GlowImp: Combining GLOW and GAN for Multivariate Time Series Imputation

Caizheng Liu[1,2], Houquan Zhou[1,2], Zhi Sun[1,2(✉)], and Guangfan Cui[2]

[1] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
{liucaizheng17b,sunzhi17b}@ict.ac.cn
[2] University of Chinese Academy of Sciences, Beijing, China

Abstract. Multivariate time series data generally contains missing values, which can confound subsequent analysis and compromise downstream applications. Commonly, an imputation method is used to estimate the missing values. The statistical and case deletion type methods can destroy the original data distribution, while RNNs contribute to a higher performance but these model may suffer from bias exploding problem. The method proposed in this paper, GlowImp is a deep sequential latent variable model that combines the advantages of both Glow-VAE with WGAN to learn the actual distribution of the hidden variable space but with high training stability and high-quality samples. Additionally, the framework integrates a variational autoencoder to reduce the dimensionality of the inputs as a way of combatting the high computational costs associated with Glow models. The result is an imputation process with state-of-the-art accuracy (ROC), as demonstrated with two public datasets.

Keywords: Data mining · Multivariate time series imputation · Deep neural network

1 Introduction

Many real-world data can be represented as multivariate time series. Heartbeats, temperature, stock prices, traffic flows - any data that changes value over time, any data captured by a sensor or measured at intervals - all of it can take the form of multivariate time series data. Unfortunately, sensor failures, power failures, human error and a multitude of other reasons frequently mean that time series data is missing some of its values. These gaps in the data obscure the original data distribution, undermining the accuracy of any subsequent analysis and the efficacy of the final application [10]. The most common solution to this problem is to impute the missing data via imputation processes. The recently works are to use the complete data of existing observations to build a model or learn the data distribution, estimate and impute in the missing values of the current time series. Once built, the model then imputes the values that are missing. The commonly-regarded "best" frameworks at present involve RNNs, and

Y. Lai et al. (Eds.): ICA3PP 2021, LNCS 13155, pp. 50–64, 2022.
https://doi.org/10.1007/978-3-030-95384-3_4

GANs [18] with autoencoders [21,34] as a more recent entrant that have been demonstrating state-of-the-art performance. Among these autoencoders methdods, [15] proposed a model based on a deep autoencoder that maps the missing values of multivariate time series data into a continuous low-dimensional hidden space, treating the low-dimensional representation as a Gaussian process. The downside is that the VAE structure can not generate real samples; the closest it can do is generate data that are near to real samples, i.e., the samples are fuzzy. Our innovation, inspired by [11,13,23] and the successful application of the Glow model to image generation, is to use a Glow-based autoencoder as an end-to-end multivariate time series imputation model. Called GlowImp, the first step in the framework is to learn the raw data distribution. Then, the raw data distribution is transformed by a series of reversible functions to derive their corresponding hidden variables (from a one-to-one correspondence). Calculating the logarithmic similarity of the training data is done accurately by selecting the appropriate reversible function transformation. Next, the generator in the filling module of the GAN structure is used to generate an accurate prediction value. Finally, the filling value discriminator evaluates the prediction value and updates the network according to the discriminant result.
In summary, our main contributions are as follows:

- We propose an imputation model for multivariate time series data Glow-based autoencoder. The model combines Glow-VAEs and GANs into a generative model that simultaneously learns to encode, generate and compare dataset samples.
- We design Glow-VAEs that infer the latent variables and evaluate and optimize the exact log-likelihood via a series of reversible function transformations instead of calculating its lower bound.
- Combining Glow-VAEs with GANs to generate the missing values encourages highly diverse data samples and prevents mode collapse of the GANs. It also exploits the VAE's capabilities at dimension reduction to lower the high overall time cost typically associated with Glow models.
- Experimental results on multiple real-world datasets show that the method achieves state-of-the-art performance.

2 Related Work

Our literature review covers work related to dealing with missing values. We begin by covering the traditional methods before moving on to discuss the more recent deep learning methods.

2.1 Traditional Methods

Statistical Methods: Simple statistical imputation methods impute missing values via statistical features, such as mean value [20], median value [1], mode value [14], the last observed valid value [2], etc.

Machine Learning Based Methods: Machine learning approaches include expectation-maximization [29], k-nearest neighbor [24], autoregressive [31], autoregressive vector imputation [5], matrix factorization [7], tensor singular value decomposition [19] and multivariate imputation by chained equations (MICE) [4]. Each of these methods relies on carefully hand-designed features or shared features that are based on the particular type of missing data.

