Time Series Data Imputation: A Survey on Deep Learning Approaches

Chenguang Fang[a,*], Chen Wang[a]

[a] *Tsinghua University, Beijing*

Abstract

Time series are all around in real-world applications. However, unexpected accidents for example broken sensors or missing of the signals will cause missing values in time series, making the data hard to be utilized. It then does harm to the downstream applications such as traditional classification or regression, sequential data integration and forecasting tasks, thus raising the demand for data imputation. Currently, time series data imputation is a well-studied problem with different categories of methods. However, these works rarely take the temporal relations among the observations and treat the time series as normal structured data, losing the information from the time data. In recent, deep learning models have raised great attention. Time series methods based on deep learning have made progress with the usage of models like RNN, since it captures time information from data. In this paper, we mainly focus on time series imputation technique with deep learning methods, which recently made progress in this field. We will review and discuss their model architectures, their pros and cons as well as their effects to show the development of the time series imputation methods.

Keywords: Time Series Imputation, Deep Learning, GAN, RNN

1. Introduction

Time series are vital in real-world applications. However, due to unexpected accidents, for example broken sensors or missing of the signals, missing values are everywhere in time series. In some datasets, the missing rate can reach 90%, which makes the data hard to be utilized [12]. The missing values significantly do harm to the downstream applications such as traditional classification or regression, sequential data integration [21] and forecasting tasks [18], leading to high demand for data imputation.

Our preliminary study [11] shows that imputing the missing values indeed helps significantly the prediction of fuel consumption. In the scenarios of fuel consumption prediction, missing values happen due to the errors of sensors. We propose an imputation approach named FuelNet to deal with such errors. The FuelNet generates proper values to impute missing data. With imputed data, the fuel consumption can be reduced by around 45.5%.

In current stages, time series data imputation is a well studied problem with different categories of methods including deletion methods, simple imputation methods and learning based methods. However, these works rarely take the temporal relations among the observations and treat the time series as normal structured data, thus losing the information from the time data.

Fortunately, with the increasing development of deep learning, a large quantity of deep learning methods are researched, among which RNN is one of the typical methods to handle sequence data. The intuition on why deep learning models could advance imputation tasks is that, they are proven to have the ability to mine information hidden in the time series. These characteristics could enable them to impute missing values with such models.

Recently, deep learning methods have been applied to multivariable time series imputation and show positive progress in imputing the missing values. In this paper, we mainly survey three papers about time series imputation with deep learning methods [7, 27, 6, 28, 25] among which RNN, GRU and GAN are adopted separately or in combination. We will review these papers about their model structure, the common parts they all adopted and the advantages and disadvantages through comparison.

The remainder of the paper is organized as follows. In the next section, we categorize existing data imputation methods and mainly give an introduction to deep learning imputation methods. Section 3 will show the definition of the problems and the symbols. Section 4 will give a detailed discussion of deep learning methods, mainly about their concrete structure, advantages and disadvantages. And finally in Section 5 we summarize the survey and give our conclusions.

2. Categorization

In this section, we will give a brief introduction of the major approaches to time series imputation. Moreover, we will classify existing time series imputation methods according to the principles and techniques they rely on.

In order to impute the missing values, researchers have proposed many imputation methods to handle the missing

*Corresponding author. Tel.: +(86)18867608173

Email addresses: fcg19@mails.tsinghua.edu.cn (Chenguang Fang), wang_chen@tsinghua.edu.cn (Chen Wang)

