Improving ED admissions forecasting by using generative AI: an approach based on DGAN

Alvarez-Chaves, H.; Spruit, M.; Moreno, M.D.R.

Computer Methods and Programs in Biomedicine 256 (2024) 108363

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: https://www.sciencedirect.com/journal/computer-methods-and-programs-in-biomedicine

Improving ED admissions forecasting by using generative AI: An approach based on DGAN

Hugo Álvarez-Chaves [a,*], Marco Spruit [b], María D. R-Moreno [a]

[a] Universidad de Alcalá, Escuela Politécnica Superior, 28805, Madrid, Spain
[b] Leiden University Medical Center, Department of Public Health and Primary Care, 2333 ZA, Leiden, The Netherlands

ARTICLE INFO

Keywords:
Generative AI
Data augmentation
Generative adversarial networks
Emergency department
DoppelGANger

ABSTRACT

Background and Objective: Generative Deep Learning has emerged in recent years as a significant player in the Artificial Intelligence field. Synthesizing new data while maintaining the features of reality has revolutionized the field of Deep Learning, proving to be particularly useful in contexts where obtaining data is challenging. The objective of this study is to employ the DoppelGANger algorithm, a cutting-edge approach based on Generative Adversarial Networks for time series, to enhance patient admissions forecasting in a hospital Emergency Department.

Methods: We employed the DoppelGANger algorithm in a sequential methodology, conditioning generated time series with unique attributes to optimize data utilization. After confirming the successful creation of synthetic data with new attribute values, we adopted the Train-Synthetic-Test-Real framework to ensure the reliability of our synthetic data validation. We then augmented the original series with synthetic data to enhance the Prophet model's performance. This process was applied to two datasets derived from the original: one with four years of training followed by one year of testing, and another with three years of training and two years of testing.

Results: The experimental results show that the generative model outperformed Prophet on the forecasting task, improving the SMAPE from 7.30 to 6.99 with the four-year training set, and from 22.84 to 7.41 for the three-year training set, all in daily aggregations. For the data replacement task, the Prophet SMAPE values decreased to 6.84 and 7.18 for four and three-year sets on the same aggregation. Additionally, data augmentation reduced the SMAPE to 6.79 for a one-year test set and achieved 8.56 for the two-year test set, surpassing the performance achieved by the same Prophet model when trained only on real data. Results for the remaining aggregations were consistent.

Conclusions: The findings of this study suggest that employing a generative algorithm to extend a training dataset can effectively enhance predictive models within the domain of Emergency Department admissions. The improvement can lead to more efficient resource allocation and patient management.

1. Introduction

The influx of patients to emergency services is a critical issue in healthcare. An increase in patient arrivals can lead to service overcrowding, potentially resulting in problems such as the transmission of infections among patients, delays in treatment, or directly an increase in the patient mortality rate, among others [1]. Moreover, resource optimization is essential to mitigate these problems and avoid the economic overcosts of the service. Effective patient flow management allows for the optimization of resource allocation, reducing overcrowding and improving both patient satisfaction and system operation.

A variety of Machine Learning-based (ML) approaches are already deployed to forecast patient arrivals. These range from more classical approaches using autoregressive or exponential algorithms [2–4], to modern techniques based on Neural Networks [3,5,6], XGBoost [6,7], Prophet [7,8], or Attention [9]. This problem has mainly been addressed from the perspective of time series, where patient influx is analyzed on an hourly, daily, weekly, or monthly basis. Nevertheless, as these data come from a sensitive health service and due to privacy restrictions, they can be difficult to obtain, leading to limited datasets.

The limitation in data availability can impact the ability of algorithms to learn and adapt optimally to the needs of emergency services. Therefore, finding methods to extend and enrich datasets while preserving the essential characteristics of the original series, is vital. This could include techniques like synthetic data generation or expanding

* Corresponding author.
E-mail addresses: hugo.alvarezc@uah.es (H. Álvarez-Chaves), m.r.spruit@liacs.leidenuniv.nl (M. Spruit), malola.rmoreno@uah.es (M.D. R-Moreno).

https://doi.org/10.1016/j.cmpb.2024.108363
Received 8 May 2024; Received in revised form 5 July 2024; Accepted 1 August 2024
Available online 8 August 2024

existing datasets with relevant and contextual information to improve the accuracy and effectiveness of predictive models and algorithms used in emergency services.

This paper explores the generation and extension of the admissions time series from a hospital Emergency Department (ED) by using a Generative Deep Learning model. For our work, we will use the algorithm DoppelGANger (DGAN) [10,11], a Generative Adversarial Network (GAN) designed specifically for time series to capture its characteristics based on attributes correlation. We aim to investigate whether DGAN can effectively predict patient influx to the ED and enrich the original dataset, thus overcoming the limitations of available data and providing a more robust and accurate model for resource management in emergency services.

To conduct our study, we will employ the Prophet forecasting model [12] as the base model, aiming to augment its effectiveness with synthetic data. Initially, we will tailor the DGAN model to optimize the format of the time series. Subsequently, we will assess the model's ability to generate representative time series that meet our contextual requirements. We also plan to use the generative model to substitute the initial dataset, a vital step that allows us to evaluate the algorithm's reliability using the Train-Synthetic-Test-Real (TSTR) method [13]. Finally, we intend to apply the generative model to augment the original dataset, followed by an analysis of the Prophet model's enhanced performance in handling this expanded dataset.

This work aims to contribute to the field of forecasting patient visits to ED by validating the use of generative techniques in improving the accuracy of prediction models. Furthermore, by using DGAN, which enables incorporating metadata to condition the generated series, we present a case study on generating data at the edge of its distribution. This involves conditioning the generated series with new values of the variables, such as different months and years, which were not encountered during the training phase.

This paper is structured as follows: the next section presents related works. Section 3 provides an insight into the data used, the algorithms (DGAN and Prophet), and the questions to face in this work. Section 4 presents the outcomes derived from the implementation of the methodology for the questions presented and the related discussion. Finally, some conclusions are outlined in Section 5.

2. Related work

Avoiding ED overcrowding is a widely studied field in the literature [14,15]. One of the main existing approaches is based on the prediction of admissions by using forecasting algorithms. In this regard, various ML models have been utilized, as can be seen in the systematic reviews of Gul and Celik [14] and Jiang et al. [15]. Several algorithms have been used to model the patients influx to the service, including Linear Regressors [7,16–18], Exponential Smoothing [2,3,7], Support Vector Regression, ARIMA [2,3,7], and Prophet [7,8]. Techniques based on Deep Learning, such as RNN [3], LSTMs [5,6], or using the attention mechanism [9], have also been used.

In the field of patient prediction, various studies have demonstrated the effectiveness of incorporating additional information into time series [5,15,19]. This strategy not only improves the accuracy of algorithms but also allows for a deeper understanding of underlying patterns. By including exogenous variables, the model is enriched with elements that extend beyond the original dataset from admissions. Among these variables, those related to the calendar, such as holidays and seasons, as well as environmental factors such as weather and air quality, have proven particularly useful [5,6,9,20–25]. Other studies utilized variables that come from information gathered from the Internet, such as search trends and socioeconomic data [9,17,20,26,27]. The incorporation of exogenous variables enables algorithms to capture information that would otherwise remain inaccessible, potentially facilitating the creation of more realistic and representative synthetic series.

Data collection in hospitals presents significant challenges, primarily due to the sensitive nature of health information and the associated privacy concerns. Recently, there has been a growing interest in the generation of synthetic data in this area, as evidenced by various studies [28,29]. Fundamental works such as that of Raghunathan et al. [30] have laid the groundwork for the use of synthetic data in multiple imputation for data disclosure limitation, demonstrating how these approaches can enhance confidentiality in public health surveys. These studies range from the creation of synthetic medical records for evaluating health policies [31] to the use of partial synthetic data in large-scale health surveys, which aids in protecting patient privacy while preserving the main data characteristics [32]. The use of synthetic data in natural language processing (NLP) has shown substantial benefits, as evidenced by the research conducted by Ive et al. [33]. The work focused on the generation of artificial mental health records to enhance the training of NLP models, thus improving the interpretation of textual data from medical records. Jiang et al. [34] employed synthetic images of COVID-19 computed tomography scans, marking a significant advancement in the simulation of medical image data during the pandemic. The study is located among various instances highlighting the generation of medical images within hospital environments [35–38]. Existing literature emphasizes the importance of synthetic data in situations where real data are limited or difficult to obtain. These examples underscore the growing importance and potential of synthetic data in the healthcare sector, not only for preserving privacy but also for enhancing the accuracy and effectiveness of medical research.

Among data generation methods, the creation of synthetic time series represents a special case extensively discussed in the literature [39,40]. They must maintain the temporal coherence of the intrinsic patterns of the series. In this domain, a diverse array of algorithms exists, ranging from the traditional and straightforward techniques such as Data Flipping, Magnitude Warping, Time Warping, and Jittering [40]. In recent years, generative approaches based on emerging architectures have been explored. Some of the most popular architectures are based on Autoencoders [41], and more specifically, by Variational Autoencoders [42], such as TimeVAE [43] or CR-VAE [44], Probabilistic AutoRegressive models like CPAR [45], Diffusion models [46], or various adaptations to time series of GANs [47–52].

DGAN [10,11] is a popular architecture based on adversarial networks to generate time series. This architecture facilitates the creation of time series by leveraging metadata intrinsic to the series, enabling conditioning based on specific external variables associated with them. Furthermore, the network design accommodates the production of multivariate series with variable lengths. DGAN has been applied in the literature to generate time series for web traffic [11,53], geographically-distributed broadband measurements [11], compute cluster usage measurements [11], economic recessions [54], energy grids [55], heat supply [56], and medical environments [57], among others.

The contribution of this paper to the state of the art is based on the application of a generative time series algorithm like DGAN to the context of patient admissions in an ED service. The context of the ED is marked by the challenge of working with typically small datasets, as a result of privacy constraints and limited time for digital records. Therefore, our aim is to improve the predictions of a forecasting model by using a generative algorithm to expand the original dataset. Most studies use generative models with large datasets; however, in this study, we will attempt to use one of them with a reduced dataset. Additionally, due to the ability to use metadata with DGAN, we will explore whether by managing such metadata, we can generate data at the distribution boundaries through metadata not seen during training. Therefore, this article seeks to provide a way to expand the dataset that allows for improved patient admissions forecasts in the ED, which could enable better service management through resource handling.

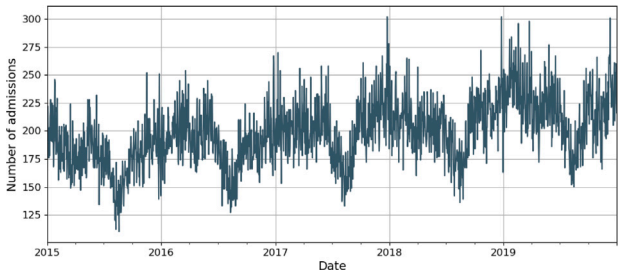

Fig. 1. Daily patient admissions in HCDGU ED service from January 1, 2015 to December 31, 2019.
Source: Figure from Álvarez-Chaves et al. [9].

3. Methods

To conduct our research, we aim to address four fundamental questions related to the data generation task. The first question concerns the dataset used and the search of the best possible data splitting to feed the generative algorithm with our data. The second question addresses whether the algorithm is capable of generating data for different ranges of time to the original data, by using metadata not previously seen during training. The third question seeks the feasibility of replacing the original dataset with an analogous synthetic dataset to test the predictive model with a fully synthetic dataset. Lastly, we extend the original dataset with synthetic data to determine whether this technique can be used for data augmentation. All these questions will be evaluated using a predictive algorithm to assess the synthetic data impact. In the next subsections, we present the data utilized for this work and its characteristics, the DGAN model for the synthetic data generation task, and the Prophet model to assess the synthetic data. Additionally, we will detail the responses to these research questions.

