ANODE-GAN: Incomplete Time Series Imputation by Augmented Neural ODE-Based Generative Adversarial Networks

Zhuoqing Chang[1], Shubo Liu[1](✉), Zhaohui Cai[1], and Guoqing Tu[2]

[1] School of Computer Science, Wuhan University, Wuhan 430072, China
{changzhuoqing,liu.shubo,zhcai}@whu.edu.com

[2] School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China

Abstract. Missing data is a commonly encountered problems in time series analysis, impeding accurate data analysis. Various methods have been proposed to impute missing values, including statistical, machine learning, and deep learning approaches. However, these methods either involve multi-steps, neglect temporal information, or are incapable of imputing missing data at desired time points. To overcome these limitations, this paper proposes a novel generative framework for imputing missing data, named the Augmented Neural Ordinary Differential Equation-assisted Generative Adversarial Network (ANODE-GAN). ANODE-GAN utilizes a Variational AutoEncoder (VAE) module to maps an incomplete time series instance to fixed-dimension initial latent vectors, generates continuous-time latent dynamics, and finally decodes them into complete data. With the aid of an additional discriminative network, ANODE-GAN can produce complete data that is closest to the original time series according to the squared error loss. By combining the generator and discriminator, ANODE-GAN is capable of imputing missing data at any desired time point while preserving the original feature distributions and temporal dynamics. Moreover, ANODE-GAN is evaluated on real-world datasets with varying missing rates by conducting the imputation task. A set of rigorous experiments show ANODE-GAN outperforms baseline methods in terms of Mean Square Error (MSE).

Keywords: Incomplete Time Series · Generative Adversarial Networks · Augment Neural ODEs

1 Introduction

The past few decades have witnessed a surge in time series data, which has become ubiquitous across a diverse range of applications including finance, healthcare, environmental science, and transportation. Consequently, the analysis of time series has gained immense popularity. However, practical time series

L. Iliadis et al. (Eds.): ICANN 2023, LNCS 14258, pp. 16–27, 2023.
https://doi.org/10.1007/978-3-031-44192-9_2

data are easily trapped into the missing values problem due to sensor malfunctions, interruptions in data transmission, and human errors. Furthermore, some data are sampled at irregular frequencies for cost-saving purposes, such as blood pressure, heart rate measurements in the medical field. These incomplete time series data hinder the application of existing data analysis methods.

Generally, there are two approaches to processing incomplete time series data. The first and simplest approach involves omitting missing data that accounts for less than 15% of the total and making inferences using only the remaining data [2]. However, this method has significant limitations as it discards historical data and ignores potentially valuable hidden information [3]. The second two-stage methodology is a natural solution for this issue. The irregular timespan is divided into uniform intervals, and missing values are imputed or filled by statistical learning [4] or machine learning-based methods [5]. An adverse effect of this discretization is inevitably destroying the measurement timing information that might be informative about latent variables [6,7]. The additional data imputation method operates independently from subsequent data analysis, often leading to suboptimal results.

These issues can be tackled by incorporating time information directly into deep learning models to model the raw data more effectively. A small trick is to concatenate time information to the input of Recurrent Neural Networks (RNNs) [6,8]. Time interval information is further incorporated into the models, allowing them to learn changes in different sampling intervals, such as Gated Recurrent Unit with Decay (GRU-D) [7] and Bidirectional recurrent imputation for time series (BRITS) [9]. Generative Adversarial Networks (GANs) have gained a lot of notoriety. Strenuous efforts have been made to employ a bidirectional RNN in the generator network to learn the distribution of the incomplete data [10,11]. Another line of work attempts to optimize the noise input vector to make the generated sample obey the original distribution, which spends much time finding the best matched input vector [12]. An end-to-end GAN-based imputation model (E2GAN) is proposed to learn high-dimensional data distributions of incomplete time series by the auto-encoder strategy, which is effective to avoid the process of noise optimization [13]. Unfortunately, these methods are unable to impute missing data at desired time points.