2.2 Deep Learning-Based Methods

RNN-based Methods: With RNN methods, recurrent components are trained on an RNN network equipped with a classification or regression model, which significantly improves accuracy. GRU-D [9]assumes that a missing variable can be computed by combining its corresponding last observed value with the global mean and some decay factors. However, this approach has many drawbacks on general datasets [9]. M-RNN, developed by [33], imputes the missing values via a bi-directional RNN. BRITS is fully based on an RNN structure and is able to perform unidirectional uncorrelated, bidirectional uncorrelated and correlated Recurrent Imputation [8]. Although the RNN yields better high accuracy, the model can suffer from problems with exploding bias [6].

VAE-based Methods: Kingma and Welling [22] proposed stochastic gradient VB, a novel estimator of the variational lower bound, for efficient approximate inference with continuous latent variables. Garnelo et al. [16,17] assume that the data is complete at the time of training and therefore missing values are only imputed during the test period. HI-VAE deals with missing data by defining an ELBO whose reconstruction error term only sums over the observed part of the data on heterogeneous and incomplete data [28]. However, HI-VAE does not exploit temporal information. GP-VAE is a deep probabilistic model that combines a VAE with Gaussian processes to capture the temporal dynamics in time series data [15]. The VAE maps the missing data from the input space into a latent space where the temporal dynamics are modeled by the Gaussian processes.

GAN-based Models: GANs were introduced by [18] as a framework for estimating generative models via an adversarial process. GAN architectures provide the discriminator with information in the form of "hints", which reveal partial information about the missingness in the original sample. In this way, the generator ensures that the samples generated stem from a true data distribution [32]. Luo et al. [25] designed a two-second stage GAN called GRUI. The generator tries to generate a realistic time series from a random noise vector z, while the discriminator tries to distinguish whether the input data is real or fake. Although this adversarial-style structure can lead to a significant increase in accuracy, a two-stage training process adds considerable time and training the "best" matched data is not always a stable undertaking, especially when the input contains random noise. E2E-GAN [26] imputes incomplete time series

via an end-to-end strategy. The solution is based on an encoder-decoder GRUI structure as the generator, which can improve the accuracy and stability when training the model. The discriminator consists of a GRUI layer and a fully connected layer working as the encoder. In a recent notable effort, [27] put forward SSGAN, a novel semi-supervised GAN with a generator, a discriminator, and a classifier to predict missing values for partially labeled time series data.

3 Method

3.1 Notations

A multivariate time series $X = \{x_1, x_2, ..., x_n\}$ is a sequence with data observed on n timestamps $T = (t_0, t_1, ..., t_{n-1})$. The i-th observation x_i contains d attributes $(x_i^1, x_i^2, ..., x_i^d)$. Time series X may contain missing values, and a binary mask vector $R^{n \times d}$ is introduced to indicate the missing positions, which is defined as:

$$M_i^j = \begin{cases} 0, & \text{if } x_i^j \text{ is null} \\ 1, & \text{otherwise} \end{cases} \tag{1}$$

If the j-th attribute of x_i is observed, M_i^j is set to 1. Otherwise, M_i^j is set to 0. Figure 1 gives an example of multivariate time series and its corresponding masking vectors.

time series X

5	/	/	/	18
12	32	9	/	76
0	/	24	/	47
x_1	x_2	.....		x_5

masking matrix M

1	0	0	0	1
1	1	1	0	1
1	0	1	0	1
m_1	m_2	.....		m_5

Fig. 1. Multivariate time series and corresponding masking vectors.

3.2 The GlowImp Architecture

To impute reasonable values in place of the missing values, a Glow-VAE/GAN model is trained to learn the original distribution of the dataset. As shown in Fig. 2, the method used is a traditional VAE method designed to make the latent vector of the data coding obey a Gaussian distribution for subsequent data generation. The VAE also optimizes the lower bound of the log-likelihood, i.e., the ELBO (evidence lower bound). It also approximates the distribution of the hidden variable space. This is done by minimizing the KL divergence. However, the model can only approximate the distribution of the hidden variable space, so the data generated will only be fuzzy. As shown in Fig. 3, using a Glow-VAE model instead of a traditional VAE solves this problem. Glow-VAE comprises an encoder, a decoder, and Glow components. Its purpose is to map the data

to low-dimensional space and then use the hidden variables to accurately model the real distribution of the data. In other words, its job is to accurately estimate the log-likelihood. The decoder then generates complete multivariate time series data. In the GAN model, a discriminator calculates the difference between the fake data and the real data. The Glow-VAE is trained jointly with the GAN with updates for the generator and discriminator derived through gradient descent.

