Beyond TimeGraph: A Comparative Analysis of Temporal Generators for Evolving Network Graphs

Edoardo Di Caro*, Nicolas Belletti*, Filippo Poltronieri*, Mauro Tortonesi*, Cesare Stefanelli*,
* Distributed Systems Research Group, University of Ferrara, Ferrara, Italy
Email: {edoardo.dicaro, nicolas.belletti, filippo.poltronieri, mauro.tortonesi, cesare.stefanelli}@unife.it

***Abstract*—The growing adoption of Artificial Intelligence (AI) in network and service management demands extensive, diverse, and high-fidelity datasets for training and evaluation. However, collecting real-world network data at scale often faces significant challenges, including privacy concerns, operational constraints, and the rarity of certain events or conditions. Generative AI offers a promising solution by synthesizing realistic data that mirrors complex network dynamics and user behavior.**

In many application domains—such as mobile connectivity, cybersecurity, and disaster recovery—realism is not only defined by accurate replication of structural features (e.g., connectivity graphs), but also by the ability to model how these features evolve over time. Capturing these temporal dynamics is critical to ensure that AI models trained on synthetic data can generalize effectively to real-world scenarios. One effective approach to this challenge is to transform raw graph data into a compact latent representation, which can then be processed by a temporal generative model. This two-stage framework enables the learning of both structural and temporal characteristics of the underlying system, offering a more comprehensive generative pipeline.

Building on previous work that employed Time-series Generative Adversarial Networks (TimeGAN) for this purpose, this paper explores an alternative temporal generative model: DoppelGANger. By integrating DoppelGANger into the graph generation pipeline, we aim to assess whether it can more accurately capture the dynamics of evolving graph structures. Furthermore, we introduce a more rigorous and detailed evaluation of the generated data by comparing decoded synthetic graph sequences against their real-world counterparts using distribution-aware and graph-structural metrics. These metrics provide a clearer picture of the quality and fidelity of the generated data, highlighting key differences between the TimeGAN and DoppelGANger approaches.

***Index Terms*—Synthetic Data Generation, Time-Generative Models, Mobile Networks**

I. Introduction

Time-evolving network data plays a crucial role in understanding the dynamic behavior of modern communication systems, such as wireless networks, sensor networks, and emerging 5G/6G infrastructures. These networks are characterized not only by the structural relationships among entities, where nodes represent devices or network elements, and edges capture the flow of information or connectivity between them, but also by the temporal dynamics through which these relationships change over time. Unlike static networks, the structure of communication networks changes continuously over time due to factors like user mobility, varying traffic loads, and network failures. Modeling these temporal dynamics is essential for a range of tasks, including traffic forecasting, anomaly detection, fault diagnosis, resource allocation, and performance optimization. By effectively capturing both the structural relationships and their evolution, time-evolving network analysis provides the foundation for building resilient, efficient, and adaptive communication networks.

However, real-world temporal graph datasets are often scarce and difficult to obtain due to several practical limitations. Privacy concerns can restrict access to sensitive data, particularly in domains such as healthcare and finance. Moreover, many observed dynamic networks are sparse, noisy, or lack consistent ground truth annotations, which hinders both the development and evaluation of robust machine learning models. These challenges highlight the growing need for high-quality synthetic data generators that can mimic the structural and temporal properties of evolving graphs while maintaining fidelity to real-world patterns. Such synthetic data can play a critical role in benchmarking algorithms, enabling privacy-preserving experimentation, and augmenting limited datasets in data-constrained scenarios.

Despite growing interest in generative modeling for time series and static graphs, relatively few approaches have explicitly targeted the generation of time-evolving graph-structured data. Dynamic graphs introduce unique challenges, including the need to model temporal dependencies alongside structural variations, such as the appearance or disappearance of nodes and edges over time. Traditional generative models are not designed to encode these relational and temporal dynamics jointly, often treating each time step in isolation or ignoring the underlying graph structure altogether. As a result, their outputs fail to capture the rich, interdependent nature of real-world evolving networks.

