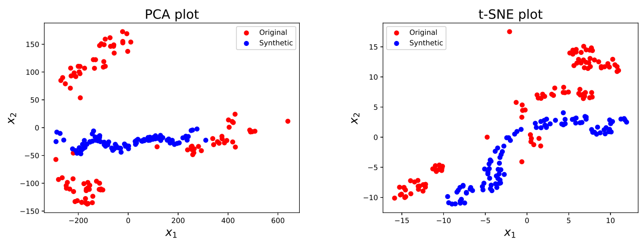
The fidelity criterion measures whether the artificial samples are distinguishable or not by calculating the discriminative score. A Recurrent Neural Network (RNN) model is trained



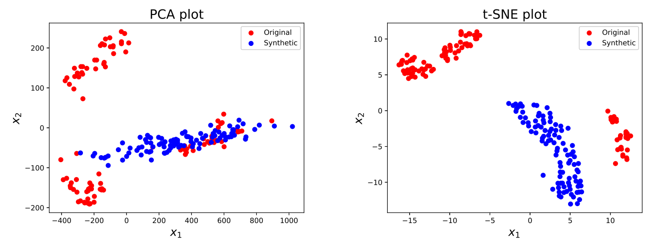
(a) VAE



(b) GAN



(c) TGAN



(d) Simulation

Fig. 5: The distribution of real and synthetic data in two dimension space using PCA and TSNE. x_1 and x_2 show the two dimensions of the reduced data.

to distinguish between real and synthetic samples. For this purpose, the original data is labelled *real*, and the synthetic samples are labelled *not real*.

The discriminative score, calculated as $|\text{classification accuracy} - 0.5|$, determines the distinguishability of the synthetic samples. When classification accuracy is 1, the discriminative score is 0.5, indicating easily detectable synthetic samples. Conversely, a classification accuracy of 0.5 results in a discriminative score of 0, indicating non-distinguishable synthetic samples. The discriminative scores are detailed in Table I

The discriminative results show that the synthetic dataset from the TimeGAN model has the lowest score, so it performs best and is most indistinguishable. However, all the methods for generating synthetic samples are well suited for creating synthetic datasets from the fidelity point of view due to their low discriminative score.

To evaluate the usefulness of synthetic time series, a sequence prediction model (2-layer LSTM) is trained on the synthetic dataset to calculate the predictive score. This model is evaluated on the real data, and the performance is measured in terms of the mean absolute error (MAE). However, to compare different models used for generating synthetic samples, we report the relative MAE where we standardize the MAE values between $[0, 1]$.

According to Table I, GAN has the lowest predictive error, i.e. higher predictive performance. TimeGAN and simulation model slightly perform better than VAE in case of usefulness. However, the initial MAE values are close for all different models.

Furthermore, the reason for the performance of the simulation-based synthetic data is that real and simulation force signals differ around index 600 according to Fig. 4(a)

TABLE I: Quantitative evaluation of Synthetic data generated from different models.

Model	Discriminative score	Predictive Score
VAE	0.082	1
GAN	0.082	**0**
TimeGAN	**0.0516**	0.854
Simulation Model	0.0944	0.96

and Fig. 4(e). Where the real data shows a bump, which is not captured in the simulation signals, the difference could indicate a physical phenomenon on the production line that is not captured accurately in the simulation model.

d) Fault detection results: To detect abnormal samples, we train a Fully Convolutional Network (FCN) with a combination of real and synthetic datasets. We use 80% of the normal real data for training and 20% for testing the FCN model. The FCN model consists of three layers with dimension of $[256, 128, 64]$. The learning rate for this model is 10^{-5}.

We examine two specific scenarios referred to case 1 and case 2. In the case 1, both real and synthetic normal datasets, along with four real abnormal samples, are employed to train the fault detection model. However, in the second case, we incorporate synthetic abnormal samples generated using various techniques into the training process of the FCN model.

Finally, to evaluate the fault detection model's performance on test data (consisting of a combination of 20% of real normal data and the latest real abnormal samples), we employ the macro F1 score criterion. The macro F1 score is calculated as the unweighted average of the F1 scores computed for each class. The F1 score itself is derived using the formula: $\frac{\text{TP}}{\text{TP}+\frac{1}{2}(\text{FP}+\text{FN})}$, where TP represents true positive. To ensure

TABLE II: Fault detection model performance with the assistance of different synthetic datasets.

Model	Case 1 (mean/std)	Case 2 (mean/std)
Only Real Data	(0.67, 0.108)	-
VAE	**(0.855, 0.056)**	**(0.746, 0.134)**
GAN	(0.704, 0.076)	(0.652, 0.095)
TimeGAN	(0.63, 0.127)	(0.603,0.092)
Simulation Model	(0.673,0.079)	-

reproducibility, we run the models for 100 iterations and report the mean and standard deviation values for the macro F1 score in Table II.

As shown in Table II, in case 1, synthetic data generated through VAE and GAN and simulation-based model, enhance the performance of fault detection model. Among these techniques, VAE demonstrates the greatest improvement, primarily due to the higher similarity between the data generated by VAE and real data.