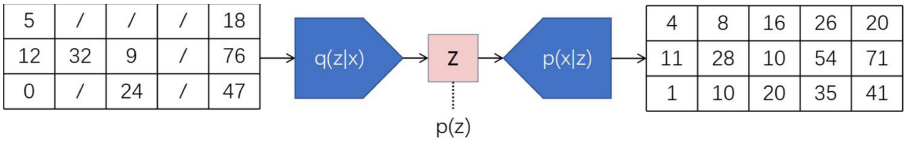


Fig. 2. The structure of the traditional VAE method. VAE method designed to make the latent vector of the data coding obey a Gaussian distribution for subsequent data generation. The VAE also optimizes the lower bound of the log-likelihood, so the data generated will only be fuzzy.

3.3 Glow-VAE

As shown in Fig. 3, Glow-VAE is designed to reconstruct target samples. In addition to mapping the original data to the hidden representations, the encoder also greatly reduces the dimensionality of the data to decrease modeling time. Unlike an encoder in a traditional VAE, Glow-VAEs encoder first learns a Gaussian distribution as hidden representations, which are passed to the Glow model. The Glow model then learns a reversible model F, where the hidden representations can be processed to produce complex data distribution beyond Gaussian distributions. The process is formulated as $z \sim p_\theta(z)$, $x \sim g_\theta(z)$, where z is a complex multivariate Gaussian hidden variable, $p_\theta(z)$ has a tractable density and g_θ is a reversible bijective function that also becomes a bijective function,i.e., $z = f_\theta(x) = g_\theta^{-1}(z)$. Thus, the result obtained is not only accurate, but also can be interpolated. To impute the missing values, it is simply a question of minimizing the following:

$$L_R = E_{z\sim p_\theta(z|x)}[q_\phi(\hat{x} \mid z)] + KL(p_\theta(z \mid x)\|p(z)), \tag{2}$$

$$L_G(D) \cong \frac{1}{N}\sum_{i=1}^{N} -\log p_\theta(x^{(i)}) + c, \tag{3}$$

$$L_{Glow-VAE} = L_R + L_G(D). \tag{4}$$

Where L_R is the loss of VAE, $L_G(D)$ is the loss of Glow model. Given dataset D, $x^{(i)} = x^{(i)} + u$, $u \sim U(0, a)$, $c = -M \cdot \log_a$, $p_\theta(z)$ is a prior distribution. a is determined by the discrimination level of the data and M is the dimensionality of x.

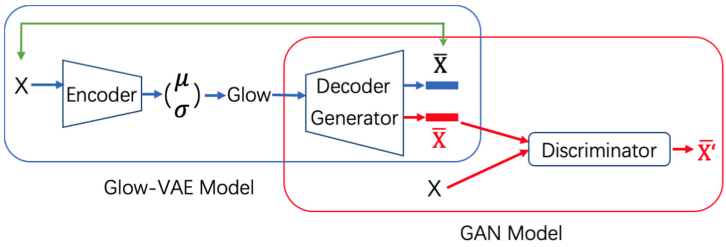


Fig. 3. The GlowImp framework of the multivariate time series missing value imputing model.

3.4 Glow Model Details

Unlike traditional VAEs, the Glow-VAE network structure is reversible. The encoder and decoder actually have similar network structures, except that they are input from different ends when the encoding and decoding functions are performed. As such, a Glow network can be seen as being divided into several large layers, each containing many depth units. As shown in Fig. 4(a), a depth unit consists of two parts, [23].

The first component is the actnorm layer that performs an affine transformation using a scale and bias parameter, similar to batch normalization. These parameters are initialized such that the post-actnorm activation have zero mean and unit variance given an initial mini-batch of data.

The second component is affine coupling layers. With careful function design, a Glow model can be learned that is both tractable and extremely flexible. This property is guaranteed by a series of Jacobi transformations (see [12] for the details of the proof). As shown Fig. 4(b), the affine coupling layers divide the feature vector z passed from actnorm in the channel dimension into two vectors to result in z1 and z2. The length of z1 is half of the feature vector z. The equations in the affine coupling layers are as follows:

$$\begin{aligned} y_1 &= z_1, \\ y_2 &= z_2 \odot exp(s(z_1)) + t(z_1), \\ z &= concat(y_1, y_2). \end{aligned} \tag{5}$$

Where the s and t denote the nonlinear transformations. Since computing the Jacobian determinant of the coupling layer operation does not involve computing the Jacobian of s or t [12], we made them deep convolutional neural networks in our experiments.

