Generation of Synthetic Aircraft Landing Trajectories Using Generative Adversarial Networks

Wijnands, Sebastiaan; Sharpanskykh, Alexei; Aly, Karim

Publication date
2024

Document Version
Final published version

Published in
14th SESAR Innovation Days, SIDS 2024

Citation (APA)
Wijnands, S., Sharpanskykh, A., & Aly, K. (2024). Generation of Synthetic Aircraft Landing Trajectories Using Generative Adversarial Networks. In *14th SESAR Innovation Days, SIDS 2024* (SESAR Innovation Days).

Generation of Synthetic Aircraft Landing Trajectories Using Generative Adversarial Networks

Sebastiaan Wijnands, Alexei Sharpanskykh, Karim Aly
Air Transport and Operations, Faculty of Aerospace Engineering
Delft University of Technology (TU Delft)
Delft, The Netherlands
s.c.p.wijnands@student.tudelft.nl, o.a.sharpanskykh@tudelft.nl, k.y.s.b.aly@tudelft.nl

***Abstract*—The increasing demand and complexity of air traffic management (ATM) systems necessitate significant advancements in automation to ensure safety and efficiency. Artificial intelligence (AI) and machine learning (ML) are emerging as promising solutions to manage this growing complexity, offering enhanced decision-making and predictive capabilities. However, the effectiveness of ML models in ATM heavily relies on the availability of extensive, high-quality data. In many cases, such data is scarce or incomplete, which presents a major barrier for training robust models. Synthetic data generation (SDG) is a viable solution to address this, enabling the creation of realistic datasets that unlock the ML value proposition. The Terminal Maneuvering Area (TMA) is a crucial segment of airspace characterized by high traffic density and diverse trajectory types, necessitating granular data to model these scenarios accurately. The main research objective of this work was to investigate the applicability of TimeGAN in generating synthetic 4-dimensional aircraft landing trajectories capable of capturing traffic patterns in this airspace, helping to analyze airspace constraints and delay propagation. The resulting synthetic trajectories were evaluated in terms of data diversity, fidelity and usefulness. The main challenge identified during the research was the imbalance in data classes, which affected the models' ability to accurately capture data patterns, particularly in less frequent scenarios. Generating synthetic data based on separate groupings showed promise in addressing these imbalances, although this approach was sensitive to the designation of groups. This work proves the capability of TimeGAN in generating diverse, realistic trajectories that are difficult to differentiate from real historical data.**

***Keywords*—Air traffic management, Deep generative models, Generative Adversarial Networks, Multivariate time series generation, Synthetic data quality evaluation**

I. Introduction

As air traffic management (ATM) systems face escalating demands and increasing complexity, advancing automation leveraging artificial intelligence (AI) and machine learning (ML) has become crucial for maintaining safety and operational efficiency. However, the effectiveness of these technologies is limited by the availability of comprehensive datasets needed for training ML models, constraining the ability of capturing complex patterns. Acquiring sufficient training data is often hampered by high costs and logistical challenges; aptly named the *data access problem*. Moreover, datasets must meet high standards of quality, adhere to data privacy laws, and promote fairness to avoid perpetuating algorithmic biases [1]. Synthetic data generation (SDG) addresses these limitation by creating realistic datasets that effectively bridge the data gap and allow for model training. By generating accurate synthetic data, one could optimize automation, improve prediction accuracy, and enhance resource allocation within ATM systems.

The Terminal Maneuvering Area (TMA) is of particular interest to model because it experiences high traffic density, complex interactions between arriving and departing aircraft, and varied trajectory patterns, making it crucial for optimizing airspace efficiency. Synthetic trajectories help examine airspace constraints and delay propagation, while also providing data for training ML models in tasks like flight delay prediction and full trajectory extension in synthetic traffic generators. This is one of the uses cases considered as part of the SynthAIr project [2]. Within the TMA, synthetic trajectories allow for detailed analysis of aircraft behavior during critical phases of flight, such as takeoff and landing. This data supports the design and evaluation of airspace procedures, air traffic control strategies, and collision avoidance systems, ensuring safety and efficiency in congested airspace [2]. Well-trained models for generating synthetic data for landing phases have a high potential for generalizability to other airports globally and to different flight phases, including en-route operations. This is because landing data offers inherent diversity, encompassing scenarios like go-arounds, holding patterns, weather impacts, capacity issues, and variability in air traffic control (ATC) directives [3].

The majority of existing literature addressing the problem of trajectory generation has been model-driven based on flight dynamic equations [3]. The objective of this research is to investigate the applicability of Generative Adversarial Networks (GANs) to the ATM setting. In this context, the inherent temporal relationships in time series data, beyond just feature distributions, increase the complexity of the generative process [4]. GANs are promising due to their abilities in generating realistic, high-dimensional synthetic data by learning from existing data distributions, but have virtually never been applied to a broad base on landing data at an airport encompassing multiple runway approach patterns, aircraft type, and operational conditions. A GAN architecture comprises a generator and discriminator and is trained through adversarial learning to generate realistic synthetic data. The generator transforms random noise into synthetic data samples, aiming to deceive the discriminator, which classifies samples as real or fake. The iterative training process involves the generator producing increasingly realistic samples while the discriminator improves at distinguishing between real and generated data [5]. TimeGAN differs from a standard GAN by using a recurrent neural network (RNN) in its architecture to capture temporal dependencies in sequential data and is trained with both supervised and adversarial losses to ensure the generated sequences are realistic and time-consistent [4]. This can be extended with conditioning mechanisms to enhance control of the generation process and enforcing diversity of generated samples. Thus, the main research question addressed in this paper: how effective are GANs in generating realistic synthetic 4-dimensional aircraft landing trajectories in an airports Terminal Maneuvering Area, including go-arounds and holding patterns? The synthetic samples are evaluated to quantify the quality, diversity, fidelity, and usefulness.

The paper commences with collating the literature and state of the art comprising the theoretical background and related work, in Section II. Subsequently, the five-step methodology of applying TimeGAN to the problem of landing trajectory generation is explained in Section III. Section IV presents the results of the experiments in terms of the established evaluation framework. Ultimately, the results from all experimentation are integrated and discussed in light of the originally proposed value proposition, in Section V. Concluding remarks alongside future implications of this work are contained in Section VI.