Table 1: Comparison of different methods addressing time series imputation

Methodologies	Sample approaches from the literature	Time interval	Value type	Time series dimension
Deletion	Listwise Deletion [51]	regular/irregular	qualitative	multidimensional
	Pairwise Deletion [29]	regular/irregular	qualitative	multidimensional
Neighbor Based	QDORC [41]	regular/irregular	quantitative/qualitative	multidimensional
	SRKN [46]	regular/irregular	quantitative/qualitative	multidimensional
Constraint Based	DERAND [43, 42]	regular/irregular	quantitative/qualitative	multidimensional
	SCREEN [44]	regular/irregular	qualitative	single dimensional
Regression Based	ARX [5]	regular	qualitative	single dimensional
	IMR [55]	regular	qualitative	single dimensional
Statistical	DPC [54]	regular	qualitative	single dimensional
	IIM [53]	regular	qualitative	multidimensional
MF Based	TRMF [52]	regular	qualitative	multidimensional
	NMF [30]	regular	qualitative	multidimensional
EM Based	EM [14]	regular	qualitative	multidimensional
	EM-GMM [32]	regular	qualitative	multidimensional
MLP Based	MLP [35]	regular	qualitative	single dimensional
	ANN [33]	regular	qualitative	single dimensional
DL Based	GRU-D [7]	regular/irregular	qualitative	multidimensional
	GRUI-GAN [27]	regular/irregular	qualitative	multidimensional
	BRITS [6]	regular/irregular	qualitative	multidimensional
	E2GAN [28]	regular/irregular	qualitative	multidimensional
	NAOMI [25]	regular/irregular	qualitative	multidimensional

values in time series. In this paper, we mainly conclude 8 kinds of the missing value imputation methods including **deletion methods**, **neighbor based methods**, **constraint based methods**, **regression based methods**, **statistical based methods**, **MF based methods**, **EM based mathods**, **MLP based mathods** and **DL based methods**. Table 1 shows the comparison of these methods we conclude. We will introduce each kind of method respectively as follows.

Deletion methods take a simple strategy that they directly erase the observations that contain missing values from the raw data [29, 51]. It is also a commonly adopted strategy when the missing value is not high and the deletion of the missing values will not influence the downstream applications. However, when the missing rate reaches some level (in [16], it is 5%), ignoring the missing values and deleting them make the data incomplete and not suitable for downstream applications.

Neighbor based methods [3, 41] find out the imputation value from neighbors, e.g., identified by clustering methods like KNN or DBSCAN. They first find the nearest neighbors of the missing values through other attributes, and then update the missing values with the mean value of these neighbors. Moreover, considering the local similarity, some methods take the last observed valid value to replace the blank [2]. SRKN (Swapping Repair with K Neighbors) [46] in our preliminary study could also be adapted to impute the missing values that are misplaced in other dimensions.

Constraint based methods [43, 42] discover the rules in dataset, and take advantage of these rules to impute. To apply to time series data, similarity rules such as differential dependencies [37, 38] or comparable dependencies [39, 40] could be employed that study the distances or similarities of timestamps as well as values [45]. More advanced constraints could be specified in a graph structure [48, 57], such as Petri net, and employed to impute the qualitative values of events in time series [49, 50]. These methods are effective when the data is highly continuous or satisfies certain patterns. For example, when the data is increasing linearly, it is effective and efficient to take simple methods or clustering methods. And when the rules or constraints are satisfied, constraints based methods outperform others in both time and accuracy [44]. However, multivariable time series in the real world are not usually satisfied with such rules, thus more general methods are required and learning based methods are researched to impute the time series automatically.

Regression based methods LOESS [9] learns a regression model from nearest neighbors for predicting the missing value referring to the complete attributes. For time series data, autoregressive (AR) models (e.g., ARX [5] and ARIMA [56]) try to predict missing values from historical data. More advanced IMR (iterative minimum repairing [55]) provides both anomaly detection and data repair for both anomalies and missing values. These methods mostly benefit from historical data as well as the accuracy of the nearest neighbors. Thus they could be applied when neighbors are reliable and the time series are highly relative.

Statistical based methods rely on statistical models to impute the missing values [24]. Simple statistical methods just utilize the data in the original data to impute the missing values, such as take the mean value or median value of the attribute to impute [1, 20]. [54] estimates probability values by statistics on speeds as well as the changes. Recently, more advanced IIM (Imputation via Individual models) [53] adaptively learns individual models for various number of neighbors. Unlike regression based methods which based on just historical data, statistical based models are learned from the whole dataset, including historical data and future data. Therefore, they may capture more information from raw data.

Matrix Factorization based methods The Matrix Factorization (MF) algorithm tries to impute the value with the Matrix Factorization and reconstruction to find the correlations among the data and complete the missing values which is a classical method of collaborative filtering [26]. In recent years MF based approaches are introduced into time series imputation fields [52, 30]. In general, MF based approaches decompose the data matrix into 2 low-dimensional matrices in the meantime extracting the features from original data. And then they try to reconstruct the original matrix and in this processing, missing values are imputed.