3.1. Emergency department data

The starting point for data generation in an ED context is the collection of historical ED patient records. In this regard, we acquired it from the Hospital Central de la Defensa Gómez Ulla (HCDGU) in Madrid, Spain. Our dataset includes five years of ED admissions data, covering the period from January 1, 2015, to December 31, 2019, with a total of 361 698 patient records and hourly aggregations. The raw series on these admissions on daily aggregations is shown in Fig. 1.

After analyzing the time series, we identified the existence of three marked patterns: daily, weekly, and yearly. We can extract these patterns through the series' temporal decomposition. The additive decomposition of the full series, shown in Fig. 2 for daily aggregation, reveals a notable upward trend with occasional peaks and drops. Moreover, this decomposition allows us to better understand seasonal variations and their impact on the overall data.

3.1.1. Daily pattern

To determine the daily pattern, we perform additive decomposition on the series forcing us to use the 24-hour daily seasonality. The outcomes are depicted in Fig. 3, illustrating the fluctuations over the course of a day. The figure shows three different periods: the first zone corresponds to the morning shift (from 8 to 15 h), where there is an increase in admissions until noon followed by a decline until 14:00.

This period has the highest concentration of emergencies; the second zone is similar to the afternoon shift (from 15 to 22 h), following a similar pattern to the morning zone but less pronounced in patients admitted, with an increase until mid-shift and a subsequent drop; the third zone concerns the night shift (22 to 8 h), and it is marked by a steady decline in patient numbers throughout the night, reversing only after 5 a.m. as patient counts begin to ascend again.

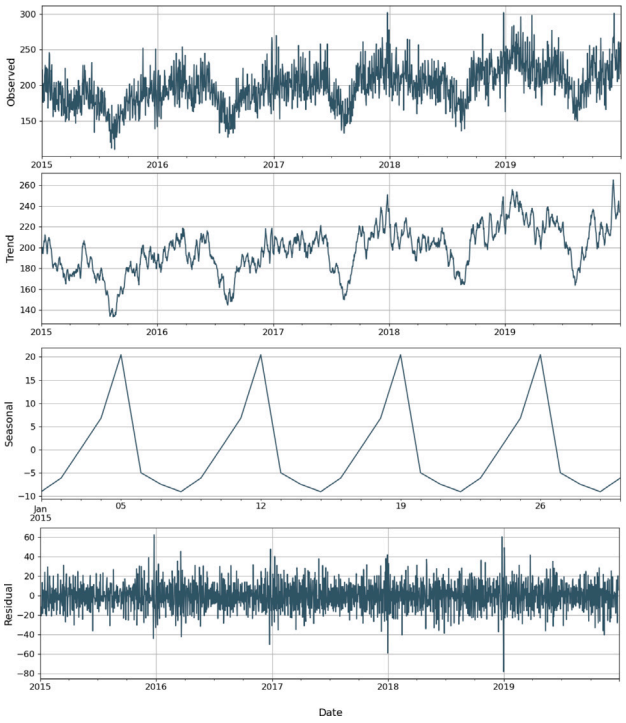

Fig. 2. Decomposition of the daily admission series from the HCDGU ED service. From top to bottom: raw series, trend, seasonality and residuals.
Source: Figure from Álvarez-Chaves et al. [9].

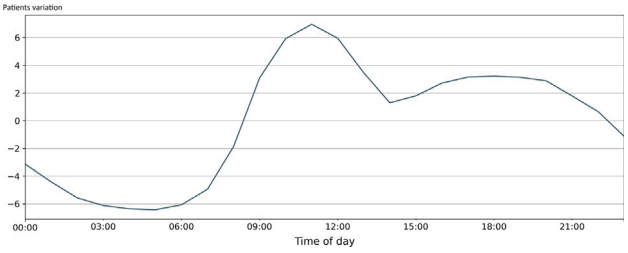

Fig. 3. Daily pattern of the HCDGU ED service. Extracted by using additive time series decomposition from the hourly admissions series.

3.1.2. Weekly pattern

Identifying the weekly pattern involves isolating the seasonal component that was identified in the previous decomposition. In Fig. 4, we illustrate the variation in patient numbers throughout the week, categorized by day. We note that Monday records the highest influx of patients, which then diminishes to reach its lowest on Thursday, before gradually climbing again towards Sunday. This observed pattern underscores the significant impact of weekend activity on the service, highlighting managers' importance in considering these trends in resource allocation.

3.1.3. Yearly pattern

To illustrate the yearly pattern, we follow a process that involves splitting the time series into the years comprising it, removing the trend, and normalizing each of the five sub-series independently. This approach standardizes the data across the same scale for better comparison. Fig. 5 shows the results of this described process, in addition to applying a 30-day centered sliding window to enhance its interpretation. It reveals that all years share similar patterns, with the most noticeable being a drop in the summer and a gradual increase towards the end of the year. This summer decline is particularly relevant in the context of the hospital's location, as the city of Madrid typically

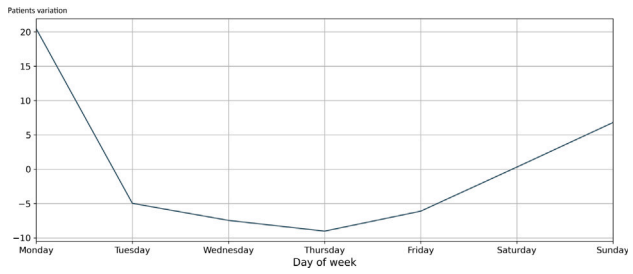

Fig. 4. Weekly pattern of the HCDGU ED service. Extracted by using additive time series decomposition from the daily admissions time series.

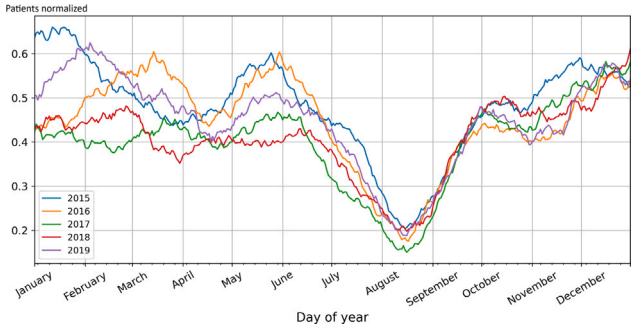

Fig. 5. Additional decomposition of the admission series from HCDGU ED service. The trend of the series was removed and it was normalized by year, applying a rolling window of 30 days to compare the years.
Source: Figure from Álvarez-Chaves et al. [9].

experiences a significant decrease in its usual population during the holiday seasons, according to reports from Instituto Nacional de Estadística (Statistics Spanish National Institute).[1]

The patterns we have detected in our study are unique to the particular scenario we are dealing with. However, our experience suggests that these foundational patterns are commonly observed in the EDs of numerous hospitals. In the study referenced as Álvarez-Chaves et al. [58], we utilized an alternative dataset and discovered evidence of the same three patterns outlined here.

As supplementary information, in Álvarez-Chaves et al. [9], the dataset used in this paper is presented in more detail, incorporating the connection with exogenous variables. The present work highlights the seasonal patterns as they will be crucial for the experiments to be detailed in the next sections.

3.1.4. Training and testing datasets

To perform our experiments, we use two different dataset configurations to test the generation capacity. The first configuration contains a train set of four years (2015–2018) and a test set with only the last year (2019). The second configuration reduces the train set down to three years (2015–2017), and uses two for testing purposes (2018–2019).

3.2. Algorithms

This work is based on synthetic data generation and its validation by using forecasting algorithms. Therefore, we will use two different algorithms, one for each purpose. For the synthetic generation of data, we will use the DGAN algorithm, since it allows us to condition the generated series through metadata. For the validation of the generated series, we will use the predictive algorithm Prophet due to its good performance in this context in previous works [7]. In the following sections, we will briefly describe both architectures.

[1] https://www.ine.es/prensa/experimental_em3.pdf

3.2.1. Generative model: DGAN

Among the generative models for time series explored, we have selected DoppelGANger (DGAN) [11], an architecture that stands out in the time series generation landscape. The model has the characteristic of being able to capture long temporal correlations and relationships between features and attributes while preventing mode collapse. Here, "attributes" refer to metadata or auxiliary information that can influence or describe the behavior of the time series—such as time labels (e.g. day of the week, month of the year...) or other categorical or continuous variables associated with the "features" (e.g. time series). DGAN was developed to improve the quality and diversity of the data generated, alleviating one of GAN's main problems. The algorithm is particularly beneficial for our objectives because it allows us to add attributes associated with the time series. We will use this feature to insert new attribute values that were not encountered during its training phase. In this regard, the algorithm is able to maintain the learned relationship between attributes and the time series. Consequently, we can manipulate these attributes to produce time series that either remain within the original dataset's time frame (replacing the original dataset with a synthetic one) or extend beyond it, moving either backward or forwards in time (data augmentation or forecasting).

The DGAN model, as a GAN architecture, consists of two different parts: the generator and the discriminator. The generator is composed of three different parts in a three-phase process to capture the relationship between time series and their respective attributes. The process begins with the attributes generation, which sets the parameters for subsequent phases. This is followed by establishing the minimum and maximum values for the time series, which is an optional part of the architecture. The first two layers each consist of a Multi-Layer Perceptron (MLP) with 2 hidden layers and 100 units in each layer. The process ends with the time series generation, also known as features. These features are conditioned by the initial attributes, which enables a more accurate and realistic control of the data produced. The feature generator comprises a single-layer LSTM network with 100 units. In our case, a sigmoid activation function is utilized for all components of the generation part. The use of LSTM units is appropriate for maintaining long-term dependencies, whereas MLP networks allow the capture of the specific dimensional and typological needs of the data. This combination effectively supports the generation of multiple time series that align with the set attributes.

The discriminator is built using an MLP-based strategy, introducing an innovative feature: the auxiliary discriminator. Therefore, DGAN contains two different discriminators. The main discriminator evaluates the entire sample, including both attributes and time series data, while the auxiliary discriminator focuses only on the attributes. This division allows for more precise feedback, which improves the generation of both high-fidelity attributes and the overall quality of the samples. In our case, we do not use the auxiliary discriminator due to we do not want to create new attributes. The discriminator is composed by 4 hidden layers and 200 units in each layer, similar to the auxiliary discriminator not used in our experiments. DGAN algorithm addresses the mode collapse problem using Wasserstein loss [59] with a gradient penalty of 10 during its training phase. This allows for improved stability and variety of its results, counteracting the limitations observed in traditional GAN models. The complete architecture is depicted in Fig. 6.

The algorithm also has the ability to create synthetic time series with varying lengths and is able to produce multivariate time series related to the same attributes. For our purposes, we only focus on univariate time series. This approach reduces complexity and aids the algorithm in learning from a smaller set of data. Additionally, the architecture presents a significant advantage in sensitive domains like healthcare acting as a defensive mechanism against Membership Inference Attacks, according to the authors. Further details on the algorithm and the discussion about its hyperparams can be explored in the work of Lin et al. [10,11].

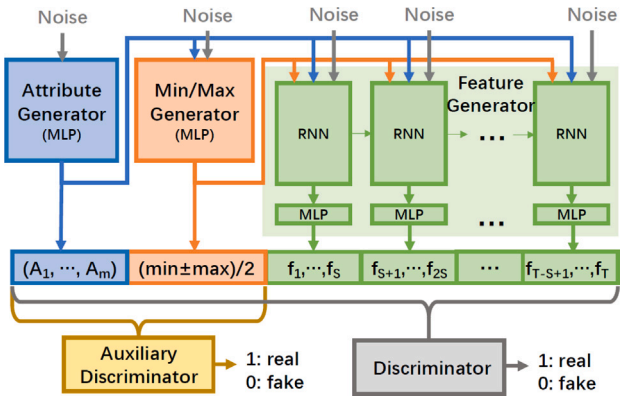

Fig. 6. DGAN model architecture.
Source: Figure from Lin et al. [11].

