I.J. Information Engineering and Electronic Business, **2025, 6, 71-80**
Published Online on December 8, 2025 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijieeb.2025.06.06

Synthetic Data Generation Using Generative Adversarial Networks for Enterprise Application

Sreevishnu A. B.
SAP Labs India Pvt. Ltd., Bangalore, 560066, India
E-mail: sreevishnu.a.b@sap.com
ORCID iD: https://orcid.org/0009-0006-9297-4559

Suman De*
SAP Labs India Pvt. Ltd., Bangalore, 560066, India
E-mail: suman.de@sap.com
ORCID iD: https://orcid.org/0000-0002-0438-2638
*Corresponding Author

Received: 14 November, 2024; Revised: 24 February, 2025; Accepted: 21 June, 2025; Published: 08 December, 2025

Abstract: This research tackles the fundamental requirement of synthetic data generation to tighten up machine learning model precision and one-shot shot learning to lessen the need to pursue data input. The project aims to develop a service that can generate reasonable synthetic data from a given dataset. It was first designed and developed, and then the project structure was set, and libraries were chosen for predevelopment analysis. This continued development process also included subsequent phases that included dataset collecting, assessment, and iterative research. Different hyperparameters were run over multiple models to select an optimal configuration. To evaluate the model's performance over produced synthetic datasets, about 1.5 and 2 times the original, synthetic data was produced, providing a basis for a robust synthetic data generating process.

Index Terms: Generative Adversarial Network, Machine Learning, One-shot Learning, Python, Synthetic Data Generation, Enterprise Applications

1. Introduction

Synthetic data holds immense promise in the development of business software. Significant progress has been made in wireframing and chatbots with Generative Adversarial Networks (GANs), however, using Generative Adversarial Networks (GANs) for generating synthetic data for business machine learning models has been less explored. With that said, this research identifies a specific opportunity for the corporate software industry to fill a gap of utilizing GANs due to the limited availability of training data. Today, the machine learning models in this domain still heavily depend on pre-processed client data, which in most cases turns out to be insufficient to adequately train those models resulting in models with poor performance and therefore low market value. This paper proposes an innovative method for synthesizing synthetic data whose properties closely resemble those of true customer data, especially for business types of applications. They introduce a 'Bring Your Own Data' model whereby users can train and synthesize data according to their needs while incorporating metrics for assessing how 'close' their data is to the ground truth. The research focuses on improving the operational efficiency of data creation procedures in industrial applications using GAN for synthetic data generation.

The main contributions of this research are to adapt GANs to domain-specific scenarios, e.g., Digital Twins specialized in the format of Industrial Internet of Things (Industrial IoT) Asset Administration Shell (AAS) through the introduction of AASGAN. [1] This work addresses a key missing piece of software engineering methodology for systems where synthetic data is generated for enterprise solutions and pioneers new techniques for improving machine learning algorithms in business software. In recent times, interest in the development of synthetic data has grown markedly in areas where data privacy and access constraints impede progress. Although GANs have the potential to generate synthetic data for many applications, the extent of GAN capability in enterprise settings remains uncertain. The primary focus of current literature is on general benefits such as improved model correctness and performance, while the few exceptions provide only partial guidance towards principled ways of combining multiple learning algorithms. However, these claims are often vague and case comparisons; the evidence for the practical success of GAN

This work is open access and licensed under the Creative Commons CC BY 4.0 License.

based synthetic data is not necessarily straightforward.

The specific issues faced when employing GANs as synthetic data generating tools for workload generation on enterprise applications are solved in this research. Specifically, it addresses the key data scarcity, privacy, and domain specific data production challenge. The purpose of this study is to show through de ned metrics and case studies how GANs can be used to create high quality synthetic data aligned with organization requirements in order to perform essential business operations and decision-making processes.

2. Literature Survey

By using synthetic data generation, it is possible to create a network of original equipment manufacturers (OEMs) that serves as the foundation for the network. It is necessary to possess a comprehensive understanding of many methodologies for generating synthetic data. A variety of open-source software solutions, like Synthetic Data Vault (SDV), facilitate the generation of such data. [2] Furthermore, within the realm of intelligent manufacturing, machine learning models and algorithms have been used to create computational models inside data-driven frameworks for digital twins. [3] Recently, Gaussian neural networks (GANs) have relocated their emphasis from image formation to the production of tabular data. Extensive research has been conducted on generative adversarial networks (GAN) in the domains of engineering and computer science, and several practical applications have been implemented. A diverse array of general-purpose network (GAN) designs, including as CycleGAN, DCGAN, and TableGAN [4], are widely used in research undertaken by academic institutions and private commercial organizations. These structures offer a robust foundation that enables the production of synthetic data. [5] Within the framework of an Industry 4.0 scenario, this study proposes an innovative approach to generate artificial data for software systems that directly engage with a network of original equipment manufacturers (OEMs). The method leverages the robust foundation provided by GAN for data collecting, along with the continuous development of Digital Twins and the achieved standards.

