A Simple Approach to Synthetic Time Series Generation

Joshua Davis
Department of Mathematics and Statistics
The University of the West Indies
St. Augustine, Trinidad and Tobago
joshua.davis2@my.uwi.edu

Patrick Hosein
Department of Electrical and Computer Engineering
The University of the West Indies
St. Augustine, Trinidad and Tobago
patrick.hosein@uwi.edu

***Abstract*—Time series data are necessary in many domains but analyses may be limited by availability, scale of data and privacy concerns. Synthetic time series may be used to supplement existing data to enable analyses and model building, in particular, machine learning applications. In this paper, a model for the generation of synthetic time series is proposed. This model is relatively easy to implement and aims to preserve statistical properties and temporal dynamics of the original time series. The model was found to be of higher quality to existing methods, producing synthetic time series which preserve the mean, variation and autocorrelation. Future work will investigate the preservation of non-negativity and seek to preserve additional statistical properties.**

***Index Terms*—time series, synthetic, energy dataset**

I. Introduction

Time series data plays a crucial role in many domains since they capture temporal dynamics of real world systems and are essential for forecasting, pattern recognition and anomaly detection [1]. However, these analyses may be limited by the availability of data, the scale of data and privacy concerns and particularly prevent the application of machine learning algorithms [2]. Synthetic time series generation provides an alternative method for obtaining time series data for analysis. These methods may use existing data and simulation to generate synthetic time series that capture the characteristics of the original data.

Synthetic time series generation also provides a method for simulation based approaches for modeling. For example, an energy grid consists of several classes of customers each of which have different load characteristics. Synthetic time series generation allows the simulation of load models while adjusting for the class of customers [3].

There are many existing methods for generating synthetic time series. Turowski et al. conducted an extensive literature review on energy time series generation [7]; however, these methods can also be applied to time series from different domains. The three most popular methods were Markov models, weighted random number generators and generative adversarial networks (GANS). In this paper, the proposed model may be classified as a weighted random number generator where the elementary time series are the original time series. Markov- and GAN-based models were utilized to assess the performance of the proposed models. The proposed model is relatively simple to implement and computationally inexpensive when compared to neural network approaches in particular.

A. Problem Formulation

Let N and M denote the number of original and synthetic time series (respectively) and T their common length. Let $x_i(t)$ denote the value of original time series $1 \leq i \leq N$ at time $1 \leq t \leq T$ and the $y_j(t)$ denote the value of synthetic time series $1 \leq j \leq M$ at time $1 \leq t \leq T$. Let $x(t)$ and $y(t)$ denote the random variable of the original and synthetic time series at time t (respectively). Define $\mu(t)$ and $\sigma^2(t)$ as the sample mean and sample variance (resp) of the N original time series at time t. That is for $1 \leq t \leq T$,

$$\mu(t) = \frac{1}{N}\sum_{i=1}^{N} x_i(t) \quad \text{and} \quad \sigma^2(t) = \frac{1}{N-1}\sum_{i=1}^{N}(x_i(t)-\mu(t))^2 \tag{1}$$

In this paper, our aim is to develop a model to generate synthetic time series while preserving the statistical properties and temporal dynamics of the original data. The objectives are (a) to generate synthetic time series using the developed models and (b) evaluate the performance of the models against existing models using different datasets.

II. Methodology

In this paper, a model is proposed and the performance of this model is compared to the performance of a Markov Model and a Generative Adversarial Network (GAN). The proposed model seeks to produce synthetic time series which have the same mean, variance and autocorrelation function as the original data.

A. Proposed Model

The proposed model generates synthetic time series $y(t)$ by using the following formula

$$y(t) = \left(\frac{\sum_{i=1}^{N}\alpha_i x_i(t)}{\sum_{i=1}^{N}\alpha_i} - \mu(t)\right)\sqrt{\frac{N(N+1)}{N-1}} + \mu(t) \tag{2}$$

for $1 \leq t \leq T$ and where the $\alpha_i > 0$ $(i = 1 \ldots N)$ are N Exponential(1) random numbers. The α_i were chosen to follow an exponential distribution instead of a uniform distribution

since it simplified analysis and allowed the use of well-known results. Additionally, the factor $\sqrt{\frac{N(N+1)}{N-1}}$ was chosen to preserve the variance of the original data. Also, note that for large sample sizes N, this scaling factor is approximately $\sqrt{N}$.