3.5 The GAN Architecture

The GAN consists of a generator (G) and a discriminator (D). G learns a mapping that tries to map the hidden vector from Glow-VAE to a predicted multivariate time series, while D tries to find a mapping that can distinguish between

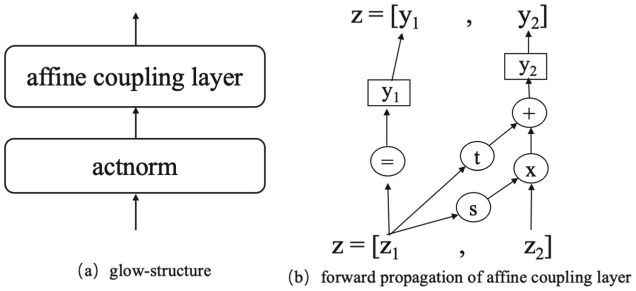


Fig. 4. The structure of the Glow module in the GlowImp framework.

the true and fake generated samples. It is noteworthy that the input of the D contains real, but incomplete, samples and fake, but complete, samples generated by G. Because of the mode collapse problem, a traditional GAN can be hard to train. WGAN is an extension of GAN that uses the Wasserstein distance and is usually easier to train than the original GAN. WGAN can improve stability in the learning stage. It can also escape from mode collapse, making the model easier to optimize. A generative model can be learned to optimize divergence notions beyond the earth mover's divergence. We chose the WGAN by [3] over a traditional GAN. The loss function of WGAN in our model follows:

$$L_{GAN} = \sum_{x \in D} \|x \otimes m - \hat{x} \otimes m\|_2 \ + \beta(E_{\hat{x} \sim q_\phi(\hat{x}|z)}[f_D(\hat{x})] - E_{x \sim P_r}[f_D(x)]). \quad (6)$$

x represents the input multivariate time series data, and m means the masking matrix. The first expression in Eq. 6 is the masked reconstruction loss that calculates the squared errors between the original sample x and the generated sample. It is noteworthy that we only calculate the non-missing part of the data. The second expression in Eq. 6 computes discriminative loss. β is a hyperparameter that controls the proportion between the masked reconstruction loss and the discriminative loss.

The overall objective function of our GlowImp model is formulated as:

$$L_{total} = L_{Glow-VAE} + L_{GAN}. \quad (7)$$

4 Experiment

To verify and measure the performance of the GlowImp framework, we imputed missing values in two real-world data sets and compared the imputations to other contemporary methods.

4.1 Dataset Description

KDD CUP 2018 Dataset: The KDD CUP Challenge 2018 dataset[1] contains air quality and weather data collected hourly collected from 20 Jan 2017 to 30 Jan 2018 in Beijing. Each record contains 12 features, including M2.5, PM10, CO, weather, temperature, etc., with roughly 15% of the values missing. We selected 11 of the 12 features and imputed the missing values in batches of one hour and 48 h.

Imputation performance was assessed in terms of root mean squared error (RMSE) and mean absolute error (MAE). RMSE is the mean value of the square root of the error between the predicted value and the true value and is a popular choice for assessing the imputation in multivariate time series. The formulation is as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x - \hat{x})^2}.$$

MAE(Mean Absolute Error) is the absolute residual, which is summed and averaged to get the MAE; it reflects the size of the error between the actual value and the imputed value of the missing value in the multivariate time series. The formulation is as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x - \hat{x}|,$$

where x is the actual value , $\hat{x}$ is the imputed value and n is the number of the time internal.

Challenge 2012 Health-care Data: The PhysioNet Challenge 2012 [2] dataset is a public collection of multivariate clinical time series data. The set consists of around 8,000 records gathered from intensive care units. Each record is a multivariate time series of about 48 h containing 42 features, including age, gender, height, heart rate, glucose levels, etc. Roughly 80.67% of the values are missing, which is too many to directly impute with any real level of accuracy. However, 554 patients are labeled as either alive (negative mortality) or dead (positive mortality), which is enough data to impute values for the remaining records and train a model in a (binary) mortality prediction task. For a fair comparison, we imputed the values using each of the different models but used the same classifier to perform the predictions. Performance was evaluated in terms of AUC (area under ROC curve) and Accuracy (ACC), calculated as:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN},$$

[1] KDDCUP. Available on: http://www.kdd.org/kdd2018/, 2018.

[2] Predicting in-hospital mortality of icu patients. Available on: The physionet/ computing in cardiology challenge 2012. In CinC (2012).

where TP is true positive, TN is true negative, FP is false positive and FN is false negative. ACC(Accuracy) refers to the proportion of correct predictions in all prediction structures. AUC is an indicator to measure the performance of a multivariate time series missing value imputing model.

4.2 Comparison Methods

We compared GlowImp to eight current imputation methods. A brief description of each follows.