3.2.2. Forecasting model: Prophet

To conduct the experiments, we need to evaluate the synthetic data generated by the DGAN model using a model trained on real data to assess its performance. We have chosen the Prophet algorithm [12] for this purpose, based on our prior successful experiences with similar datasets to the ones utilized in this study [7].

Prophet is a model developed by Facebook and uses an approach based on the classic components of time series analysis: trend, seasonality, and noise. Additionally, it allows the introduction of exogenous variables such as events and holidays. The way Prophet models these components varies among them: for trend, it employs change points that allow for variable trends, and it can use logistic or linear models depending on the characteristics of the series; for seasonality, it uses Fourier series to extract different yearly, weekly, or daily seasonalities; for exogenous variables, the algorithm assigns a specific weight in training, which is used to make predictions for a date where the exogenous variable is present.

The Prophet algorithm includes hyperparameters for configuring its features, such as trend change points, which seasonalities to extract, and how these seasonalities are combined, among others. To simplify experimentation, we have decided to keep these hyperparameters stable throughout all fittings. This approach ensures a more realistic evaluation of the synthetic data since modification of the hyperparameters could result in models that are sufficiently different from each other to be incomparable. The hyperparameters we have chosen are detailed in Table 1 and correspond to a linear trend (growth param) with 25 trend points (n_changepoints param), a variation of 0.05 (changepoint_prior_scale param), a range of 80% for the changepoint range, and without specific changepoints. We use automatic seasonality detection (correctly detected by Prophet the daily, weekly, and yearly seasonalities) with a variation of 10 (seasonality_prior_scale param). All components are combined in an additive manner. In our work, we do not use the possibility of adding exogenous variables to the model. More details of the algorithm are available in the original paper [12].

3.3. Model evaluation questions

In this section, we explore critical questions that guide our inquiry into the effective deployment of the DGAN model for synthetic data generation. We examine the dataset's optimal partitioning strategy, the model's capability to generate data for unseen attribute values, the feasibility of replacing the real dataset with synthetic data for model validation, and the potential of data augmentation to improve model performance. These inquiries are pivotal for not only assessing the synthetic data's quality and utility but also for understanding how different data handling strategies affect the performance and reliability of our forecasting models.

Table 1
Prophet hyperparameters used in the entire experimentation.

Parameter	Value
Growth	Linear
n_changepoints	25
changepoint_prior_scale	0.05
specified_changepoints	False
changepoint_range	0.8
daily_seasonality	Auto
weekly_seasonality	Auto
yearly_seasonality	Auto
seasonality_prior_scale	10.0
All Components mode	Additive

3.3.1. Dataset split analysis

The first question concerns how to train the DGAN model to best fit our data. GAN models require multiple instances to extract its distribution during the training process, so we divide the entire time series into smaller fragments. In this way, and based on seasonality, we use different series partitions to try to capture the main features of the original series. Additionally, we make use of the attributes to condition the generated series, by using the temporal location of each series.

In this regard, we opt to divide the original series into three different splits according to seasonality:

1. Daily: 365 or 366 series of 24 values per year. The attributes introduced for this division are day, day of the week, month, and year.
2. Weekly: 53 series of 168 values per year. The attributes in this case are the week of the year, month, and year.
3. Yearly: 24 series of 365 or 366 values corresponding to an entire year at a specific hour. The attributes introduced are hour, year, and leap year, to indicate that the generated series should have an extra day.

To assess this question, we perform the remaining experiments using each of the detailed data-splitting methods. This approach allows us to explore the impact of seasonality on the performance of the algorithm, identifying which one best suits our needs and research objectives. By conducting a comparative analysis of the results, we aim not only to evaluate the training of the model generated using each splitting method but also to understand how the selection of a particular split may influence the adaptability of the DGAN model to our data.

It is noteworthy that the same attributes introduced for the DGAN architecture are also inserted into the Prophet algorithm as exogenous variables.

DGAN training

The DGAN training process requires the prior setting of its hyperparameters. In this regard, we use a batch size to fit a year's data, i.e., 365 for daily splits, 53 for weekly splits, and 24 for yearly splits. We employ a data normalization between [0–1], and a learning rate for both the generator and discriminator of $1e-4$. In addition, we do not use the attribute discriminator in our study, as generating synthetic attributes is not required. Regarding the number of epochs, they have been selected experimentally by using a two-step process. First, we set an initial number of 100K epochs by using a training set of the period 2015–2017, a validation set of one year (2018), and another year for testing (2019). We detected some changepoints at the epochs 5K for daily splits, and 10K for weekly and yearly splits, indicating a possible overfitted for a long number of epochs. Fig. 7 shows the evolution of the MAE metric for the long training and the number of epochs selected for subsequent more exhaustive analysis.

The second step is based on a visual inspection of the generated data patterns. We train DGAN with the full train (2015–2018 and 2015–2017, depending on the dataset used) during the epochs previously selected and save the model every 20 epochs. Then, we explore every

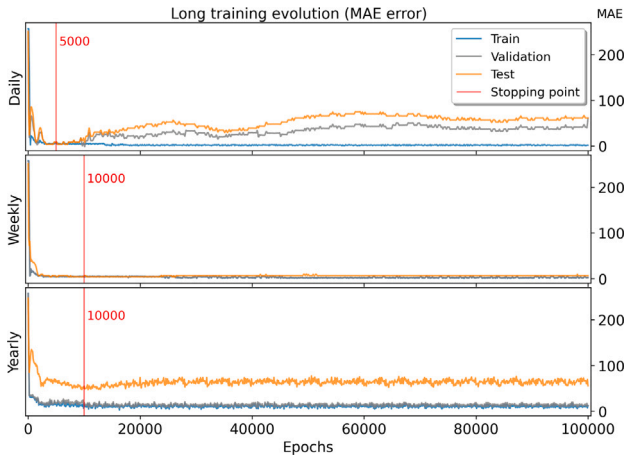

Fig. 7. Training evolution for the DGAN algorithm. We use three years of the training set (2015–2017), one as the validation set (2018), and one as the testing set (2019). MAE error was computed in each epoch to evaluate the first stopping point of our training process.

Table 2
Number of epochs selected for each partition and train set.

Train set	Partition	Number of epochs
2015–2018	Daily	2960
	Weekly	9400
	Yearly	9480
2015–2017	Daily	3540
	Weekly	5320
	Yearly	9700

model saved and generate 1000 synthetic series (for the training set range) to analyze their patterns. Finally, we select the model from the epoch where the patterns most closely match the actual data. The final epochs chosen are presented in Table 2. In addition, Fig. 8 shows the result of the seasonal decompose of synthetic data generated by the DGAN selected model for the daily division and the dataset with four years of training. We can observe that all patterns in real data are correctly learned. For the weekly splitting, we obtain similar results in training than the daily division. In contrast, the yearly splits could not learn the daily seasonality for the same dataset as shown in Fig. 9, but the yearly seasonality is captured more precisely.

Additionally, to evaluate the similarity between the synthetic data and the real training data, we compared the distributions for the same period as the training data. Table 3 shows the values of the Wasserstein-1 distance and Pearson correlation for an hourly aggregation. We observe that daily and weekly splits present consistent values, though sufficiently distant to indicate that the patterns are being learned without memorizing specific instances from the training set. In the case of the yearly partition, the metrics are considerably worse, which was expected due to the lack of learning the daily seasonality.

Using the training approach, we prioritize capturing the primary characteristics of the data while disregarding outliers that may be present in the series. To reconstruct the synthetic series, it is necessary to reassemble the series by concatenating the various fragments generated by the algorithm. For instance, to create a series with daily segmentation, we must concatenate each day's output produced by our method.

3.3.2. Data generation for unseen attribute values

The second issue concerns the behavior of DGAN with unseen attribute values. In our case, this feature is the key to extending the original time series, since we use it to establish the temporal location

Table 3
Wasserstein-1 distance and Pearson correlation (ρ) for the training set real against synthetic data for the same period and 1H aggregation. All ρ values are significant ($p_{value} < 0.05$). For Wasserstein-1 distance, lower values are better. For ρ higher valuer are better.

Train set	Partition	Wasserstein	ρ
2015–2018	Daily	0.87	0.82
	Weekly	0.79	0.83
	Yearly	1.27	−0.05
2015–2017	Daily	0.84	0.82
	Weekly	0.74	0.83
	Yearly	3.33	0.38

of the generated series. Consequently, attributes enable us to generate new series that extend both backward and forward beyond the original dataset.

In this scenario, to evaluate the model's performance with unseen attribute values, we utilize the initial data partition for both training and testing. We generate synthetic data for the temporal attributes associated with the test set to evaluate the model's predictive performance. Then, we assess its effectiveness and compare it to the performance of the Prophet algorithm in the forecasting task.

3.3.3. Replacing original data

The third question of our study aims to further strengthen the validity of the generated data. To this end, we apply the Train-Synthetic-Test-Real (TSTR) technique [13] using the Prophet model as a validation algorithm. This technique consists of replacing the original training dataset with synthetic data to determine if the generative model can replicate similar data without losing information from the original series during the training process. Afterward, once the Prophet model has been fitted with the synthetic data, it is tested using the real test set. To carry out this task, we use the temporal attributes of the training set to create a similar series that will be used to train the Prophet model.

3.3.4. Data augmentation

The last question arises from the previous ones and relates to extending the original training dataset. In this case, the goal is to expand the training dataset by going back in time and generating a new training set where real and synthetic data are merged for training purposes.

Nevertheless, increasing the original dataset requires a previous analysis. Introducing synthetic data should be done with the consideration that we might be biasing the forecasting model, preventing it from generalizing properly later on. To avoid this issue, we performed multiple fittings with different training datasets, wherein each fitting we increased the dataset. Thus, we analyze the model behavior when 1, 2, 3, and 4 years are added to the original training set. Thus, in the most long dataset case, we have 8 years of training for the case with 4 original years and 7 for the one with 3.

4. Results and discussion

In this section, we present the results obtained by the different tasks detailed in the previous section. First, we show the metrics assessment for each single question, and then we discuss the answers to the questions raised in this study.

4.1. Results

To visualize the results, we must first set some metrics to understand the scope of the results. In this case, we use the following metrics for each task:

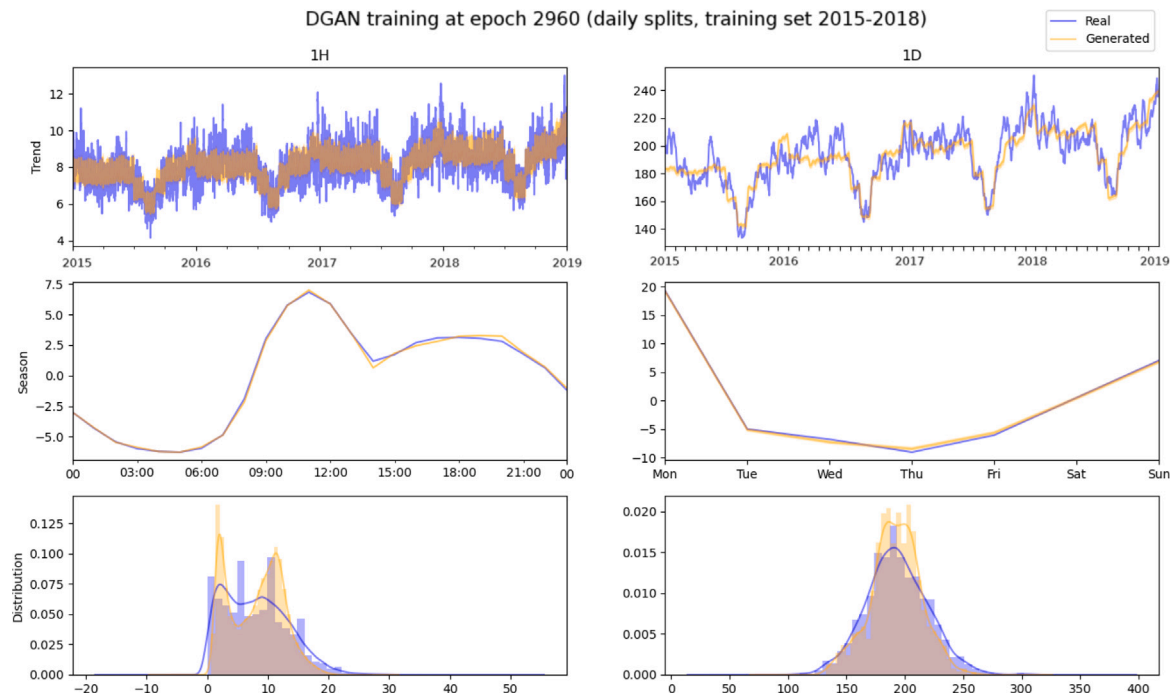

Fig. 8. DGAN seasonal decomposition for the training process at epoch 2960 for the daily split and the four-year training dataset. The algorithm captures correctly all seasonalities: daily (presented in 1H aggregations), weekly (presented in 1D aggregations), and yearly (presented in the trend).