In this paper, we present an extended synthetic data generation framework for time-evolving networks, building directly upon the TimeGraph [1] architecture. Our work aims to evolve and deepen the understanding of TimeGraph's modular design by systematically exploring the effect of alternative temporal generative models within its framework. Specifically, we replace the original time-series generation component with DoppelGANger [2], an established model for realistic sequential data synthesis. This substitution allows us to investigate how different temporal modeling choices influence the quality, realism, and downstream utility of the generated dynamic graph sequences. Importantly, our approach retains the core

978-3-903176-75-1 ©2025 IFIP

TimeGraph principle of learning a latent temporal representation and decoding it into a sequence of graph snapshots. This consistency enables controlled experimentation and fair comparison across different temporal models, shedding light on the strengths, limitations, and trade-offs inherent to each method in the context of dynamic graph generation.

II. Background and Related Works

The synthesis of realistic time-evolving data has become increasingly important in all those domains where temporal dynamics play a central role. Time-generative models are a class of machine learning techniques designed to capture and replicate these temporal dependencies, enabling the creation of synthetic time series that mirror the statistical and sequential properties of real-world data. Among these models, TimeGAN [3] and DoppelGANger have emerged as two prominent approaches, each offering distinct advantages in how they handle multivariate dependencies, conditional distributions, and sequence coherence.

TimeGAN introduces a novel approach to time-series generation by integrating the generative power of GANs with the temporal consistency of supervised sequence models. Unlike prior GAN-based models for time-series, TimeGAN incorporates a supervised loss that explicitly encourages accurate modeling of stepwise temporal transitions. This is achieved by a hybrid training framework that combines adversarial loss with supervised sequence modeling, enabling it to preserve both the marginal distributions and temporal correlations of the original data. The architecture consists of four key components: an embedding network, a recovery network, a generator, and a discriminator. These networks operate in a shared latent space learned via an autoencoding mechanism, enabling efficient adversarial training and preserving long-range temporal dependencies. The supervised loss is computed in the latent space to further discipline the generator toward realistic dynamics. Studies confirm the importance of each architectural and training component, particularly the joint supervision and embedding structure.

DoppelGANger is a generative framework tailored for synthetic tabular and time-series data, particularly in domains with complex dependencies such as IoT, finance, and healthcare. DoppelGANger introduces a structured, multi-stage generation process that separates the generation of static and temporal attributes while preserving their dependencies. The model consists of two main GAN components: one for generating static attributes and high-level temporal features, and another for generating detailed time-series conditioned on those features. This hierarchical decomposition enables the model to capture both global and local patterns more effectively. It also incorporates explicit dependency modeling between variables and uses auxiliary discriminators to stabilize training and improve fidelity across different data types. This modular design allows DoppelGANger to capture global structure and inter-variable dependencies more explicitly, and makes it scalable to longer sequences and larger datasets.

In the context of these models, input data are inherently time-series, requiring any structured data to be first transformed into a sequential format. To enable this, autoencoders [4] are commonly employed, as they learn compact, informative embeddings from structured inputs. These embeddings, when ordered over time, can then serve as effective inputs to time-generative models. In this work, the original data consist of connectivity networks represented as graphs. To bridge the gap between these graph-structured inputs and the temporal modeling framework, we leverage Graph Neural Networks (GNNs)—specifically, graph autoencoders [5] [6]—which produce meaningful low-dimensional representations of the graphs. These representations are then organized temporally to interface with the downstream time-series generative model.

III. Architecture and Methodology

The architecture at the base of the TimeGraph framework, as illustrated in Figure 1, is composed of two key models that operate in synergy to form a comprehensive and robust synthetic data generation pipeline. These two components are designed to play complementary roles, with each addressing a distinct phase of the generation process while relying on one another for optimal overall performance.