- Statistical imputation methods [30], where the missing values are simply replaced with zero, mean and median.
- KNN [24], which uses a k-nearest neighbor algorithm to find neighboring data. Unobserved data is imputed as the weighted average of k neighbors.
- MF [7], which factorizes an incomplete matrix into low-rank matrices to fill the missing values.
- SVD [19], which uses matrix completion through iterative singular value decomposition (SVD).
- GP-VAE [15], a method that combines ideas from VAEs and Gaussian processes to capture temporal dynamics for time series imputation.
- BRITS [8], one of methods that uses bidirectional RNN to impute time series.
- GRUI [25], which uses a based two-stage GAN to impute missing values.
- E2E-GAN [26], one of the state-of-the-art methods. It relies on an end-to-end GAN network to impute the missing values in time series data.

4.3 Implementations Details

All experiments were conducted under the same conditions with the same settings. The hardware platform was an Intel i7 9700k PC with 48GB memory and an NVIDIA GTX 1080 8 GB video card. The deep learning framework was PyTorch 1.7 and TensorFlow 1.15.0. The Glow model contained five actnorm and affine coupling layers.

To maintain the same number of training and test samples as the contemporary method, the dataset was split with 80% of the records used for the training set and the remaining 20% used for the test set. All values are normalized in the range of 0 to 1. For training, 50% of the KDD data in the training set was deleted. When testing, we deleted between 10% and 80% of the KDD data, tested each method at a range of levels of missing data. For PhysioNet dataset, we imputed the values using each of the different models but used the RNN to perform the predictions.

4.4 Experiment Results

KDD Dataset Results: The results with the KDD dataset and a missing value ratio of 10% appear in Table 1. Here, GlowImp returned significantly fewer errors than the other methods in RMSE and MAE.

Table 1. The RMSE and MAE results of the GlowImp and other methods on the KDD datasets(lower is better)

Method	Zero	Mean	Median	KNN	MF	SVD	GP-VAE	BRITS	GRUI	E2E-GAN	GlowImp
RMSE	1.009	1.009	1.079	0.853	0.683	1.553	0.994	0.196	0.179	0.153	**0.136**
MAE	1.004	1.005	1.038	0.924	0.826	1.246	0.989	0.168	0.142	0.114	**0.089**

Generally, the proportion of missing data in multivariate time series may be uncertain. Data with a higher proportion of missing values means that it is more difficult to correctly predict the missing values. Therefore, the ability to correctly predict data with different missing ratios is an important basis for evaluating model performance. To assess the frameworks with different levels of missing data, we then conducted the same experiment with the BRITS, GRUI, E2E-GAN and GlowImp, varying the ratios of missing values from 10% to 80% in steps of 10%. The results are shown in Fig. 5. Again, GlowImp returned the fewest errors.

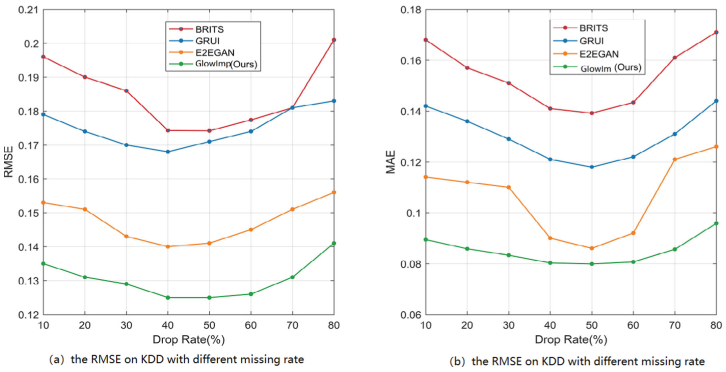
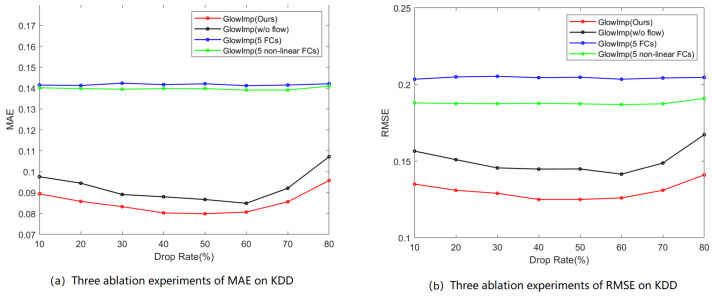


Fig. 5. The RMSE and MAE results of the GlowImp and other methods on the KDD dataset(the lower, the better).