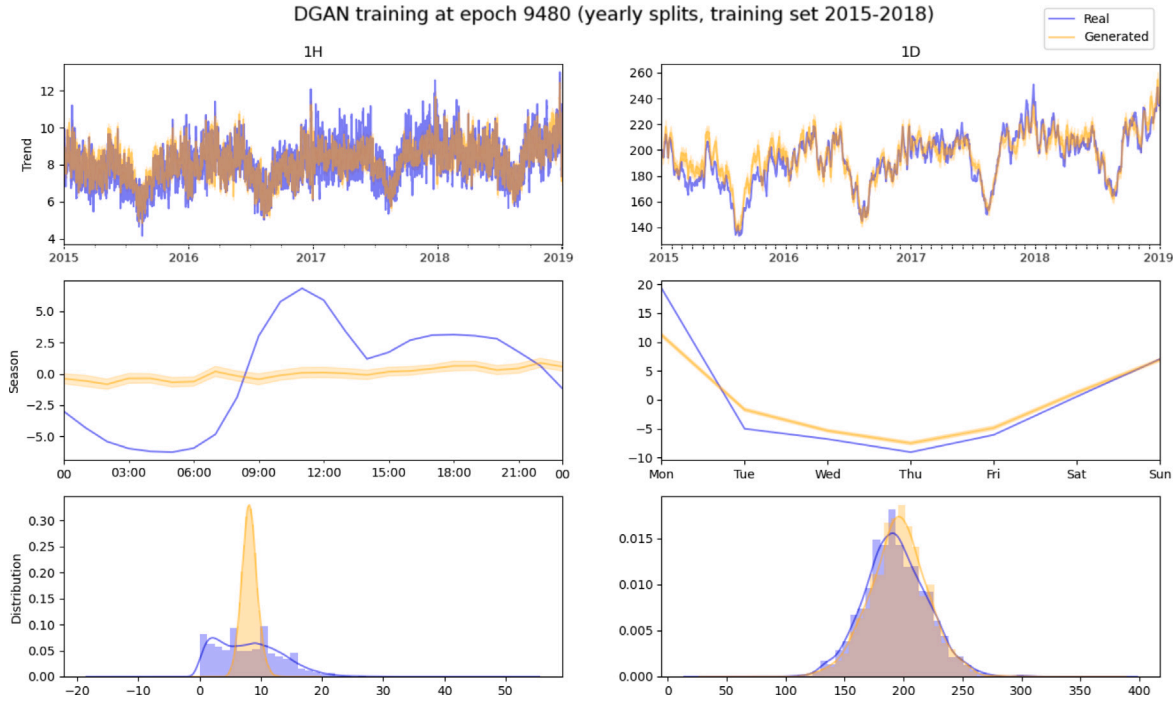

Fig. 9. DGAN seasonal decomposition for the training process at epoch 9480 for the daily split and the training dataset. The algorithm captures correctly two seasonalities: weekly (presented in 1D aggregations), and yearly (presented in the trend); but it is not able to capture the daily seasonality.

- SMAPE (Symmetric Mean Absolute Percentage Error). The error is presented as a percentage, computed according to Eq. (1). This approach addresses the issue encountered with the Mean Absolute Percentage Error (MAPE) when the actual value is zero. It facilitates the comparison of model accuracy across different use cases and datasets. Lower values indicate better accuracy.
- MAE (Mean Absolute Error). Refers to the mean of patient deviation (absolute value) in the forecasting task. This method provides a straightforward understanding of the error and is computed as in Eq. (2). Lower values are better, indicating less error.
- R^2 (Coefficient of Determination). It is the percentage of variation explained by the relationship between two variables (predicted

Table 4
Summary table of the best results obtained for the use of DGAN model as a predictor. We compare it to the Prophet forecasting algorithm for the same testing set (actual data). All Pearson correlations are significant ($p_{value} < 0.05$). All results are available in Appendix A.1. Best results are highlighted in bold.

Training set	Testing set	Aggregation	Forecasting Method	Split	SMAPE	MAE	R^2	MBE	ρ
2015–2018	2019	1H	**DGAN**	**Daily**	**35.98**	**2.48**	**0.68**	**0.03**	**0.82**
			Prophet	–	36.98	2.55	0.67	−0.21	**0.82**
		8H	**DGAN**	**Weekly**	**12.92**	**7.85**	**0.92**	**−0.81**	**0.96**
			Prophet	–	16.81	9.27	0.89	−1.71	0.95
		1D	**DGAN**	**Weekly**	**6.99**	**15.23**	**0.50**	**−2.44**	**0.71**
			Prophet	–	7.30	15.96	0.45	−5.13	0.70
		1W	**DGAN**	**Weekly**	**4.38**	**62.83**	**0.86**	**−16.78**	**0.93**
			Prophet	–	5.12	74.13	0.80	−35.32	0.91
2015–2017	2018–2019	1H	**DGAN**	**Daily**	**37.47**	**2.53**	**0.65**	**0.05**	**0.81**
			Prophet	–	46.72	3.35	0.48	−2.25	0.80
		8H	**DGAN**	**Daily**	**14.00**	**8.15**	**0.91**	**0.40**	**0.95**
			Prophet	–	32.58	19.21	0.60	−18.01	0.93
		1D	**DGAN**	**Daily**	**7.41**	**15.62**	**0.43**	**1.20**	**0.68**
			Prophet	–	22.84	54.39	−4.22	−54.04	0.55
		1W	**DGAN**	**Daily**	**5.07**	**72.76**	**0.75**	**8.32**	**0.87**
			Prophet	–	22.60	375.73	−4.61	−375.73	0.74

and actual values). It provides a measure of how well-observed outcomes are replicated by the model. It is computed as in Eq. (3). Higher values (closer to 1) indicate better performance.

- MBE (Mean Bias Error). It represents the signed difference between the predicted values and the actual values, being useful to know if the model is overestimating or underestimating the predictions. It is computed as in Eq. (4). Values closer to zero are better as they indicate no bias.
- ρ (Pearson Correlation). It measures the linear relationship between the predicted values and the actual values. It is computed as in Eq. (5). Values closer to 1 or −1 indicate strong correlation, with 1 being a perfect positive correlation.

$$\text{SMAPE}(y, \hat{y}) = \frac{100\%}{n} \sum_{i=0}^{n-1} \frac{2 * |y_i - \hat{y}_i|}{|y| + |\hat{y}|} \tag{1}$$

$$\text{MAE} = \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{2}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{\sum_{i=1}^{n} (y_i - \overline{y_i})^2} \tag{3}$$

$$\text{MBE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i) \tag{4}$$

$$\rho = \frac{\sum (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum (x_i - \overline{x})^2 \sum (y_i - \overline{y})^2}} \tag{5}$$

In addition to the metrics, we also present the results for four temporal aggregations: the first related to the hourly format of the data (1H); the second in an aggregation similar to the shifts of the service workers (8H); the third in daily aggregation to see what happens when the intraday pattern is removed (1D); the fourth in weekly aggregation to determine the best model when the weekly pattern is removed (1W). All aggregations are performed after the models made their forecasts in hourly format.

Among the four issues identified, the first concerns determining the optimal data split. This is addressed by applying subsequent experiments to all the divisions created. Therefore, the results for the other three issues are presented in the following subsections.

4.1.1. Data generation for unseen attribute values

After conducting the experiments described in Section 3.3.2, the best results are depicted in Table 4. The table compares the predictive performance of the DGAN algorithm against the baseline Prophet algorithm. For the one-year test set (2019), the generative model achieves results similar to the baseline, with slight improvements across all metrics.

The differences between the two models for 1H are not so remarkable, however, in the 8H aggregation case, the differences are notable, with SMAPE improving from 16.81, MAE from 9.27, and R^2 from 0.89 with Prophet to 12.92, 7.85, and 0.92, respectively. The generative model also shows slight improvements in other metrics and aggregations, then the best DGAN models are, at least similar that Prophet or even better. Notably, the larger the data aggregation, the more significant its improvement over the baseline, especially in bias (with an MBE of −16.78 for the generative model against −35.32 for Prophet in 1W aggregation). Fig. 10 shows an example of DGAN predictions for each split, comparing with the Prophet base model for each aggregation used. It is observed DGAN is not able to capture the daily and weekly seasonality when is trained with yearly splits.

Interestingly, the two-year test set presents a notable result. With one less year of training data, the Prophet algorithm has difficulties fitting the data. Conversely, DGAN effectively extracts and extrapolates the seasonal patterns from the original series for future predictions (at least, for daily and weekly splits, although daily split always achieves better metrics). Thus, it achieves a SMAPE of 37.47, MAE of 2.53, R^2 of 0.65, and ρ of 0.81, compared to Prophet's 46.72, 3.35, 0.48, and 0.80, respectively, for 1H aggregation. The difference between the two models increases with larger aggregations, with DGAN reaching a SMAPE of 5.07, MAE of 72.76, R^2 of 0.75, and ρ of 0.87 against Prophet's 22.60, 375.73, −4.61, and 0.74, respectively, for 1W aggregation. In the case of 1W aggregation, the bias showed by MBE metric really improve from −375.73 to 8.32 which means that the model is able to fit the data. Fig. 11 presents an example of DGAN forecasts for each split and aggregation.

When analyzing the dataset splitting method for training DGAN, we find the daily division as the best performance among the studied in terms of hourly aggregation for the one-year test dataset, while for all other scenarios, the weekly division is superior. However, the difference between daily and weekly divisions is not significant in any case as we can see in Fig. 10, which indicates that no partition clearly outperforms the others. Nevertheless, the yearly division demonstrates significantly lower performance compared to the other two. For the two-year test dataset, the daily division proves to be the most effective of all the splits analyzed.

The detailed results of all executions for this issue can be found in Appendix A.1.

4.1.2. Replacing original data

Results from substituting training data with synthetic data and applying the TSTR technique are presented in Table 5. In the scenario with a one-year test dataset, we observe similar results for training with both real and synthetic data for the 1H and 8H aggregations. Extending

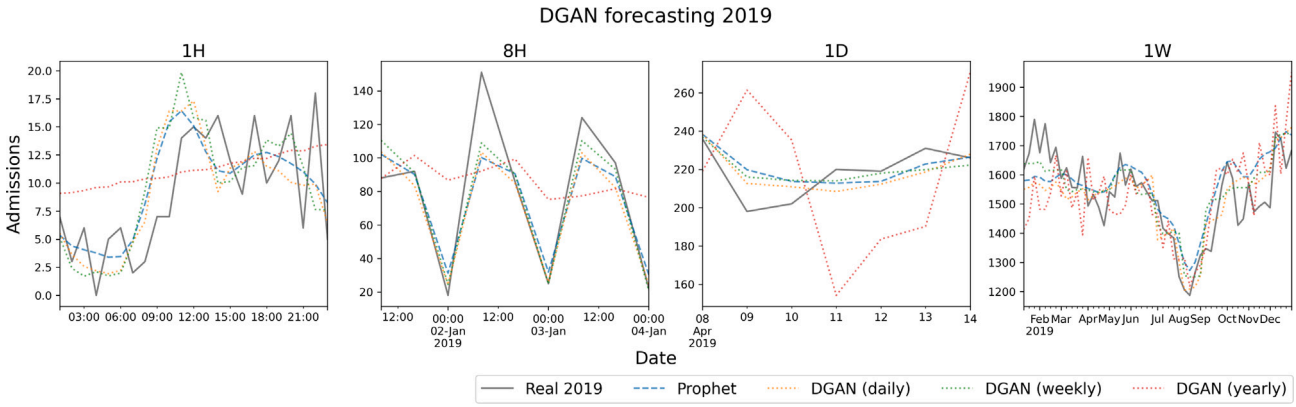

Fig. 10. DGAN forecasting for the year 2019 by using different splits of the data to feed it. Predictions are compared to the Prophet-based model forecasting. The first day of the test set was selected to check the daily seasonality in 1H and 8H; a random week to assess the weekly seasonality in 1D; and the entire test set to evaluate the yearly seasonality.