The first of these component is an Encoder/Decoder model, which serves as the foundational building block of the entire system. Its primary function is to transform the original, real-world data into a compact and meaningful representation within a latent space. This embedding process is of critical importance: any inaccuracies, inefficiencies or distortions introduced at this stage have the potential to propagate through the subsequent stages of the pipeline. Over time, such errors can become amplified, significantly compromising both the realism and the quality of the final synthetic data output. Therefore, selecting and training a high-performing and expressive model is essential. Given the graph-like structure of the input data, a Graph Autoencoder is employed. This type of model is well-suited for learning the underlying patterns and relationships within structured graph data. It enables effective encoding into a low-dimensional, continuous latent space while preserving the topological integrity and feature richness of the original data.

The second major component, which forms the core of the synthetic data generation process, is a generative model specifically tailored for time-series data. This model is responsible for capturing the temporal dynamics, correlations, and dependencies that are inherent in time-evolving datasets. Its role is to learn how the data evolves over time within the latent space and subsequently generate realistic synthetic sequences that accurately reflect the statistical properties of the original dataset. Since the inputs to this model are the embeddings produced by the Encoder, it must be capable of handling multivariate time-series data. This requirement arises from the multidimensional nature of the latent space, since encoding the full complexity of the original data in just a single dimension (which would correspond to a univariate time series) is rarely

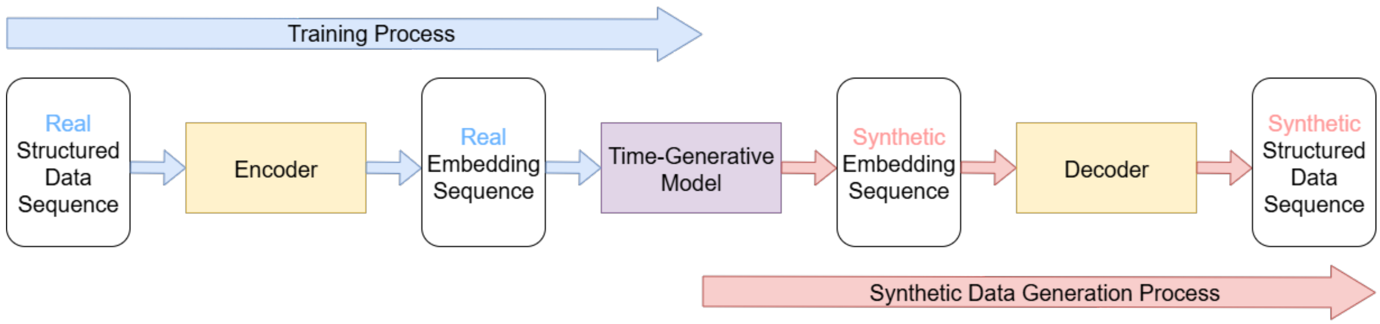

Fig. 1. Full architecture and pipeline of the TimeGraph framework, highlighting the Training and the Synthetic Data Generation workflows.

sufficient. By supporting multivariate modeling, the generative model can preserve the intricate interdependencies among different features, ensuring that the generated sequences retain the realism and diversity of the source data.

However, while increasing the dimensionality of the latent space often enhances representational capacity, it also introduces significant challenges. Higher-dimensional data increases the complexity of the generative task, often resulting in longer training times and greater difficulty in achieving convergence. Beyond a certain point, this added complexity may lead to diminishing returns or even a degradation in performance, where the generated data no longer closely follows the desired distribution [7]. Therefore, careful dimensionality control is vital: the goal is to find a balance between expressive power and practical learnability, keeping the dimensionality as low as possible without sacrificing the fidelity of the data representation.

In this paper, we leveraged the modular architecture of the TimeGraph framework, which allows for seamless substitution of its core components without disrupting the overall pipeline or requiring significant adjustments to other parts. Taking advantage of this flexibility, we chose to retain the previously trained Autoencoder due to its strong performance. Specifically, this model achieved an Average Precision (AP) of 0.9 and an Area Under the Curve (AUC) of 0.91, all while maintaining a latent space of just one dimension per graph node.

