CEUR-WS.org/Vol-4085/paper34.pdf

KGSynX: Knowledge Graph and Explainable Feedback Guided LLMs for Synthetic Tabular Data Generation

Ke YU[1], Shigeru Ishikura[2], Yukari Usukura[2], Yuki Shigoku[2] and Teruaki Hayashi[1]

[1]*Department of Systems Innovation, School of Engineering, the University of Tokyo*
[2]*Infomart Corporation*

Abstract

Synthetic tabular data is vital for augmentation, privacy, and performance under limited data, yet most work targets marginal statistics, neglecting downstream utility and explainability in scarce-data scenarios. We propose KGSynX, which builds a knowledge graph from table records and derives graph embeddings to inform LLM prompts. A SHAP-guided feedback loop measures attribution differences between real and generated data and injects targeted corrections into subsequent prompts. Evaluated under the Train-on-Synthetic, Test-on-Real (TSTR) protocol on heart disease, enterprise invoice, and telco churn datasets, KGSynX consistently outperforms baseline in accuracy, F1, and AUC while closing the SHAP attribution gap. By explicitly modeling structure and semantics, KGSynX produces more reliable synthetic datasets for downstream tasks.

Keywords

Synthetic Data, LLM, Explainable AI, Knowledge Graph

1. Introduction

Synthetic tabular data generation has emerged as a critical technique in scenarios where access to real datasets is limited by privacy, regulatory, or logistical constraints—for example, in healthcare [16], finance, and telecommunications [4, 9]. By creating high-quality synthetic records, practitioners can augment scarce data, share information without exposing sensitive details [18], and improve model training under low-resource conditions. However, most state-of-the-art approaches—ranging from generative adversarial networks (GANs) [1, 12, 13] and diffusion models [14, 15, 6] to Large Language Model (LLM) based generators [8] primarily focus on matching marginal feature distributions or low-order statistics. While these methods can reproduce individual column histograms or pairwise correlations, they often fail to capture higher-order semantic relationships present in the joint distribution. As a result, synthetic samples may exhibit unrealistic combinations of features, leading to degraded performance in downstream tasks and undermining user trust [5]. And these techniques still rely on handcrafted objectives or black-box signals, making it difficult to trace how structural or semantic errors persist in the synthetic data.

To address these challenges, we present **KGSynX**, which integrates knowledge graphs (KG) [10] and explainable AI feedback to steer LLM-based synthesis. Our key contributions are: First, KGSynX constructs a knowledge graph in which each record is represented as an entity node and each feature-value pair as an attribute node; edges encode the semantic dependencies inherent in the original table. We then extract structure-aware embeddings via Node2Vec [3] and incorporate them into LLM prompts, ensuring that sample generation respects the encoded graph topology. Next, we implement a SHAP-driven refinement loop [2]: after each generation round, we compute the attribution gap between real and synthetic data, identify the top-k discrepant features, and automatically inject targeted instructions into the prompt to correct those errors. This explainable feedback mechanism both improves downstream utility [19] and provides clear diagnostics for auditing.

We validate KGSynX under the Train-on-Synthetic, Test-on-Real (TSTR) protocol [11] on three benchmark datasets. Compared to baselines, our method achieves substantial gains in accuracy, F1 score, and AUC [20], while progressively narrowing the SHAP attribution gap. These results demonstrate

ISWC 2025 Companion Volume, November 2–6, 2025, Nara, Japan

© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

that explicitly modeling semantic structure and leveraging interpretable feedback are key to producing reliable synthetic data for practical applications.