Our third set of experiments was an ablation study, designed to assess the contribution of the Glow module. It comprised four tests: the first with the Glow module, the second where we simply removed the Glow module; a third where we replaced the Glow module with a 5-layer fully connected layer; and the last where we replaced the Glow module with a five-layer nonlinear fully connected layer. All tests were conducted with a range of missing value ratios. Figure 6 shows the results. The tests with the Glow module returned substantially fewer errors, verifying Glow's contribution to the framework.

Figure 7 and Fig. 8 show the imputation results of Tongzhou and Mentougou districts, respectively. The blue point is the ground true time series and the read curve is the imputed values. We could see that GlowImp could capture the evolution trend and impute the missing values pretty well. The GlowImp could



(a) Three ablation experiments of MAE on KDD

(b) Three ablation experiments of RMSE on KDD

Fig. 6. Three ablation experiments of the GlowImp framework on RMSE and MAE on KDD dataset(the lower, the better).

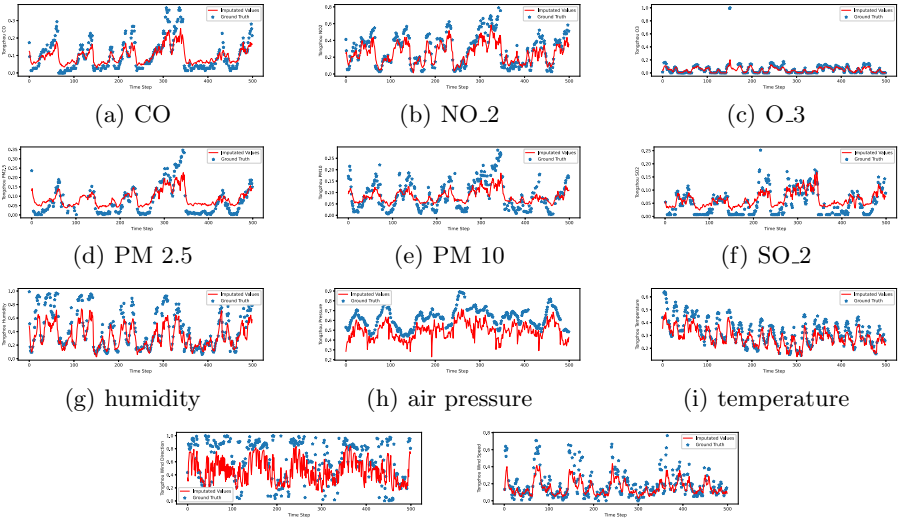
capture the potential probability density distribution of multivariate time series and make full use of the interactive information of multivariate time series. We can observe that in Fig. 7(b) NO_2 and Fig. 7(c) O_3 maintain better performance than the Fig. 7(j) wind direction and Fig. 7(k) wind speed, as the change pattern of NO_2 and O_3 are relatively regular, while the wind changes in a chaotic manner, verifying it is difficult to capture the potential probability density distribution perfectly with chaotic behavior.

PhysioNet Dataset Results: We evaluated reliability in terms of ACC and AUC with the PhysioNet dataset and the binary prediction task. For a fair comparison, we imputed the values using each of the different models but used the same RNN classifier to perform the predictions. The results appear in Fig. 9, with GlowImp demonstrating very competitive scores to the other methods.

PhysioNet dataset roughly 80.67% of the values are missing and we conduct ablation study by randomly deleting the observed data varying the ratios of missing values about 80% and 90%. It comprised three tests: the first with the Glow module, the second where we replaced the Glow module with a 5-layer fully connected layer; and the last where we replaced the Glow module with a five-layer nonlinear fully connected layer. Table 2 and Table 3 shows the results. The tests with the Glow module returned higher ACC and AUC, verifying Glow's contribution to the framework.

Table 2. The AUC results of the GlowImp and other methods on the PhysioNet datasets

Method	5 FCs	5 non-FCs	Ours(GlowImp)
80%	0.751	0.872	**0.881**
90%	0.737	0.859	**0.864**



