

GAN-Based Synthetic Sequence Generation for Money Collection Histories

Bora Ozuguzel

Graduate Program in Data Science
Ozyegin University, Istanbul, Türkiye
bora.ozuguzel@ozu.edu.tr

Ekrem Duman

Department of Industrial Engineering
Ozyegin University, Istanbul, Türkiye
ekrem.duman@ozyegin.edu.tr

Abstract

This study examines how generative adversarial network (GAN)-based synthetic-data generation can be used for money collection histories. Two provided synthetic historical collection datasets are used in this study. Customer records are organized as irregular histories: some customers have one observed collection state, while others have multiple collection states ordered mainly by delinquency days and related repayment, balance, and collection-action information. The workflow reviews data quality, removes leakage-prone fields, builds ordered sequence windows, and develops MoneyCollectionTimeGAN, a compact sequence-generation model that adapts selected TimeGAN ideas to generate collection behavior windows. Simpler prediction and forecasting checks are kept as supporting evidence. Related GAN methods for tabular data and minority-class augmentation provide background for the design choices. The generated histories are evaluated using reconstruction and temporal losses, separability AUC, statistical and temporal validation, proxy prediction tests, range-validity checks, and duplicate screening. Results show meaningful but partial temporal similarity, with generated sequences still distinguishable from the provided synthetic input process. The findings are reported as bounded synthetic-to-synthetic evidence rather than validation on live operational customer data.

Keywords

Generative adversarial networks; Money collection; Synthetic data generation; Collection analytics

1. Introduction

Money collection concerns the recovery of overdue receivables from accounts whose repayment behavior changes over time. A collection record is therefore not only a single classification row, it can also be one observed state inside a customer history. Recent debt-collection predictive analytics research shows that account and contact-center information can support repayment-related decisions, which supports treating the domain as a data-driven behavioral prediction problem in addition to account scoring (Sanchez et al. 2022).

This study asks whether GAN-based synthetic sequence generation can represent money collection behavior over ordered customer histories. The question is important because row-level synthetic data may preserve separate feature distributions while still missing the way delinquency, repayment, balance, action history, and risk signals move together over time. The research setting is deliberately bounded: both input datasets are provided synthetic historical collection datasets, and the results are interpreted relative to that provided synthetic input process rather than live operational customer data.

The contribution is a compact synthetic-to-synthetic sequence-generation and evaluation workflow for money collection histories. It constructs customer-history windows ordered mainly by delinquency progression, applies a local TimeGAN-inspired model named MoneyCollectionTimeGAN, and evaluates whether generated windows preserve enough temporal structure to be credible within the provided synthetic input process. The result is not a deployment claim; it is a methodological study of whether this type of sequence generation is promising for the domain.

The paper is organized as follows. Section 2 defines the problem and objectives. Section 3 reviews the method sources and research gap. Section 4 describes the data and preprocessing. Sections 5 and 6 present the method and results. Section 7 interprets the statistical evidence and validates the objectives, Section 8 states limitations, and Section 9 concludes.

2. Problem Statement and Objectives

Money collection histories contain repeated or single observed states for customers, including delinquency days, repayment behavior, balance changes, action counts, product information, risk-related fields, and observed collection outcomes. The problem addressed in this study is how to generate synthetic histories that preserve behavior across ordered collection states instead of creating unrelated individual rows.

The study has four objectives. First, it reviews the two provided synthetic historical datasets so data quality, target balance, repeated Customer_ID histories, and sequence readiness are understood before generation. Second, it builds leakage-aware modeling inputs by excluding fields that would create misleading prediction signal. Third, it uses MoneyCollectionTimeGAN to generate constant-length collection-history windows ordered mainly by DLQ_Days. Fourth, it evaluates whether generated windows are statistically, temporally, visually, and practically credible within the provided synthetic input process.

The central evaluation question is whether the generated windows become difficult to distinguish from the provided synthetic input windows while keeping meaningful temporal structure. The evaluation therefore combines reconstruction and temporal errors, separability, visual overlap, scaled-range validity, and duplicate screening. These criteria are read together because a generator can produce plausible individual values while still failing to preserve ordered behavior.