II. Related Work

Methods for generating synthetic aircraft trajectories can be broadly classified into model-driven and data-driven approaches. Model-driven methods use mathematical models of aircraft dynamics and environmental constraints to generate trajectories, where Base of Aircraft Data (BADA) [6] and OpenAP [7] are the most well-known. These deterministic methods often involve optimization processes to achieve objectives like minimizing fuel consumption or flight time [8]. However, their deterministic nature limits their ability to incorporate uncertainty and randomness [3]. Flight plan database statistical extrapolation uses historical data to generate synthetic traffic sets, creating trajectories that replicate real-world traffic patterns. By leveraging existing data, this approach can produce realistic synthetic trajectories without requiring detailed physical modeling of aircraft dynamics, though its efficacy depends considerably on the availability and quality of historical flight data [9]–[11]. Markov chain models introduce stochastic processes into trajectory generation, enabling the modeling of uncertainty and randomness. This has been used for estimating conflict probabilities in uncontrolled airspace, as it represents the trajectory as a sequence of states with probabilistic transitions, capturing the inherent variability [12].

Gaussian Mixture Models (GMMs) identify representative trajectory patterns by clustering trajectories [13]. They generate trajectories reflecting the statistical properties of observed flight paths, capturing the diversity and variability of real flight trajectories. By fitting a mixture of Gaussian distributions to the trajectory data, this method effectively represents different common flight paths and deviations from those paths, making it suitable for scenarios where the data exhibits multi-modal distributions [14]. Most recently, deep generative modeling techniques have been applied to the generation of synthetic aircraft trajectories. A GAN was applied to generate aircraft trajectories for detecting atypical approaches at Paris Orly Airport. The model produced realistic synthetic trajectories for the last 25 NM for a single runway and aircraft type, though facing challenges with mode collapse and required smoothing to remove unrealistic noise [15]. The Temporal Convolutional Variational Autoencoder (TCVAE) presented in [3] represents the acme of the state-of-the-art, serving as the benchmark for this study. The authors were capable of generating 4-dimensional landing paths at Zurich Airport for estimating conflict probabilities through Monte Carlo simulations. This model integrates a temporal convolutional network (TCN) to capture sequential dependencies and uses a VampPrior, a variational mixture of posteriors, for better handling complex latent distributions. While high-density latent space regions produced realistic trajectories, low-density areas generated less realistic paths.

III. Methodology

This study systematically investigates the applicability of TimeGAN [4] to aircraft trajectory generation, following five stages: (A) data collection and preprocessing, (B) trajectory labeling and clustering, (C) TimeGAN model implementation, (D) the evaluation framework, and (E) model training and optimization through experiments.

A. Data - Landing Trajectories

This study makes use of the available dataset of landing trajectories at Zurich Airport (CH), parsed from Automatic Dependent Surveillance-Broadcast (ADS-B) data originally sourced from the OpenSky Network [16]. The circa 18.000 recorded landing trajectories occurred in the months of October and November in 2019, each initiating when the aircraft is within a 40 nautical mile distance from the airport. The vast majority ($\sim$90%) of landings within this two month period were on runway 14, with remainder usually allocated to runway 28 ($\sim$7%) and 34 ($\sim$2%). Some of the trajectories contain go-arounds and holding patterns.

Compared to take-offs, landing paths are much more complex and varied due to arrivals appearing from any radial and the crucial role of air traffic controllers in sequencing aircraft through varying climate and capacity conditions. This is reflected in occurrences of go-arounds and holding patterns. A go-around (missed approach) is a standard procedure for pilots when landing conditions are unsafe, involving full power, retracting flaps and gear if necessary, and climbing to a safe altitude to re-enter airspace, crucial for preventing runway accidents [17]. Holding patterns are predefined flight paths used by air traffic control to delay aircraft at congested airports or during airspace congestion. Aircraft fly in a racetrack pattern above a fix, awaiting approach clearance, helping to manage traffic flow and ensure safe separation [18].

Given the variation in approach procedures at Zurich Airport, the 18.000 trajectories are safely deemed representative of arrivals at other major airports subject to equal operational constraints due to local terrain, noise abatement, and emission regulation [3]. The value proposition of a synthetic trajectory generator is therefore safeguarded by this hypothesized generalizability to other geographic locations and flight phases (take-offs and en-route). To ensure consistency in trajectory representation, each trajectory is resampled at a fixed rate and condensed to a state vector of longitude, latitude, and altitude at each timestamp. The resampling and simplification of the input dataset was varied during the experimentation/training of the GANs to gain insight into performance sensitivity.

B. *Trajectory Labeling & K-Means Clustering using DTW*

The labeling of landing trajectories is essential for discerning between different archetypical landing behaviors. The two benefits thereof are the ability to conduct separate analysis on various landing sequences to boost explainability, and it opens the door to incorporating conditioning mechanisms to the generative modeling process. The latter becomes especially relevant for addressing lacking mode coverage of GANs by attempting to exploit transfer-learning between labeled classes. Trajectories are first categorized into three categories: trajectories involving a go-around, involving holding pattern, and *normal* procedures. Go-arounds are detected based on multiple Instrument Landing System (ILS) alignments of the aircraft, separated by a climbing phase. In some cases, the aircraft lands on a different runway on the second attempt. Holding patterns identification is based on the total change in track angle of the aircraft's flight path, and setting a minimum cumulative track angle threshold for a trajectory to be labeled as such. Additionally, each trajectory is further labeled based on the specific runway used for landing.

Complementary to operational-based labeling, a more flexible approach was taken. Trajectories were clustered by employing the k-means clustering algorithm with Dynamic Time Warping (DTW) as the dedicated distance metric [19], [20]. DTW finds an optimal match between points in the two series by stretching or compressing them along the time axis, which is especially useful for comparing sequences with similar overall patterns but differing temporal dynamics. This approach ensures that similar-shaped trajectories are grouped together, even if there are temporal misalignments. The clustering was performed using the longitude and latitude coordinates alone. Omission of altitude proved effective as this lowered the dimensionality of the space. The altitude profile is also rather consistent between all trajectories, rendering this feature more informative. Figure 1 depicts the clustering of nominal trajectories on runway 14 for the case of four clusters centers, as an example.