This paper proposes a novel generative framework, called Augmented Neural Ordinary Differential Equation-assisted Generative Adversarial Network (ANODE-GAN), to overcome the aforementioned challenges. ANODE-GAN is designed based on a GAN network, consisting of a VAE-based generator and a discriminator. In the generator, a time-aware LSTM (TA-LSTM) learns to encode incomplete time series instance into fixed-dimension initial latent vectors by perceiving the impact of the previous observation. An Augmented Neural Ordinary Differential Equation (ANODE) infers continuous-time latent dynamics by utilizing the posterior distribution of the initial latent state. A fully connected network decodes the latent dynamics into a complete time series. Additionally, a fully connected layer serves as the discriminator to distinguish true from false elements. The generator not only spares no effort to learn continuous latent dynamics of time series and construct the complete time series data, but also deceives the discriminator. By using the generator and the discriminator, ANODE-GAN can automatically learn the original feature distributions and temporal dynamics from incomplete time series, generate latent states at any desired time points and reconstruct these temporal data. This model improves imputation accuracy by more effectively modeling latent dynamics, primarily due to the suitable encoder, stable flow learned from time series to continuous hidden dynamic in an augmented space, and the efficacy of additional discriminator network. The proposed method is evaluated on real-world datasets with varying missing rates. Extensive experiments indicate ANODE-GAN achieves superior imputation accuracy compared to existing methods.

To conclude, the contributions of this paper can be summarized as follows:

- This paper proposes a novel generative framework, ANODE-GAN, to address the issue of missing data in real-world time series data. The proposed framework is capable of imputing data at any desired time and simplifying the data imputation process.
- This is the first work that imputes missing data of incomplete time series by applying ANODEs to the GAN framework, the generator of which employs a TA-LSTM network to compute approximate posterior from partial observations and an ANODE to model more complex functions using simpler flows with improved stability and generalization.
- The ANODE-GAN model is applied to three real-world time series datasets with varying degrees of missing data for the purpose of imputation. Results demonstrate that this method outperforms existing methods in terms of imputation accuracy.

2 Proposed Method

This section presents the proposed ANODE-GAN algorithm to address the issue of incomplete time series analysis. As the framework is shown in Fig. 1, the algorithm is designed under the GAN network, which consists of a generator and a discriminator playing a min-max game [18].

2.1 Generator Network Architecture

A d-dimensional time series $X = \{x_1, x_2, \ldots, x_n\} \in R^{d \times n}$ is observed at $T = \{t_1, t_2, \ldots, t_n\}$, where t is the sampling timestep and x_t is the t^{th} observation. Missing data happens sometimes. Missing values can be indicated by a mask matrix $M = \{m_1, m_2, \ldots, m_n\} \in \{1, 0\} \in R^{d \times n}$ with the same dimension, where m_t is 1 if x_t is revealed and 0 if x_t is missing.

The generator of proposed method is presented in Fig. 1, which is designed based on a VAE framework including a TA-LSTM encoder, an ANODE network and a fully connected decoder.