2. Method Overview

2.1. Framework

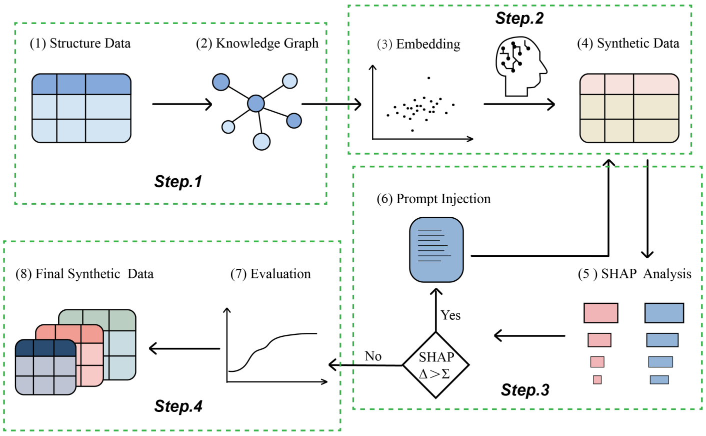

Figure 1: KGSynX Framework: (Step 1) KG Construction, (Step 2) Embedding & Initial Synthesis, (Step 3) SHAP Analysis and Prompt Feedback Loop, (Step 4) Final Optimized Synthetic Data Generation.

Figure 1 presents KGSynX as a four-step pipeline. (Step 1) *KG Construction*: we lift raw tables into a knowledge graph where each row becomes an entity node and each attribute-value is linked via typed relations, exposing domain rules and constraints. (Step 2) *Embedding & Initial Synthesis*: we compute Node2Vec embeddings over the KG and inject these structure-aware vectors into prompts for ChatGPT-4o [7] to generate an initial batch of synthetic records. (Step 3) *SHAP Analysis & Prompt Feedback Loop*: we train classifiers on real and synthetic data and use SHAP [17] to measure feature-importance gaps; these gaps are automatically translated into targeted prompt edits, and Steps 2–3 are repeated until misalignment falls below a preset threshold. (Step 4) *Final Optimized Synthetic Data Generation*: with the refined prompts, we produce the final synthetic dataset that best matches the real data in both statistics and decision-logic semantics. This modular design keeps knowledge extraction, generation, feedback, and refinement decoupled while preserving end-to-end semantic guidance.

2.2. Core Components

Knowledge Graph Construction. We construct a knowgraph graph $G = (V, E)$ where

$$V = V_{\text{entity}} \cup V_{\text{attribute}}, \quad E = \{(e, a) \mid \text{record } e \text{ has attribute } a\}.$$

Here, V_{entity} represents the set of sample entity nodes and $V_{\text{attribute}}$ represents the set of feature-value nodes. The edge set E captures associations between entities and their attributes, thus encoding the structural dependencies inherent in the original tabular data.

SHAP Attribution Gap. We quantify semantic alignment by computing

$$D_{\text{SHAP\_cos}} = 1 - \frac{\phi_{\text{real}} \cdot \phi_{\text{syn}}}{\|\phi_{\text{real}}\| \, \|\phi_{\text{syn}}\|},$$

where ϕ_{real} and ϕ_{syn} are the normalized SHAP attribution vectors for the real and synthetic datasets. The cosine distance $D_{\text{SHAP\_cos}}$ measures the angular dissimilarity between these vectors, with values closer to 0 indicating that the synthetic data's attribution pattern closely aligns with that of the real data.

Prompt Refinement. Given an initial prompt $\mathscr{P}_t$, we iteratively refine it by updating based on the top-k attribution discrepancies $\Delta\phi_k$:

$$\mathscr{P}_{t+1} = \mathscr{P}_t \oplus \{\text{emphasize features in } \Delta\phi_k\}.$$

The operator $\oplus$ denotes the appending of targeted instructions to the existing prompt. Through this SHAP-guided feedback loop, the LLM is steered to generate samples whose feature importance distributions progressively converge to those of the real dataset.

2.3. Prompt Example

Prompt Examples

Initial Prompt:

```
"Using the knowledge graph context, generate synthetic records ensuring the
following attribute dependencies: [KG summary]."
```

After SHAP Feedback:

```
"Prioritize matching the distribution of {Feature_A} and reduce
overrepresentation of {Feature_B}."
```

The first prompt instructs the LLM to adhere to the structural relationships embedded within the knowledge graph during the generation of new records. The second prompt encourages the model to refine its output by prioritizing features exhibiting the most significant attribution discrepancies.