Our intervention, therefore, focused instead on the time-series generative model. We opted to replaced the original TimeGAN with DoppelGANger, based on a series of observations regarding the characteristics of the synthetic data generated by TimeGAN. Earlier high-level analysis using Principal Component Analysis (PCA) [8] confirmed that TimeGAN-generated data points remained within the overall distribution of the real data. However, these data points were found to be notably concentrated near the center of the distribution. A more thorough and detailed analysis, discussed in the following section, revealed that TimeGAN has a tendency to "smooth out" the data distribution excessively, limiting its ability to accurately reflect the full diversity of the source data. This limitation motivated our decision to explore an alternative generative model that could more accurately capture the underlying structure of the data.

IV. Synthetic Data Evaluation

A. Setup

To ensure consistency with previous TimeGraph experiments and enable a fair comparison, both models were trained using the same dataset and identical time windows of data. The selected dataset is Vignette 2 of the Anglova Scenario [9], a highly detailed and realistic emulated network environment specifically designed to replicate the complex dynamics encountered during military operations. It addresses a wide range of operational conditions and constraints, including mobility, terrain variability, intermittent connectivity, and limited resources. These characteristics make it an exceptional source of realistic mobility and connectivity patterns in Mobile Ad-hoc Networks (MANETs) [10], particularly in harsh and unpredictable environments—a domain where high-quality, representative data is notoriously difficult to obtain or synthesize accurately.

For the purposes of this study, the four companies—each consisting of 24 mobile nodes—have been analyzed independently. This resulted a dataset composed of a series of temporally correlated graphs, where each graph captures the connectivity state of a single company at a specific point in time. These individual graphs, representing snapshots of the network, were then grouped into sequences of temporally evolving graphs over fixed-duration windows of 10, 30, 60, 120, and 300 seconds. This approach allows the models to capture and learn the dynamics of the network over time, enabling an evaluation of their performance and the realism of the generated data across different temporal granularities.

To evaluate the quality and realism of the generated data, we adopted a two-stage analysis approach. First, we applied PCA to the learned embeddings in order to obtain a high-level view of the distribution of the synthetic versus real data. This dimensionality reduction step helped reveal whether the generative model captured the overall structure and variability of the original dataset. Following this, we performed a more in-depth evaluation by decoding the embeddings back into sequences of evolving graph structures. From these reconstructed graphs, we computed two key metrics over each time window: mean connectivity and mean churn rate. These metrics provide insight into both the structure and dynamics of the network. Mean connectivity quantifies how well-connected the nodes are by measuring, for each node, the number of other nodes it maintains direct connections with. Mean churn rate captures the network's dynamicity by evaluating how much each node's local neighborhood changes over time. Specifically, it accounts for the number of new neighbors that appear and the number that disappear between consecutive seconds. Both metrics are calculated by taking the mean across all 24 nodes of the network over the whole duration of the chosen time window. By averaging these metrics across nodes, we obtain a compact summary of the network's behavior during each period, getting insight into both the structural density and temporal stability of the networks. We then compared the distributions of these metrics in the synthetic data against those in the real dataset, enabling a quantitative assessment of how closely the generated sequences match the statistical properties of the original network behavior.

B. Results

Figure 2 presents the results of applying PCA to visualize the distributions of synthetic data generated by DoppelGANger and TimeGAN, in comparison to real data. The analysis focuses on time windows of 10, 30, and 300 seconds, as these intervals revealed the most pronounced differences and are therefore especially useful in highlighting key distinctions between the two approaches. Both models succeeded in generating synthetic data that largely falls within the distribution of the real data. However, notable differences emerge between them. The synthetic data produced by TimeGAN appears more tightly clustered than that of DoppelGANger, occupying a smaller portion of the real data's distributional space. This suggests that while both models are capable of producing realistic samples, TimeGAN tends to concentrate on the central regions of the data distribution and struggles more with representing outliers or edge cases. This tendency is further supported by quantitative evaluations using metrics such as mean connectivity and mean churn rate, which confirm the more conservative nature of TimeGAN's data generation.