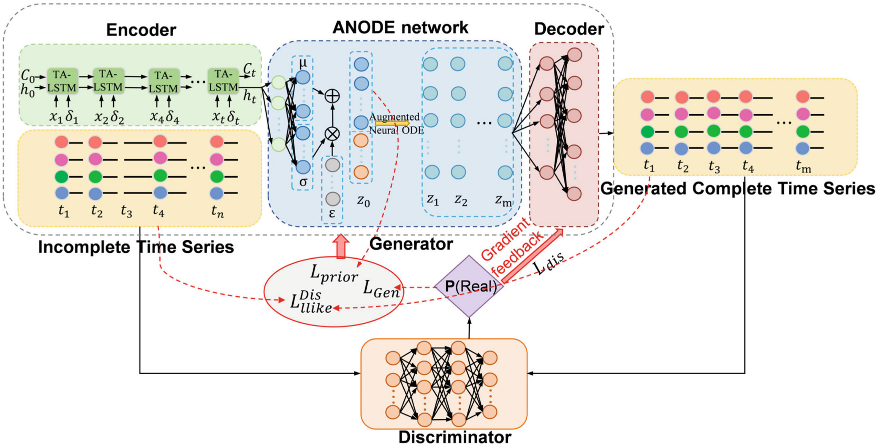

Fig. 1. The proposed ANODE-GAN framework. The generator generates complete data by learning the continuous-time latent dynamics of the incomplete time series. The discriminator employs a fully connected network to predict the truth probability.

Time information is especially important in incomplete time series. On one hand, the time points are closely related to missing rates, on the other, nonuniform time intervals clearly reflect the impact of the previous observation. The RNNs cannot handle nonuniform time interval between successive elements. Inspired by [14], TA-LSTM is employed as the encoder to learn the posterior distribution from partially-observed trajectories. It is of great significance to control the influential decay of the past observation on the current moment. It will introduce a time decay function $\beta(\delta_i^t)$ to modify memory cells, which is validated by previous works [14,19]. Compared with LSTM, the previous memory cell C_{t-1} is TA-LSTM is divided into a short-term memory C_{t-1}^S and a long-term memory C_{t-1}^L. $\beta(\delta_i^t)$ is used to decay the historical influence of the past data on C_{t-1}^S got through a linear network. C_{t-1}^L is obtained by interpolation of C_{t-1} and C_{t-1}^S. An adjusted previous memory $C_{t-1}^{'}$ is established by the sum of C_{t-1}^L and $\beta(\delta_i^t)C_{t-1}^S$. The rest of the calculation process is consistent with the traditional LSTM, only replacing C_{t-1} with $C_{t-1}^{'}$. The update function of TA-LSTM is summarized below:

$$\beta(\delta_i^t) = \frac{1}{\log(e + \delta_i^t)} \tag{1}$$

$$C_{t-1}^S = \tanh(\omega_s C_{t-1} + b_s) \tag{2}$$

$$C_{t-1}^L = C_{t-1} - C_{t-1}^S \tag{3}$$

$$C_{t-1}^{'} = C_{t-1}^L + \beta(\delta_i^t) \times C_{t-1}^S \tag{4}$$

$$f_t = \sigma(v_f x_i^t + u_f h_i^{t-1} + b_f) \tag{5}$$

$$i_t = \sigma(v_i x_i^t + u_i h_i^{t-1} + b_i) \tag{6}$$

$$o_t = \sigma(v_o x_i^t + u_o h_i^{t-1} + b_o) \tag{7}$$

$$\widetilde{C} = \tanh(v_c x_i^t + u_c h_i^{t-1} + b_c) \tag{8}$$

$$C_t = f_t * C_{t-1}^{'} + i_t * \widetilde{C} \tag{9}$$

$$h_t = o_t \circ \tanh(C_t) \tag{10}$$

where the matrices v_p, u_p, and vectors b_p for $p \in \{f, i, o, c\}$ are training parameters. A linear network maps the hidden state of the last moment into a corresponding multivariate normal distribution $\mathcal{N}(\mu_z, \sigma_z)$ in latent space, where μ_z is the mean and σ_z is the standard deviation. To support backpropagating the gradient during training, the initial state z_0 is defined by the sum of a deterministic variable and an auxiliary independent random variable ε.

$$z_0 = \mu_z + \sigma_z \circ \varepsilon \tag{11}$$

where $\circ$ defines the element-wise product and $\varepsilon \sim \mathcal{N}(0, I)$.

Recently, Neural ODEs have been flagged as a possible solution for the continuous-time dynamics modeling, describing the input to output variable transformation by a continuous representation of trajectory through a vector field defined by a neural network [15,16]. Motivated by [17], ANODEs are used to generate latent states of any desired time $T^{'} = \{t_1, t_2, \ldots, t_n\}$, which use the additional dimensions in latent states to avoid trajectories intersecting each other. As presented in Fig. 1, the initial state learn from the last hidden state of TA-LSTM encoder is concatenated with a vector of zeros, which enables to learn more complex latent dynamics using simpler flows.