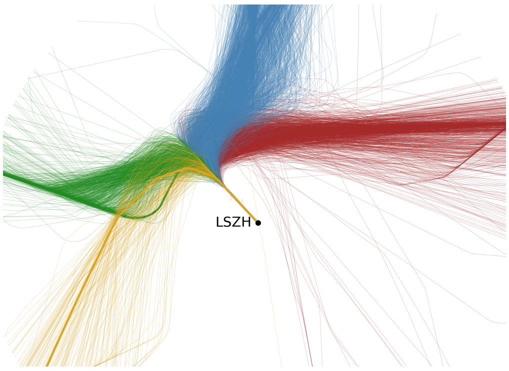


Figure 1. K-means clustering with DTW as a distance metric, for nominal landings on runway 14 (N = 4), as an example

C. *TimeGAN Model Architecture*

TimeGAN [4] synthesizes realistic time-series data using a combination of autoregressive and adversarial methods. The architecture consists of four main components (depicted in Figure 2): the embedding network, the recovery network, the sequence generator, and the sequence discriminator. The embedding network converts high-dimensional time-series data into a compact latent space using RNN layers such as LSTMs or GRUs. The recovery network reconstructs the original time-series data from the latent space representation, ensuring accurate data preservation. The sequence generator produces synthetic latent sequences, which are then mapped back to the original data space by the recovery network. The sequence discriminator differentiates real sequences from synthetic ones, providing feedback to improve the generator's performance.

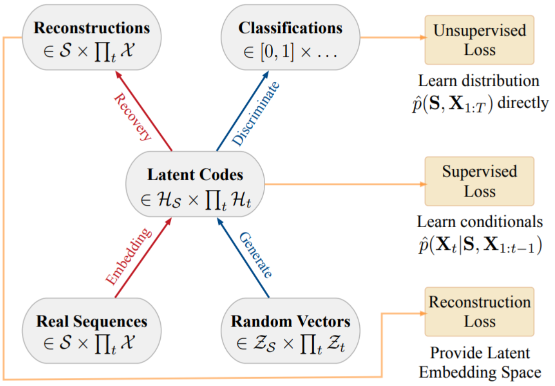

Reconstructions
$\in \mathcal{S} \times \prod_t \mathcal{X}$
Classifications
$\in [0,1] \times \ldots$
Unsupervised Loss
Learn distribution
$\hat{p}(\mathbf{S}, \mathbf{X}_{1:T})$ directly
Recovery
Discriminate
Latent Codes
$\in \mathcal{H}_\mathcal{S} \times \prod_t \mathcal{H}_t$
Supervised Loss
Learn conditionals
$\hat{p}(\mathbf{X}_t | \mathbf{S}, \mathbf{X}_{1:t-1})$
Embedding
Generate
Real Sequences
$\in \mathcal{S} \times \prod_t \mathcal{X}$
Random Vectors
$\in \mathcal{Z}_\mathcal{S} \times \prod_t \mathcal{Z}_t$
Reconstruction Loss
Provide Latent Embedding Space

Figure 2. TimeGAN model architecture, retrieved from [4]. S, X, and Z are the static, temporal, and random vector spaces, respectively

TimeGAN uses a combination of supervised and unsupervised learning to capture both data distribution and temporal dynamics. The reconstruction loss ($\mathcal{L}_R$) ensures that the embedding and recovery networks accurately reconstruct the original data from the latent space. This loss is defined as the sum of the differences between the original $(\mathbf{s}, \mathbf{x}_{1:T})$ and reconstructed $(\tilde{\mathbf{s}}, \tilde{\mathbf{x}}_{1:T})$ data points, formulated in Equation 1 (from [4]). Where s and t refer to values of vectors in the static and temporal spaces.

$$\mathcal{L}_{\mathrm{R}} = \mathbb{E}_{\mathbf{s},\mathbf{x}_{1:T}\sim p}\left[\|\mathbf{s}-\tilde{\mathbf{s}}\|_2 + \sum_t \|\mathbf{x}_t - \tilde{\mathbf{x}}_t\|_2\right] \tag{1}$$

The generator is trained in two modes: open-loop and closed-loop. In the autoregressive mode, it uses its previous synthetic outputs ($\hat{h}_S$ and $\hat{h}_{1:t-1}$) to generate the next synthetic vector ($\hat{h}_t$). This unsupervised loss aims to trick the discriminator, and is formulated in Equation 2 (from [4]). y_S and y_t are the discriminator's outputs for real data, and $\hat{y}_S$ and $\hat{y}_t$ for synthetic data.

$$\begin{aligned}\mathcal{L}_U = \mathbb{E}_{s,x_{1:T}\sim p}&\left[\log y_S + \sum_t \log y_t\right] \\ &+ \mathbb{E}_{s,x_{1:T}\sim \hat{p}}\left[\log(1-\hat{y}_S) + \sum_t \log(1-\hat{y}_t)\right]\end{aligned} \tag{2}$$

In tandem, a supervised loss ($\mathcal{L}_S$) ensures that the generator models step-by-step relationships in time-series by using real data embeddings ($h_{1:t-1}$), computed by the embedding network to generate the next latent vector, in Equation 3 (from [4]).

$$\mathcal{L}_S = \mathbb{E}_{s,x_{1:T}\sim p}\left[\sum_t \|h_t - g_X(h_S, h_{t-1}, z_t)\|^2\right] \tag{3}$$

D. Evaluation Framework

The evaluation framework used to determine the quality of synthetic samples rests on four pillars (following a similar approach as proposed in [4]): data diversity, fidelity, usefulness, and a statistical e-distance metric.

To assess *diversity* of synthetic time series data, a dimensionality reduction approach was implemented to visualize and compare the distribution of original and synthetic samples, using Principal Component Analysis (PCA) [21] and t-Distributed Stochastic Neighbor Embedding (t-SNE) [22]. PCA provided a 2D representation of the data, facilitating a comparison of distributions, whereas t-SNE emphasized local structures and neighborhood relationships. By examining these visualizations, one can determine whether the synthetic data effectively captures the original diversity [4].

To assess the *fidelity* of synthetic time series data, a discriminator model was employed to evaluate the ability of a network to distinguish between original and synthetic data. This model, comprising three LSTM layers followed by a dense output layer, was compiled with the Adam optimizer and binary cross-entropy loss function. It was trained to classify samples as either original or synthetic [4]. Following training, the discriminator was evaluated on separate test sets of original and synthetic data. The model's performance was quantified by its accuracy through the generation of a confusion matrix, specifically the True Positive Rate (TPR), True Negative Rate (TNR), and Accuracy. In this context, positive indicates synthetic and negative indicates real samples.