During the analysis of this model, the observed time series $x_i(t)$ are treated as constant and the α_i are random variables. Proofs establishing the expected value (mean) and the covariance structure of the generated time series $y(t)$ are performed first, then the variance is derived.

In these proofs, the distribution of transformed random variable $\beta_i = \frac{\alpha_i}{\sum_{i=1}^{N} \alpha_i}$ is used since formula for $y(t)$ may be equivalently written as

$$y(t) = \left(\sum_{i=1}^{N} \beta_i x_i(t) - \mu(t) \right) \sqrt{\frac{N(N+1)}{N-1}} + \mu(t) \tag{3}$$

Since the α_i are independent and identically distributed exponential variables, the $\beta_1, \ldots \beta_N$ have a joint Dirichlet($\vec{1}_N$) distribution with the following mean and covariance structure [8]:

$$E[\beta_1] = \frac{1}{N} \tag{4}$$

$$Var[\beta_1] = \frac{\frac{1}{N}\left(1 - \frac{1}{N}\right)}{N+1} = \frac{N-1}{N^2(N+1)} \tag{5}$$

$$Cov[\beta_1, \beta_2] = \frac{-\frac{1}{N^2}}{N+1} = \frac{-1}{N^2(N+1)} \tag{6}$$

1) Mean: For $t = 1 \ldots T$, note that

$$\begin{aligned} &E[y(t)] \\ &= \left(\sum_{i=1}^{N} E\left[\beta_i\right] x_i(t) - \mu(t) \right) \sqrt{\frac{N(N+1)}{N-1}} + \mu(t) \\ &= \left(\sum_{i=1}^{N} \frac{1}{N} x_i(t) - \mu(t) \right) \sqrt{\frac{N(N+1)}{N-1}} + \mu(t) \\ &= (\mu(t) - \mu(t)) \sqrt{\frac{N(N+1)}{N-1}} + \mu(t) \\ &= \mu(t) \end{aligned} \tag{7}$$

Therefore, the mean of the original data is preserved.

2) Covariance Structure: Let $t_1, t_2 = 1 \ldots T$ with the possibility that both are equal. Define the sample covariance of the N original time series as

$$\sigma(t_1, t_2) = \frac{1}{N-1} \sum_{i=1}^{N} (x_i(t_1) - \mu(t_1))(x_i(t_2) - \mu(t_2)) \tag{8}$$

Then,

$$\begin{aligned} &Cov[y(t_1), y(t_2)] \\ &= \frac{N(N+1)}{N-1} Cov\left[\sum_{i=1}^{N} \beta_i x_i(t_1), \sum_{i=1}^{N} \beta_i x_i(t_2) \right] \\ &= \frac{N(N+1)}{N-1} \left(\begin{array}{l} \sum_{i=1}^{N} x_i(t_1) x_i(t_2) Var[\beta_i] \\ + \sum_{i \neq j} x_i(t_1) x_j(t_2) Cov[\beta_i, \beta_j] \end{array} \right) \\ &= \frac{N(N+1)}{N-1} \left(\begin{array}{l} Var[\beta_1] \sum_{i=1}^{N} x_i(t_1) x_i(t_2) \\ + Cov[\beta_1, \beta_2] \sum_{i \neq j} x_i(t_1) x_j(t_2) \end{array} \right) \\ &= \frac{1}{N(N-1)} \left(\begin{array}{l} (N-1)(N-1)\sigma(t_1, t_2) \\ + (N-1)N\mu(t_1)\mu(t_2) \\ - N(N-1)\mu(t_1)\mu(t_2) \\ + (N-1)\sigma(t_1, t_2) \end{array} \right) \\ &= \sigma(t_1, t_2) \end{aligned} \tag{9}$$