2.4. Semantic Alignment Convergence

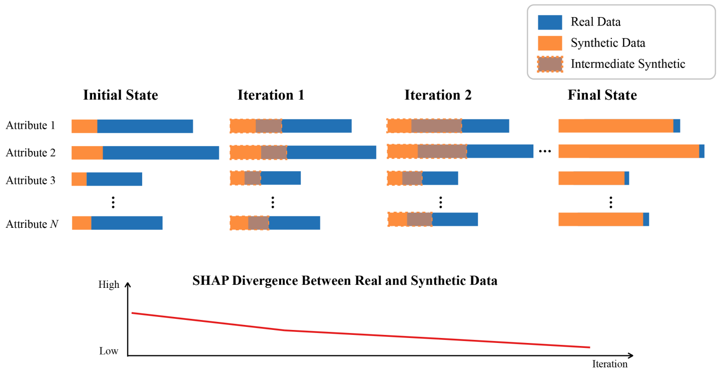

Figure 2: SHAP attribution gap D_{SHAP} over iterative feedback cycles

As shown in Figure 2, at each iteration we measure the SHAP divergence between real and synthetic models and update the prompts accordingly. This loop terminates when the semantic-alignment gap falls below ε (default 0.1) or the maximum number of rounds T is reached (default 5). In practice, convergence is typically achieved within 3–4 rounds.

3. Experiments & Results

3.1. Datasets and Classifiers

We used the three benchmark datasets in our experiments. The UCI Heart Disease dataset contains 303 samples with 13 clinical features, and is evaluated using a RandomForest classifier to capture non-linear interactions. The Enterprise Invoice Usage dataset comprises 500 enterprise transaction records with 11 attributes, for which we employ XGBoost due to its robustness on structured financial data. Finally, the Telco Churn dataset (7,043 samples, 20 features) is tested with LightGBM to leverage its high efficiency and accuracy in large-scale customer churn prediction. All classifiers are trained with default hyperparameter settings and 5-fold cross-validation to ensure a fair comparison.

3.2. Performance Comparison

Our KGSynX consistently outperforms CTGAN and vanilla LLM generators, achieving the best F1 and Area Under the Curve (AUC) scores across the board (Table 1). In the Heart Disease dataset, KGSynX boosts Accuracy from 0.667 (CTGAN) to 0.767 and improves F1 from 0.474 to 0.750. On the Enterprise dataset, it reaches the highest accuracy (0.900) and F1 (0.904), demonstrating its ability to model complex enterprise data. For Telco Churn, KGSynX attains the top AUC (0.853) and a balanced F1 (0.611), confirming its robustness in large-scale customer prediction tasks. These results validate that integrating knowledge-graph embeddings with SHAP-driven prompt refinement yields synthetic data with downstream utility and semantic fidelity.

Table 1
Performance comparison across three datasets

Method	Heart Disease[1]			Enterprise Invoice[2]			Telco Churn[3]		
	Accuracy	F1	AUC	Accuracy	F1	AUC	Accuracy	F1	AUC
Real	0.867	0.826	0.929	0.867	0.826	0.929	0.833	0.713	0.867
MedGAN	0.664	0.384	0.527	0.725	0.724	0.818	0.730	0.515	0.294
CTGAN	0.667	0.474	0.746	0.655	0.670	0.628	0.726	0.332	0.557
TabDDPM	0.541	0.380	0.498	0.425	0.357	0.544	0.721	0.603	0.772
LLM	0.350	0.361	0.278	0.765	0.766	0.838	0.626	0.584	0.810
LLM+KG	0.600	0.625	0.741	0.865	0.868	**0.943**	0.760	0.326	0.824
Ours	**0.767**	**0.750**	**0.827**	**0.900**	**0.904**	0.942	**0.776**	**0.611**	**0.853**

4. Conclusion & Future Work

In this work, we introduced KGSynX, a framework that seamlessly integrates knowledge-graph embeddings with SHAP-driven feedback to guide large language models in generating synthetic tabular data. Our method explicitly models the structural dependencies of tabular data and iteratively refines generation prompts based on feature attribution discrepancies. Our experiments, conducted under the TSTR protocol on UCI Heart Disease, Enterprise Invoice Usage, and Telco Churn datasets, demonstrate that KGSynX outperforms GAN-base models, TabDDPM, LLM-only, and LLM+KG baselines in classification accuracy, F1 score, and AUC, while preserving semantic fidelity and interpretability.