Results regarding the mean connectivity distributions are illustrated in Figure 3. This plots are obtained by computing the probability density estimation on the data comprising each dataset, namely real, synthetically generated with DoppelGANger and synthetically generated with TimeGAN. As such, the value on the y-axis only corresponds to the number of samples with a specific mean connectivity value. Such value it is given only by the fact that the real dataset has more examples than the synthetic ones, as it is common practice to generate a smaller set of data than the original. This means that the "height" of the graph is not relevant to the result, while the important results are carried by the actual shape of the curves.

Results regarding the mean connectivity distributions are illustrated in Figure 3. These plots are obtained by computing the probability density estimation on the data from each dataset, namely the real dataset, the one synthetically generated with DoppelGANger, and the one synthetically generated with TimeGAN. Accordingly, the values on the y-axis correspond only to the number of samples associated with

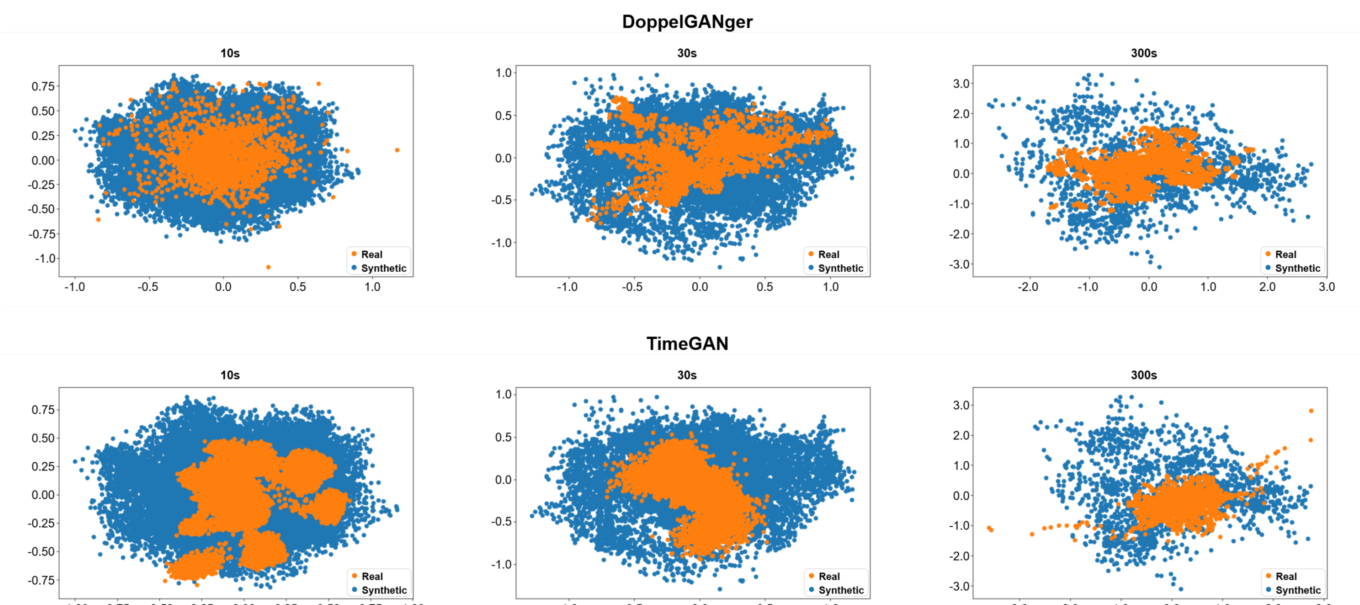

Fig. 2. PCA-based visualization of data distributions for real and synthetic datasets across three time windows: 10, 30, and 300 seconds (left to right). The top row shows synthetic data generated by DoppelGANger, while the bottom row displays synthetic data from TimeGAN.