The *usefulness* of data was evaluated by measuring the predictive performance of an LSTM model trained on synthetic versus real data, relying on a time series regression approach. During training, the model was optimized using the Adam optimizer and mean absolute error (MAE) as the loss function. After training, the performance of each model was evaluated on the real test data to measure how well the synthetic data-trained model generalized to unseen real data. The MAE of predictions from models trained on both synthetic and real data was compared, providing insight into the predictive usefulness of the synthetic data [4]. This process facilitated a comparison of the predictive performance and overall usefulness of the synthetic data relative to the real data.

The *energy distance* metric was used to evaluate the similarity between synthetic and real time series data. The energy distance measures the divergence between the distributions of two sets of random vectors [23]. It approaches a positive constant if the distributions are similar and increases towards infinity if they differ significantly. Energy distances were computed over 100 random subsets from both the true and synthetic datasets. The mean energy distance was then used to assess which generation method produces synthetic trajectories whose distribution is closest to the observed data [3]. A lower mean energy distance indicates that the synthetic data more closely approximates the distribution of the real data.

E. Experiments

The exploratory nature of this study is reflected in the design of three experiments. These are aimed at investigating data quality of synthetic samples when TimeGAN is trained on either clustered groups (using k-means with DTW), separate runway assignments, and flight operation (normal, go-around, holding pattern). The following experiments were performed:

- *Experiment 1: Cluster-Based Generation* - Investigated TimeGAN's ability to generate trajectories when trained on clustered groups of flight data. Trajectories were grouped into five clusters using k-means with Dynamic Time Warping (DTW), with each cluster representing different flight patterns. The model was trained separately on each cluster to learn specific trajectory characteristics.
- *Experiment 2: Minority Runway-Specific Generation* - Investigated TimeGAN's ability to generate landing trajectories associated with different runway assignments. It focused on simulating landings on two distinct runways - 28 and 34 - each with unique approach patterns. Whereas runway 14 has a high frequency of use, data for runways 28 and 34 is limited, presenting a heavy data class imbalance challenge for the model generalization.
- *Experiment 3: Rare Operational Scenarios* - Investigated TimeGAN's ability to generate data for less frequent and challenging flight operations (i.e. go-arounds and holding patterns). These patterns contains more abrupt and sustained maneuvers and constitute a small minority in historical data.

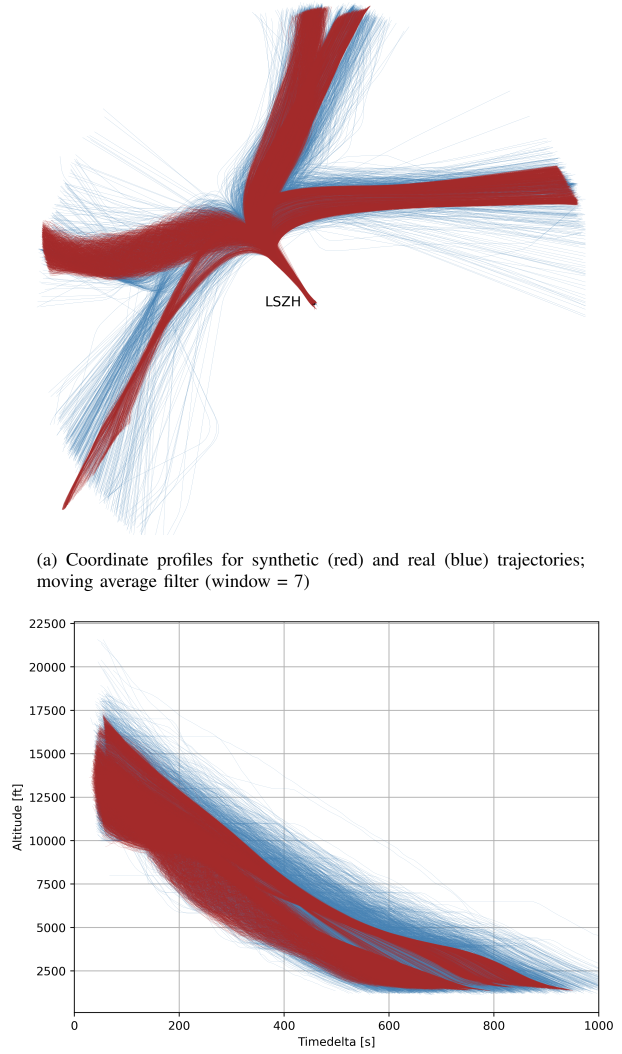
IV. Results

This section provides an analysis and evaluation of the synthetic data generation capabilities of TimeGAN through the three experiments described in Section III.F. Insights into data quality and model adaptability under different conditions are presented.

A. *Experiment 1: Cluster-Based Generation*

For the set of landing trajectories on runway 14, sampled at 100 data points, filtered for a maximum cumulative track angle change of 180 degrees, TimeGAN was trained on each cluster separately. The data imbalance for trajectories per cluster is evident considering the range of 364 to 1212 trajectories per cluster. The complexity of aircraft trajectories closer to the 40 nautical mile cut-off from an airport is significantly higher. Within this critical distance, aircraft must adhere to Standard Terminal Arrival Routes (STARs), which are intricate with multiple way points, altitude restrictions, and speed constraints to ensure safe and orderly arrivals. Adverse weather conditions and noise abatement procedures further complicate the approach phase, requiring deviations from standard procedures and additional vectoring, all of which contribute to noisy and intricate data. To remove unwanted noise, a smoothing filter, such as a simple moving average, may be applied during post-processing. The obtained synthetic trajectories for the separate classes are combined and plotted in Figure 3, after the application of filter with a rolling window of seven data points.