since

$$\begin{aligned} &\sum_{i=1}^{N} x_i(t_1) x_i(t_2) \\ &= (N-1)\sigma(t_1, t_2) + N\mu(t_1)\mu(t_2) \end{aligned} \tag{10}$$

and

$$\begin{aligned} &\sum_{i \neq j} x_i(t_1) x_j(t_2) \\ &= \left(\sum_{i=1}^{N} x_i(t_1) \right) \left(\sum_{i=1}^{N} x_i(t_2) \right) - \sum_{i=1}^{N} x_i(t_1) x_i(t_2) \\ &= N^2 \mu(t_1)\mu(t_2) \\ &\quad - [(N-1)\sigma(t_1, t_2) + N\mu(t_1)\mu(t_2)] \\ &= N(N-1)\mu(t_1)\mu(t_2) - (N-1)\sigma(t_1, t_2) \end{aligned} \tag{11}$$

Therefore, this model preserves the autocovariance structure of the original data. Additionally, the variance is preserved since

$$Var[y(t)] = Cov[y(t), y(t)] = \sigma(t, t) = \sigma^2(t) \tag{12}$$

B. Markov method

A second order non-homogeneous Markov method based on the work of Labeeuw and Deconinck was employed to evaluate the performance of the proposed models [3]. The Markov chain was calculated using the following steps:

1) The empirical cumulative distribution function (CDF) was constructed from the values of the original time series $x_i(t)$
2) The limits of 20 states of equal probability were computed using the empirical CDF.

3) The values of the original time series were transformed using the limits of the 20 states to produce a time series $s_i(t)$
4) A second order non-homogeneous Markov chain was calculated based on the state time series

$$P(s(1) = i), \quad P(s(2) = i \mid s(1) = j)$$
$$P(s(t+2) = i \mid s(t+1) = j,\ s(t) = k) \tag{13}$$

for $t = 1, \ldots (T-2)$. The transition probabilities were computed empirically as the ratio of the number of transitions to the total number of transitions from each respective state.

Note that the method in [3] has an initial clustering step; however, preliminary investigation indicated that no clustering was necessary for the chosen datasets. Specifically, the k-means algorithm was applied to the time series for values of $k = 1, \ldots 10$ and both the elbow and Silhouette methods indicated that there were no distinct clusters. In order to generate a synthetic time series using the Markov model, a random time series of states was simulated using the Markov chain and each state was realized using a value by randomly drawing within the limits of their respective states according to the empirical CDF.

C. TimeGAN

The Time-series Generative Adversarial Network developed by Yoon and Jarrett was also employed to evaluate the performance of the proposed models [5]. This model differs from previous GANs as it preserves the temporal dynamics of the time series. It achieves this by consisting of four components: an encoder, decoder generator and discriminator. The model simultaneously trains the auto-encoding components (encoder/decoder) and the adversarial components (generator/discriminator). Temporal representations in the latent space are dynamics are trained by using a recurrent neural network.

TABLE I
TIMEGAN MODEL HYPERPARAMETERS

Hyperparameter	Value
`module`	`gru`
`hidden_dim`	24
`num_layer`	3
`iterations`	10000
`batch_size`	128

The implementation in the associated GitHub repository `https://github.com/jsyoon0823/TimeGAN` is utilized. The hyperparameters used were the defaults and are specified in Table I. Since this implementation produces the same number of synthetic time series as the original time series, the model was run multiple times until the desired number of synthetic time series was obtained for evaluation.

III. RESULTS

A. Datasets

Two source datasets were used to assess the proposed model: household energy consumption and solar irradiance data from the Baseline Surface Radiation Network (BSRN) [9]. These two datasets were chosen since they are known to have a fairly consistent daily pattern, which supports the goal of synthesizing real world time series while preserving the underlying data structure. Two datasets were used from the BSRN source, one containing only time series from the first January to March and another containing all months of the year.