3. Literature Review

GAN-based synthetic data generation begins with the generator-discriminator training idea introduced by Goodfellow et al. (2014). Conditional GANs add conditioning information to the generation process (Mirza and Osindero 2014), and Wasserstein GANs provide a different objective for stabilizing adversarial learning (Arjovsky et al. 2017). These sources define the adversarial foundation, but they do not by themselves solve the ordered money collection setting.

Time-dependent synthetic data requires attention to sequence dynamics. TimeGAN combines embedding, recovery, adversarial learning, and supervised temporal objectives so generated sequences can preserve temporal relationships, and it evaluates generated data using visual, discriminative, and predictive evidence (Yoon et al. 2019). Earlier recurrent conditional GAN work also supports synthetic real-valued sequence generation (Hyland et al. 2017). Recent synthetic time-series evaluation research emphasizes that there is no single universal quality metric, which supports using several diagnostics together (Stenger et al. 2024).

Money collection records are also mixed row-and-column data. CTGAN is relevant because it addresses tabular data with continuous and discrete columns (Xu et al. 2019). FinGAN-related customer and financial work is relevant only as context for GAN-based augmentation and analytical customer relationship management settings (Kate et al. 2022; Vuletic et al. 2024). Financial synthetic-data studies provide useful evaluation context such as comparing statistical properties, similarities, and correlations between input and generated data (Ramzan et al. 2024). These sources are method context, not direct money collection implementations.

Classical and supervised tools provide supporting evidence. ARIMA supplies a simple univariate time-series benchmark for delinquency movement (Hyndman and Khandakar 2008). Logistic regression and random forest

baselines check whether leakage-safe fields retain useful collection-outcome prediction signal, with random forests providing nonlinear supervised comparison (Schonlau and Zou 2020). ROC-AUC supports classifier interpretation (Carrington et al. 2023), class-imbalance literature supports balanced metrics (Johnson and Khoshgoftaar 2019), and PCA supports low-dimensional visual inspection while not proving generated-data quality by itself (Jolliffe and Cadima 2016).

The research gap addressed here is the lack of a bounded sequence-generation and evaluation workflow for irregular money collection histories, where customer states must be ordered by delinquency progression and generated windows must be tested for temporal similarity and separability. TimeGAN provides the sequence-generation design language, CTGAN provides mixed tabular context, and FinGAN-related work provides augmentation context. The local contribution is to adapt selected sequence-generation ideas to collection histories and then discuss whether the resulting generated histories are credible enough for this domain.

4. Data and Preprocessing

The study uses two provided synthetic historical money collection datasets. The 60K dataset has 64,016 rows and 43 columns. It contains Customer_ID, DLQ_Days, DLQ_Date, delinquency, action, product, repayment, risk, profit, and Target fields. The 75K dataset has 75,481 rows and 42 columns. It contains Customer_ID, DLQ_Days, Transaction, delinquency, action, product, repayment, risk, profit, and Target fields.

The datasets are treated as irregular collection histories. One row is one observed collection state, not a complete daily panel. Some customers have one row and some have multiple rows. Histories are grouped by Customer_ID and ordered mainly by DLQ_Days before sequence windows are created. Transaction is treated as an identifier rather than a payment variable. Target means money collected or not, while Target_Churn is a churn label and is excluded from the collection-focused modeling pipeline.

The datasets support a bounded synthetic-data methodology study because they contain repeated customer histories, temporal ordering variables, target labels, and mixed numeric/categorical collection fields.

Preprocessing converts the provided records into a leakage-aware sequence modeling Table 1. It handles numeric parsing and missing values, builds ordered customer windows, encodes categorical fields for support baselines, and excludes identifiers, churn labels, provider score or probability fields, and the collection Target from prediction features. This prevents proxy baselines and diagnostics from learning from fields that reveal the answer.

The data preparation choices also define the meaning of the results. Grouping by Customer_ID makes the unit of generation a customer-history window rather than an isolated row, and ordering mainly by DLQ_Days uses the available delinquency progression as the practical time axis. These choices narrow the study, but they make the reported sequence-generation result more interpretable as collection-history generation rather than churn modeling or provider-score reproduction.