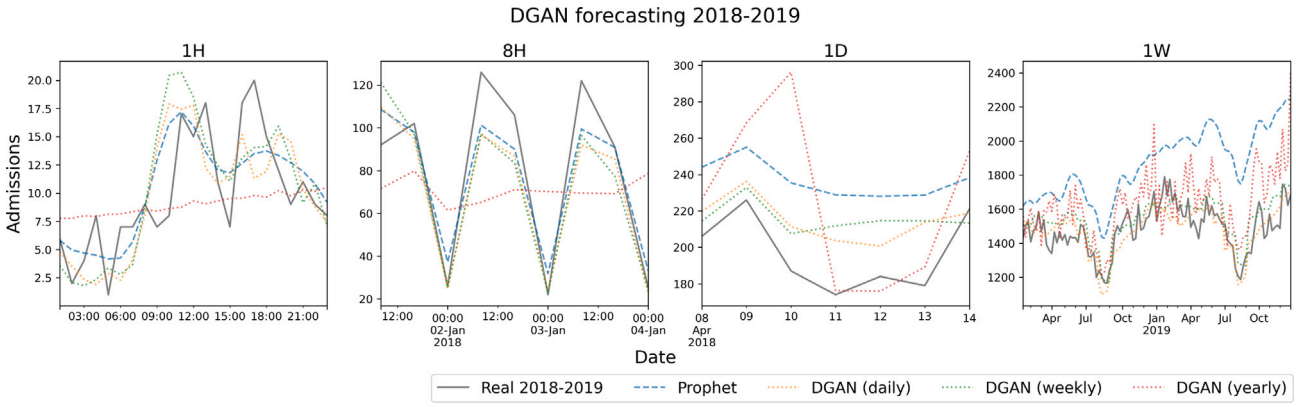

Fig. 11. DGAN forecasting for the years 2018–2019 by using different splits of the data to feed it. Predictions are compared to the Prophet-based model forecasting. The first day of the test set was selected to check the daily seasonality in 1H and 8H; a random week to assess the weekly seasonality in 1D; and the entire test set to evaluate the yearly seasonality.

Table 5
Summary table of the best results obtained for replacing original data in train compared by using synthetic data generated with the DGAN algorithm (TSTR test). All Pearson correlations are significant ($p_{value} < 0.05$). All results are available in Appendix A.2. Best results are highlighted in bold.

Training set	Testing set	Aggregation	Train data	Split	SMAPE	MAE	R^2	MBE	ρ
2015–2018	2019	1H	**Synthetic**	**Daily**	**36.61**	**2.52**	**0.67**	**0.06**	**0.82**
			Original	–	36.98	2.55	**0.67**	−0.21	**0.82**
		8H	**Synthetic**	**Daily**	**16.16**	**9.13**	**0.89**	**0.46**	**0.95**
			Original	–	16.81	9.27	**0.89**	−1.71	**0.95**
		1D	**Synthetic**	**Weekly**	**6.84**	**14.91**	**0.51**	**−3.80**	**0.74**
			Original	–	7.30	15.96	0.45	−5.13	0.70
		1W	**Synthetic**	**Weekly**	**4.42**	**62.83**	**0.86**	**−26.20**	**0.94**
			Original	–	5.12	74.13	0.80	−35.32	0.91
2015–2017	2018–2019	1H	**Synthetic**	**Daily**	**37.79**	**2.50**	**0.66**	**0.12**	**0.82**
			Original	–	46.72	3.35	0.48	−2.25	0.80
		8H	**Synthetic**	**Daily**	**16.51**	**8.90**	**0.89**	**1.00**	**0.95**
			Original	–	32.58	19.21	0.60	−18.01	**0.93**
		1D	**Synthetic**	**Daily**	**7.18**	**15.19**	**0.45**	**2.99**	**0.68**
			Original	–	22.84	54.39	−4.22	−54.04	0.55
		1W	**Synthetic**	**Daily**	**4.73**	**68.36**	**0.76**	**20.82**	**0.88**
			Original	–	22.60	375.73	−4.61	−375.73	0.74

this aggregation to 1D, the advantage of using synthetic data becomes more apparent. With a 1W aggregation, the performance significantly improves, changing from a SMAPE of 5.12, an MAE of 74.13, an R^2 of 0.80, an MBE of −35.32, and a ρ of 0.91 with real data training, to 4.42, 62.83, 0.86, −26.20, and 0.94, respectively, with synthetic data training. Fig. 12 shows a comparison between the Prophet fitted with real data and synthetic data (for each split), where all the seasonalities are correctly captured except the daily pattern for the yearly division.

Considering the two-year test dataset, the difference between training with real versus synthetic data is substantial, with synthetic data showing a considerable advantage. The Prophet model is capable of capturing both the seasonality and trend of the series, reflecting these in its predictions (as shown in Fig. 13). As a result, there is a significant improvement across all aggregations, more pronounced with larger aggregations. The SMAPE values for synthetic data training are 37.79 for 1H aggregation, 16.51 for 8H, 7.18 for 1D, and 4.73 for 1W,

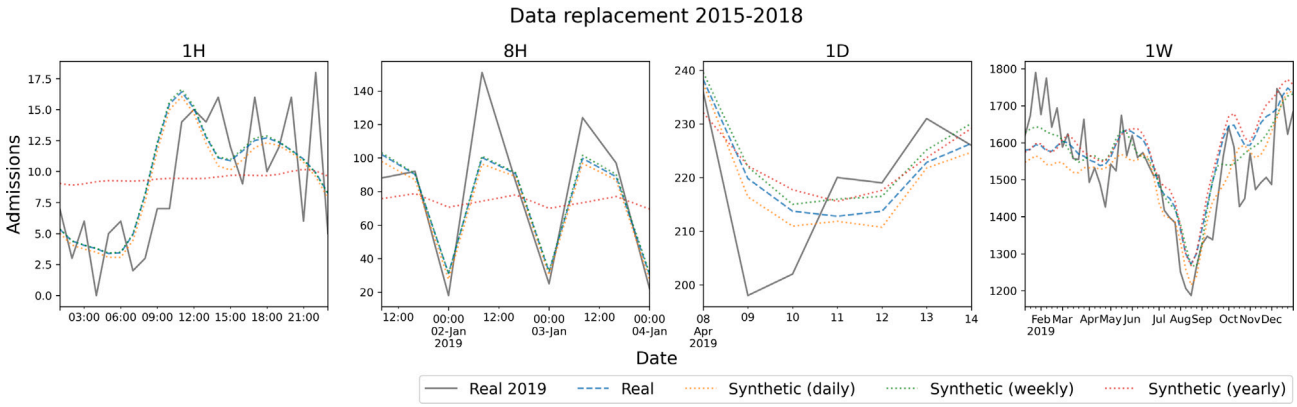

Fig. 12. Prophet model trained using synthetic data to replace the 2015–2018 training set. 1H and 8H aggregations demonstrate daily seasonality, while the 1D and 1W aggregations exemplify weekly and yearly seasonality, respectively.

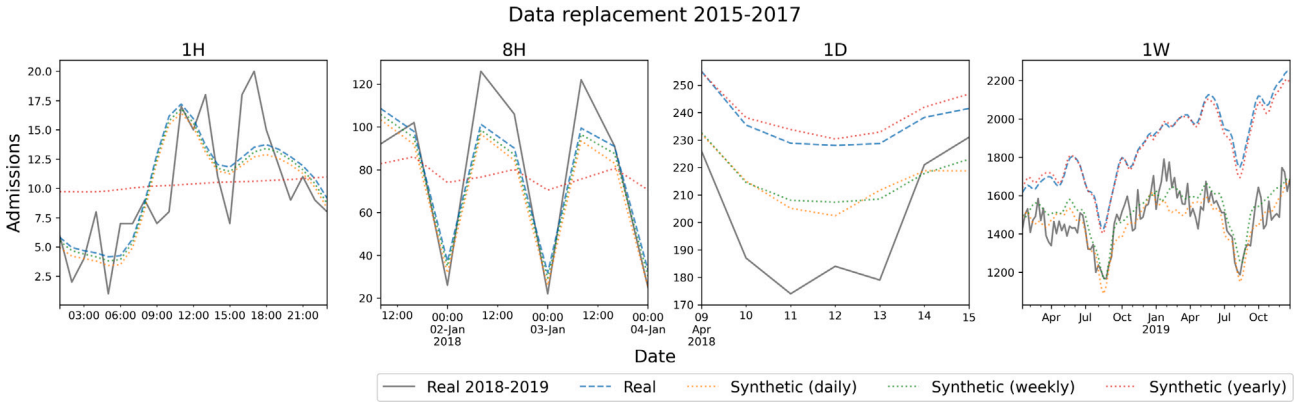

Fig. 13. Prophet model trained using synthetic data to replace the 2015–2017 training set. 1H and 8H aggregations demonstrate daily seasonality captured, while the 1D and 1W aggregations exemplify weekly and yearly seasonality detected, respectively.

compared to 46.72, 32.58, 22.84, and 22.60 respectively when trained with real data.

Concerning the impact of data partitions, daily aggregation yields the best performance for 1H and 8H aggregations within the one-year test set, and for all aggregations within the two-year test set. For 1D and 1W aggregations in the one-year test set, weekly splits demonstrate superior performance. Although daily and weekly partitions show similar performance in all cases, the yearly split is significantly behind them, failing to capture the trend for the two-year test set, similar to the outcomes observed when training with real data (present in Fig. 13).

The detailed results of the runs for this problem can be found in Appendix A.2.

4.1.3. Data augmentation

Regarding data augmentation by extending the original training dataset with synthetic data, the most significant results are presented in Table 6. For the one-year test dataset, the difference in using data augmentation with synthetic data as opposed to training with the original training set becomes more pronounced with larger aggregations. For the 1H and 8H aggregations, adding an extra year to the original training dataset achieves the best results. Nevertheless, adding more years to the training dataset does not enhance the Prophet model's ability to learn additional information, and in some instances, it even results in a loss of precision. For the 1D aggregation, improvements reach up to the addition of two years, declining in accuracy if more years are added. With a 1W aggregation, the best results are achieved by incorporating the previous four years into the training set (doubling the original training dataset). This enhances the SMAPE from 5.12 to 3.72, the MAE from 74.13 to 55.87, the R^2 from 0.80 to 0.87, the MBE to from −35.32 to 1.01, and the ρ from 0.91 to 0.93. Fig. 14 depicts an example of all predictions of Prophet model by using data augmentation for each split, aggregation, and the one-year test set. For daily and weekly aggregations, data augmentation enhances the capture of weekly seasonality, while the performance on other seasonalities remains similar. However, for yearly splits, although it significantly improves the capture of yearly patterns (being selected for 1W aggregation), it causes a loss of accuracy in detecting other seasonalities. For example, it is noticeable how the daily pattern gradually diminishes as more synthetic data is introduced for yearly splits.