Expectation-Maximization based methods Expectation-Maximization (EM) based methods have been successfully applied to missing data imputation problems [32, 13, 14]. EM based methods follow a two-stage strategy consisting of the E (Expectation) step and the M (Maximization) step which iteratively imputes the missing values with the statistical model parameters and then updates the statistical model parameters to maximize the possibility of the distribution of the filled data.

Multi-Layer Perceptron based methods Multi-Layer Perceptron (MLP) based methods employee MLP, which is also called fully connected networks. MLP tries to predict missing value by complete values. It can be divided into 3 parts: input layers, hidden layers and output layers. In this approach, by minimizing the loss function, the perceptron learns a function to impute missing values by input variables. In [35], MLP is used to predict missing values in neural network-based diagnostic systems. And in [33], MLP is employed to impute Population Census.

Recently, **deep learning based methods** [7, 27, 28, 6] mainly deploy Recurrent Neural Network (RNN), since RNN is capable of capturing the time information. In these papers, time information is handled separately and attached with more importance. To impute the time series, not only RNN is used, they also combine the models like Gated Recurrent Unit (GRU) [7, 27, 28] to extract the long-term information, Generative Adversarial Networks (GAN) [27, 28] to generate the imputed values and Bidirectional Recurrent Networks to improve the accuracy [6].

According to the above classification, due to the length, the methods for time series imputation are too many to give a detailed introduction. Since among these methods, deep learning based ones are the latest and most powerful, we will discuss 3 latest deep learning methods for time series imputation, find the connections and the differences among them.

3. Preliminary

In this section, we first give our formalization of the imputation tasks. It is because when introducing the aforesaid deep learning methods, they formalize the imputation tasks with different symbols and formulas. And in our research, we review them and explain their methods with uniform definitions.

Definition 1 (Multivariable Time Series)**.** *We first denote a timestamp lists* $\mathbf{T} = (t_0, t_1, ..., t_{n-1})$*, and the time series* $\mathbf{X} = \{\mathbf{x_{t_0}}, \mathbf{x_{t_1}}, ..., \mathbf{x_{t_{n-1}}}\}^T$ *as a sequence of n observations. The i-th observation of* $\mathbf{X}$ *is* $\mathbf{x_{t_i}}$*, which consists of d attributes* $\{x^0_{t_i}, x^1_{t_i}, ..., x^d_{t_i}\}$*.*

After defining the multivariable time series, we use mask matrix $\mathbf{M}$ to denote the missing values.

Definition 2 (Mask Matrix)**.** *Mask Matrix* $\mathbf{M}$ *represents the missing values in* $\mathbf{X}$*, i.e.,* $\mathbf{M} \in \mathbb{R}^{n \times d}$*. And each element of* $\mathbf{M}$ *is defined as below*

$$\mathbf{M}^j_{t_i} = \begin{cases} 0 & \text{if } x^j_{t_i} \text{ is not observed, i.e. } x^j_t = None \\ 1 & otherwise \end{cases} \tag{1}$$

To utilize the time information, the time intervals should be recorded with an extra structure. Therefore, we introduce the time lag, a matrix to represent the time intervals between two adjacent observed values of $\mathbf{X}$.

Definition 3 (Time Lag)**.** *We use* $\delta \in \mathbb{R}^{n \times d}$ *to record the time lag, and we calculate it in an iterative way as follows.*

$$\delta^j_{t_i} = \begin{cases} t_i - t_{i-1}, & \mathbf{M}^j_{t_{i-1}} = 1 \\ \delta^j_{t_{i-1}} + t_i - t_{i-1}, & \mathbf{M}^{j-1}_{t_{i-1}} == 0 \& i > 0 \\ 0, & i == 0 \end{cases} \tag{2}$$

Example 1. We now give an example of the time series $\mathbf{X}$, and corresponding timestamp lists $\mathbf{T}$

$$\mathbf{X} = \begin{bmatrix} 1 & 6 & None & 9 \\ 7 & None & 7 & None \\ 9 & None & None & 79 \end{bmatrix}, \mathbf{T} = \begin{bmatrix} 0 \\ 5 \\ 13 \end{bmatrix} \tag{3}$$

And we can thus compute the mask matrix $\mathbf{M}$ and the time lag δ.

$$\mathbf{M} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \delta = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 5 & 5 & 5 & 5 \\ 8 & 13 & 8 & 13 \end{bmatrix} \tag{4}$$