The continuous time latent states finally flow to another fully connected layer, which will decode the latent state to a generated data $X^{'}$.

The ANODE-GAN mainly performs the imputation task, which is designed based a VAE framework. The loss function of the generator includes two parts, one is Kullback-Leibler (KL) penalty $L_{prior} = D_{KL}[q_\phi(z \mid x) \mid\mid p_\theta(z)]$ and the other is the reconstruction error $E_{z \sim q_\phi(z|x)}[log p_\theta(x \mid z)]$. The following describes the total loss L_{Gen}.

$$L_{Gen} = -E_{z \sim q_\phi(z|x)}[log p_\theta(x \mid z)] + D_{KL}[q_\phi(z \mid x) \mid\mid p_\theta(z)] \tag{12}$$

2.2 Discriminator Network Architecture

The discriminator consists of three fully connected layers to learn a global contextual information. It aimed to identify whether each data in the completed time series is real or imputed rather than distinguish the whole completed vector is true of fake. To get an estimated probability that illustrates the degree of authenticity, *tanh* is used as the activation function in the first two layers and sigmoid activation function is adopted in the last layer. The discriminator updates a set of parameters that generates large probability when real data is encountered and low probability when false data is coming. Thus, the loss function of can be expressed as follows.

$$L_{Dis} = -(E[logD(X)] + E[log(1 - D(X^{'}))]) \tag{13}$$

where D is the estimated mask probability of the discriminator. The discriminator attempts to make log output of prediction on real data close to 1 (the first term) and minizine the loss for imputed data (the second term).

The generator aims at the imputed data closer to the truth ones by fooling the discriminator D. The adversarial loss of generator can be calculated as follows:

$$L_{adv} = E[\log(1 - D(X^{'}))] \tag{14}$$

2.3 Training of ANODE-GAN

The total training loss of ANODE-GAN is expressed as the following shows.

$$L_{ANODE-GAN} = \lambda_G L_{Gen} + L_{adv} \tag{15}$$

where λ_G is hyper-parameter. The lose function can be trained by the BPTT algorithm. The ANODE-GAN provides an adversary strategy for incomplete time series imputation. The discriminator D is trained with the truth data and imputed completed data, and the mask matrix effectively provides supervision on the imputed data, making the generated data closer to the real data.

3 Experimental

In this section, the proposed ANODE-GAN method is evaluated on three real-world datasets with missing values. Results of this experiment are provided and further analyzed in details.

3.1 Datasets and Baseline Models

Three datasets including a Gas dataset, a GAMS dataset and an Electricity dataset are used to evaluate the proposed ANODE-GAN algorithm.
Gas Sensor Array Temperature Modulation Dataset[1]: The dataset contains 13 text files that corresponds to a different measurement day. Each file records a time-dependent multivariate response of 14 MOX gas sensors to the different gas stimuli every 0.3 s. The evaluation data selected in this paper is recorded on October 5, 2016, with more than 90,909 pieces of data.

[1] Gas sensor array temperature modulation Data Set. Available on: https://archive.ics.uci.edu/ml/datasets/Gas+sensor+array+temperature+modulation.

GAMS[2]**:** It is a public complete air quality dataset published by the gams Environmental Monitoring company (denoted as GAMS). The GAMS indoor data is selected in this paper, air quality of which is collected every minute between 2016/11/21 to 2017/3/28. CO_2, Humidity, PM10, PM2.5, Temperature, and Voc in GAMS are measured more than 130,000 times as independent variables.
Electricity[3]**:** It is a widely-used University of California Irvine (UCI) public dataset. The electricity consumption in kWh is recorded every 15 min between 2012/01/01 to 2014/12/31 for 321 clients, which has no missing data. The data in this paper is converted to express hourly consumption.