For the two-year test set, we observe that results improve as more data is added, to the point of having more synthetic data than real data in the training set. This allows the Prophet algorithm to better fit the original time series. With 1H aggregation, results significantly improve with the addition of synthetic data, moving from a SMAPE of 46.72, an MAE of 3.35, an R^2 of 0.48, an MBE of −2.25, and a ρ of 0.80 to 39.03, 2.62, 0.65, −0.52, and 0.82, respectively. For 8H aggregation, results also improve notably, changing from a SMAPE of 32.58, an MAE of 19.21, an R^2 of 0.60, an MBE of −18.01, and a ρ of 0.93 to 19.07, 9.93, 0.88, −4.13, and 0.95, respectively. In 1D aggregation, the improvement is even more evident, moving from a SMAPE of 22.84, an MAE of 54.39, an R^2 of −4.22, an MBE of −54.04, and a ρ of 0.55 to 8.56, 18.21, 0.28, −12.38, and 0.70, respectively. With 1W

Table 6
Summary table of the best results obtained for data augmentation by using synthetic data generated with the DGAN algorithm and the Prophet model as the forecasting method. All Pearson correlations are significant ($p_{value} < 0.05$). Best results are highlighted in bold.

Training set	Testing set	Aggregation	Split	SMAPE	MAE	R^2	MBE	ρ
2014–2018	2019	1H	**Daily**	**36.43**	**2.51**	**0.67**	0.24	**0.83**
2015–2018			–	36.98	2.55	**0.67**	**−0.21**	0.82
2014–2018		8H	**Daily**	**15.77**	**9.04**	**0.89**	1.95	**0.95**
2015–2018			–	16.81	9.27	**0.89**	**−1.71**	**0.95**
2013–2018		1D	**Weekly**	**6.79**	**14.91**	**0.50**	5.16	**0.75**
2015–2018			–	7.30	15.96	0.45	**−5.13**	0.70
2011–2018		1W	**Yearly**	**3.72**	**55.87**	**0.87**	**1.01**	**0.93**
2015–2018			–	5.12	74.13	0.80	−35.32	0.91
2011–2017	2018–2019	1H	**Daily**	**39.03**	**2.62**	**0.65**	**−0.52**	**0.82**
2015–2017			–	46.72	3.35	0.48	−2.25	0.80
2011–2017		8H	**Daily**	**19.07**	**9.93**	**0.88**	**−4.13**	**0.95**
2015–2017			–	32.58	19.21	0.60	−18.01	0.93
2011–2017		1D	**Daily**	**8.56**	**18.21**	**0.28**	**−12.38**	**0.70**
2015–2017			–	22.84	54.39	−4.22	−54.04	0.55
2011–2017		1W	**Daily**	**6.70**	**98.22**	**0.55**	**−86.10**	**0.89**
2015–2017			–	22.60	375.73	−4.61	−375.73	0.74

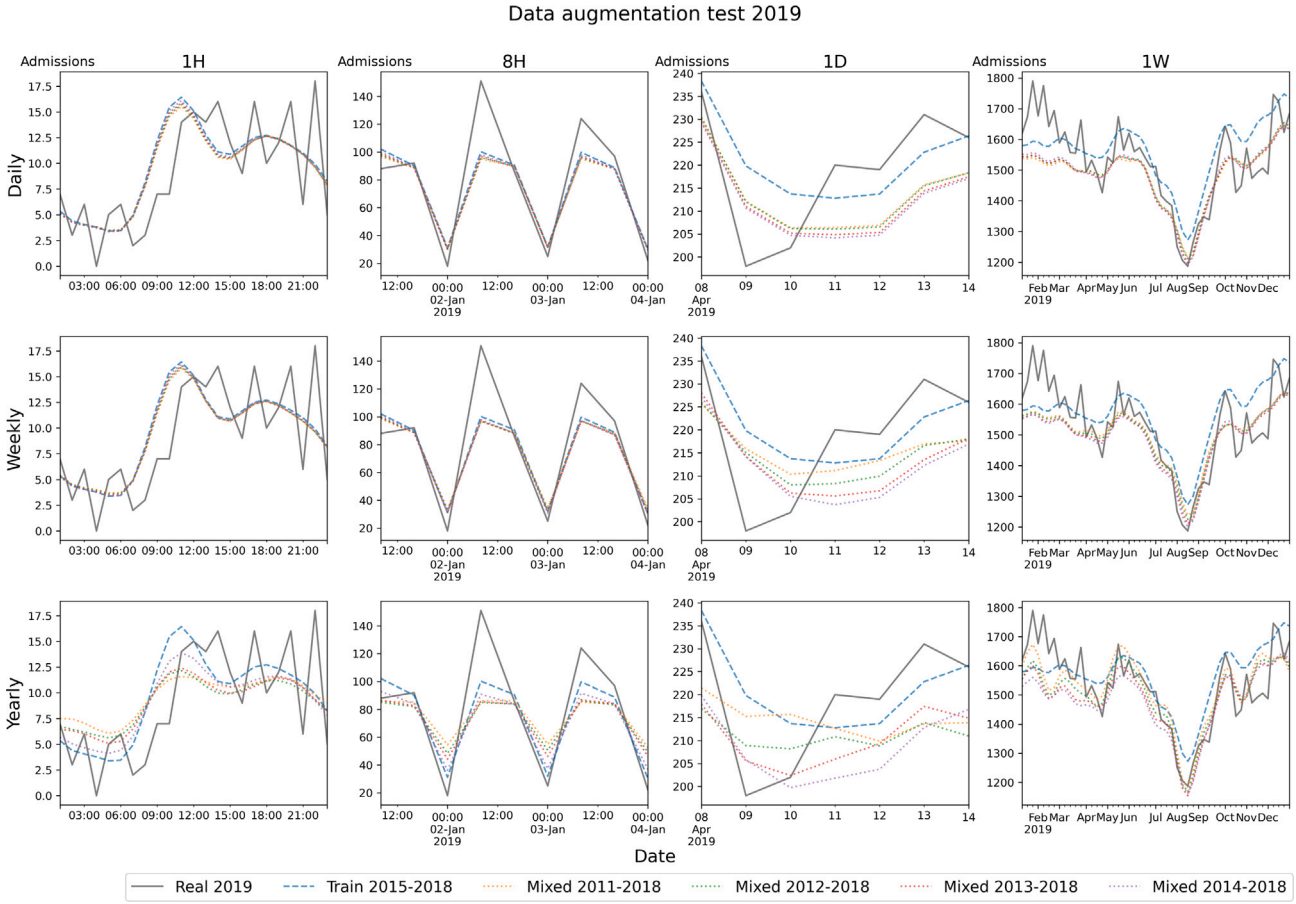

Fig. 14. Prophet forecasting model by using data augmentation with synthetic data. All plots show actual data for the year 2019, Prophet base model trained with 2015–2018 original data as a baseline, and the Prophet fitted for different ranges with mixed data (synthetic and real).

aggregation, we observe an improvement from a SMAPE of 22.60, an MAE of 375.73, an R^2 of −4.61, an MBE of −375.73, and a ρ of 0.74 to 6.70, 98.22, 0.55, −86.10, and 0.89, respectively. These results demonstrate how the algorithm's bias is more pronounced with larger aggregations. Fig. 15 shows an example of predictions for the two-year test set. In this scenario, the alignment with the weekly pattern and yearly trend is enhanced as additional synthetic data is incorporated into the training dataset. However, the capability to capture the daily pattern gradually diminishes in the weekly and yearly divisions, while, the daily split maintains this ability.

An extension of the results summary presented in this section can be found in Appendix A.3. Additionally, the complete results are consulted in Appendix A.4, including all executions for all years and all divisions.

4.2. Discussion

Based on the results presented in the previous section and considering the questions raised in this study we discuss an answer for each of them.

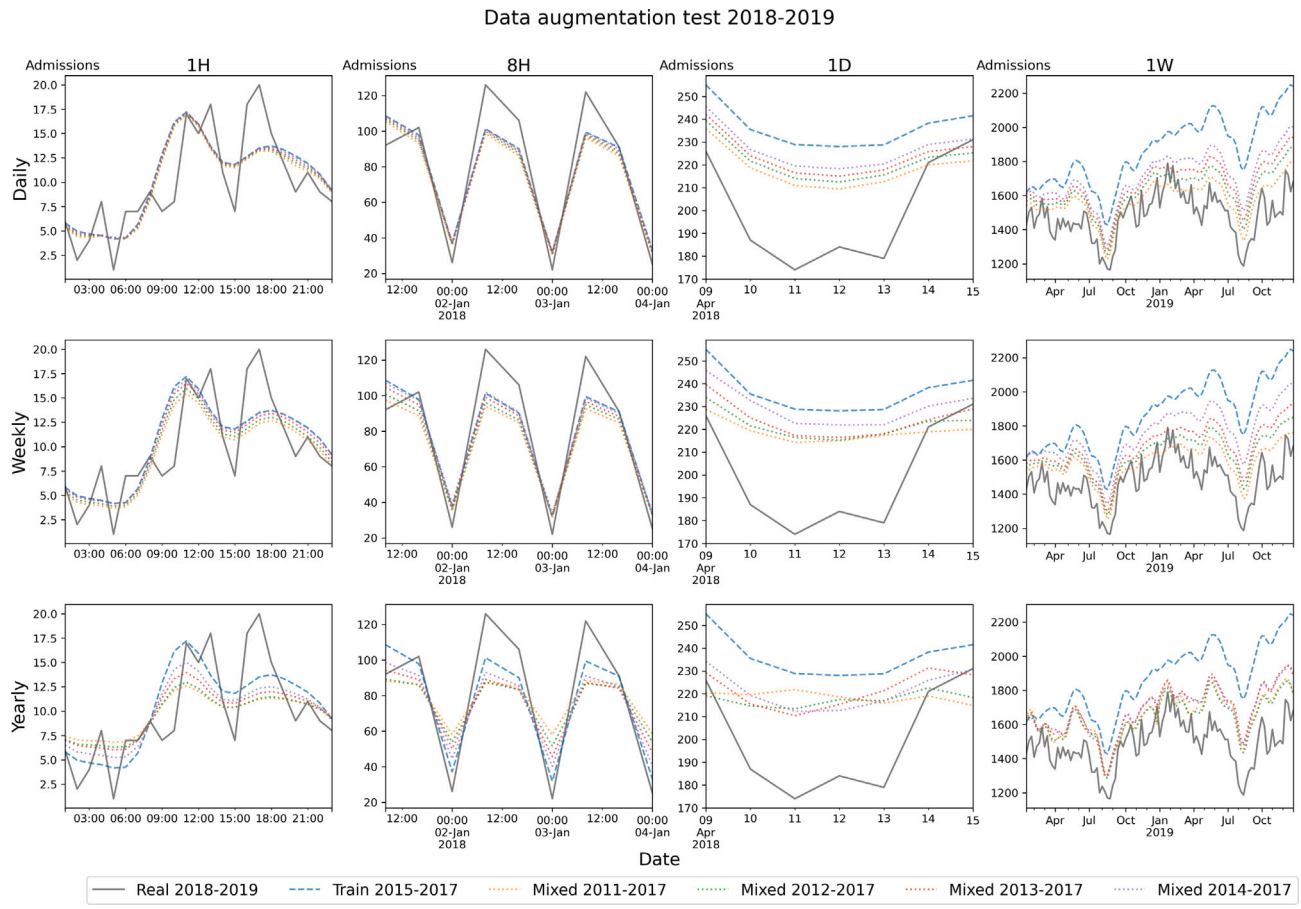

Fig. 15. Prophet forecasting model by using data augmentation with synthetic data. All plots show actual data for the years 2018–2019, Prophet base model trained with 2015–2017 original data predictions as a baseline, and the Prophet fitted for different ranges with mixed data (synthetic and real).

Table 7
Splits selected for the data generation process, based on their performance for each task and aggregation, indicating the training set used.