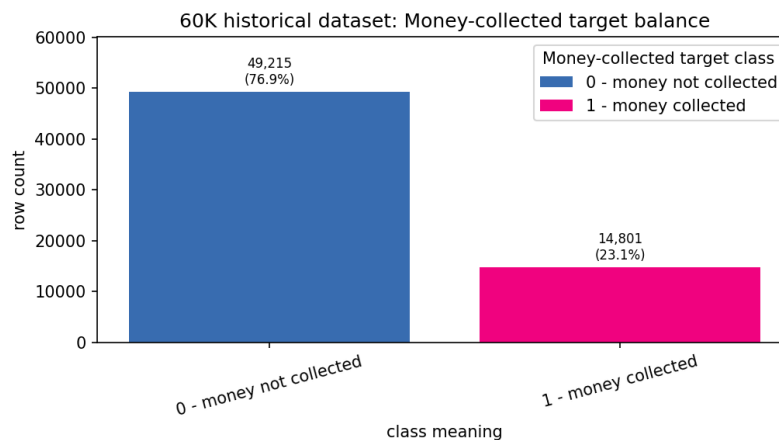


Figure 1. Money-collected target balance in the 60K historical dataset.

Figure 1 shows the Target balance for the money-collected label. It motivates imbalance-aware interpretation because plain accuracy can look good even when the smaller outcome class is modeled poorly.

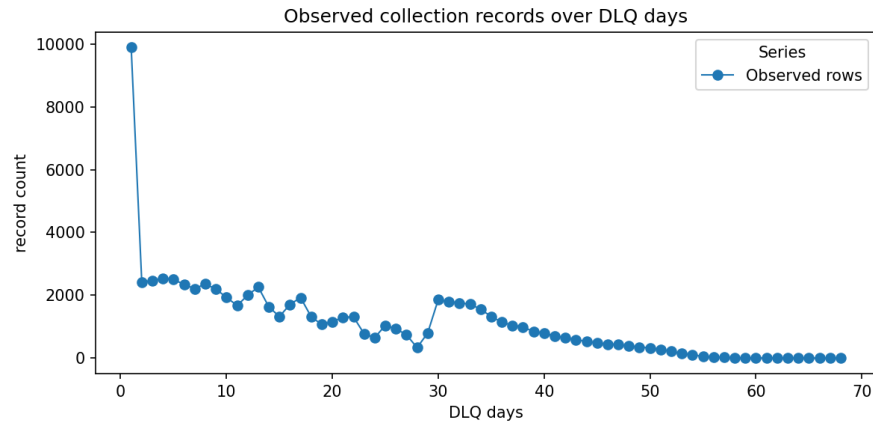


Figure 2. Observed collection records over delinquency-day progression.

Figure 2 shows the distribution of observed collection records along the DLQ_Days ordering axis. It is an input-data check: the sequence model can only learn ordered behavior where the provided synthetic histories contain usable coverage.

5. Methodology

The methodology has four connected parts. First, the datasets are reviewed and cleaned. Second, leakage-safe supervised baselines and simple forecasting checks are used as supporting diagnostics. Third, Customer_ID histories are transformed into constant-length sequence windows. Fourth, MoneyCollectionTimeGAN generates sequence windows and the generated output is evaluated against the provided synthetic input windows.

MoneyCollectionTimeGAN is the implemented sequence-generation branch in this paper. It adapts selected TimeGAN concepts: representation, recovery, temporal movement, generated sequence output, and input-versus-generated evaluation (Yoon et al. 2019). The implementation is compact and local, and CTGAN and FinGAN-related studies are used as bounded context for mixed tabular data and augmentation ideas.

A sequence window is a constant-length ordered block of customer history. The current run uses [46714, 4, 12] input windows and generates [1000, 4, 12] synthetic windows. Recovery MSE measures reconstruction after representation and recovery. Temporal supervisor MSE measures one-step movement inside the learned representation. Input-versus-generated AUC measures how easily a classifier can separate generated windows from provided input windows, where values closer to 0.5 are better for indistinguishability.

The compact implementation is selected because the immediate research task is to test the collection-history framing and evaluation protocol. A full neural sequence generator would require additional training and tuning decisions beyond the current scope. In this paper, the simpler implementation makes it easier to trace which parts of the evidence come from reconstruction, temporal movement, baseline comparison, and separability. This is appropriate for an initial synthetic-data study, as long as the paper does not overstate the model as a production-ready TimeGAN system.