The trajectory distribution is wider and sparser at the start of the trajectories when the aircraft is in the air, 40 NM from the runway, reflected in the difficulty of TimeGAN to properly capture this. The latter half of each trajectory is often considerably smoother, whereas the start sometimes contains jagged paths (resembling noise) due to overfitting of the model. With aircraft ground speed being a decreasing function over time and sampling performed at fixed time intervals, data points at the start of each trajectory are more separated. This translates into a differential complexity along a trajectory that the GAN must capture. The distribution of longitude, latitude, and altitude become more narrow with a function of time. GANs are notoriously unstable during training due to the adversarial dynamics between the generator and discriminator, which can lead to significant variations in performance and realism across different clusters of landing trajectories. When training GANs with the same parameters across various clusters, the instability can become pronounced because each cluster may have distinct characteristics and data distributions. This discrepancy can cause the GAN to perform well on some clusters while struggling with others, as the generator might be better at capturing the features of more common or less complex clusters but fail to generalize to more varied or less frequent ones. In Figure 3, it is evident that TimeGAN struggles the most with flights approaching from the west as there is a higher concentration of noise. Flights from the southwest are less frequent, and TimeGAN falls victim to mode collapse. The remaining three clusters are well captured by the model, and the synthetic samples thereof are of high quality.



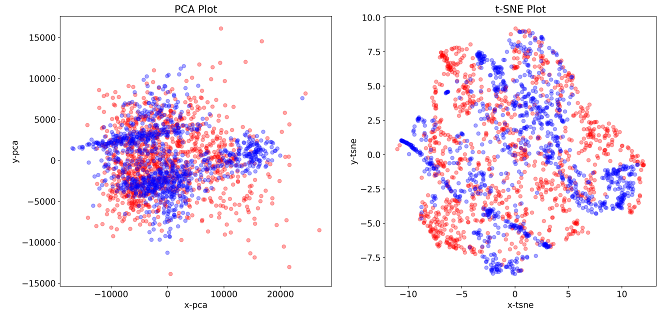
(a) Coordinate profiles for synthetic (red) and real (blue) trajectories; moving average filter (window = 7)

(b) Altitude profiles for synthetic (red) and real (blue) trajectories; moving average filter (window = 7)

Figure 3. TimeGAN synthetic samples (per cluster)

Pertaining to the full set, both PCA and t-SNE analysis demonstrate reasonable coverage of the trajectory distribution, largely steering clear of mode collapse (visualized in Figure 4). In both cases however, clear clustering is observed, as opposed to the more uniform spread in the original data. During hyperparameter tuning, the risk of mode collapse proved to be proportional to the number of training steps. Simultaneously, an inadequate number of training steps leads to notable mode collapse at the start of a trajectory, where the model seems to resort to an averaging of the real data, with a minuscule standard deviation in the starting point.

The results for the assessment of data fidelity and usefulness are provided in Table I. Pertinent to data usefulness, the Train-Real-Test-Real (TRTR) and Train-Synthetic-Test-Real (TSTR) n-step (n = 5) ahead prediction performance figures are included. The quoted value is a percentage change in



(a) PCA (all landings)

(b) t-SNE (all landings)

Figure 4. PCA & t-SNE analysis for the combined set of trajectories; synthetic (blue) and real (red) trajectories

performance (in MAE) of the TSTR compared to the TRTR evaluation paradigm. Lastly, e-distance provides a quantitative metric for the distance between the synthetic and real trajectory distributions complementary to the qualitative PCA and t-SNE analyses. The mediocre accuracy indicates the trained discriminator is deceived often. Real and synthetic samples are difficult to differentiate, and the higher TPR indicates that this largely stems from synthetic samples misclassified as real. The low e-distance value suggests a high degree of similarity between the real and synthetic data distributions.

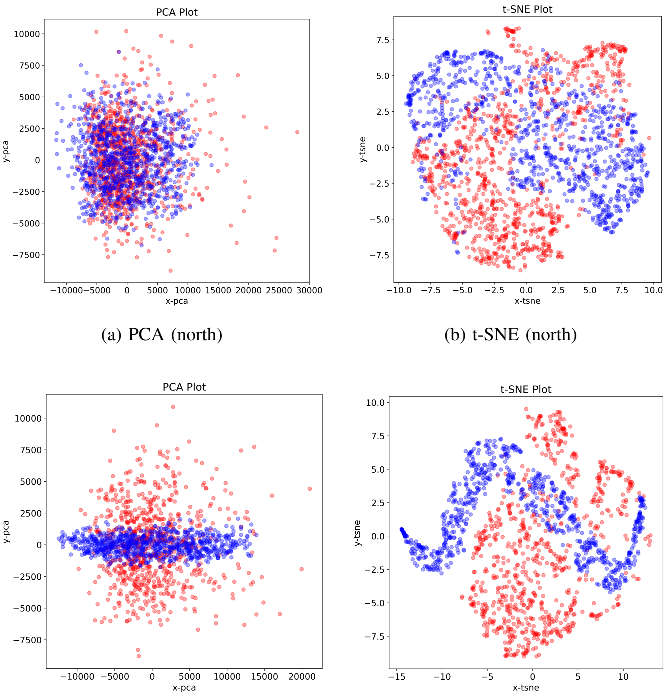
With regard to usefulness, the trained LSTM seems to learn similar patterns for both datasets. Simultaneously, this diminishes the *usefulness* of the augmentation with synthetic samples for predictive power.

TABLE I. PERFORMANCE METRICS (ALL LANDINGS)

Metric	Value
Accuracy	0.608
True Positive Rate (TPR)	0.947
True Negative Rate (TNR)	0.269
TSTR vs. TRTR	-0.473%
E-distance	1.482

The varying performance of TimeGAN for different clusters is clearly visible in PCA and t-SNE plots for separate clusters. Figure 5 depicts the results for two clusters with a large discrepancy in diversity and similar coverage of the synthetic samples. This coverage can be matched with visual inspection of the trajectories in Figure 3. Whereas PCA aligns, the t-SNE plot suggest considerable latent differences between the real and synthetic samples in both clusters. Table II contains the remaining performance indicators for each of the five clusters, containing similar variations. In some cases, figures are less representative due to fewer samples within a cluster.

In many cases, the LSTM model performing binary classification cannot accurately distinguish between real and synthetic samples. In the case of flights approaching from the southwest, the real and synthetic sample are seen to be most accurately distinguished. Interestingly, the results for the TSTR and TSTR evaluation metrics prove an indifference in prediction ability (except for flights originating from the southwest).