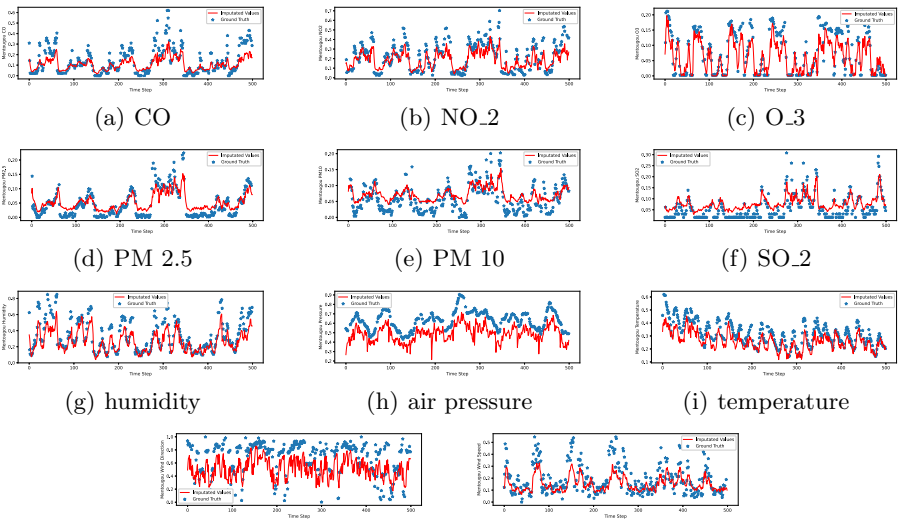
(a) CO (b) NO_2 (c) O_3

(d) PM 2.5 (e) PM 10 (f) SO_2

(g) humidity (h) air pressure (i) temperature

(j) wind direction (k) wind speed

Fig. 7. The KDD dataset imputed results. The ground true(blue) and the imputed values(red) in Tongzhou, Beijing of KDD dataset. The X axis indicates time step. The Y axis indicates imputed values. (Color figure online)



(a) CO (b) NO_2 (c) O_3

(d) PM 2.5 (e) PM 10 (f) SO_2

(g) humidity (h) air pressure (i) temperature

(j) wind direction (k) wind speed

Fig. 8. The KDD dataset imputed results. The ground true(blue) and the imputed values(red) in Mentougou, Beijing of KDD dataset. The X axis indicates time step. The Y axis indicates imputed values. (Color figure online)

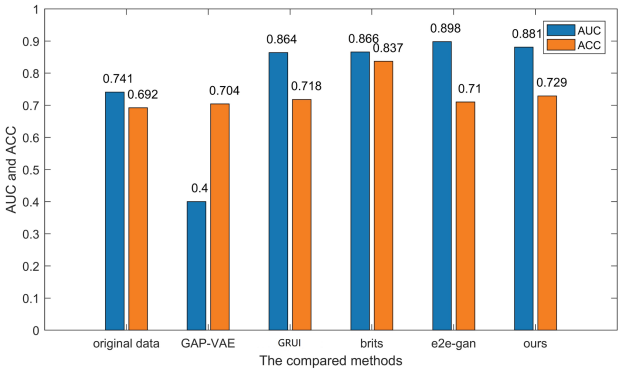


Fig. 9. Comparison of ACC and AUC results with previous methods on the Physionet data set. The RNN model that uses the dataset imputed by our method achieves very competitive scores.

Table 3. The ACC results of the GlowImp and other methods on the PhysioNet datasets

Method	5 FCs	5 non-FCs	Ours(GlowImp)
80%	0.718	0.722	**0.729**
90%	0.671	0.699	**0.711**

5 Conclusion

In this paper, we proposed a novel model for imputing missing values from multivariate time series data. The model, based on Glow-VAE, solves the problem with traditional VAEs where the distribution of the hidden variable space can only be learned approximately. The combination of Glow-VAE and a WGAN structure means that only a small amount of training data is needed to achieve the same accuracy as most previous methods. Experiments with two public datasets show our proposed method imputes missing values with state-of-the-art accuracy. Although in our paper we focused on time-series data arising in KDD and PhysioNet datasets, we believe that our approaches will be widely useful for a variety of time-series tasks. We will explore our model to characterize missing-not-at-random data and we will conduct theoretical analysis to understand the behaviors of existing solutions for missing values in the further works.

Acknowledgements. This paper is partially supported by Strategic Priority Research Program of Chinese Academy of Sciences, Grant No. XDA19020400, National Science Foundation of China under Grant No. U1911401,91746301, 61772498, 61872206, 61802370.