Table 1. Compact method summary for MoneyCollectionTimeGAN.

Element	Reported value	Interpretation
Sequence input	[46714, 4, 12]	Customer_ID histories are grouped into constant-length windows ordered mainly by DLQ_Days.
Representation and recovery	recovery MSE 0.000506	The compact latent representation can reconstruct the scaled input

		windows with low error.
Temporal movement	temporal MSE 0.002987	One-step movement in the learned representation is measured separately from reconstruction.
Generated output	[1000, 4, 12]	The generator produces collection-history windows in the same scaled sequence format.
Boundary	TimeGAN-inspired local implementation	The method adapts selected TimeGAN ideas rather than reproducing the full original architecture.

The supporting methods answer narrower questions. Logistic regression and random forest ask whether leakage-safe fields contain collection-outcome prediction signal. Naive, moving-average, and ARIMA checks ask whether delinquency movement is forecastable. Independent feature drawing asks what happens when feature values are generated without sequence learning. PCA gives a visual projection. These checks support the main synthetic sequence question, but they are not the main generator.

Method Pseudocode

Algorithm 1. Money collection GAN workflow used in this study.

Input: the 60K and 75K provided synthetic historical collection datasets and the leakage-exclusion list.

Output: leakage-safe baselines, DLQ_Days-ordered sequence windows, MoneyCollectionTimeGAN generated windows, and generated-data diagnostics (Table 2).

Table 2. Pseudocode summary for the money collection GAN workflow.

Step	Action	Purpose
1	Load the 60K and 75K provided synthetic historical collection datasets.	Start from both collection-history sources.
2	Review structure, missing values, target balance, repeated Customer_ID histories, and DLQ_Days coverage.	Establish sequence readiness before generation.
3	Keep Target as the collection label and exclude churn, identifiers, provider scores, Transaction, and other leakage-prone fields.	Prevent misleading model inputs.
4	Clean decimal text, impute missing values, encode categorical fields, and run leakage-aware baselines.	Create support checks for prediction signal.
5	Build Customer_ID histories ordered mainly by DLQ_Days and convert them into fixed-length sequence windows.	Represent irregular rows as ordered histories.
6	Train MoneyCollectionTimeGAN with representation, recovery, temporal movement, and generated-window output.	Generate ordered collection-history windows.
7	Evaluate generated windows with losses, temporal/statistical checks, PCA, separability tests, validity rules, and duplicate screening.	Report quality and boundaries without overclaiming.

6. Results and Evaluation

The results are best read in three layers: input-sequence evidence, support baselines, and generated-sequence evidence. Input-sequence evidence supports the time-dependent framing. The strongest lagged features include:

- Number_of_Actions_ever: lag-1 autocorrelation 1, temporal gap 0.0103.
- Payment_ratio_last3months: lag-1 autocorrelation 1, temporal gap 0.
- Delinquent_amount: lag-1 autocorrelation 1, temporal gap 0.

These values show that several collection-history features are persistent across ordered states. That persistence is exactly what a sequence generator should try to preserve.

The ARIMA benchmark for DLQ_Days uses 240 train points and 60 test points, with RMSE 9.373 and MAE 8.077. This does not generate histories; it only provides a classical reference for delinquency movement.

The strongest leakage-safe Target baseline is Random Forest with ROC-AUC 0.853. This means the remaining fields retain useful collection-label signal after leakage columns are removed. It also means generated data should be evaluated beyond marginal statistics, because useful collection behavior includes predictive structure (Table 3).

Table 3. Proxy baseline metrics.

Target	Model	Acc.	Bal. acc.	F1	ROC-AUC
Target	Logistic Regression	0.783	0.754	0.793	0.815
Target	Random Forest	0.867	0.741	0.853	0.853

MoneyCollectionTimeGAN produces measurable generated windows in the expected shape. Recovery MSE is 0.000506, temporal supervisor MSE is 0.002987, and the direct input-versus-generated AUC is 0.701. The recovery and temporal errors are useful, but the AUC shows that generated windows are still distinguishable from the provided synthetic input windows.