■

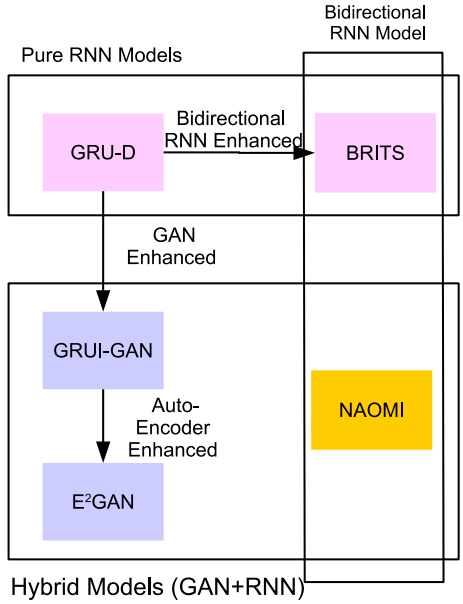


Figure 1: The relationships among methods we mainly surveyed.

4. Methods

In this section, we will first give an overall review of the relationships among the given approaches and comparisons of them and then discuss them individually with details. The main deep learning methods we researched for time series imputation are GRU-D [7], GRUI-GAN [27], E^{2}GAN [28], BRITS [6] and NAOMI [25]. All of them are deep learning approaches published recently for time series imputation tasks. Among these methods, recurrent neural network (RNN) and generative adversarial network (GAN) are main architectures that are adopted. The reason is that RNN and its variations (e.g., LSTM, GRU) have been proven powerful in modeling sequence data, while GAN has been successfully applied to generation and imputation tasks.

To describe the relationships among these methods, we illustrate the dependencies and common structures of them in Figure 1. In Figure 1, we use arrows to describe the dependencies, for example GRUI-GAN improves the work by using GAN while E^{2}GAN is the updated version of GRUI-GAN. And we use boxes to describe the common structures among the methods, for example GRU-D and BRITS are both pure RNN models and BRITS and NAOMI both adopt bidirectional RNN structures. This can help us to understand how the time series imputation task is systematically modeled, how the solutions are developed and what progress people make in this process. In the following sections, we will take a progressive order to review them.

4.1. Characteristics of Chosen Methods

In this section, we give the characteristics of the chosen methods in Table 2 to give a brief introduction and a taxonomy of the chosen methods we reviewed. We consider the following criteria:

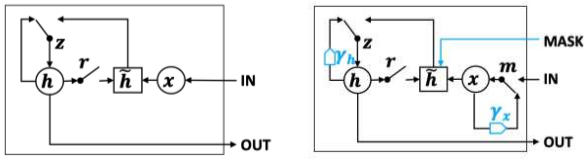
- *Irregular Time Series Awareness*: time series including regular time series with fixed time interval and irregular time series. Both of them are common kinds which are important for classifying the using condition of the methods [54, 44].
- *Model Prototype*: model prototype concludes the overall kind of model in the methods, e.g., RNN, GAN and CNN. It is a basic information to classify the model type. If the model prototype is hybrid, it means more than 1 kind of prototype is employed.
- *Specific Models*: specific models introduce the specific kinds of model adopted in the methods. The specific models may relate to the basic idea of the methods.
- *Auto-Encoder Enhanced*: auto-encoder structure is an approach that can be applied in the imputation of the data. With the structure of encoder and decoder, it extracts the features from low-dimensional layers and recovery missing values by decoder. Therefore, it can serve as a feature of methods.
- *Adversarial Training Enhanced*: adversarial training adopts adversarial structure (e.g., GAN [15] and CGAN [31]) to enhance the model. It takes the idea of generative adversarial structure with generator and discriminator. Large amount of models can be enhanced with such idea.
- *Bidirectional Enhanced*: Bidirectional RNN trains 2 models in forward direction and backward direction respectively with RNN and then combines them into the same loss function [17]. This idea is vital in data imputation tasks since both previous series and future series of missing values are known. Therefore, bidirectional structure benefits from both backward and forward training processing. Such idea is adopted in [25, 6].