(a) PCA (north)

(b) t-SNE (north)

(c) PCA (southwest)

(d) t-SNE (southwest)

Figure 5. PCA & t-SNE analysis for separate sets of trajectories; synthetic (blue) and real (red) trajectories

TABLE II. PERFORMANCE METRICS (PER CLUSTER)

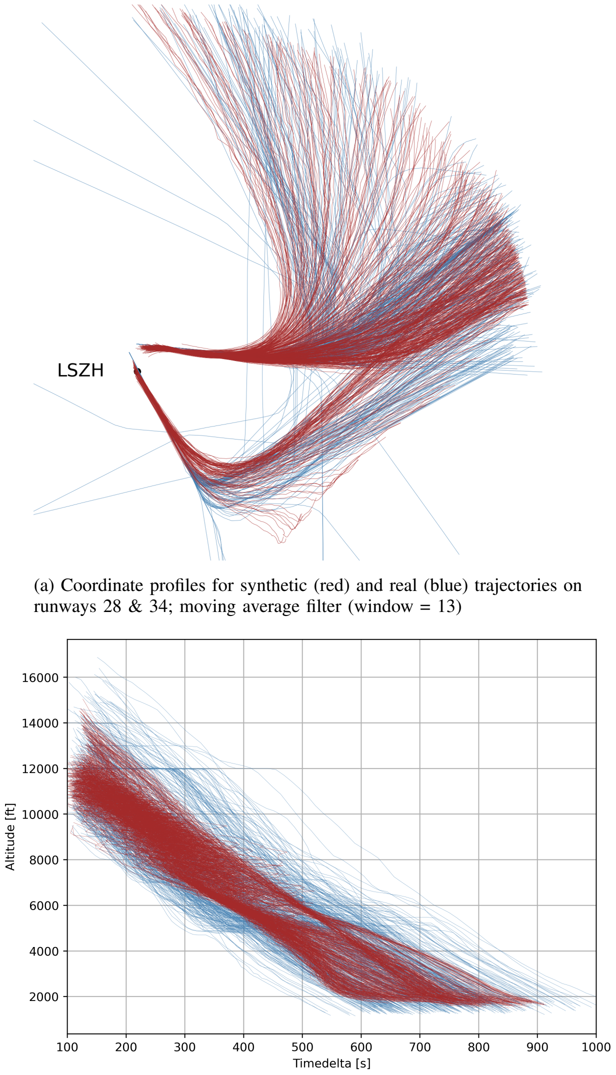
Metric	N	NE	E	SW	W
Accuracy	0.737	0.651	0.704	0.869	0.728
TPR	0.707	0.697	0.722	0.893	0.745
TNR	0.767	0.603	0.685	0.844	0.709
TSTR vs. TRTR	-0.224%	-0.127%	-0.152%	-2.401%	-0.122%
E-distance	2.880	4.621	5.500	4.856	3.210

The e-distance values all approach a constant positive value, which by the design of this test point to similarity in the distributions. This ensures sampling from the learned model generates consistently realistic results.

B. *Experiment 2: Minority Runway-Specific Generation*

The results for runway 28 and runway 34 (depicted in Figure 6), demonstrated the effectiveness of TimeGAN in generating realistic synthetic landing trajectories despite the limited data available. Performance can partially be attributed to the choices made during pre and post-processing stages. The complexity was capped with a maximum cumulative track angle (180 degrees) and low sampling rate (75).

Despite the lower data count for runway 34 leading to some noise during the final turn towards runway alignment, the combination of the preprocessing techniques helped mitigate data sparsity issues and allowed TimeGAN to generate trajectories that were closely aligned with real landing patterns. The altitude profiles suffered more from noise, even with the application of a moving average filter. This noise concentrated at the start of the trajectories and points at a differential over-



(a) Coordinate profiles for synthetic (red) and real (blue) trajectories on runways 28 & 34; moving average filter (window = 13)

(b) Altitude profiles for synthetic (red) and real (blue) trajectories on runways 28 & 34; moving average filter (window = 13)

Figure 6. TimeGAN synthetic samples (per runway)

underfitting phenomenon along the temporal dimension, as well as between geographic and altitude features. Trajectories landing on runway 28 were more diverse according to both PCA and t-SNE analysis. This can largely be attributed to the fact that runway 28 contained nearly four times as many historical samples to train from. Additionally, flights approached it from more varied angles. This higher variation on runway 34 combined with its inherent rarity for landings put strain on the stability of the training process. Overall, the data fidelity and usefulness analysis demonstrates that TimeGAN performed relatively consistently across both runways, with similar accuracy metrics and slight differences in e-distance (III). This finding is significant because it indicates that the model's performance does not degrade substantially with fewer samples. The lower E-distance for runway 28 suggests that the synthetic trajectories for this runway were slightly closer to the real data, supported by the diversity assessment results.

TABLE III. Performance metrics (per runway)

Metric	Runway 28	Runway 34
Accuracy	0.694	0.704
TPR	0.528	0.722
TNR	0.861	0.685
TSTR vs. TRTR	-0.129%	-0.144%
E-distance	3.198	3.795

C. *Experiment 3: Rare Operational Scenarios*

This experiment highlights significant challenges in incorporating complex flight maneuvers like holding patterns and go-arounds into TimeGAN training. Holding patterns, as in Zurich's airspace, involve intricate circular routes with variable entry points depending on daily conditions. The problem intensifies when integrating them with the full trajectory. Aircraft can enter the same holding pattern from various directions, creating a high-dimensional space that the current dataset struggles to cover adequately. A similar conclusion is made with regard to the mere 50 go-around samples available in the dataset. This led to difficulties in capturing the full range of maneuver behaviors, with TimeGAN prone to overfitting noise from the remaining trajectory and failing to generate realistic representations. Instead, following a similar procedure as laid out in [14], go-arounds were pre-processed to start when the aircraft initiates the go-around, and ends when it has landed. With fewer than 50 samples it remained infeasible to generate realistic samples and resulting trajectories were unflyable. To address these issues, transfer learning offers a valuable strategy. By leveraging models pre-trained on standard landings, TimeGAN can apply learned representations to better handle the complexities of go-arounds and holding patterns.