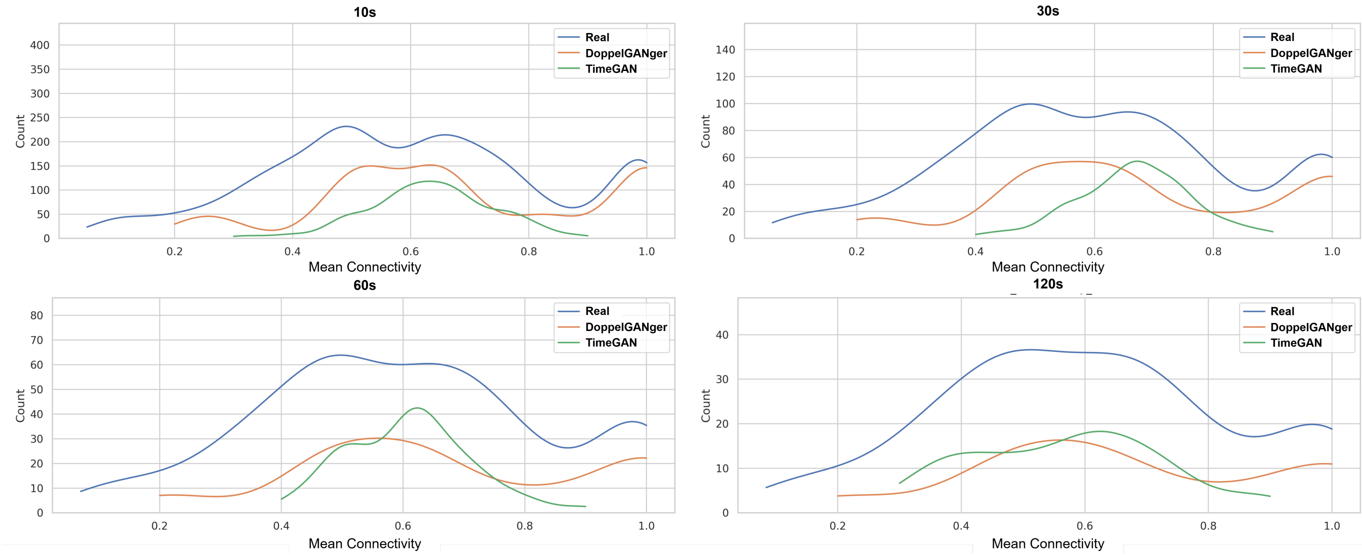

Fig. 3. Mean connectivity distributions for time frames of 10, 30, 60 and 120 seconds, comparing real data (blue) and synthetically generated using DoppelGANger (orange) and TimeGAN (green).

a specific mean connectivity value. This value is influenced only by the fact that the real dataset contains more examples than the synthetic ones, as it is common practice to generate a smaller set of synthetic data compared to the original. Therefore, the "height" of the plot is not relevant to the result; instead, the meaningful insights are conveyed through the actual shape of the curves.

According to the observations made during the PCA analysis of the results, it becomes evident that TimeGAN has a tendency to "smooth out" the data, generating the highest number of examples near the central peak of the original distribution and gradually decreasing the number of occurrences as it moves further away from these values. In contrast, DoppelGANger demonstrates strong performance by following the overall shape of the original distribution more closely and by successfully representing samples closer to the edge cases. Notably, the difference between the two models becomes less pronounced as the examined time window increases. Moreover, the fact that DoppelGANger better follows the original data distribution does not necessarily make it the preferred model for every application scenario. For instance, edge cases with a mean connectivity equal to 1.0 represent graphs that remain fully connected for the entire duration of the sample. Since this configuration is already present in the real data, generating additional examples of it brings only marginal benefit, as it contributes little to no new or significant information.

The other key metric evaluated is the mean churn rate, which complements mean connectivity by reflecting the dynamicity of the network. If connectivity patterns are well preserved but churn rates differ significantly, it suggests that the model has correctly captured structural features but not temporal dependencies with the same accuracy. A narrower churn rate distribution than the original indicates overly static behavior, while a wider one suggests excessive dynamicity. This could imply that consecutive graphs have drastically different connections, resulting in transitions that are abrupt or unrealistic, and therefore lacking plausible temporal correlation. Figure 4 compares the mean churn rate distributions between the real and synthetic datasets across different time windows. Once again, DoppelGANger demonstrates better performance than TimeGAN; however, in this case, the improvement is less pronounced than what was observed with the mean connectivity metric. Moreover, the difference between the two models continues to diminish as the length of the time window increases. Notably, especially at lower time windows, TimeGAN tends to generate samples that exhibit greater dynamicity. Despite this, the generated churn rate values remain close enough to those of the original data to be considered reasonable and within an acceptable range of variation.