The proposed ANODE-GAN method is compared with the following baselines.
Gated Recurrent Unit (GRU) [1]**:** Gated Recurrent Unit. An improved version of RNN that is capable of capturing long-term temporal dependencies.
GRU-D [7]**:** A hidden state decay mechanism is introduced in the GRU model to deal with the missing data issue.
Time-Aware LSTM (T-LSTM) [14]**:** Another kind of LSTM that improves the internal structure to tackle the irregular time interval issue.
Latent ODE [15]**:** It is one of advanced imputation method based on VAE, where continues time latent states generated by neural ODEs according to the initial state learned from a RNN encoder will be decoded to impute the missing values.
ODERNN-VAE [16]**:** An additional ODE is used in the encoder to better learn the approximate posterior than RNN on sparse data.
ANODE [17]**:** An additional vector of zeros is concatenated with the initial state to avoid latent trajectories intersecting each other.
Adversarial Joint-Learning Recurrent Neural Network (AJ-RNN) [20]**:** It is GAN-based imputation method that trained in an adversarial and joint learning manner where a discriminator is introduced to minimize the negative impact of missing data in the generator.

3.2 Creating Missing Data

The missing data in this paper assumes the missing completely at random (MCAR) regime. For imputation task, this paper randomly drops out k percent of time series, where $p \in \{10, 20, \ldots, 70\}$. The incomplete time series are imputed by the proposed ANODE-GAN method and comparison methods. MSE is used to calculated the imputation accuracy.

[2] GAMS Indoor Air Quality Dataset. Available on: https://github.com/twairball/gams-dataset.

[3] ElectricityLoadDiagrams20112014 Data Set. Available on: https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014.

3.3 Imputation Performance Comparison for Incomplete Time Series

In implementation, for all baseline models, batch size is 50, learning rate is 0.001, dropout rate is 0.5, and training epoch is 1500. The hyper-parameters λ_G of the generator is 1 for Gas and 0.5 for Gams and 0.3 for Electricity, which is determined based on the results of the ablation study below. The dimensionality of the latent states in Gas, Gams and Electricity is 30, 60, and 132. The ADAM algorithm is adopted to train all the networks. For all the experiments, input data are normalized with zero mean and unit variance, and 80% of dataset as selected as train set and the rest 20% as test set.

Table 1. The MSE results of ANODE-GAN and comparison methods on the Gas dataset.

Missing Rate (%)	GRU	GRU-D	T-LSTM	Latent ODE	ODERNN-VAE	ANODE	AJ-RNN	ANODE-GAN
0	.0565	.0482	.0144	.0321	.0224	.0195	.1004	.0166
10	.0703	.0624	.1076	.0166	.0170	.0199	.1171	.0169
20	.0962	.0737	.2031	.0162	.0164	.0177	.1162	.0153
30	.1250	.1153	.2961	.0246	.0226	.0345	.1403	.0161
40	.1614	.1518	.3892	.0223	.0211	.0281	.1442	.0159
50	.2017	.1907	.4818	.0285	.0194	.0206	.1468	.0165
60	.2572	.3056	.5799	.0259	.0275	.0251	.1675	.0160
70	.3245	.4011	.6742	.0290	.0287	.0286	.2377	.0165

Table 1 presents the imputation results on Gas dataset by using ANODE-GAN and baseline methods including GRU, GRUD, T-LSTM, Latent ODE, ODERNN-VAE, Augmented ODE, and AJ-RNN. The first column in Table 1 is missing rate that represents how many percent values are randomly dropped and other columns and other columns denote MSE results of corresponding imputation methods in terms of corresponding data missing rate. It can be observed that ANODE-GAN achieves the best imputation accuracy in high missing-rate cases. Additionally, imputation accuracy of most traditional methods worsens with increasing missing rate.