Digital Twins are used to manufacture digital replicas of tangible objects in the physical world. The field of Synthetic Data Generation has received little research attention, despite its importance for corporations to effectively address challenges in the Automotive Industry and Industry 4.0. The present work presents a novel concept named AASGAN. It employs the knowledge model of a Digital Twin data via an Asset Administration Shell, using a technique for generating synthetic data known as Generative Adversarial Network (GAN) to produce data that closely resembles authentic data. The significance lies in the existence of a recognized research deficiency regarding the standardization of the knowledge model of Digital Twins and the generation of fabricated data for a network of manufacturers, often referred to as Original Equipment Manufacturers (OEMs).

This project attempts to leverage GANs as opposed to transformer-based models like GPT – 3, etc. This enables data to be generated without the need to “teach” a model progressively. It intends to use data to provide sufficiently similar synthetic data in reasonable spans of time. The project also focuses on enterprise application over general purpose data, thereby breaking new ground.

3. Proposed Idea

The solution consists of mainly two parts:

- The UI, and
- The backend service.

The process of data generation can often be time-consuming. So, both components are designed with asynchronous process flow in mind. The UI is the user-facing component which allows upload of data, tweak metadata of source data, checking status of running generation jobs, and displaying metrics of generated data. In order to start a data generation job, the user uploads a data, which is the only mandatory input. However, the solution also accepts two optional inputs – hyperparameters and categorical columns. In case neither are provided, a default value will be used for hyperparameters (which are epochs and retention period, at the moment) and the solution will infer the kind of columns which the user can the edit in the UI.

The backend service comprised of 5 components:

- API server
- GAN worker
- Cleanup worker
- Database
- Filesystem

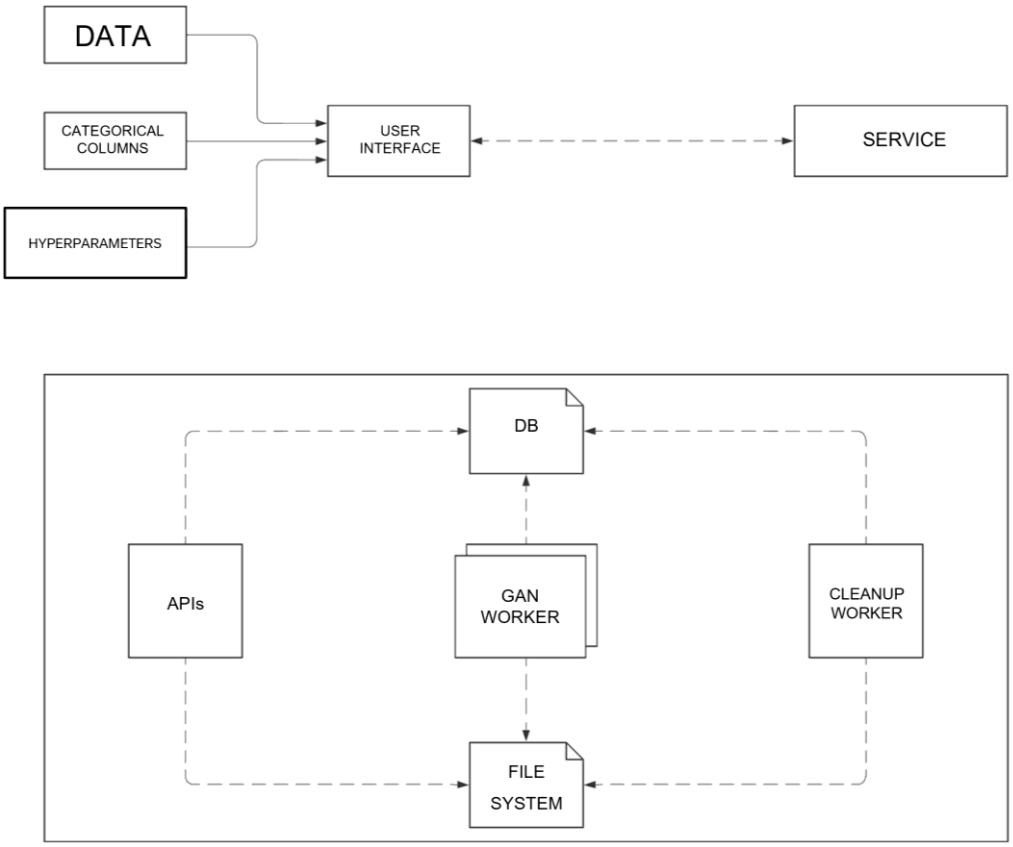

Fig. 1. Solution Architecture of the proposed Idea

The API Server exposes endpoint to the frontend UI for all the functionalities, which include:

- Data management (creation, deletion, and updating)
- Job management (deletion of job metadata, model, and metrics)

API for data management allows users to delete source data, generated data, or both. Job management API allows the deletion job information persisted in the service like start time, duration, etc. as well as the metrics for a particular job. Metrics and models are associated with a particular job as opposed to a particular dataset as the same data can be used to run multiple jobs with different hyperparameters.