V. Discussion

TimeGAN demonstrated the potential to generate high quality synthetic aircraft landing trajectories, but encountered challenges, especially when dealing with data imbalance. Specifically, there was a significant challenge in adequately capturing the underlying trajectory distribution in sparser areas, notably comparatively rare operational conditions. Training and generation based on individual clusters proved to be a promising approach. However, this method was highly sensitive to the number of cluster centers, meaning that careful consideration is needed when clustering the data to avoid poor generation quality in less populated clusters. Generation of landings on specific runways could also be effectively trained for separately, but operational conditions (go-arounds and holding patterns) proved infeasible. Sparser regions of trajectory data distributions could be learned by optimizing the sampling rate parameter in the preprocessing step. A lower sampling rate reduces the dimensionality and computational burden, simplifying the data and making it easier for the GAN to capture temporal dependencies, particularly in shorter sequences. However, this simplification may result in the loss of important details, producing less accurate and varied outputs. Therefore, finding the optimal sampling rate is crucial:

a higher rate enhances detail and complexity, while a lower rate reduces complexity but risks oversimplification.

Previous work centered around go-arounds used a two-step process of dimensionality reduction and multivariate statistical distribution modeling [14]. By filtering and normalizing a dataset of 407 trajectories and applying a method to map two-dimensional aircraft paths to a single dimension, the authors achieved more realistic synthetic data with Gaussian mixture models (GMM) and statistical copulas. However, the study's limitations included its specificity to a single airport and runway, the exclusion of irregular flight movements, and the overall lacking potential for generalizability. Compared to the 400 go-arounds in their dataset, the 45 available in this case proved insufficient. In [15], the authors applied a convolutional GAN to generate landing trajectories and detect anomalies at Paris Orly Airport, focusing on Airbus A320 aircraft. Similar to TimeGAN in some cases, this approach required smoothing to remove high-frequency noise. The authors relied on trajectories that started at 25 nautical miles from the airport, inevitably decreasing complexity and variation compared to the 40 nautical mile cut-off used in this study. No clustering or labeling of trajectories was performed, limiting the controllability of the generation process. The most recent work involved a integrated a temporal convolutional network (TCN) to capture sequential dependencies and a Variational Mixture of Posteriors (VampPrior) for the prior distribution, in a VAE architecture [3]. The authors used the same data but excluded go-arounds from the set. Sparser regions of the learned latent space resulted in unrealistic trajectories. The e-distance values obtained for the TCVAE are lower than for TimeGAN, which could indicate greater similarity. Compared to all previous work, this study expanded the evaluation framework of synthetic samples. The inclusion of assessments on data diversity, fidelity, and usefulness were particularly valuable in revealing varying performance among classes.

This study's assumptions, while necessary for model development, introduce limitations that impact the generalizability and broader applicability of its findings. The focus on a specific airport and trajectory type assumes that these data are representative of other airports, which simplifies the modeling process but restricts the model's applicability to diverse environments. This assumption may not hold true across airports with different layouts, traffic patterns, or environmental conditions, potentially limiting the effectiveness of the generated synthetic data in broader ATM scenarios. Further research is needed to validate these models across diverse datasets and explore alternative or complementary approaches that can better capture the complexities of different ATM scenarios. The concept of a Universal Time Series Generator (UTG) provides the ultimate form of generalization. By training on data from a few specific airports, the UTG is able to generate synthetic flight path data for entirely different airports. This capability allows for the modeling and optimization of operations at unfamiliar and even hypothetical airports without needing real-world data from those specific locations. Incorporating controlled noise in generative models for flight trajectories could enable the synthesis of plausible, yet unobserved paths that anticipate future changes in airspace management. This approach challenges traditional notions of data fidelity and distribution-matching, suggesting a broader utility for models that extend beyond historical data. However, the unclear relationship between dataset characteristics and optimal clustering calls for further empirical research.

Conditioning Extension & Transfer Learning

Conditioning in generative models involves guiding the data generation process based on specific labels or constraints to produce outputs that adhere to certain criteria. This has been explored as an extension to work presented thus far, using the TTS-CGAN model [24]. In attempting to generate synthetic data for aircraft landing trajectories using TTS-CGAN, several technical challenges emerged that hindered the model's ability to produce realistic outputs. While conditioning on specific labels and utilizing advanced model architectures like self-attention layers certainly seem promising aiming to benefit from transfer-learning across classes, the model ultimately struggled to handle the complexity and variability of the data, failing to generate meaningful results. One significant reason the TTS-CGAN model may have failed to produce the desired results in synthetic data generation is due to the mismatch between the model's complexity and the available data. TTS-CGAN consists of a combination of a conditional GAN and self-attention mechanisms to generate time-series data that adhere to specific conditional labels, such as different clusters, runways, or operational conditions. The self-attention layers are particularly adept at capturing long-range dependencies within the data, theoretically allowing the model to generate more contextually consistent sequences. However, in the context of generating synthetic aircraft landing trajectories, the complexity of the self-attention mechanism can become a drawback. Given the high dimensionality and variability of the data—characterized by diverse trajectories under varying operational conditions, the model's self-attention mechanism may have become overwhelmed, resulting in outputs that are unexplainable and uninterpretable, overfitting to noise rather than capturing true flight dynamics. This limitation is compounded by the GAN's inherent training instability, particularly when the data distribution is complex and multi modal. The integration of more domain-specific knowledge (e.g. flight trends) or constraints directly into the model architecture, could be more effective in this application.

VI. Conclusions & Future Work

This research explored the effectiveness of GANs in generating realistic synthetic aircraft trajectories during the landing phase. Increased complexity in landing trajectories, represented by greater variability and higher dimensionality, posed significant challenges for the GAN models. The risk of mode collapse and overfitting was notable, emphasizing the need for a balanced approach in model training and evaluation. The results demonstrate that TimeGAN can successfully generate realistic synthetic aircraft trajectories, particularly when data is clustered and trained separately for each class. This clustering was essential in managing the complexity during training and ensuring that the generated trajectories were diverse and reflective of varied real-world operations. Key strategies that proved effective included setting a maximum cumulative track angle threshold to manage trajectory complexity and reducing the sampling rate to make the training process more computationally feasible. Post-processing techniques, by applying smoothing filters, further refined the synthetic data. Even in the event of severe minority classes, as was the case for runway use, TimeGAN proved effective at capturing the patterns in data, but required significant smoothing of the output time series. With respect to go-arounds and holding patterns, the available real datasets proved too limited for the model to capture the diversity yet simultaneously maintain realism. The variation in performance of TimeGAN between groups of data underscores the sensitivity of the training procedure to the underlying real distribution. Further testing of models across various airports and operational environments, as well as conditioning on factors such as weather and specific airport data, can provide additional insight into their adaptability and robustness. Incorporating flight simulators to assess flyability and physical realism, including complex trajectories, will further enhance the evaluation framework. However, a major challenge remains: the limited availability of diverse real-world data to train on. This paradox highlights the need for rich real data to produce high-quality synthetic data, emphasizing the intricate balance between innovation and reality in the field of trajectory generation.