The `power-real` dataset was derived from smart meter readings of 25 households from September to November 2023. For the purposes of this paper, the time series were downsampled from a frequency of 15 minutes to a frequency of one hour and chunked by day (time series of length 24). The hourly power consumption was then extracted and invalid data were removed to produce the `power-real` data set which consisted of 2049 time series. Note that the real power can be negative, indicating net power injection into the grid.

The `brsn` and `bsrnQ1` datasets were derived from the direct solar irradiance readings at the Boulder, Colorado, USA from January 2015 to April 2020. The data was downsampled from a frequency of 1 minute to a frequency of one hour and chunked by day (time series of length 24). The hourly direct solar irradiance values were then extracted and invalid data were removed to produce `bsrn` dataset. The time series in January-March from 2016 to 2020 were used to form the `bsrnQ1` dataset. This was done since solar irradiance patterns during similar calendar periods are more likely to be similar, which will facilitate reliable synthesis. The `bsrn` dataset consisted of 1932 time series whereas the `bsrnQ1` dataset consisted of 537 time series. Since the solar irradiance values are non-negative, two versions of synthetic time series were evaluated, one with the negative values and one another formed by replacing negative values with zero.

B. Metrics

The performance of the models were assessed using the mean, standard deviation and autocorrelation function. Then, the mean of the R^2 values were computed to evaluate the overall performance of each model.

For each synthetic time series dataset, the mean $\mu_{syn}(t)$ and standard deviation $\sigma_{syn}(t)$ were calculated for each time $1 \leq t \leq T$. Then, the R^2 value was used to compare the synthetic mean to the true mean (and likewise for the standard deviation). That is the following two metrics were used:

$$R^2_{mean} = 1 - \frac{\sum_{t=1}^{T}(\mu_{orig}(t) - \mu_{syn}(t))^2}{\sum_{t=1}^{T}(\mu_{orig}(t) - \overline{\mu_{orig}})^2} \tag{14}$$

$$R^2_{sd} = 1 - \frac{\sum_{t=1}^{T}(\sigma_{orig}(t) - \sigma_{syn}(t))^2}{\sum_{t=1}^{T}(\sigma_{orig}(t) - \overline{\sigma_{orig}})^2} \tag{15}$$

The autocorrelation function (ACF) measures the temporal dependence of values within a time series. The ACF $\rho(k)$ for time series $x(t)$ with lag $k = 1 \ldots K(= 20)$ is defined as the correlation between $x(t)$ and $x(t+k)$. Since this function is defined per time series, the R^2 value of the mean

autocorrelation function was used to compare the original and synthetic datasets.

C. Discussion

The proposed models, the Markov model and the TimeGAN model were implemented and each were used to produce synthetic time series for each dataset. The metrics were applied to synthetic time series to assess each model's performance; Tables II-IV contain the results for each dataset where the best (smallest) values are in bold.

TABLE II
POWER-REAL DATASET RESULTS

	Proposed	Markov	TimeGAN
R^2_{mean}	0.999	0.990	0.992
R^2_{sd}	0.998	0.829	0.964
R^2_{acf}	0.995	0.971	0.973
average	0.998	0.930	0.976

TABLE III
BSRN DATASET RESULTS

	Proposed	Proposed (w/o Negatives)	Markov	TimeGAN
R^2_{mean}	1.000	0.994	1.000	0.995
R^2_{sd}	1.000	0.977	1.000	0.999
R^2_{acf}	0.997	0.992	1.000	0.965
average	0.999	0.988	1.000	0.987

TABLE IV
BSRN Q1 DATASET RESULTS

	Proposed	Proposed (w/o Negatives)	Markov	TimeGAN
R^2_{mean}	1.000	0.996	1.000	0.996
R^2_{sd}	1.000	0.984	1.000	0.999
R^2_{acf}	0.996	0.991	1.000	0.997
average	0.999	0.990	1.000	0.997

For `power-real` dataset, the proposed model produces synthetic time series with the best R^2 values across all statistical properties (Table II). Specifically, the R^2 values for the synthetic time series produced by the proposed model are $R^2_{mean} = 0.999$, $R^2_{sd} = 0.998$ and $R^2_{0.995}$. This indicates that the proposed model produces time series which preserve the mean and variation of the original data as well as its auto-correlation structure. The Markov model produced time series which had reasonably accurate mean ($R^2_{mean} = 0.990$) and autocorrelation structure ($R^2_{acf} = 0.971$), but underperformed for the standard deviation ($R^2_{sd} = 0.829$). The TimeGAN also performed reasonably well with $R^2_{mean} = 0.992$, $R^2_{sd} = 0.964$ and $R^2_{acf} = 0.973$.