Despite these encouraging results, the current implementation relies on heuristic prompt adjustments, which may require manual tuning and domain expertise. Additionally, SHAP-based attribution computations introduce substantial computational overhead, limiting scalability in resource-constrained environments. Future work will focus on developing reinforcement-learning-based or differentiable optimization techniques for automated prompt refinement to reduce reliance on heuristics. We also plan to explore efficient SHAP approximation methods and extend our approach to multi-label, multi-modal knowledge graphs and streaming data scenarios to enhance applicability.

[1] https://archive.ics.uci.edu/dataset/45/heart+disease

[2] provided by Infomart Corporation

[3] https://www.kaggle.com/datasets/blastchar/telco-customer-churn

Supplemental Material Statement

The source code, real and synthetic datasets, and reproducible pipeline for KGSynX are available online via

- GitHub

Acknowledgments

This study was supported by the joint research project with Infomart Corporation and JST PRESTO Grant Number JPMJPR2369.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT in order to: Grammar and spelling check, Paraphrase and reword. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] L. Xu, A. Skoularidou, S. Wu, and G. Ermon, "Modeling tabular data using conditional GAN," in *NeurIPS*, 2019.

[2] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *NeurIPS*, 2017.

[3] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *KDD*, 2016.

[4] E. De Cristofaro, "Synthetic Data: Methods, Use Cases, and Risks," *arXiv preprint arXiv:2303.01230*, 2024.

[5] T. Marwala, E. Fournier-Tombs, and S. Stinckwich, "The Use of Synthetic Data to Train AI Models: Opportunities and Risks for Sustainable Development," *arXiv preprint arXiv:2309.00652*, 2023.

[6] M. Kotelnikov, P. Blinov, D. Baranchuk, et al., "TabDDPM: Modeling Tabular Data with Diffusion Models," *arXiv preprint arXiv:2302.07984*, 2023.

[7] OpenAI, "GPT-4 Technical Report," *arXiv preprint arXiv:2303.08774*, 2023.

[8] X. Fang, W. Xu, F. A. Tan, J. Zhang, Z. Hu, Y. Qi, S. Nickleach, D. Socolinsky, S. Sengamedu, and C. Faloutsos, "Large Language Models (LLMs) on Tabular Data: Prediction, Generation, and Understanding—A Survey," *arXiv preprint arXiv:2402.17944*, 2024.

[9] N. Patki, R. Wedge, and K. Veeramachaneni, "The Synthetic Data Vault," in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 2016.

[10] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, A.-C. Ngonga Ngomo, et al., "Knowledge Graphs," *ACM Computing Surveys (CSUR)*, vol. 54, no. 4, pp. 1–37, 2021.

[11] C. Esteban, S. L. Hyland, and G. Rätsch, "Real-valued (medical) time series generation with recurrent conditional GANs," *arXiv preprint arXiv:1706.02633*, 2017.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, pp. 2672–2680, 2014.

[13] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[14] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," *arXiv preprint arXiv:1503.03585*, 2015.

[15] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," in *Advances in Neural Information Processing Systems*, vol. 32, pp. 11895–11907, 2019.

[16] E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart, and J. Sun, “Generating multi-label discrete electronic health records using generative adversarial networks,” in *Proceedings of the 2nd Machine Learning for Healthcare Conference*, pp. 286–305, 2017.

[17] E. Mosca, F. Szigeti, S. Tragianni, D. Gallagher, and G. Groh, “SHAP-based explanation methods: a review for NLP interpretability,” *Proceedings of the 29th International Conference on Computational Linguistics*, pp. 4593–4603, 2022.

[18] E.-J. van Kesteren, “To democratize research with sensitive data, we should make synthetic data more accessible,” *Patterns*, vol. 5, no. 9, 2024.

[19] J. Achterberg, M. Haas, B. van Dijk, and M. Spruit, “Fidelity-agnostic synthetic data generation improves utility while retaining privacy,” *Patterns*, 2025.

[20] M. S. M. Sajjadi, O. Bachem, M. Lucic, O. Bousquet, and S. Gelly, “Assessing generative models via precision and recall,” in *Advances in Neural Information Processing Systems*, vol. 31, 2018.