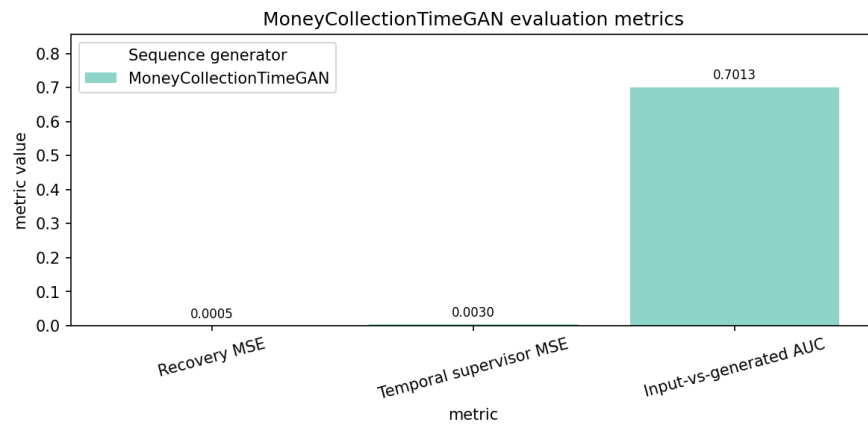


Figure 3. MoneyCollectionTimeGAN reconstruction, temporal, and separability metrics

Figure 3 summarizes the core sequence-branch metrics. The main interpretation is mixed: reconstruction and temporal movement are credible, but separability remains above the ideal value of 0.5.

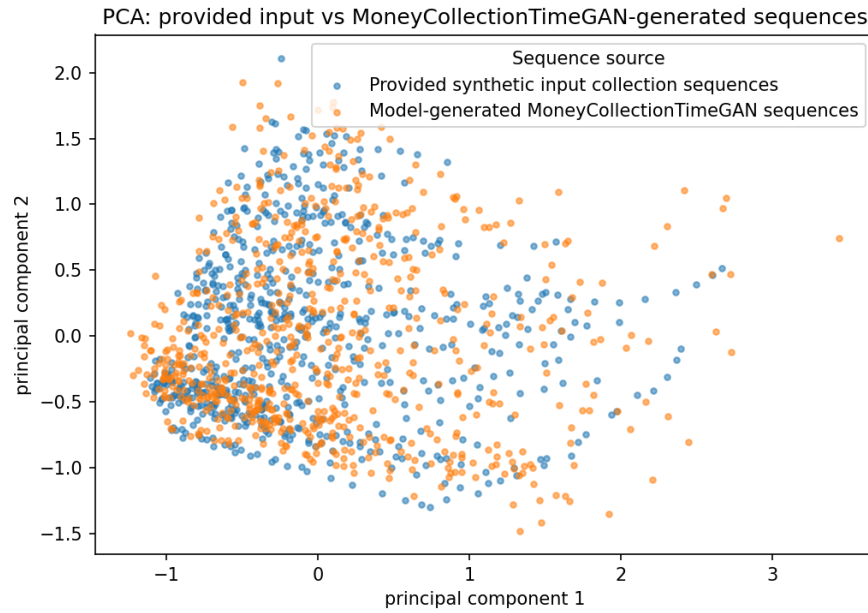


Figure 4. PCA diagnostic for provided input sequences and generated sequences

Figure 4 projects input and generated sequence windows into two principal components. The plot is useful for visual overlap or separation, but it is not treated as proof of quality because PCA removes detail and does not test step-by-step temporal behavior.

The simple generation comparison clarifies the result. MoneyCollectionTimeGAN has standardized input-versus-generated AUC 0.646, while independent feature drawing has AUC 0.803. Since values closer to 0.5 indicate weaker separability, the sequence-aware branch is better than independent feature drawing in this comparison. It is still not indistinguishable.

This comparison gives the result its mixed interpretation. Independent feature drawing is a useful negative reference because it can match some individual values while breaking ordered behavior. MoneyCollectionTimeGAN has a better temporal profile than that reference, especially on lag and temporal-gap diagnostics. However, the remaining separability means that better temporal profile is not the same as complete generated-data quality. The result is therefore promising for research development but limited for operational interpretation.