Conventional algorithms including GRU, GRU-D and T-LSTM gain relatively high imputation accuracy when missing rate is less than 10%. With higher missing rate, these methods cannot accurately infer the missing values, because they impute missing values by hidden state of previous moment. AJ-RNN does not generate imaginatively accurate imputation values, because it utilizes RNNs to model the captured data and cannot learn the distribution with missing values. Imputation results using ODE-based methods maintain good within varying missing rates. Such methods can model the original incomplete data distribution by a VAE structure and learn the continuous-time latent dynamics via an ODE model. The proposed method takes advantage of GANs to model the incomplete data distribution to generate more accurate imputed values.

Table 2 also presents the imputation performance on the Gams dataset using the proposed ANODE-GAN and various comparison methods. As expected, when the amount of missing data exceeds 10%, the proposed method wins others methods in most cases. This demonstrates that the augmented neural ODE coupled with an additional discriminative network has much higher utility for the missing data imputation.

Table 2. Performance (MSE) of imputation task using the Gams dataset.

Missing Rate (%)	GRU	GRU-D	T-LSTM	Latent ODE	ODERNN-VAE	ANODE	AJ-RNN	ANODE-GAN
0	.0058	.0060	.0050	.0484	.0478	.0531	.0933	.0481
10	.0473	.0448	.1059	.0457	.0461	.0532	.1002	.0477
20	.0908	.0886	.2110	.0497	.0506	.0538	.1138	.0483
30	.1367	.1275	.3105	.0528	.0577	.0717	.0957	.0514
40	.1804	.1752	.4182	.0557	.0546	.0580	.0982	.0535
50	.2329	.2216	.5195	.0526	.2322	.0575	.1073	.0513
60	.2831	.2639	.6280	.0598	.0669	.5689	.0971	.0577
70	.3306	.3081	.7198	.0623	.0596	.0657	.1227	.0543

Table 3. Performance comparison on different imputation models in the Electricity dataset with different missing rates.

Missing Rate (%)	GRU	GRU-D	T-LSTM	Latent ODE	ODERNN-VAE	ANODE	AJ-RNN	ANODE-GAN
0	.0917	.0922	.0835	.2176	.2363	.1896	.3417	.1836
10	.1281	.1598	.1936	.2134	.2681	.1643	.3230	.1667
20	.1635	.2169	.2955	.2548	.2741	.1806	.3071	.1669
30	.2031	.2349	.3946	.2401	.2826	.1905	.2887	.1858
40	.2539	.2770	.4886	.2160	.2408	.1713	.2846	.1691
50	.3164	.3329	.5729	.5937	.2456	.2187	.2884	.1811
60	.3935	.4142	.6571	.5448	.3467	.2405	.2757	.1780
70	.5048	.5274	.7431	.5929	.2540	.1803	.2889	.1773

In order to further verify the effectiveness of ANODE-GAN, the imputation task is also tested on a larger dataset with 321 dimensions. Table 3 shows MSE results tested on the Electricity dataset with different missing rates using ANODE-GAN and baseline algorithms. There is no doubt that the experimental results are consistent with the previous conclusion. The TA-LSTM is capable of learning the unequal time interval and effectively inferring the posterior distribution of irregularly-sampled time series data. The additional discriminative network makes the imputed complete time series closest to the original incomplete data. Furthermore, to present the efficiency of each module in the proposed method, ablation study will be presented in the latter part of this paper.

3.4 Model Analysis

Impact of Hyper-parameter
This paper explores the influence of hyper-parameter λ_G in the imputation task as presented in Fig. 2. The black line represents the MSE result of the imputation task of the hyper-parameter influence in the Gas dataset, and the red line and blue line are impact of the hyper-parameter on the Gams dataset and the Electricity dataset, respectively. The imputation performances vary with the change of hyper-parameter. Too large or too small hyper-parameter makes the MSE indicator rise. This indicates that the loss function of the discriminator makes great contribution to the imputation result. As can be seen, the blue curve reaches the bottom when λ_G is 0.1. The red and black curves get the lowest point when λ_G is 0.5 and 1. Specifically, the large dimension of the electricity dataset make loss of the generator L_{Gen} much larger than the loss of the discriminator L_{adv}, and the hyper-parameter need to reduce the L_{Gen} to make the adversarial network work when the imputation task is executed. On the contrary, the dimensions of Gams and Gas datasets are relatively small, and λ_G needs to be set a little larger. Conclusion can be drawn that it is necessary to adjust the loss proportion of generator and discriminator loss in order to maximize imputation accuracy of the ANODE-GAN method.