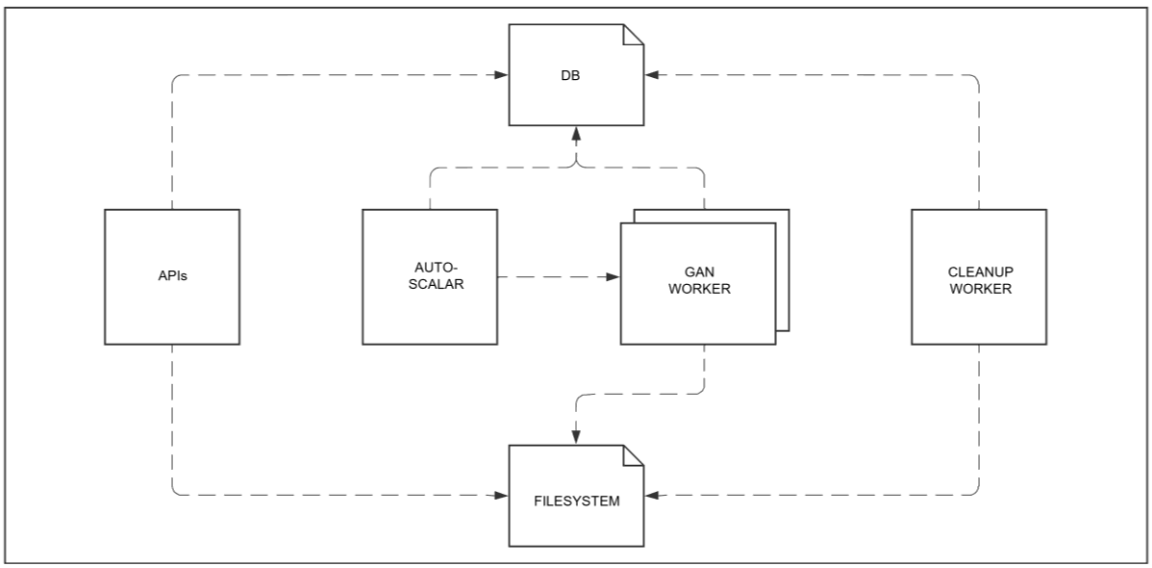

Fig. 2. Alternative Architecture where GAN agents are scaled horizontally based on several requests.

The GAN worker runs the actual training job and creates all the output artefacts, like output data, metrics, etc. The number of GAN worker instance can be scaled up and scaled down, if needed, although it is preconfigured in the above architecture. The cleanup worker runs periodic jobs to clean up all entities, uploaded or generated, to ensure that no data is retained for longer than is needed. The cleanup worker checks the retention period against the time creation and cleans up dataset and model that have breached their user-provided or default retention period.

The service's database acts as the single source of truth that binds all the above components together. Each create request to the API adds a new entry in the table, which is then picked up by the GAN workers. The cleanup worker also periodically reads and cleans up expired entries. The filesystem stores the artefacts uploaded and generated. For the sake of simplicity, a server's native filesystem is used here.

The above diagram depicts an alternative architecture where GAN workers are scaled horizontally based on several requests that need to be processed. It involves a fourth component, the Auto-scaler, which communicates with the database and depending on the volume of unprocessed jobs scales up and assigns jobs to the GAN workers. Due to time constraints, the previous architecture has been chosen.

4. Implementation Resources

The testing setup and production environment of this work used different technologies and tools. The experiments were conducted on a UNIX-based operating system, such as Ubuntu or macOS, which facilitated the management of processes, tasks, and system resources. These elements are crucial for effectively handling high-performance computing jobs, particularly those involving deep learning and data-intensive procedures. The research used a system equipped with 8 GB of RAM, sufficient for managing medium-sized datasets and training models using the CT GAN library. Although the memory capacity was limited, it proved sufficient to execute the applications without significant memory issues, particularly when using effective memory management techniques. The computer was equipped with an octa-core CPU, enabling it to process files simultaneously and expedite program execution. This configuration greatly facilitated the acceleration of model training, particularly during the pre-processing and execution phases, therefore enhancing the overall efficiency and speed of the process.