4.2. GRU-D

GRU-D is proposed by [7] as one of the early attempts to impute time series with deep learning models. It is the first one among the 5 researched paper to systematically model missing patterns into RNN for time series classification problems. It is also the first research to exploit that, RNN can model multivariable time series with the informativeness from the time series. Former works like [23, 8] attempted to impute missing values with RNN by concatenating timestamps and raw data, i.e., regard timestamps as one attribute of raw data. But in [7], the concept **time lag** is first proposed. In this paper, Gated Recurrent Unit (GRU) is first adopted to generate missing values. In each layer of GRU, since the input can contain missing values, they replace the input $x_{t_i}^j$ with a combination of

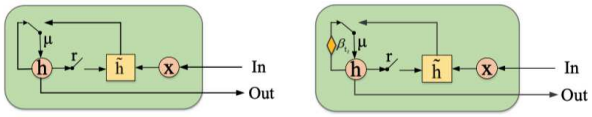
Table 2: Characteristics of the chosen methods

Methodologies	Model Prototype	Specific Models	Auto-Encoder Enhanced	Adversarial Training Enhancednced	Bidirectional Enhanced
GRU-D	RNN	GRU	–	–	–
GRUI-GAN	Hybrid	GRU+GAN	–	yes	–
E2GAN	Hybrid	GRU+GAN	yes	yes	–
BRITS	RNN	Bidirectional RNN	–	–	yes
NAOMI	Hybrid	RNN+GAN	–	yes	yes



(a) GRU (b) GRU-D

Figure 2: Model of GRU and GRU-D. Images extracted from [7].



(a) GRU (b) GRU-I

Figure 3: Model of GRU and GRU-I. Images extracted from [27].

the existing values $x^j_{t_i}$ and statistical values, element-wise multiplied with $\mathbf{M}$ and $\mathbf{1}-\mathbf{M}$ respectively.

$$x^j_{t_i} \leftarrow m^j_{t_i} x^j_{t_i} + \left(1 - m^j_{t_i}\right) \tilde{x}^j$$

where $\tilde{x}$ can be one of the mean value, last observed value or concatenation of $[\mathbf{x_i}; \mathbf{m_i}; \delta_i]$.

The main contribution of this paper is the GRU based model GRU-D and the proposition of **decay rate**. To address the imputation of the missing values, they discover that

- The missing variables tend to be close to some default value if its last observation happens a long time ago.
- The influence of the input variables will fade away over time if the variable has been missing for a while.

And then they propose **decay rate** γ, which is defined as below

$$\gamma_{t_i} = \exp(-\max(\mathbf{0}, \mathbf{W}_\gamma \delta_{\mathbf{t_i}}))$$

The decay rate tries to model the impact of the other values have on the missing values. In brief, it guarantees that the larger the time intervals are, the less their influence on imputing the missing values. And then they replace the input variable as

$$x^j_{t_i} \leftarrow m^j_{t_i} x^j_{t_j} + \left(1 - m^j_{t_i}\right) \gamma^j_{\boldsymbol{x}_{t_i}} x^j_{t'_i} + \left(1 - m^j_{t_i}\right) \left(1 - \gamma^j_{\boldsymbol{x}_{t_i}}\right) \tilde{x}^j$$

Therefore, as illustrated in Figure 2, the GRU-D model is proposed with 2 different trainable decays $\gamma_{\boldsymbol{x}}$ and $\gamma_{\boldsymbol{h}}$, where $\gamma_{\boldsymbol{x}}$ is the input decay rate and the $\gamma_{\boldsymbol{h}}$ is the decay rate for the hidden state.

4.3. GRUI-GAN

In [27], GRU-I is proposed as the recurrent unit to capture the time information. As Figure 3 illustrates, it follows the structure of GRU-D in Section 4.2 with the removal of the input decay. Therefore, there is no innovation in the RNN part as well as the decay rate.

The main contribution of this paper locates in the GAN structure. Figure 4 shows the structure. The Generative Adversarial Network (GAN) structure is made up of a generator (G) and a discriminator (D). The G learns a mapping $G(z)$ that tries to map the random noise vector z to realistic time series. The D tries to find a mapping $D(.)$ that tells us the input data's probability of being real. Therefore, in this paper, the model takes a random noise as the input of the GAN model, which means the generating is a random process. Both G and D are based on GRU-I, and it takes lots of time to train the model to get the data imputed.