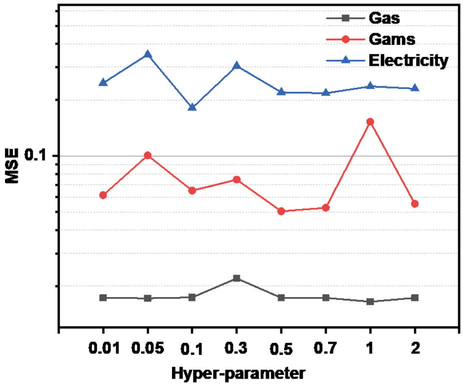

Fig. 2. The influence of hyper-parameter in the imputation task when missing rate is 50%. (Color figure online)

Ablation Study
An ablation study is provided to evaluate the insights and impact of the discriminator and the latent state augment of proposed models. Table 4 presents the imputation results on the Gas dataset, the Gams dataset and the Electricity dataset with a variety of missing rates (measured by MSE). The first ablated model is ANODE-GAN with no latent state augment (ANODE-GAN-no-augment) and the second is ANODE-GAN with no discriminator (ANODE-GAN-no-D). The last model is the proposed ANODE-GAN model. As expected,

the discriminator and the latent state augment are of vital significance to missing data imputation. Evaluated on three datasets, the imputation accuracy decreases without the discriminator or latent augment.

Table 4. The ablation study of discriminator and latent state augment.

Model	Dataset	Missing Rate (%)							
		0	10	20	30	40	50	60	70
ANODE-GAN -no-augment	Gas	.0252	.0178	.0175	.0173	.0220	.0309	.0313	.0342
	Gams	.0458	.0572	.0522	.0684	.0623	.0513	.0630	.0612
	Electricity	.2891	.2895	.1677	.2525	.2768	.2505	.1949	.1803
ANODE-GAN -no-D	Gas	.0241	.0214	.0214	.0201	.0194	.0171	.0231	.0225
	Gams	.0484	.0464	.0542	.0570	.0598	.1526	.0685	.0710
	Electricity	.2745	.2321	.1679	.5990	.1839	.2365	.1894	.1778
ANODE-GAN	Gas	.0166	.0169	.0153	.0161	.0159	.0165	.0160	.0165
	Gams	.0481	.0477	.0483	.0514	.0535	.0505	.0577	.0543
	Electricity	.1836	.1667	.1669	.1858	.1691	.1811	.1780	.0543

4 Conclusion

This paper introduces a novel generative framework called ANODE-GAN, which is capable of imputing missing data at any desired time point. To the best of our knowledge, this paper is the first to use ANODEs in a GAN framework for data imputation. With an additional discriminative network, ANODE-GAN employs the KL penalty, the reconstruction error, and the adversarial loss to generate complete time series that are closest to the original incomplete data. Extensive empirical studies on real-world datasets show that the proposed method improves the accuracy of incomplete time series imputation. Investigating how to make combination of neural controlled differential equations and the GAN framework suitable for dealing with incomplete graph time series data remains to be done in the future work.

Acknowledgments. This work was supported by the Major Projects of Technical Innovation of Hubei Province under Grant 2018AAA046.