Task	Training set	Aggregation	Split
Forecasting	2015–2018	[1H]	Daily
		[8H, 1D, 1W]	Weekly
	2015–2017	[1H, 8H, 1D, 1W]	Daily
Data replacement	2015–2018	[1H, 8H]	Daily
		[1D, 1W]	Weekly
	2015–2017	[1H, 8H, 1D, 1W]	Daily
Data augmentation	2014–2018	[1H, 8H]	Daily
	2013–2018	[1D]	Weekly
	2011–2018	[1W]	Yearly
	2011–2017	[1H, 8H, 1D, 1W]	Daily

The first question referred to the best way to feed data into the generative algorithm, and which is the best method to facilitate capturing the distribution and characteristics of the original series. Based on the results, it depends on the specific task to solve, the aggregation used, and the length of the training dataset. Thus, we find that daily and weekly divisions perform closely in most of our tasks. However, in one specific case, the yearly division outperforms them despite losing adaptability to the daily and weekly pattern. Table 7 summarizes the best splitting method for each task and aggregation used in our study, also indicating the training set used.

The second question, concerning how the generative algorithm performs on unseen attribute values (also presented as using the generative algorithm as a forecasting method), demonstrates that the model operates effectively, even surpassing the results achieved by a forecasting model like Prophet. This finding suggests that the model can produce realistic data beyond the attribute values encountered during training, implying the potential to explore areas outside the original distribution's boundaries.

The synthetic data also contains the main characteristics of actual data, a finding corroborated by the TSTR test (employed to tackle the third question regarding replacing original data in training). In this assessment, we once again observe that the outcomes are even better to those obtained using real data in training with Prophet, being able to remove some noise that could be confusing the model but maintaining the main features of the original time series. It is important to highlight that, in this research, the DGAN algorithm was not expressly trained to simulate with precision the original distribution; rather, the main intention was to utilize it for generating data to extend the original dataset.

The last issue related to data augmentation provides interesting results. We observe that in this scenario, it is possible to enhance the performance of a forecasting model by extending the dataset with synthetic data. However, better results are not always achieved with the addition of more synthetic data, suggesting that an excess of synthetic data might weaken the model. This could occur for two reasons: firstly, if the synthetic data patterns are strongly marked, the model might lose its ability to generalize in some cases where instances do not fit these strict patterns. This limitation could be due to the way we are generating the synthetic time series for our tests, by averaging 1000 generated series and losing values that highly deviate from the patterns. Secondly, while synthetic data can enhance the capture of certain patterns, it may lead to the loss of others. For instance, in the case of yearly splits, although the capture of the yearly pattern and trend from the original series is improved, this advantage is countered by a diminished accuracy in identifying daily and weekly patterns.

Therefore, it is essential to carefully balance the improvements in certain characteristics with the potential detriments to others.

Using synthetic data to feed a forecasting model may go beyond extending the time series itself. For instance, employing a forecasting model based on Neural Networks, the generator could be used as a method to prevent overfitting by alternating between the real series and multiple synthetic series for the same range during training epochs. Similarly, it could serve as an imputation method to generate segments of the series that are missing.

It is important to emphasize the significance of the results obtained, given we are dealing with particularly small datasets. Generative algorithms typically are trained on a large number of instances to capture their distribution. In our scenario, the model achieves a good adjustment to the data distribution with a training dataset of only 4 years and even with just 3 years. The challenge addressed in this study is intrinsically linked to the nature of the data used, and adapting the approach to a different dataset requires an initial analysis of its particular characteristics, such as seasonality, trends, or potential attributes that could be introduced.

Considering a management perspective, Generative Deep Learning has promising implications for health system management. Generating synthetic data similar to real data can improve the accuracy of models predicting patient arrivals, as we experimented, thus optimizing resource planning and enhancing operational efficiency. Additionally, it enables the use of different ML techniques in situations where obtaining data is difficult or limited. Generative models could also democratize access to high-quality data, preserving privacy and providing public data without compromising confidentiality. This could boost research and development, encourage interdisciplinary collaboration, and facilitating the adoption of new technologies. The generation of synthetic data for specific contexts could also lead to customized solutions, improving current processes and opening new lines for the research community.

5. Conclusions and future lines

In this section, we present the conclusions extracted from our work and possible future works that can be derived from it.

5.1. Conclusions

In this work, we utilized DGAN, a Generative Deep Learning algorithm based on GANs specifically designed for time series, to assess whether it can be used to extend a limited dataset obtained from ED hospital admissions. To this end, we adopted a sequential methodology that involved addressing four distinct questions. The first question underscored the critical role of appropriately partitioning the original data fed into the generative algorithm, ensuring optimal performance for each task. The second question demonstrated DGAN's capability to generate synthetic series successfully, even when conditioned with attribute values that were not seen during the training process. Additionally, this point revealed that the generative algorithm could act as a predictive algorithm in certain scenarios. The third question validated the generated data using a TSTR test. Thus, combining the insights from the second and third questions, we can conclude that it is possible to generate synthetic data similar to real data for both replacing and extending the training set. Finally, by expanding the original dataset, we verified that it is feasible to improve the outcomes obtained with the predictive model using a mix of synthetic and actual data, compared to training only with real data.

This study aims to present an approach to explore improvements in the forecasting task in environments where data are limited. We observed that applying generative architectures to create synthetic data can enhance the fitting process of a forecasting algorithm. The study is also significant because it demonstrates the strong performance of a generative algorithm for time series on a limited training set, as we have presented the fittings for two datasets composed by 4 and 3 years respectively, outperforming the Prophet algorithm even in forecasting tasks.

From a macro perspective, a study like the one presented could guide policymakers in implementing actions such as data collection and publication to stimulate more comprehensive research on hospital resource management. The sensitive context of hospital emergency services requires efforts to prevent overcrowding that could cause the system to overflow. Therefore, studying and understanding resource allocation is a general social interest for a region or country. Publishing datasets would enable the research community to address this issue, although confidentiality issues often arise. Synthetic data generation, by using techniques such as DGAN, could offer a solution by allowing publication of data that replicate real data without compromising privacy.

Finally, it is important to note that a study of this nature is deeply linked to the available data and should not lead to conclusions beyond what can be reasonably inferred for this specific case. More comprehensive studies would be recommended, though we believe we have established a solid starting point. Additionally, the limited size of our datasets restricted our ability to introduce more attributes for the time series, as they quickly led to overfitting, preventing the generation of reliable values beyond the range of the original series. With a larger dataset, it would be worth considering the addition of attributes related to weather or calendar events to generate even more realistic data.

5.2. Future lines

In light of the results obtained, several future research directions can be proposed. The first could focus on studying and comparing with other generative algorithms, such as TimeVAE [43] or CPAR [45], evaluating their performance and applicability in different hospitals. Another area of interest would be the application of generative algorithms specifically for special periods, such as holidays or festivities, where patient influx varies significantly. In this context, we think that creating an ensemble of algorithms could improve accuracy by combining the individual strengths of each one. Additionally, it would be beneficial to tailor the algorithms to specific patient characteristics, such as origin, age, or reason for admission, as well as to the emergency department services that attend to them, such as traumatology, pediatrics, or psychiatry. This personalization could increase the effectiveness of the modeling.

CRediT authorship contribution statement

Hugo Álvarez-Chaves: Writing – original draft, Visualization, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Marco Spruit:** Writing – review & editing, Validation, Supervision, Project administration, Conceptualization. **María D. R-Moreno:** Writing – review & editing, Validation, Supervision, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study was funded by Junta de Comunidades de Castilla-La Mancha (SBPLY/19/180501/000024) and Comunidad de Madrid (PIPF-2023/COM-29922). We would like to express our gratitude to the clinical support of Dr. Helena Hernández Martínez, Dr. María Isabel Pascual Benito, and Mr. Francisco López Martínez, and the collaboration of Jim Achterberg for his valuable comments and for sharing information about this topic. We also want to acknowledge the support of the Hospital Central de la Defensa Gómez Ulla (HCDGU), Madrid, Spain.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cmpb.2024.108363.

References

[1] N.R. Hoot, D. Aronsky, Systematic review of emergency department crowding: causes, effects, and solutions, Ann. Emerg. Med. 52 (2) (2008) 126–136, http://dx.doi.org/10.1016/j.annemergmed.2008.03.014.

[2] J. Boyle, M. Jessup, J. Crilly, D. Green, J. Lind, M. Wallis, P. Miller, G. Fitzgerald, Predicting emergency department admissions, Emerg. Med. J. 29 (5) (2012) 358–365, http://dx.doi.org/10.1136/emj.2010.103531.

[3] C.N. Rocha, F. Rodrigues, Forecasting emergency department admissions, J. Intell. Inf. Syst. (ISSN: 0925-9902) (2021) 1–20, http://dx.doi.org/10.1007/s10844-021-00638-9.

[4] V. Rema, K. Sikdar, Time series modelling and forecasting of patient arrivals at an emergency department of a select hospital, in: Recent Trends in Signal and Image Processing: ISSIP 2020, Springer, 2021, pp. 53–65, http://dx.doi.org/10.1007/978-981-33-6966-5_6.

[5] V.K. Sudarshan, M. Brabrand, T.M. Range, U.K. Wiil, Performance evaluation of emergency department patient arrivals forecasting models by including meteorological and calendar information: A comparative study, Comput. Biol. Med. (ISSN: 0010-4825) 135 (2021) 104541, http://dx.doi.org/10.1016/j.compbiomed.2021.104541.

[6] Y. Zhang, J. Zhang, M. Tao, J. Shu, D. Zhu, Forecasting patient arrivals at emergency department using calendar and meteorological information, Appl. Intell. 52 (10) (2022) 11232–11243, http://dx.doi.org/10.1007/s10489-021-03085-9.

[7] H. Álvarez-Chaves, P. Muñoz, M.D. R-Moreno, Machine learning methods for predicting the admissions and hospitalisations in the emergency department of a civil and military hospital, J. Intell. Inf. Syst. 61 (3) (2023) 881–900, http://dx.doi.org/10.1007/s10844-023-00790-4.

[8] J. Park, B. Chang, N. Mok, 144 Time series analysis and forecasting daily emergency department visits utilizing facebook's prophet method, Ann. Emerg. Med. 74 (4) (2019) S57, http://dx.doi.org/10.1016/j.annemergmed.2019.08.149.

[9] H. Álvarez-Chaves, I. Maseda-Zurdo, P. Muñoz, M.D. R-Moreno, Evaluating the impact of exogenous variables for patients forecasting in an emergency department using attention neural networks, Expert Syst. Appl. 240 (2024) 122496, http://dx.doi.org/10.1016/j.eswa.2023.122496.

[10] Z. Lin, A. Jain, C. Wang, G. Fanti, V. Sekar, Using gans for sharing networked time series data: Challenges, initial promise, and open questions, in: Proceedings of the ACM Internet Measurement Conference, 2020, pp. 464–483, http://dx.doi.org/10.48550/arXiv.1909.13403.

[11] Z. Lin, A. Jain, C. Wang, G. Fanti, V. Sekar, Generating high-fidelity, synthetic time series datasets with doppelganger, 2019, http://dx.doi.org/10.48550/arXiv.1909.13403, arXiv preprint arXiv:1909.13403.

[12] S.J. Taylor, B. Letham, Forecasting at scale, Amer. Statist. 72 (1) (2018) 37–45, http://dx.doi.org/10.1080/00031305.2017.1380080.

[13] C. Esteban, S.L. Hyland, G. Rätsch, Real-valued (medical) time series generation with recurrent conditional gans, 2017, http://dx.doi.org/10.48550/arXiv.1706.0263, arXiv preprint arXiv:1706.02633.

[14] M. Gul, E. Celik, An exhaustive review and analysis on applications of statistical forecasting in hospital emergency departments, Health Syst. (ISSN: 20476973) 9 (2020) 263–284, http://dx.doi.org/10.1080/20476965.2018.1547348.