The closest statistical validation rows also support partial temporal similarity:

Range-validity checks report no missing values and no scaled-range violations in the generated windows. The nearest-neighbor duplicate screen reports exact duplicate count 0 in the scaled sequence representation. These checks are necessary for interpreting the generated output, but they are not privacy guarantees.

The range-validity and duplicate checks should be read as basic validity screens, not as final success criteria. They show that the generated tensor can be inspected and compared without obvious numerical failure. They do not show that the generated histories are realistic enough for collection policy use, and they do not prove that sensitive information would be protected under a real-data setting. This is why the evaluation uses several diagnostics together rather than treating any single pass/fail check as decisive (Table 4).

Table 4. Evidence interpretation and separability results

Evidence	Value	Interpretation
Direct MoneyCollectionTimeGAN separability	0.701	Generated windows remain distinguishable from the provided synthetic input process.

Standardized sequence comparison	MCTGAN 0.646; independent draw 0.803	The sequence-aware branch is closer to 0.5 than independent feature drawing, so it improves temporal plausibility but does not solve separability.
Temporal similarity	lag diff 0.033; gap diff 0.016	Lag and temporal-gap differences are much smaller than the independent feature draw baseline.
Bootstrap and permutation check	AUC CI 0.582 to 0.639; p=0.005	A self-contained separability test still detects a real difference between input and generated windows.
Range-validity and duplicate screen	range flags 0/0; duplicates 0	Generated scaled windows pass basic validity checks and do not appear as exact copies in this representation.

The evidence table brings the main results together. The next section interprets this evidence as a statistical and research-validation result.

7. Statistical Analysis, Discussion, and Research Validation

Overall, the model should be interpreted as partially successful rather than fully successful. The clearest positive evidence is that MoneyCollectionTimeGAN preserves more temporal behavior than independent feature drawing. The standardized sequence comparison has lower separability for MoneyCollectionTimeGAN than for independent feature drawing, and its lag and temporal-gap differences are much smaller. This matters because money collection histories are ordered behavioral records. If a generator only samples unrelated rows or unrelated features, it can miss the development of delinquency, repayment, balance, action, and risk patterns across the same customer history.

The second positive result is that the generated windows pass basic validity and duplicate checks. The output has no missing values, no scaled-range violations, and no exact duplicates in the nearest-neighbor screen. This supports the use of the workflow as a sandbox for method development, scenario exploration, and later evaluation design. It also shows that the generated output is usable enough for diagnostic comparison rather than failing at the basic validity level.

The AUC values answer related but different statistical questions. The direct input-versus-generated AUC of 0.701 is the separability result from the MoneyCollectionTimeGAN branch. The standardized comparison AUC of 0.646 places MoneyCollectionTimeGAN beside lightweight local generation baselines under a common comparison setup. The independent feature drawing AUC of 0.803 is higher, which means it is easier to separate from the input sequence windows. Read together, these values say that MoneyCollectionTimeGAN improves over a weak non-sequential generator, but still does not reach the ideal indistinguishability region near 0.5.

The hypothesis-style test supports the same interpretation. The centroid-based separability score has a bootstrap confidence interval above 0.5 and a small permutation p-value, so the input and generated windows are statistically distinguishable in this diagnostic representation. This is not a negative result by itself. It gives the paper a clearer success boundary: the method captures some temporal behavior and produces valid sequence-shaped outputs, but the remaining separability shows that the generator is not yet strong enough for claims of full synthetic replacement.

The main weakness is also clear: the generated sequences remain distinguishable. A direct input-versus-generated AUC of 0.701 is not close enough to 0.5 to claim that generated and input windows are indistinguishable. The bootstrap and permutation evidence points in the same direction. Therefore, the results should not be described as fully successful synthetic replication. The correct interpretation is that the method captures part of the temporal structure but still leaves measurable differences from the provided synthetic input process.

This discussion also explains why row-level baselines and figures are kept in supporting roles. Proxy prediction and ARIMA show that the data contain usable prediction and temporal signals. PCA gives a visual summary. However,

the main research question is not whether a classifier can predict the collection outcome or whether one univariate series can be forecast. The main question is whether generated customer-history windows preserve ordered collection behavior. For that question, separability, temporal similarity, validity, and duplicate screening must be discussed together.