For the `bsrn` dataset (Table III), Markov model produces synthetic time series with perfect $R^2 = 1$ across all statistical properties (Table III). Since this BSRN dataset contains irradiance data which must be non-negative and the proposed model produced some negative values, the performance of the proposed model was assessed before and after setting these negative values to zero. Initially, the proposed model had synthetic time series which preserved the mean and variance

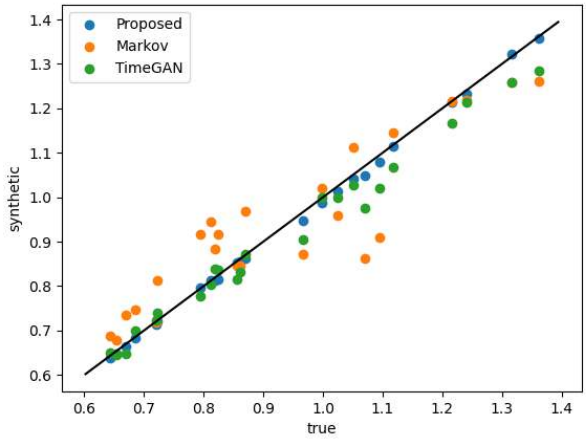

Fig. 1. Power-real dataset synthetic vs true standard deviation

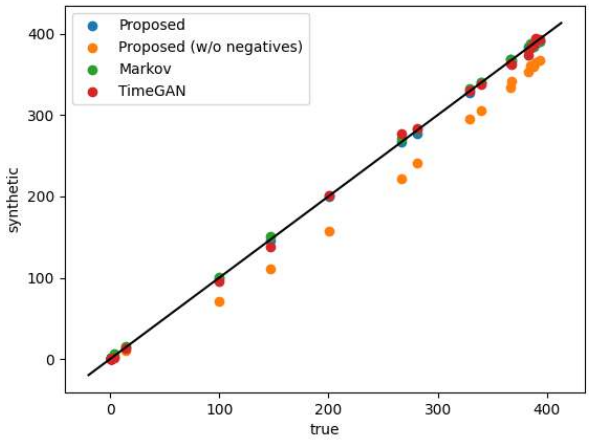

Fig. 2. BSRN dataset synthetic vs true standard deviation

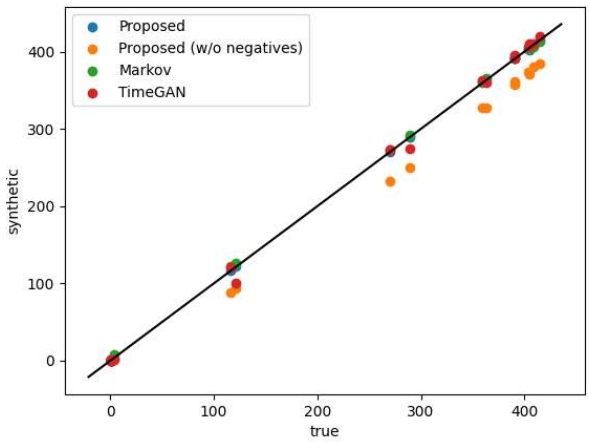

Fig. 3. BSRN Q1 dataset synthetic vs true standard deviation

($R^2_{mean} = R^2_{sd} = 1$) and had similar autocorrelation structure ($R^2_{acf} = 0.997$), but after setting the negative values to zero, the performance degraded and the synthetic time series had $R^2_{mean} = 0.994$, $R^2_{sd} = 0.977$ and $R^2_{acf} = 0.988$. These results were comparable to the synthetic time series produced by the TimeGAN which had $R^2_{mean} = 0.995$, $R^2_{sd} = 0.999$ and $R^2_{acf} = 0.965$.