This behavior highlights an important previous observation: while DoppelGANger generally achieves better alignment with the original data and may be more suitable for most application scenarios, TimeGAN can still be a valuable tool when specific requirements are present. For instance, in situations where it is desirable to train on highly dynamic networks—perhaps to simulate rapidly evolving systems or to stress-test models under frequent structural changes—TimeGAN's tendency to generate more temporally active graphs can be advantageous. In such cases, its ability to introduce greater variation in graph evolution over time can complement the original dataset, particularly if those dynamic behaviors are underrepresented in the real data. Thus, while overall performance might favor DoppelGANger, the choice of model should ultimately be guided by the goals and constraints of the target application.

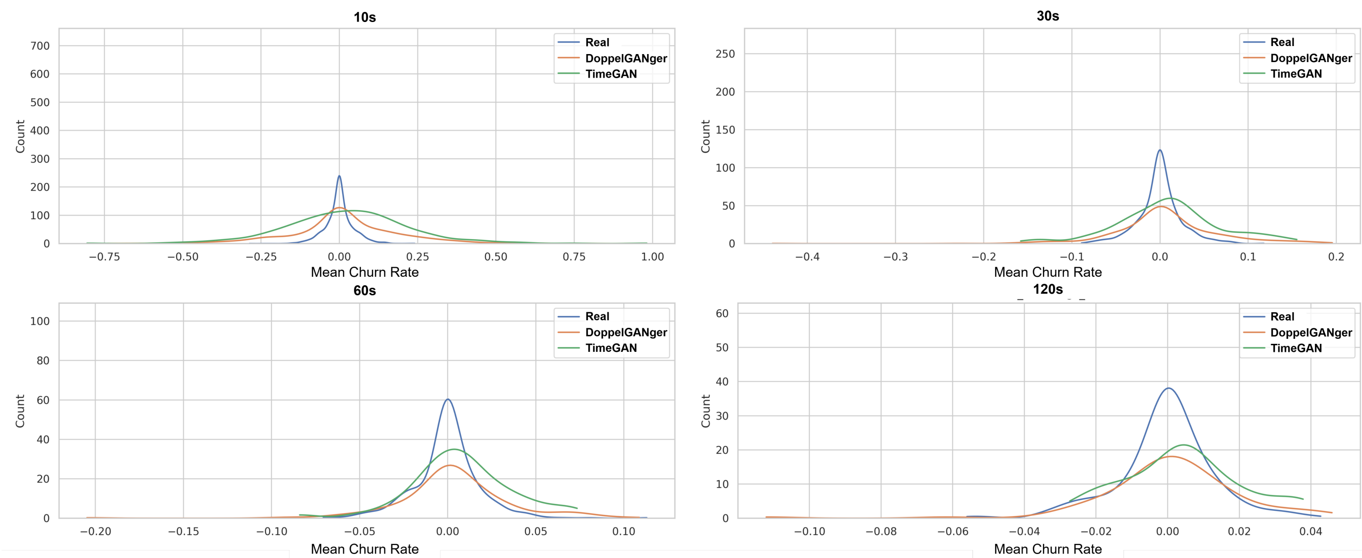

Fig. 4. Mean churn rate distributions for time frames of 10, 30, 60 and 120 seconds, comparing real data (blue) and synthetically generated using DoppelGANger (orange) and TimeGAN (green).

V. Conclusions and Future Works

In this paper, we extended the TimeGraph framework for synthetic generation of time-evolving network graphs by integrating DoppelGANger as an alternative temporal generative model to TimeGAN. Our goal was to assess whether DoppelGANger could better capture the structural and dynamic properties of real-world mobile networks, particularly under the challenging conditions of the Anglova scenario.