The GRUI-GAN takes advantage of the ability of GAN in imputation, which has been proven powerful in image imputation such as [34]. And the adversarial structure improves accuracy. Moreover, the paper adopts a WGAN structure, which improves the stability of the learning stage, get out of the problem of mode collapse and makes it easy for the optimization of the GAN model.

However, this model is not practical since the accuracy of the generative model seems not stable with a random noise input. And it also makes the model hard to converge.

4.4. BRITS

Unlike former methods, BRITS [6] is totally based on RNN structure and proposes imputation with unidirectional dynamics. Time lag (corresponding to "time gaps" in [6]) is also employed since the time series may be irregular. Similar to the idea of decay rate γ from GRU-D introduced in Section 4.2, they propose **temporal decay**

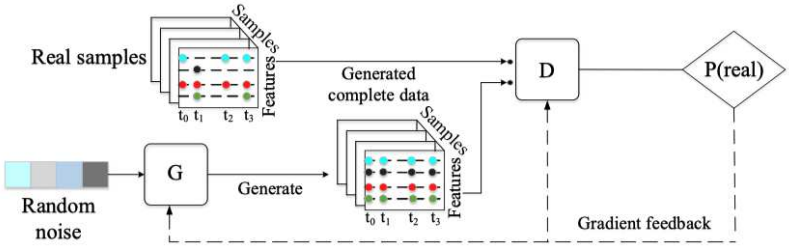


Figure 4: The structure of the GRUI-GAN. Image extracted from [27].

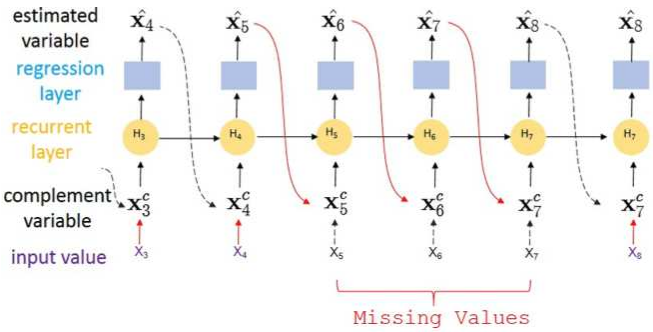


Figure 5: The structure of the BRITS. Image extracted from [6].

factor $\gamma_t = \exp\left(-max\left(0, \mathbf{W}_\gamma \delta_t + \mathbf{b}_\gamma\right)\right)$. Compared to GRU-D where the time lags are considered in input and serve as the decay rate, in BRITS the hidden states update with the decay rate γ. It means when updating the hidden state, the old hidden state decays according to the time duration recorded in the time lags. Hence, the model is updated by:

$$\begin{aligned}
\hat{\mathbf{x}}_t &= \mathbf{W}_x \mathbf{h}_{t-1} + \mathbf{b}_x \\
\mathbf{x}_t^c &= \mathbf{m}_t \odot \mathbf{x}_t + (1 - \mathbf{m}_t) \odot \hat{\mathbf{x}}_t \\
\gamma_t &= \exp\left\{-\max\left(0, \mathbf{W}_\gamma \delta_t + \mathbf{b}_\gamma\right)\right\} \\
\mathbf{h}_t &= \sigma\left(\mathbf{W}_h \left[\mathbf{h}_{t-1} \odot \gamma_t\right] + \mathbf{U}_h \left[\mathbf{x}_t^c \circ \mathbf{m}_t\right] + \mathbf{b}_h\right) \\
\ell_t &= \left\langle \mathbf{m}_t, \mathcal{L}_e\left(\mathbf{x}_t, \hat{\mathbf{x}}_t\right)\right\rangle
\end{aligned} \tag{5}$$

The former model named RITS is the unidirectional version of the proposed methods in [6]. As the bidirectional version, BRITS employs bidirectional RNN by utilizing the bidirectional recurrent dynamics, i.e., they train 2 models in forward direction and backward direction respectively [17]. Thus consistency loss is introduced to take the losses of both directions into consideration.

To conclude, in BRITS, time lags are still adopted to deal with irregular time series. Only RNN is used to model the time series. We can also conclude from the model and the experiments that bidirectional RNN contributes to a higher performance since the unidirectional model may suffer from bias exploding problem [4].