For the `bsrnQ1` dataset (Table IV), similar results were obtained when compared to the `bsrn` dataset. The proposed and Markov models performed similarly, while the TimeGAN model produced more accurate time series. The proposed model with negative values replaced with zeros, had more accurate mean $R^2_{mean} = 0.994$ and variance $R^2_{sd} = 0.984$ when compared to the full BSRN dataset. The TimeGAN also had more accurate mean $R^2_{mean} = 0.996$ and autocorrelation $R^2_{acf} = 0.997$. These results indicate the importance of clustering time series with similar characteristics in order to produce more accurate synthetic time series.

Figures 1-3 contain scatter plot of the synthetic standard deviation vs the true standard deviation for $t = 1, \dots 24$ for each of the datasets. Across all three datasets, the standard deviation of the synthetic time series produced by the proposed and TimeGAN models closely mirror that of the true data. However, for the Markov model there is a discrepancy of standard deviation of the Markov model is not systematic and occurs for varying values of the true standard deviation (Figure 1). This indicates that the model is not able to reproduce the standard deviation of the original `power-real` dataset. However, the Markov model is able to reproduce the standard deviation in the `bsrn` and `bsrnQ1` datasets. Additionally, from Figures 2 and 3, replacing the negative values produced by the proposed model with zeros systematically reduces the standard deviation of the synthetic time series. This is to be expected since replacing negative values by zeros, effectively reduces the range of the synthetic time series produced, which will reduce the variation of the data.

Across all three datasets, the proposed model has the most accurate and consistent results. The proposed model had an average $R^2 = 0.999$ whereas the Markov and TimeGAN models had average $R^2 = 0.977$ and $R^2 = 0.987$ respectively. The proposed model also had higher average $R^2 = 0.992$ when the synthetic data produced was modified to maintain the non-negativity of the original dataset.

IV. Conclusion

Overall the proposed model produces synthetic time series of higher quality than the Markov and TimeGAN models. In particular, the mean, variation and autocorrelation function of the proposed model are of higher quality than the Markov and TimeGAN models. Further work is necessary to determine a method which preserves the non-negativity of the original time series while maintaining the high quality of the synthetic time series. Additionally, future work can also include the matching of higher-order moments of the original time series; in particular, the skewness (third moment) of the original time series is not necessarily preserved by the models proposed in this paper.

References

[1] Lim, Bryan, and Stefan Zohren. "Time-series forecasting with deep learning: a survey." Philosophical Transactions of the Royal Society A 379.2194 (2021): 20200209.

[2] Zhang, Chi, et al. "Generative adversarial network for synthetic time series data generation in smart grids." 2018 IEEE international conference on communications, control, and computing technologies for smart grids (SmartGridComm). IEEE, 2018.

[3] W. Labeeuw and G. Deconinck, "Residential Electrical Load Model Based on Mixture Model Clustering and Markov Models," in IEEE Transactions on Industrial Informatics, vol. 9, no. 3, pp. 1561-1569, Aug. 2013, doi: 10.1109/TII.2013.2240309.

[4] Candanedo, L. (2017). Appliances Energy Prediction [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5VC8G.

[5] Yoon J, Jarrett D, Van der Schaar M. Time-series generative adversarial networks. Advances in neural information processing systems. 2019;32.

[6] Gretton, Arthur, et al. "A kernel two-sample test." The journal of machine learning research 13.1 (2012): 723-773.

[7] Turowski, M., et al. "Generating synthetic energy time series: A review." Renewable and Sustainable Energy Reviews 206 (2024): 114842.

[8] Shumway, Robert H., and David S. Stoffer. Time series analysis and its applications: with R examples. New York, NY: Springer New York, 2006.

[9] Augustine, John (2020): Basic measurements of radiation at station Boulder, SURFRAD [dataset]. NOAA - Air Resources Laboratory, Boulder, PANGAEA.