The experimental results support this hypothesis. DoppelGANger produced synthetic embeddings with greater distributional diversity, as confirmed through PCA visualization, and demonstrated better alignment with the original dataset in both mean connectivity and churn rate metrics. This suggests an improved ability to represent both core structural patterns and edge cases, leading to higher-fidelity synthetic graph sequences. Notably, while TimeGAN exhibited a tendency to “smooth out” the temporal dynamics—often favoring central patterns—its output remains valuable, especially in scenarios with specific requirements.

This line of research opens several promising avenues for future work. First, extending the framework to incorporate node and edge attributes would enable the generation of richer and more application-specific synthetic graphs, expanding its applicability to more complex network structures. Additionally, integrating a time-series model that supports both generation and forecasting could significantly enhance the utility of TimeGraph. Such a hybrid framework could serve both in offline contexts—augmenting datasets for training and testing AI models—and in online scenarios, enabling real-time forecasting of network evolution. This, in turn, would support proactive downstream tasks such as anomaly detection, routing optimization, and failure prediction, allowing systems to adapt dynamically to anticipated changes.

References

[1] E. Di Caro, M. Brina, N. Belletti, F. Poltronieri, M. Tortonesi, and C. Stefanelli, “Timegraph: Synthetic generation of graph sequences for realistic mobile connectivity models,” in *2025 IEEE 11th International Conference on Network Softwarization (NetSoft)*, 2025, pp. 249–256.

[2] Z. Lin, A. Jain, C. Wang, G. Fanti, and V. Sekar, “Using gans for sharing networked time series data: Challenges, initial promise, and open questions,” in *Proceedings of the ACM Internet Measurement Conference*, ser. IMC ’20. New York, NY, USA: Association for Computing Machinery, 2020, p. 464–483. [Online]. Available: https://doi.org/10.1145/3419394.3423643

[3] J. Yoon, D. Jarrett, and M. van der Schaar, “Time-series generative adversarial networks,” in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/c9efe5f26cd17ba6216bbe2a7d26d490-Paper.pdf

[4] P. Li, Y. Pei, and J. Li, “A comprehensive survey on design and application of autoencoder in deep learning,” *Applied Soft Computing*, vol. 138, p. 110176, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494623001941

[5] T. N. Kipf and M. Welling, “Variational graph auto-encoders,” 2016. [Online]. Available: https://arxiv.org/abs/1611.07308

[6] S. J. Ahn and M. Kim, “Variational graph normalized autoencoders,” in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, ser. CIKM ’21. New York, NY, USA: Association for Computing Machinery, 2021, p. 2827–2831. [Online]. Available: https://doi.org/10.1145/3459637.3482215

[7] A. Crespo Márquez, *The Curse of Dimensionality*. Cham: Springer International Publishing, 2022, pp. 67–86. [Online]. Available: https://doi.org/10.1007/978-3-030-97660-6_7

[8] F. L. Gewers, G. R. Ferreira, H. F. D. Arruda, F. N. Silva, C. H. Comin, D. R. Amancio, and L. D. F. Costa, “Principal component analysis: A natural approach to data exploration,” *ACM Comput. Surv.*, vol. 54, no. 4, May 2021. [Online]. Available: https://doi.org/10.1145/3447755

[9] N. Suri, J. Nilsson, A. Hansson, U. Sterner, K. Marcus, L. Misirlioğlu, M. Hauge, M. Peuhkuri, B. Buchin, R. in't Velt, and M. Breedy, “The angloval tactical military scenario and experimentation environment,” in *2018 International Conference on Military Communications and Information Systems (ICMCIS)*, 2018, pp. 1–8.

[10] J. Hoebeke, I. Moerman, B. Dhoedt, and P. Demeester, “An overview of mobile ad hoc networks: applications and challenges,” *Journal-Communications Network*, vol. 3, no. 3, pp. 60–66, 2004.