Python served as the primary programming language utilized throughout the project. The CT GAN library, along with its suite of machine learning tools including TensorFlow, Keras, and PyTorch, facilitated the construction and evaluation of many generative adversarial network (GAN) models. To ensure consistent functionality across various operational environments, the application was "containerized" using Docker. This enabled the team to partition the whole application into individual containers, including all of its dependencies, frameworks, and configurations. This reduced the likelihood of compatibility issues and facilitated the implementation and expansion stages. The Kubernetes framework was used to systematically arrange and manage the configuration of Docker containers within a cluster. Kubernetes facilitated the seamless expansion of the service and efficient management of resource sharing. This ensured optimal use of CPU resources for executing demanding GAN training algorithms. Google Colab, a cloud-based platform, was used for the construction and evaluation of models. The provision of GPU and TPU resources at no cost significantly accelerated the training process of deep learning models such as CT GAN. This reduced the duration of training and enhanced processing efficiency in comparison to operating on local hardware with restricted resources. Creating and refining code was largely carried out in Visual Studio Code, the primary integrated development environment (IDE). The development process was facilitated by the presence of several add-ons dedicated to Python, Docker, and Kubernetes, which integrated the techniques for development, testing, and release into a single tool. Furthermore, the software had user-friendly remote programming capabilities that enabled the execution of code directly on cloud platforms such as Google Colab. This configuration ensured a convenient and efficient method for conducting tests, supporting seamless development cycles, and optimizing the use of hardware and cloud-based resources.

5. Results & Discussions

At the time of writing, steps 1 through 3 have been completed, with still being refined upon. Step 4 is in progress with sufficient progress being made in the initial evaluation of performance of model created using 100% source data vs 50% source data + 50% synthetic data vs 40% source data + 60% synthetic data vs 34% source data + 66% synthetic data. The results are as follows in Fig. 3:

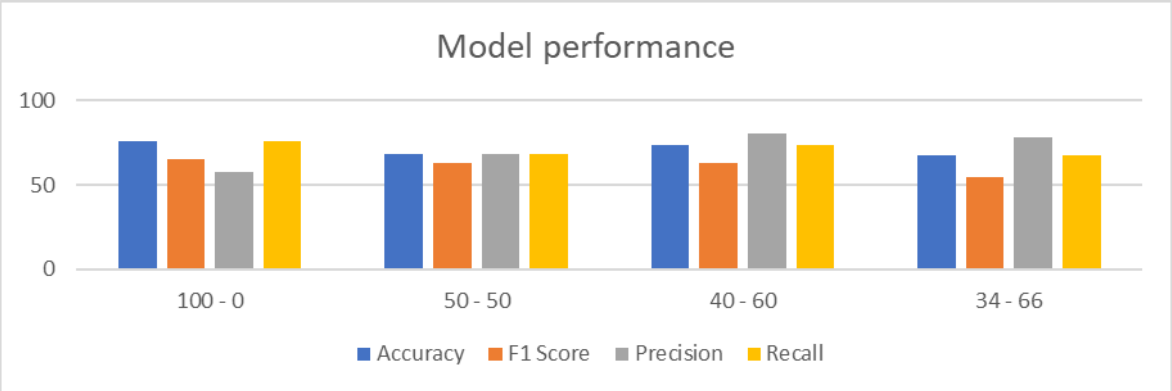

Fig. 3. Model Performance

Major challenges were observed as part of the results which can be explored as follows:

- Model Training Complexity: Training GANs is a trap deeply rooted in complexity, computational resources, and expertise. A major problem is that we must balance the generator and discriminator during training to prevent things like mode collapse.
- Finding quality datasets: Good-quality datasets that provide reasonable performance without much tweaking were hard to come by, especially since the focus domain is industrial data.
- Constant changes to the library: This project leveraged a library called CTGAN for GAN creation and data generation. During the initial phase, the library was still in 0.x.x releases. So, there were constant API changes, some of which were incompatible with the previous version, requiring significant amounts of time for rewrites.

As part of the research, potential risks and their mitigation techniques are covered that require consideration during integration with any software solutions:

- GDPR-compliance: Handling data containing personally identifiable data can be a legal issue. This is mitigated using a retention period, after which data is cleaned. Furthermore, data can be deleted on demand as well.
- Resource management: If too many training jobs run in parallel, there might be a chance that the system will run out of resources, blocking the whole flow. Mitigated by limiting the number of jobs that can run concurrently.

It is shown that real and synthetic are strongly related, not indifferent overall data variances. Generator, discriminator, and auxiliary losses are combined with the loss function of CTGAN totals. [13] Iterative update of both generator and discriminator networks, with gradient-based optimization based on reducing their respective losses striving for equilibrium. The generator loss is a sum of Adversarial and Reconstruction loss. In CTGAN, the loss functions for generating synthetic data can be articulated as follows:

$$l_D = \frac{1}{m}\sum_{i=1}^{m}[D(x'^{(i)}) - D(x^{(i)})] \quad (1)$$

$$l_G = -\frac{1}{m}\sum_{i=1}^{m}[D(x'^{(i)})] + H \quad (2)$$

While equations (1) and (2) say x is actual data and x′ is bogus data, x represents what we observed and x' represents what we do not. The output D(x) corresponds to the output of the Discriminator in real data, and the output D(x') for synthetic data. This typically advantageous h is equal to the cross-entropy score. This is juxtaposed with distributed real and artificial data using the table_evaluator Python module, [14, 15] where the slope of the graphs shows the disparity or similarity between the distributions of real and generated data during the distribution comparisons. Several statistical methods are used to determine the slope, i.e., the relationship between the x-axis (attribute values) and y-axis (frequency or count). Comparison of source data against each of 600, 900, and 1200 rows of generated synthetic data for this research work is shown as follows in Fig. 4, Fig. 5, and Fig. 6:

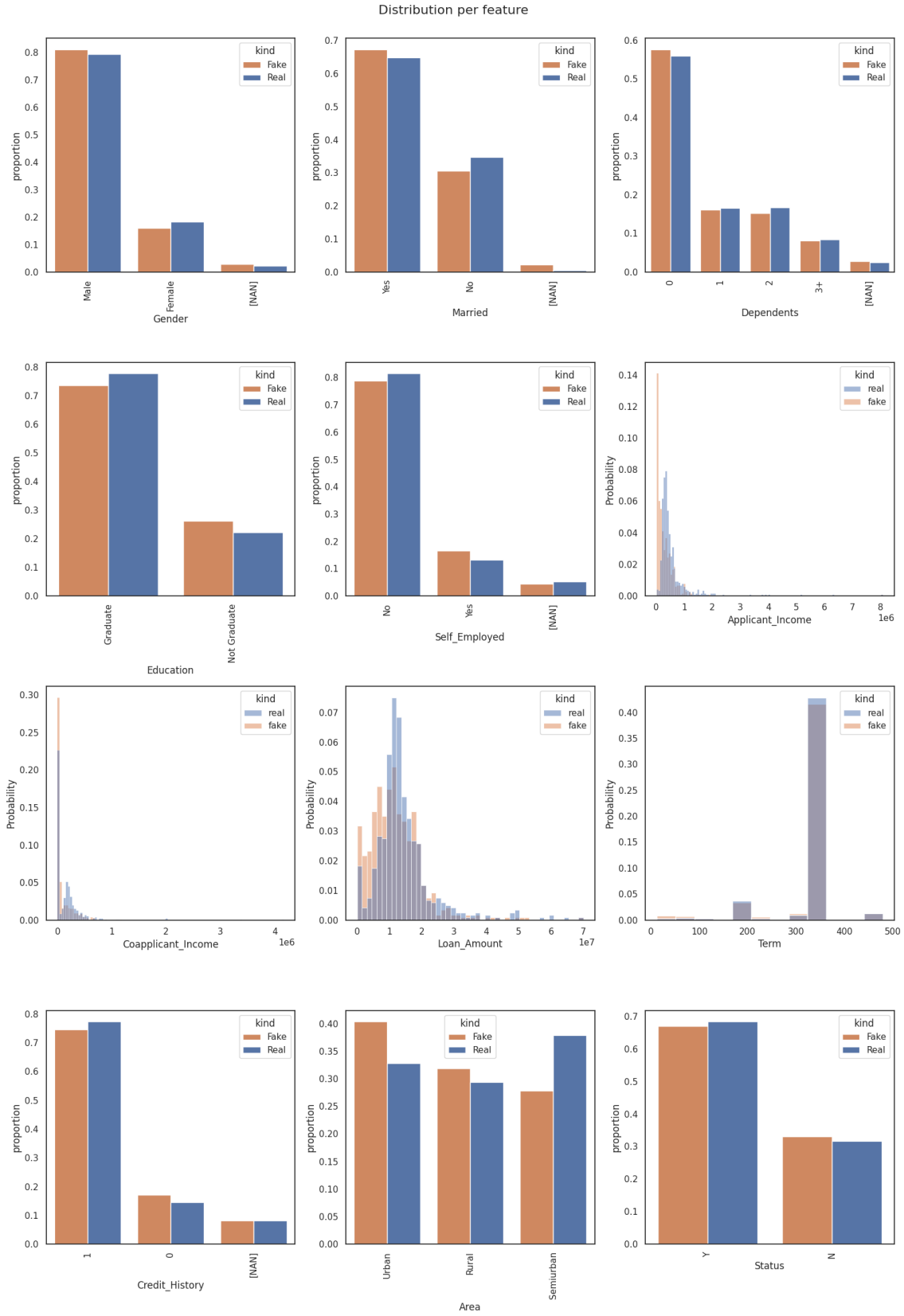

Fig. 4. Comparison of source data against each of 600 rows of synthetic data