[15] S. Jiang, Q. Liu, B. Ding, A systematic review of the modelling of patient arrivals in emergency departments, Quant. Imaging Med. Surg. (ISSN: 22234292) (2022) http://dx.doi.org/10.21037/QIMS-22-268/COIF.

[16] A. Aroua, G. Abdul-Nour, Forecast emergency room visits–a major diagnostic categories based approach, Int. J. Metrol. Qual. Eng. 6 (2) (2015) 204, http://dx.doi.org/10.1051/ijmqe/2015011.

[17] A. Ekström, L. Kurland, N. Farrokhnia, M. Castrén, M. Nordberg, Forecasting emergency department visits using Internet data, Ann. Emerg. Med. 65 (4) (2015) 436–442, http://dx.doi.org/10.1016/j.annemergmed.2014.10.008.

[18] Q. Xu, K.-L. Tsui, W. Jiang, H. Guo, A hybrid approach for forecasting patient visits in emergency department, Qual. Reliab. Eng. Int. 32 (8) (2016) 2751–2759, http://dx.doi.org/10.1002/qre.2095.

[19] H. Álvarez-Chaves, D.F. Barrero, M. Cobos, M.D. R-Moreno, Patients forecasting in emergency services by using machine learning and exogenous variables, in: International Conference on Innovative Techniques and Applications of Artificial Intelligence, Springer, 2021, pp. 167–180, http://dx.doi.org/10.1007/978-3-030-91100-3_15.

[20] M. Fralick, J. Murray, M. Mamdani, Predicting emergency department volumes: A multicenter prospective study, Am. J. Emerg. Med. 46 (2021) 695–697, http://dx.doi.org/10.1016/j.ajem.2020.10.047.

[21] H.S. Almeida, M. Sousa, I. Mascarenhas, A. Russo, M. Barrento, M. Mendes, P. Nogueira, R. Trigo, The dynamics of patient visits to a public hospital pediatric emergency department: a time-series model, Pediatr. Emerg. Care. 38 (1) (2022) e240–e245, http://dx.doi.org/10.1097/PEC.0000000000002235.

[22] W. Whitt, X. Zhang, Forecasting arrivals and occupancy levels in an emergency department, Oper. Res. Health Care 21 (2019) 1–18, http://dx.doi.org/10.1016/j.orhc.2019.01.002.

[23] I. Marcilio, S. Hajat, N. Gouveia, Forecasting daily emergency department visits using calendar variables and ambient temperature readings, Acad. Emerg. Med. 20 (8) (2013) 769–777, http://dx.doi.org/10.1111/acem.12182.

[24] N.B. Menke, N. Caputo, R. Fraser, J. Haber, C. Shields, M.N. Menke, A retrospective analysis of the utility of an Artificial Neural Network to predict ED volume, Am. J. Emerg. Med. 32 (6) (2014) 614–617, http://dx.doi.org/10.1016/j.ajem.2014.03.011.

[25] Y. Sun, B.H. Heng, Y.T. Seow, E. Seow, Forecasting daily attendances at an emergency department to aid resource planning, BMC Emerg. Med. 9 (1) (2009) 1–9, http://dx.doi.org/10.1186/1471-227X-9-1.

[26] M. Yousefi, M. Yousefi, M. Fathi, F.S. Fogliatto, Patient visit forecasting in an emergency department using a Deep Neural Network approach, Kybernetes 49 (9) (2020) 2335–2348, http://dx.doi.org/10.1108/K-10-2018-0520.

[27] A.F.W. Ho, B.Z.Y.S. To, J.M. Koh, K.H. Cheong, Forecasting hospital emergency department patient volume using Internet search data, IEEE Access 7 (2019) 93387–93395, http://dx.doi.org/10.1109/ACCESS.2019.2928122.

[28] A. Gonzales, G. Guruswamy, S.R. Smith, Synthetic data in health care: a narrative review, PLOS Digit. Health 2 (1) (2023) e0000082, http://dx.doi.org/10.1371/journal.pdig.0000082.

[29] M. Giuffrè, D.L. Shung, Harnessing the power of synthetic data in healthcare: innovation, application, and privacy, NPJ Digit. Med. 6 (1) (2023) 186, http://dx.doi.org/10.1038/s41746-023-00927-3.

[30] T.E. Raghunathan, J.P. Reiter, D.B. Rubin, Multiple imputation for statistical disclosure limitation, J. Off. Stat. 19 (1) (2003) 1.

[31] P. Davis, R. Lay-Yee, J. Pearson, Using micro-simulation to create a synthesised data set and test policy options: The case of health service effects under demographic ageing, Health Policy 97 (2–3) (2010) 267–274, http://dx.doi.org/10.1016/j.healthpol.2010.05.014.

[32] B. Loong, A.M. Zaslavsky, Y. He, D.P. Harrington, Disclosure control using partially synthetic data for large-scale health surveys, with applications to cancors, Stat. Med. 32 (24) (2013) 4139–4161.

[33] J. Ive, N. Viani, J. Kam, L. Yin, S. Verma, S. Puntis, R.N. Cardinal, A. Roberts, R. Stewart, S. Velupillai, Generation and evaluation of artificial mental health records for natural language processing, NPJ Digit. Med. 3 (1) (2020) 69.

[34] Y. Jiang, H. Chen, M. Loew, H. Ko, COVID-19 CT image synthesis with a conditional generative adversarial network, IEEE J. Biomed. Health Inf. 25 (2) (2020) 441–452.

[35] H.-C. Shin, N.A. Tenenholtz, J.K. Rogers, C.G. Schwarz, M.L. Senjem, J.L. Gunter, K.P. Andriole, M. Michalski, Medical image synthesis for data augmentation and anonymization using generative adversarial networks, in: Simulation and Synthesis in Medical Imaging: Third International Workshop, SASHIMI 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Proceedings 3, Springer, 2018, pp. 1–11.

[36] L. Middel, C. Palm, M. Erdt, Synthesis of medical images using gans, in: Uncertainty for Safe Utilization of Machine Learning in Medical Imaging and Clinical Image-Based Procedures: First International Workshop, UNSURE 2019, and 8th International Workshop, CLIP 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 17, 2019, Proceedings 8, Springer, 2019, pp. 125–134.

[37] F. Garcea, A. Serra, F. Lamberti, L. Morra, Data augmentation for medical imaging: A systematic literature review, Comput. Biol. Med. 152 (2023) 106391.

[38] A. Kebaili, J. Lapuyade-Lahorgue, S. Ruan, Deep learning approaches for data augmentation in medical imaging: A review, J. Imaging 9 (4) (2023) 81.

[39] G. Iglesias, E. Talavera, Á. González-Prieto, A. Mozo, S. Gómez-Canaval, Data augmentation techniques in time series domain: a survey and taxonomy, Neural Comput. Appl. 35 (14) (2023) 10123–10145, http://dx.doi.org/10.1007/s00521-023-08459-3.

[40] B. Fu, F. Kirchbuchner, A. Kuijper, Data augmentation for time series: traditional vs generative models on capacitive proximity time series, in: Proceedings of the 13th ACM International Conference on Pervasive Technologies Related To Assistive Environments, 2020, pp. 1–10, http://dx.doi.org/10.1145/3389189.3392606.

[41] D. Bank, N. Koenigstein, R. Giryes, Autoencoders, Mach. Learn. Data Sci. Handb. Data Min. Knowl. Discov. Handb. (2023) 353–374, http://dx.doi.org/10.1007/978-3-031-24628-9_16.

[42] D.P. Kingma, M. Welling, Auto-encoding variational bayes, 2013, http://dx.doi.org/10.48550/arXiv.1312.6114, arXiv preprint arXiv:1312.6114.

[43] A. Desai, C. Freeman, Z. Wang, I. Beaver, Timevae: A variational auto-encoder for multivariate time series generation, 2021, http://dx.doi.org/10.48550/arXiv.2111.08095, arXiv preprint arXiv:2111.08095.

[44] H. Li, S. Yu, J. Principe, Causal recurrent variational autoencoder for medical time series generation, 2023, http://dx.doi.org/10.48550/arXiv.2301.06574, arXiv preprint arXiv:2301.06574.

[45] K. Zhang, N. Patki, K. Veeramachaneni, Sequential models in the synthetic data vault, 2022, http://dx.doi.org/10.48550/arXiv.2207.14406, arXiv preprint arXiv:2207.14406.

[46] L. Lin, Z. Li, R. Li, X. Li, J. Gao, Diffusion models for time-series applications: a survey, Front. Inf. Technol. Electron. Eng. (2023) 1–23, http://dx.doi.org/10.1631/FITEE.2300310.

[47] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, 2014, http://dx.doi.org/10.48550/arXiv.1406.2661.

[48] J. Yoon, D. Jarrett, M. Van der Schaar, Time-series generative adversarial networks, Adv. Neural Inf. Process. Syst. 32 (2019).

[49] E. Brophy, Z. Wang, Q. She, T. Ward, Generative adversarial networks in time series: A systematic literature review, ACM Comput. Surv. 55 (10) (2023) 1–31, http://dx.doi.org/10.1145/3559540.

[50] X. Li, A.H.H. Ngu, V. Metsis, Tts-cgan: A transformer time-series conditional gan for biosignal data augmentation, 2022, http://dx.doi.org/10.48550/arXiv.2206.13676, arXiv preprint arXiv:2206.13676.

[51] G. García-Jara, P. Protopapas, P.A. Estévez, Improving astronomical time-series classification via data augmentation with generative adversarial networks, Astrophys. J. 935 (1) (2022) 23, http://dx.doi.org/10.3847/1538-4357/ac6f5a.

[52] C. Lu, C.K. Reddy, P. Wang, D. Nie, Y. Ning, Multi-label clinical time-series generation via conditional gan, IEEE Trans. Knowl. Data Eng. (2023) http://dx.doi.org/10.1109/TKDE.2023.3310909.

[53] M.H. Naveed, U.S. Hashmi, N. Tajved, N. Sultan, A. Imran, Assessing deep generative models on time series network data, IEEE Access 10 (2022) 64601–64617, http://dx.doi.org/10.1109/ACCESS.2022.3177906.

[54] S. Dannels, Creating disasters: Recession forecasting with GAN-generated synthetic time series data, 2023, http://dx.doi.org/10.48550/arXiv.2302.10490, arXiv preprint arXiv:2302.10490.

[55] X. Zheng, N. Xu, L. Trinh, D. Wu, T. Huang, S. Sivaranjani, Y. Liu, L. Xie, A multi-scale time-series dataset with benchmark for machine learning in decarbonized energy grids, Sci. Data 9 (1) (2022) 359, http://dx.doi.org/10.1038/s41597-022-01455-7.

[56] X. Cai, Z. Luo, X. Lin, N. Zhang, Y. Mao, X. Lin, W. Zhong, Data self-expansion and DoppelGANger-based time-series modeling for realistic steam data generation, in: 2023 8th International Conference on Power and Renewable Energy, ICPRE, IEEE, 2023, pp. 1969–1974, http://dx.doi.org/10.1109/ACCESS.2022.3177906.

[57] I. Isasa, M. Hernandez, G. Epelde, F. Londoño, A. Beristain, A. Alberdi, P. Bamidis, E. Konstantinidis, Effect of incorporating metadata to the generation of synthetic time series in a healthcare context, in: 2023 IEEE 36th International Symposium on Computer-Based Medical Systems, CBMS, IEEE, 2023, pp. 910–916, http://dx.doi.org/10.1109/CBMS58004.2023.00341.

[58] H. Álvarez-Chaves, D.F. Barrero, H.H. Martínez, M.I.P. Benito, An analysis of the time aggregation influence on patients forecasting in emergency services, in: 2021 XLVII Latin American Computing Conference, CLEI, IEEE, 2021, pp. 1–10, http://dx.doi.org/10.1109/CLEI53233.2021.9640117.

[59] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 214–223, URL https://proceedings.mlr.press/v70/arjovsky17a.html.