References

1. Qing, T., Fang, L., Yong, L., Denis, S.: Air pollution forecasting using a deep learning model based on 1D convnets and bidirectional GRU. IEEE Access **7**(5), 76690–76698 (2019)
2. Graham, J.W.: Missing data analysis: making it work in the real world. Annu. Rev. Psychol. **60**(1), 549–576 (2009)
3. Tian, W., Haoxiong, K., Alireza, J., Sheng, W., Mohammad, H.S., Shuqiang, H.: Missing value filling based on the collaboration of cloud and edge in artificial intelligence of things. IEEE Trans. Industr. Inf. **18**(8), 5394–5402 (2021)

4. Bengio, Y., Gingras, F.: Recurrent neural networks for missing or asynchronous data. In: 9th Conference on Neural Information Processing Systems (NeurIPS), vol. 8 (1995)
5. Hudak, A.T., Crookston, N.L., Evans, J.S., Hall, D.E., Falkowski, M.J.: Nearest neighbor imputation of species-level, plot-scale forest structure attributes from LiDAR data. Remote Sens. Environ. **112**(5), 2232–2245 (2008)
6. Lipton, Z.C., Kale, D., Wetzel, R.: Directly modeling missing data in sequences with RNNs: improved classification of clinical time series. In: Machine Learning for Healthcare Conference, pp. 253–270 (2016)
7. Che, Z., Purushotham, S., Cho, K., Sontag, D., Liu, Y.: Recurrent neural networks for multivariate time series with missing values. Sci. Rep. **8**(1), 1–12 (2018)
8. Choi, E., Bahadori, M.T., Schuetz, A., Stewart, W.F., Sun, J.: Doctor AI: predicting clinical events via recurrent neural networks. In: Machine Learning for Healthcare Conference, pp. 301–318 (2016)
9. Cao, W., Wang, D., Li, J., Zhou, H., Li, L., Li, Y.: BRITS: bidirectional recurrent imputation for time series. In: 32th Conference on Neural Information Processing Systems (NeurIPS), vol. 31 (2018)
10. Miao, X., Wu, Y., Wang, J., Gao, Y., Mao, X., Yin, J.: Generative semi-supervised learning for multivariate time series imputation. In: 35th Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), vol. 35 (2021)
11. Gupta, M., Beheshti, R.: Time-series imputation and prediction with bi-directional generative adversarial networks. arXiv preprint arXiv:2009.08900 (2020)
12. Yoon, J., Jordon, J., Schaar, M.: Gain: missing data imputation using generative adversarial nets. In: 35th International Conference on Machine Learning (ICML), pp. 5689–5698 (2018)
13. Luo, Y., Zhang, Y., Cai, X., Yuan, X.: E^2GAN: end-to-end generative adversarial network for multivariate time series imputation. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI), pp. 3094–3100 (2019)
14. Baytas, I.M., Xiao, C., Zhang, X., Wang, F., Jain, A.K., Zhou, J.: Patient subtyping via time-aware LSTM networks. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 65–74 (2017)
15. Chen, R.T., Rubanova, Y., Bettencourt, J., Duvenaud, D.K.: Neural ordinary differential equations. In: 32th Conference on Neural Information Processing Systems (NeurIPS), vol. 31 (2018)
16. Rubanova, Y., Chen, R.T., Duvenaud, D.K.: Latent ordinary differential equations for irregularly-sampled time series. In: 33th Conference on Neural Information Processing Systems (NeurIPS), vol. 32 (2019)
17. Dupont, E., Doucet, A., Teh, Y.W.: Augmented neural ODEs. In: 33th Conference on Neural Information Processing Systems (NeurIPS), vol. 32 (2019)
18. Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., Bharath, A.A.: Generative adversarial networks: an overview. IEEE Signal Process. Mag. **35**(1), 53–65 (2018)
19. Zhang, Y.: ATTAIN: attention-based time-aware LSTM networks for disease progression modeling. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI), pp. 4369–4375 (2019)
20. Ma, Q., Li, S., Cottrell, G.W.: Adversarial joint-learning recurrent neural network for incomplete time series classification. IEEE Trans. Pattern Anal. Mach. Intell. **44**(4), 1765–1776 (2020)