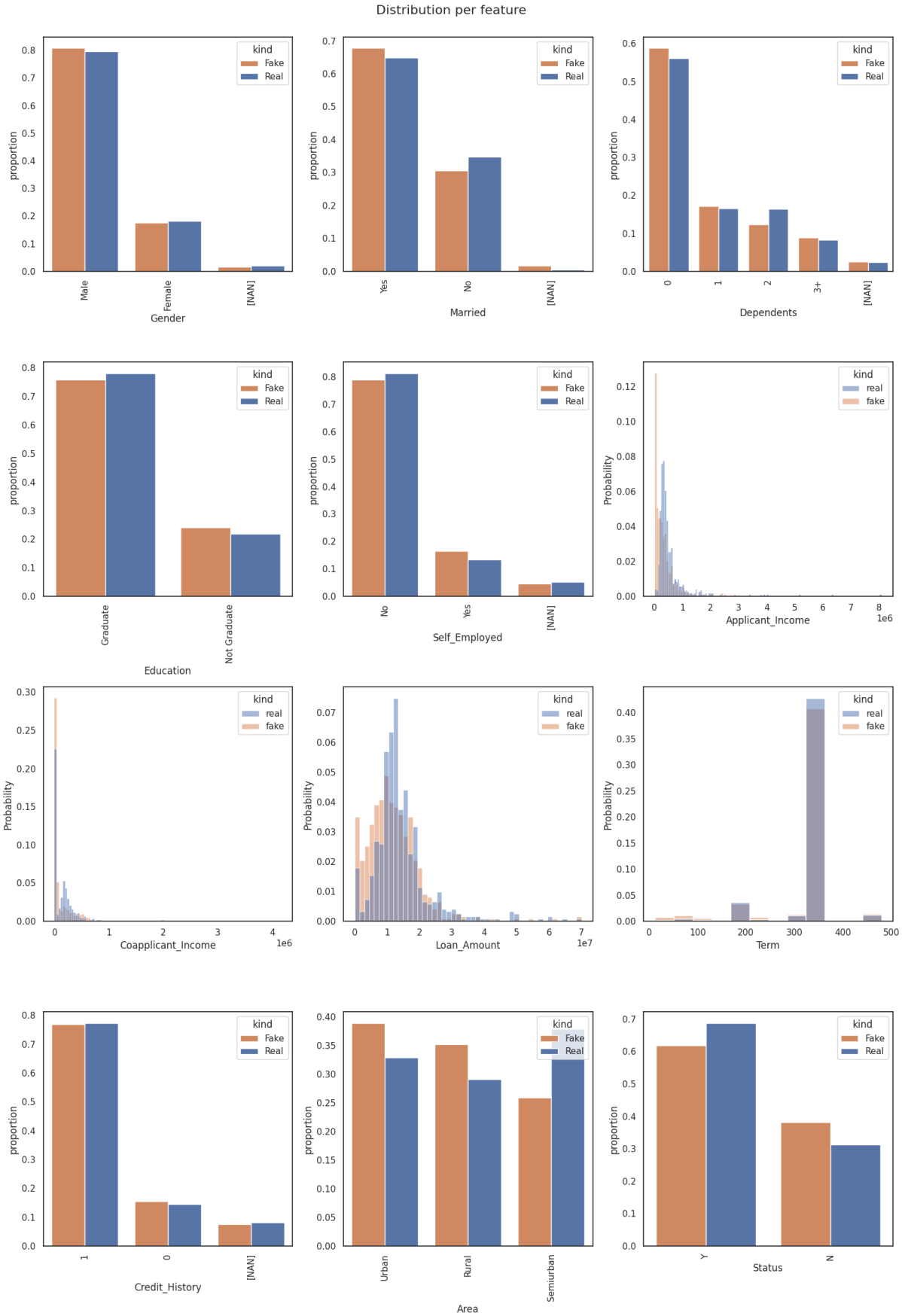

Fig. 5. Comparison of source data against each of 900 rows of synthetic data on more attributes