REFERENCES

[1] Eurocontrol and Partners, “Fly ai report - demystifying and accelerating ai in aviation/atm,” European Aviaiton High Level Group on AI, Tech. Rep., 2020. [Online]. Available: https://www.eurocontrol.int/publication/fly-ai-report

[2] SESAR Joint Undertaking, “Sesar innovation pipeline - air traffic management research and innovation,” Publications Office of the European Union, Tech. Rep., 2023. [Online]. Available: https://www.sesarju.eu/innovation-pipeline

[3] T. Krauth, A. Lafage, J. Morio, X. Olive, and M. Waltert, “Deep generative modelling of aircraft trajectories in terminal maneuvering areas,” *Machine Learning with Applications*, vol. 11, p. 100446, March 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666827022001219

[4] 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

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems*, vol. 27. Curran Associates, Inc., June 2014. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[6] A. Nuic, D. Poles, and V. Mouillet, “Bada: An advanced aircraft performance model for present and future atm systems,” *International Journal of Adaptive Control and Signal Processing*, vol. 24, no. 10, pp. 850–866, May 2010. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/acs.1176

[7] J. Sun, J. M. Hoekstra, and J. Ellerbroek, “Openap: An open-source aircraft performance model for air transportation studies and simulations,” *Aerospace*, vol. 7, no. 8, July 2020. [Online]. Available: https://www.mdpi.com/2226-4310/7/8/104

[8] X. Olive, J. Sun, M. Murça, and T. Krauth, “A framework to evaluate aircraft trajectory generation methods,” in *14th USA/Europe Air Traffic Management Seminar, ATM2021*, September 2021. [Online]. Available: https://research.tudelft.nl/en/publications/a-framework-to-evaluate-aircraft-trajectory-generation-methods

[9] T. Gräupl, “Validation of simulated synthetic air traffic against recorded air traffic in germany,” in *Integrated Communications Navigation and Surveillance (ICNS)*, January 2017, pp. 2063–2068. [Online]. Available: https://elib.dlr.de/111344/

[10] F. Hoffmann, U. Epple, M. Schnell, and U.-C. Fiebig, “Feasibility of ldacs1 cell planning in european airspace,” in *2012 IEEE/AIAA 31st Digital Avionics Systems Conference (DASC)*, October 2012, pp. 1–17. [Online]. Available: https://ieeexplore.ieee.org/document/6383053

[11] C.-H. Rokitansky, M. Ehammer, and T. Graupl, “Newsky - building a simulation environment for an integrated aeronautical network architecture,” in *2007 IEEE/AIAA 26th Digital Avionics Systems Conference*, 2007, pp. 4.B.4–1–4.B.4–11. [Online]. Available: https://ieeexplore.ieee.org/document/4391905

[12] D. Jacquemart and J. Morio, “Conflict probability estimation between aircraft with dynamic importance splitting,” *Safety Science*, vol. 51, no. 1, pp. 94–100, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925753512001300

[13] M. C. Rocha Murça and M. de Oliveira, “A data-driven probabilistic trajectory model for predicting and simulating terminal airspace operations,” in *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*, October 2020, pp. 1–7. [Online]. Available: https://ieeexplore.ieee.org/document/9256644

[14] T. Krauth, J. Morio, X. Olive, B. Figuet, and R. Monstein, “Synthetic aircraft trajectories generated with multivariate density models,” *Engineering Proceedings*, vol. 13, no. 1, 2021. [Online]. Available: https://www.mdpi.com/2673-4591/13/1/7

[15] G. Jarry, N. Couellan, and D. Delahaye, “On the use of generative adversarial networks for aircraft trajectory generation and atypical approach detection,” in *Air Traffic Management and Systems IV*, 2021, pp. 227–243. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-33-4669-7_13

[16] X. Olive, “traffic, a toolbox for processing and analysing air traffic data,” *Journal of Open Source Software*, vol. 4, p. 1518, 2019.

[17] A. Khattak, J. Zhang, P.-W. Chan, F. Chen, and C. M. Matara, “Ai-supported estimation of safety critical wind shear-induced aircraft go-around events utilizing pilot reports,” *Heliyon*, vol. 10, no. 7, p. e28569, January 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405844024046000

[18] K. Artiouchine, P. Baptiste, and C. Dürr, “Runway sequencing with holding patterns,” *European Journal of Operational Research*, vol. 189, no. 3, pp. 1254–1266, September 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221707005978

[19] R. Tavenard, J. Faouzi, G. Vandewiele, F. Divo, G. Androz, C. Holtz, M. Payne, R. Yurchak, M. Rußwurm, K. Kolar, and E. Woods, “Tslearn, a machine learning toolkit for time series data,” *Journal of Machine Learning Research*, vol. 21, no. 118, pp. 1–6, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-091.html

[20] D. J. Berndt and J. Clifford, “Using dynamic time warping to find patterns in time series.” in *KDD Workshop*. AAAI Press, April 1994, pp. 359–370. [Online]. Available: http://dblp.uni-trier.de/db/conf/kdd/kdd94.html#BerndtC94

[21] F. Bryant and P. Yarnold, “Principal-component analysis and exploratory and confirmatory factor analysis,” January 1995. [Online]. Available: https://psycnet.apa.org/record/1995-97110-004

[22] L. van der Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, August 2008. [Online]. Available: http://jmlr.org/papers/v9/vandermaaten08a.html

[23] G. Szekely and M. Rizzo, “Testing for equal distributions in high dimension,” *InterStat*, vol. 5, November 2004. [Online]. Available: https://www.researchgate.net/publication/228918499_Testing_for_equal_distributions_in_high_dimension

[24] X. Li, A. H. H. Ngu, and V. Metsis, “Tts-cgan: A transformer time-series conditional gan for biosignal data augmentation,” *arXiv preprint arXiv:2206.13676 [cs]*, June 2022. [Online]. Available: https://arxiv.org/abs/2206.13676