Each objective is met within this bounded setting. The data are reviewed and described as provided synthetic historical collection records. Leakage-prone fields are excluded from the modeling inputs. Customer_ID histories are converted into DLQ_Days-ordered sequence windows. MoneyCollectionTimeGAN generates windows in the expected shape. The generated output is evaluated through reconstruction, temporal movement, separability, lightweight baselines, PCA, range-validity checks, and duplicate screening. The validation result is mixed rather than purely positive: the workflow is promising for synthetic sequence experimentation, but stronger modeling and validation are needed before it can support deployment-oriented claims.

Practically, the method is most useful as a controlled experimentation and method-development environment. It can help test sequence-generation logic before sensitive operational collection data are available. It can also help define what evidence should be required in a future real-data study. It is not yet evidence that generated histories can replace real data, support operational collection decisions, or satisfy formal privacy requirements.

For industrial engineering and operations management, this partial-success result is still meaningful. Collection operations involve repeated decisions about timing, contact actions, repayment behavior, and risk. A synthetic sequence generator that begins to preserve ordered behavior can support experimentation with data preparation, feature engineering, evaluation protocols, and decision-support prototypes before institutionally sensitive data can be shared. The value is therefore methodological: the study identifies a workable sequence framing, shows which diagnostics are informative, and shows where the current generator still falls short.

A fully successful result would need stronger evidence than the current paper reports. The generated windows would need lower separability, closer temporal and statistical validation across more features, and downstream utility tests where models trained on generated labeled sequences perform well on held-out input sequences. If real or institutionally verified anonymized data become available, the same evaluation design should be repeated under stricter data-governance controls. Until then, the correct claim is not that the generated histories are operational substitutes, but that GAN-based sequence generation is a plausible research direction for money collection histories. This distinction is important for interpreting the figures and tables. The figures show the minimum evidence chain: target balance, delinquency-order coverage, core MoneyCollectionTimeGAN metrics, and PCA overlap. The tables likewise focus on method design, proxy prediction signal, and separability interpretation. This keeps the paper centered on the research question while avoiding repetitive dataset inventory and implementation-setting detail.

8. Limitations

The first limitation is the data setting. Both inputs are provided synthetic historical collection datasets, so the study evaluates whether MoneyCollectionTimeGAN can model the provided synthetic input process. It should not be interpreted as evidence that the same results would hold on live customer collections without additional validation.

The second limitation is model scope. MoneyCollectionTimeGAN is a compact local adaptation of selected TimeGAN ideas, focused on representation, recovery, temporal movement, generated sequence output, and separability evaluation. Broader neural architectures, larger tuning studies, and longer sequence settings remain future work.

The third limitation is evaluation scope. CTGAN, FinGAN-related work, ARIMA, proxy classifiers, PCA, and TSTR-style reasoning are used to position or support the sequence-generation study. They help interpret the results, but the central evidence remains the MoneyCollectionTimeGAN input-versus-generated comparison on the provided synthetic histories.

9. Conclusion

This paper presents a compact GAN-based synthetic sequence-generation workflow for money collection histories. The results are promising but only partially successful. The model improves temporal behavior compared with independent feature drawing, and generated windows pass basic validity and duplicate checks. At the same time,

generated windows remain distinguishable from the provided synthetic input windows, so the findings should be reported as partial temporal similarity rather than complete distributional equivalence.

The main research value is that the study frames money collection as an ordered customer-history generation problem rather than only a row-level synthetic-data problem. This framing is useful for a domain where delinquency, repayment, balance, actions, and risk signals evolve together. Future work should test stronger neural TimeGAN variants, compare against richer sequence baselines, validate on real or institutionally verified anonymized collection histories, and evaluate downstream utility when generated labels are available.

Acknowledgments

Special thanks are extended to Nazli Mehrabi for providing the synthetic money collection datasets and domain context that made this project possible. The analysis, method adaptation, and interpretations are the author's responsibility.