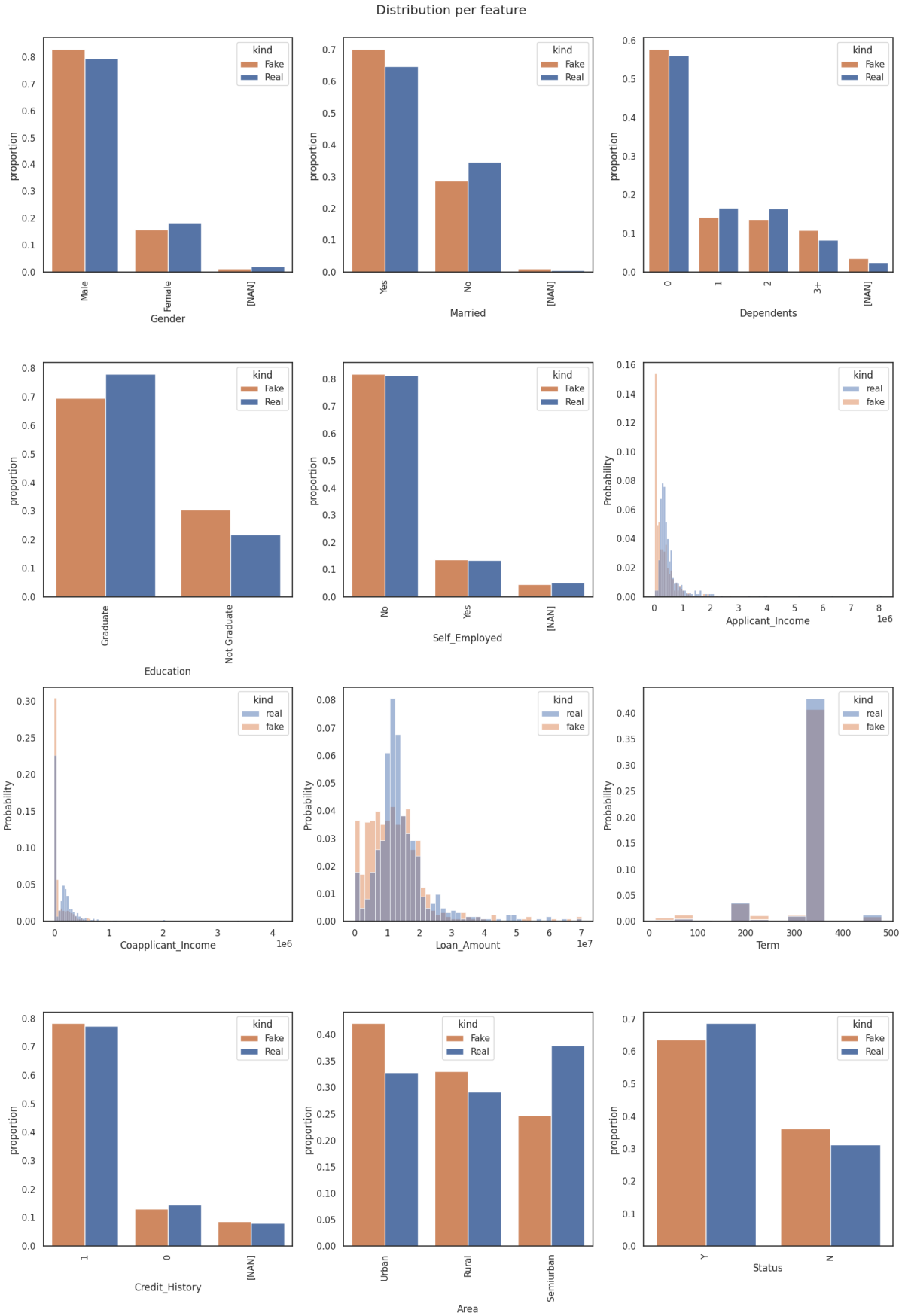

Fig. 6. Comparison of source data against each of 1200 rows of synthetic data on more attributes

6. Conclusion

Generative adversarial networks, often known as GAN, are a cutting-edge approach to machine learning that has a wide range of applications across a variety of fields. The usefulness of Generative Adversarial Networks (GANs) in extrapolating data from a very small sample size has been shown by the GANs on their own. Other Machine Learning (ML) models, such as BERT and Large Language Models, are used to improve and expand the capabilities of Generative Adversarial Networks (GANs). When these models are combined, the creation of data via the use of Generative Adversarial Networks (GANs) has the potential to demonstrate even greater efficacy. This also involves a few challenges as discussed in the previous sections but overall provides several benefits to applications of different industries. Industries like Aerospace, Architecture, Agribusiness, Banking, Aerospace, Chemicals, Consumer Products, Construction, Real Estate, Automotive, Aerospace, Power and Energy, Aerospace, Infrastructure, Aerospace, Retail, Aerospace, Healthcare, Aerospace, High Technology, Aerospace, Defense and Security, Aerospace, Fashion, Aerospace, Utilities, Aerospace, Sports and Entertainment, Aerospace, Travel, and Aerospace, Material and Energy, that would have potential application in their digital twins to solve the need for synthetic data. The domains of Sustainability and Human Resource Management contain domains in which data of real-life things are replicated, or should be, and considered; synthetic data is developed for this, and the suggested concept is important.

The application that was developed as a part of this research has the potential to be enhanced in the future to incorporate the characteristics that were specified above, provided that a significant amount of work and financial resources is invested. This application has the potential to become a viable option to produce synthetic data that more closely mimics the data that is being collected.

7. Future Works

Within the field of generative artificial intelligence, both Language Models (LLMs) and Generative Adversarial Networks (GANs) are regarded as subcategories. The incorporation of Language Models (LLMs), such as GPT, along with other language models like BERT and USE, can significantly enhance the generation of synthetic data by harnessing Generative Adversarial Networks (GANs). Generational Adversarial Networks (GANs) may be utilized and combined with language models to improve their performance and enable language models to boost their overall capabilities. The advancement of generative artificial intelligence can transform data production while simultaneously mitigating the associated risks. The incorporation of Language Models (LLMs) in this project will greatly enhance the current capabilities of Generative Adversarial Networks (GANs).

Acknowledgment

This paper was possible thanks to the support of our parent organization, SAP Labs India. We would like to thank techNxt for creating a research community for budding authors and our managers and colleagues for extending their support in our research journeys.

References

[1] S. De and P. Mitra, "A Novel Technique of Synthetic Data Generation for Asset Administration Shells in Industry 4.0 Scenarios," in IEEE Transactions on Artificial Intelligence, doi: 10.1109/TAI.2024.3409516.