References

1. Acuña, E., Rodriguez, C.: The treatment of missing values and its effect on classifier accuracy. In: Classification, Clustering, and Data Mining Applications, pp. 639–647 (2004)
2. Amiri, M., Jensen, R.: Missing data imputation using fuzzy-rough methods. Neurocomputing **205**, 152–164 (2016)
3. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 214–223 (2017)
4. Azur, M.J., Stuart, E.A., Frangakis, C., Leaf, P.J.: Multiple imputation by chained equations: what is it and how does it work? Int. J. Methods Psychiatr. Res. **20**(1), 40–49 (2011)
5. Bashir, F., Wei, H.L.: Handling missing data in multivariate time series using a vector autoregressive model-imputation (VAR-IM) algorithm. Neurocomputing **276**, 23–30 (2018)
6. Bengio, S., Vinyals, O., Jaitly, N., Shazeer, N.: Scheduled sampling for sequence prediction with recurrent neural networks. In: Proceedings of the 28th International Conference on Neural Information Processing Systems, vol. 1, pp. 1171–1179 (2015)
7. Li, C., et al.: Recommending missing sensor values. In: 2015 IEEE International Conference on Big Data (Big Data), pp. 381–390 (2015)
8. Cao, W., Wang, D., Li, J., Zhou, H., Li, L., Li, Y.: BRITS: bidirectional recurrent imputation for time series. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
9. Che, Z., Purushotham, S., Cho, K., Sontag, D., Liu, Y.: Recurrent neural networks for multivariate time series with missing values. Sci. Rep. **8**(1), 6085 (2018)
10. Cheema, J.R.: A review of missing data handling methods in education research. Rev. Educ. Res. **84**(4), 487–508 (2014)
11. Dinh, L., Krueger, D., Bengio, Y.: Nice: non-linear independent components estimation (2015)
12. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real NVP. CoRR abs/1605.08803 (2016)
13. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real NVP (2017)
14. Donders, A.R.T., van der Heijden, G.J., Stijnen, T., Moons, K.G.: Review: a gentle introduction to imputation of missing values. J. Clin. Epidemiol. **59**(10), 1087–1091 (2006)
15. Fortuin, V., Baranchuk, D., Raetsch, G., Mandt, S.: GP-VAE: deep probabilistic time series imputation. In: Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, vol. 108, pp. 1651–1661 (2020)
16. Garnelo, M., et al.: Conditional neural processes. In: Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 1704–1713 (2018)
17. Garnelo, M., et al.: Neural processes. arXiv:1807.01622 [cs, stat] (2018)
18. Goodfellow, I.J., et al.: Generative adversarial nets. In: Proceedings of the 27th International Conference on Neural Information Processing Systems, vol. 2, pp. 2672–2680 (2014)
19. He, J., Sun, G., Zhang, Y., Geng, T.: Data recovery in heterogeneous wireless sensor networks based on low-rank tensors. In: 2016 IEEE Symposium on Computers and Communication (ISCC), pp. 616–620 (2016)
20. Kantardzic, M.: Data Mining: Concepts. Methods, and Algorithms, Models (2011)

21. Kingma, D.P., Welling, M.: Auto-encoding variational bayes (2014)
22. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, 14–16 April 2014, Conference Track Proceedings (2014)
23. Kingma, D.P., Dhariwal, P.: Glow: generative flow with invertible 1×1 convolutions. In: Advances in Neural Information Processing Systems, vol. 31. Curran Associates, Inc. (2018)
24. Liew, A.W.C., Law, N.F., Yan, H.: Missing value imputation for gene expression data: computational techniques to recover missing data from available information. Brief. Bioinf. **12**(5), 498–513 (2011)
25. Luo, Y., Cai, X., Zhang, Y., Xu, J., Yuan, X.: Multivariate time series imputation with generative adversarial networks. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, pp. 1603–1614 (2018)
26. 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 Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, pp. 3094–3100 (2019)
27. Miao, X., Wu, Y., Wang, J., Gao, Y., Mao, X., Yin, J.: Generative semi-supervised learning for multivariate time series imputation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 10, pp. 8983–8991 (2021)
28. Nazabal, A., Olmos, P.M., Ghahramani, Z., Valera, I.: Handling Incomplete Heterogeneous Data using VAEs. arXiv:1807.03653 [cs, stat] (2020)
29. Nelwamondo, F.V., Mohamed, S., Marwala, T.: Missing data: a comparison of neural network and expectation maximization techniques. Curr. Sci. **93**(11), 1514–1521 (2007)
30. Rubinsteyn, A., Feldman, S.: fancyimpute: an imputation library for python (2016). https://github.com/iskandr/fancyimpute
31. Sridevi, S., Rajaram, S., Parthiban, C., SibiArasan, S., Swadhikar, C.: Imputation for the analysis of missing values and prediction of time series data. In: 2011 International Conference on Recent Trends in Information Technology (ICRTIT), pp. 1158–1163 (2011)
32. Yoon, J., Jordon, J., van der Schaar, M.: GAIN: missing data imputation using generative adversarial nets. In: Proceedings of the 35th International Conference on Machine Learning, vol. 80, pp. 5689–5698 (2018)
33. Yoon, J., Zame, W., Schaar, M.: Multi-directional recurrent neural networks: a novel method for estimating missing data (2017)
34. Zhao, J., Kim, Y., Zhang, K., Rush, A.M., LeCun, Y.: Adversarially regularized autoencoders (2018)