References

- Arjovsky, M., Chintala, S. and Bottou, L., Wasserstein generative adversarial networks, *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 214-223, 2017.
- Carrington, A. M., Manuel, D. G., Fieguth, P. W., Ramsay, T., Osmani, V., Wernly, B., Bennett, C., Hawken, S., Magwood, O., Sheikh, Y., McInnes, M. and Holzinger, A., Deep ROC analysis and AUC as balanced average accuracy, for improved classifier selection, audit and explanation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 329-341, 2023.
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., Generative adversarial nets, *Proceedings of the 27th International Conference on Neural Information Processing Systems*, vol. 2, pp. 2672-2680, 2014.
- Hyland, S. L., Esteban, C. and Ratsch, G., Real-valued medical time series generation with recurrent conditional GANs, Available: <https://arxiv.org/abs/1706.02633>, Accessed on May 18, 2026.
- Hyndman, R. J. and Khandakar, Y., Automatic time series forecasting: The forecast package for R, *Journal of Statistical Software*, vol. 27, no. 3, pp. 1-22, 2008.
- Johnson, J. M. and Khoshgoftaar, T. M., Survey on deep learning with class imbalance, *Journal of Big Data*, vol. 6, article 27, 2019.
- Jolliffe, I. T. and Cadima, J., Principal component analysis: A review and recent developments, *Philosophical Transactions of the Royal Society A*, vol. 374, article 20150202, 2016.
- Kate, P., Ravi, V. and Gangwar, A., FinGAN: Generative adversarial network for analytical customer relationship management in banking and insurance, Available: <https://arxiv.org/abs/2201.11486>, Accessed on May 18, 2026.
- Mirza, M. and Osindero, S., Conditional generative adversarial nets, Available: <https://arxiv.org/abs/1411.1784>, Accessed on May 18, 2026.
- Ozturk, C., Enhancing financial time-series analysis with TimeGAN: A novel approach, *Proceedings of the 2024 9th International Conference on Computer Science and Engineering*, pp. 447-450, Antalya, Türkiye, September 11-13, 2024.
- Ramzan, F., Sartori, C., Consoli, S. and Recupero, D. R., Generative adversarial networks for synthetic data generation in finance: Evaluating statistical similarities and quality assessment, *AI*, vol. 5, no. 2, pp. 667-685, 2024.
- Sanchez, C., Maldonado, S. and Vairetti, C., Improving debt collection via contact center information: A predictive analytics framework, *Decision Support Systems*, vol. 159, article 113812, 2022.
- Schönlau, M. and Zou, R. Y., The random forest algorithm for statistical learning, *The Stata Journal*, vol. 20, no. 1, pp. 3-29, 2020.
- Stenger, M., Leppich, R., Foster, I., Kounev, S. and Bauer, A., Evaluation is key: A survey on evaluation measures for synthetic time series, *Journal of Big Data*, vol. 11, article 66, 2024.
- Vuletic, M., Prenzel, F. and Cucuringu, M., Fin-GAN: Forecasting and classifying financial time series via generative adversarial networks, *Quantitative Finance*, vol. 24, no. 2, pp. 175-199, 2024.
- Xu, L., Skoularidou, M., Cuesta-Infante, A. and Veeramachaneni, K., Modeling tabular data using conditional GAN, *Advances in Neural Information Processing Systems*, vol. 32, pp. 7333-7343, 2019.
- Yoon, J., Jarrett, D. and van der Schaar, M., Time-series generative adversarial networks, *Advances in Neural Information Processing Systems*, vol. 32, pp. 5509-5519, 2019.

Biographies

Bora Ozuguzel is a graduate student in the Graduate Program in Data Science at Ozyegin University in Istanbul, Türkiye. He received his B.Sc. in Computer Science from Ozyegin University. His academic interests include machine learning, synthetic data generation, time-series analysis, and applied data science for customer-behavior problems. His current research focuses on using generative adversarial network concepts to model and evaluate synthetic money collection histories.

Dr. Ekrem Duman received his B.S. degree from the Electrical and Electronics Engineering Department at Bogazici University in 1990, and his M.S. and Ph.D. degrees from the Industrial Engineering Department of the same university in 1994 and 1998, respectively. After several years of industrial experience, he worked as a professor in the Industrial Engineering Department of Dogus University between 2001 and 2011. Besides his theoretical studies, he is interested in applied operations research, data analytics, and decision-support applications, and he has been involved in many industrial projects, mostly in banking, as a coordinator or consultant. In this study, he supervised the research framing, method selection, validation design, and academic reporting of the money collection GAN workflow.