In Case 2, only the data generated by VAE contributes to enhancing the fault detection performance. However, it should be noted that since the deep generative models utilized in this study still heavily rely on data volume, their performance is hindered by the limited availability of abnormal samples. Consequently, the synthetic abnormal samples themselves do not yield an improvement in the fault detection performance

V. Conclusion

Addressing data scarcity involves generating synthetic samples and augmenting the training set. This study compared various deep generative models and a simulation model for synthetic dataset generation. Three deep generative models — VAE, GAN, and TimeGAN — were assessed for their industrial applicability. Additionally, a simulation model based on finite element modeling was developed to create diverse scenarios.

The qualitative analysis revealed that VAE achieved a higher score due to the better overlap between the generated synthetic samples and the distribution of real data. This suggests that VAE can accurately estimate the parameters of the real data distribution, such as mean and standard deviation. Conversely, the quantitative results indicated that the synthetic data generated by TimeGAN had a lower discriminative score, implying that it was less distinguishable. The predictive score favored the synthetic data generated by GAN, suggesting that GAN was effective in preserving the temporal dependencies present in the real data.

The fault detection model results indicated that VAE significantly improved the model's performance, especially when only normal synthetic samples were used. However, generating abnormal synthetic samples with limited data posed challenges. In conclusion, VAE proved effective in closely estimating real data parameters and enhancing fault detection.

Acknowledgment

We express our sincere gratitude to Novo Nordisk A/S for providing the physical system, conducting meticulous data collection. Furthermore, we are deeply grateful to Innovation Foundation Denmark for generously funding the MADE FAST project, of which this research forms an integral part.

References

[1] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—part i: Fault diagnosis with model-based and signal-based approaches," *IEEE transactions on industrial electronics*, vol. 62, no. 6, pp. 3757–3767, 2015.

[2] D. Miljković, "Fault detection methods: A literature survey," in *Proceedings of the 34th International Convention MIPRO*, 2011, pp. 750–755.

[3] M. Fernandes, J. M. Corchado, and G. Marreiros, "Machine learning techniques applied to mechanical fault diagnosis and fault prognosis in the context of real industrial manufacturing use-cases: a systematic literature review," *Applied Intelligence*, vol. 52, no. 12, pp. 14246–14280, 2022.

[4] A. Paleyes, R.-G. Urma, and N. D. Lawrence, "Challenges in deploying machine learning: a survey of case studies," *ACM Computing Surveys*, vol. 55, no. 6, pp. 1–29, 2022.

[5] S. I. Nikolenko, *Synthetic data for deep learning*, vol. 174, Springer, Cham, 2021.

[6] A. Oussidi and A. Elhassouny, "Deep generative models: Survey," in *International Conference on Intelligent Systems and Computer Vision (ISCV)*, 2018, pp. 1–8.

[7] M. Xia, H. Shao, D. Williams, S. Lu, L. Shu, and C. W. de Silva, "Intelligent fault diagnosis of machinery using digital twin-assisted deep transfer learning," *Reliability Engineering & System Safety*, vol. 215, pp. 107938, 2021.

[8] P. Han, A. L. Ellefsen, G. Li, F. T. Holmeset, and H. Zhang, "Fault detection with lstm-based variational autoencoder for maritime components," *IEEE Sensors Journal*, vol. 21, no. 19, pp. 21903–21912, 2021.

[9] G. Gins, P. Van den Kerkhof, J. Vanlaer, and J. F. Van Impe, "Improving classification-based diagnosis of batch processes through data selection and appropriate pretreatment," *Journal of Process Control*, vol. 26, pp. 90–101, 2015.

[10] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5990–5998, 2017.

[11] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[12] A. Antoniou, A. Storkey, and H. Edwards, "Data augmentation generative adversarial networks," *arXiv preprint arXiv:1711.04340*, 2017.

[13] J. Yoon, D. Jarrett, and M. van der Schaar, "Time-series generative adversarial networks," in *Advances in Neural Information Processing Systems*, 2019.

[14] J. Huang, J. Wen, H. Yoon, O. Pradhan, T. Wu, Z. O'Neill, and K. S. Candan, "Real vs. simulated: Questions on the capability of simulated datasets on building fault detection for energy efficiency from a data-driven perspective," *Energy and Buildings*, vol. 259, pp. 111872, 2022.

[15] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[16] Gebhard R., "'A resilient modeling strategy' - Technical presentation," Tech. Rep., Solid Edge University, 2013.

[17] A. Maćkiewicz and W. Ratajczak, "Principal components analysis (pca)," *Computers & Geosciences*, vol. 19, no. 3, pp. 303–342, 1993.

[18] G. E. Hinton and S. Roweis, "Stochastic neighbor embedding," *Advances in neural information processing systems*, vol. 15, 2002.

[19] F. Kakavandi, C. Gomes, R. de Reus, J. Badstue, J. L. Jensen, P. G. Larsen, and A. Iosifidis, *Towards Developing a Digital Twin for a Manufacturing Pilot Line: An Industrial Case Study*, pp. 39–64, Springer Nature Singapore, Singapore, 2023.

[20] F. Kakavandi, R. de Reus, C. Gomes, N. Heidari, A. Iosifidis, and P. G. Larsen, "Product quality control in assembly machine under data restricted settings," in *IEEE 20th International Conference on Industrial Informatics (INDIN)*, 2022, pp. 735–741.