[2] Synthetic Data Vault (SDV), Version 0.17.2, Jan 24, 2023, Available: https://sdv.dev/SDV/index.html, Last Accessed: 29.01.2023.

[3] Friederich, J., Francis, D. P., Lazarova-Molnar, S., & Mohamed, N. (2022). A framework for data-driven digital twins of smart manufacturing systems. Computers in Industry, 136, 103586. https://doi.org/10.1016/j.compind.2021.103586

[4] Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. Data Synthesis based on Generative Adversarial Networks. PVLDB, 11 (10): 1071-1083, 2018. DOI: https://doi.org/10.14778/3231751.3231757

[5] Figueira, A.; Vaz, B. Survey on Synthetic Data Generation, Evaluation Methods and GANs. Mathematics 2022, 10, 2733. https://doi.org/10.3390/math10152733.

[6] Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. Data Synthesis based on Generative Adversarial Networks. PVLDB, 11 (10): 1071-1083, 2018. DOI: https://doi.org/10.14778/3231751.3231757.

[7] M. Naveen and S. De, "Efficient Digital Assistant Workflow for Ticket Monitoring and Dispatching," 2022 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), Bangalore, India, 2022, pp. 1-6, doi: 10.1109/SMARTGENCON56628.2022.10083722.

[8] Figueira, A.; Vaz, B. Survey on Synthetic Data Generation, Evaluation Methods and GANs. Mathematics 2022, 10, 2733. https://doi.org/10.3390/math10152733.

[9] César Quilodrán-Casas, Vinicius L.S. Silva, Rossella Arcucci, Claire E. Heaney, YiKe Guo, Christopher C. Pain, Digital twins based on bidirectional LSTM and GAN for modelling the COVID-19 pandemic, Neurocomputing, Volume 470, 2022, Pages 11-28, ISSN 0925-2312, DOI: 10.1016/j.neucom.2021.10.043.

[10] Sanitago Gomez, “Demystifying the CTGAN Loss Function | Synthetic Data Modeling”, sdv-team, Github Repository, Available: https://github.com/sdv-dev/SDV/discussions/980, Last Accessed: 01.06.2023.

[11] Table Evaluator, Available: https://baukebrenninkmeijer.github.io/table-evaluator/, Last Accessed: 31.01.2023.
[12] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, Kalyan Veeramachaneni, "Modeling Tabular Data using Conditional GAN", October 2019, Available: https://arxiv.org/pdf/1907.00503.pdf, Last Accessed: 01.06.2023.
[13] Sanitago Gomez, "Demystifying the CTGAN Loss Function | Synthetic Data Modeling", sdv-team, Github Repository, Available: https://github.com/sdv-dev/SDV/discussions/980, Last Accessed: 01.06.2023.
[14] Table Evaluator, Available: https://baukebrenninkmeijer.github.io/table-evaluator/, Last Accessed: 31.01.2023.
[15] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, Kalyan Veeramachaneni, "Modeling Tabular Data using Conditional GAN", October, 2019, Available: https://arxiv.org/pdf/1907.00503.pdf, Last Accessed: 01.06.2023.

Authors' Profiles

Sreevishnu A. B. is a Developer at SAP Labs India, and has over five years of experience in software development. Sreevishnu, specializing in Python and problem-solving, has advanced from Scholar to Associate Developer, and currently holds the position of Developer, contributing to many projects. He possesses a Master of Technology in Computer Software Engineering from the Birla Institute of Technology and Science, Pilani, and a Bachelor of Computer Application from the Amrita School of Arts and Science, Kochi. Sreevishnu participates actively in coding communities, with his projects displayed on GitHub and GitLab.

Suman De is a Product Manager and Scrum Trainer at SAP working for the Global Adoption & Experience CoE unit. He is also pursuing a Ph.D. at IIT Kharagpur in the Centre for Computational and Data Sciences, as a working professional. In addition, he holds the position of Visiting Professor at the Symbiosis Institute of Business Management, where his areas of expertise include Business Analytics, Cloud Computing, and Marketing. He has obtained his Master of Technology degree in Software Engineering from BITS Pilani and has authored 49 research papers, which have been published in prestigious journals such as IEEE, Elsevier, and Springer. Additionally, he has contributed to four book chapters on cutting-edge industrial subjects. Suman is an accomplished Scrum Trainer and mentor for Research initiatives. He has gained recognition as a thought leader for his valuable contributions to external conferences. He has served as a Keynote speaker at six research conferences and is affiliated with other IEEE and Springer Conferences as a member of the Technical Program Committees. Additionally, he has chaired sessions at International Conferences and was recently nominated as the Youth Engineering Icon of India by the Institution of Engineering and Technology (IET).

How to cite this paper: Sreevishnu A. B., Suman De, "Synthetic Data Generation Using Generative Adversarial Networks for Enterprise Application", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.17, No.6, pp. 71-80, 2025. DOI:10.5815/ijieeb.2025.06.06