4.5. E²GAN

E^{2}GAN [28] is another work based on GAN. While the GRUI-GAN in Section 4.3 takes a random noise vector as input, which takes lots of time to train, E^{2}GAN adopts an auto-encoder structure based on GRUI to form the generator. The overall structure of their model is in Figure 6.

In E^{2}GAN, concepts including mask, time lag, decay rate and GRUI are all reserved without improvement, thus there is no innovation in the GRUI structure. The main contribution is the auto-encoder structure they adopt in the generator. This is a common strategy taken by image generation and imputation such as Context-Encoder [34], PixelGANs [19], but not a common strategy in RNN based GAN. Since the input of the model is the original time series, the model compresses the input incomplete time series $\mathbf{X}$ into a low-dimensional vector z with the help of the GRUI. And then the reconstructing part will reconstruct the complete time series $\mathbf{X}'$ to fool the discriminator. And the discriminator of the method attempts to distinguish actual incomplete time series $\mathbf{X}$ and the fake but complete sample $\mathbf{X}'$ through the adoption of recursive neural network. The framework of the discriminator is also an encoder.

E^{2}GAN takes an encoder-decoder RNN based structure as the generator, which tackles the difficulty of training the model and the accuracy. So far, according to the experiments in the paper, E^{2}GAN has achieved state-of-the-art and outperforms other existing methods.

4.6. NAOMI

NAOMI (**N**on-**A**ut**O**regressive **M**ultiresolution **I**mputation [25]) proposes a non-autoregressive model which conditions both previous values but also future values, i.e., equipped with bidirectional RNN like BRITS introduced in Section 4.4. Since in the imputation tasks, future values and historical values are both observed, the intuition is to take advantage of both values and train bidirectional models for them. As illustrated in Figure 7, f_f and f_b are forward and backward RNN respectively, thus the hidden state h_t is a joint hidden state concatenated by h_t^f and h_t^b.

Moreover, a special predicting strategy is performed in this paper. They adopt a *divide and conquer strategy*. As it is shown in Figure 7, with 2 known values x_1 and x_5, they first predict the midpoint x_3 by x_1 and x_5 with proposed bidirectional RNN models, and then x_3 is updated and utilized to predict x_2 and x_4 respectively. Thus a fine-grained prediction is performed. Finally, adversarial training is taken to enhance the model.

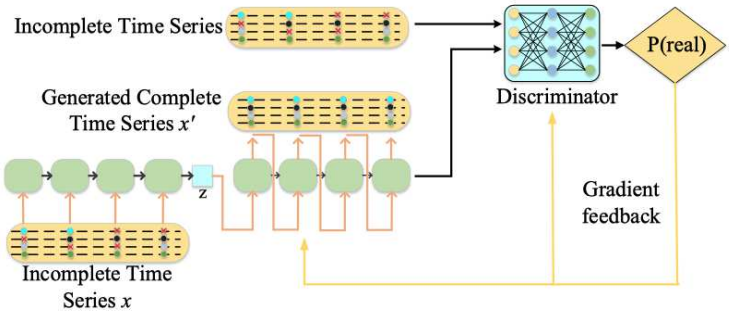


Figure 6: The structure of the E^{2}GAN. Image extracted from [28].

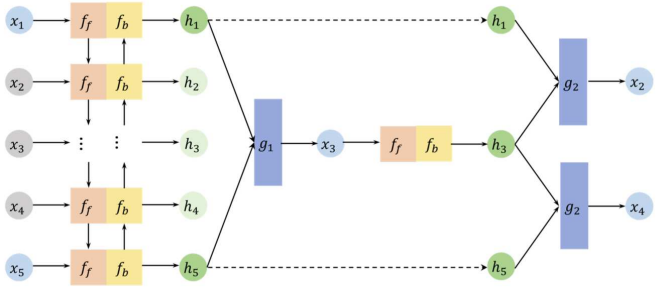


Figure 7: The structure of the NAOMI. Image extracted from [25].

However, in NAOMI, time gaps are ignored and the data is injected into the RNN model without timestamps. It suggests the model is not aware of irregular time series although we can still take them as input by removing their timestamps directly.

5. Conclusion

In this paper, we give a brief introduction to the imputation methods for time series. We propose that existing methods can be classified into 3 main classed: deletion methods, traditional methods, and learning based methods. And we introduce our classification in detail. Moreover, we investigate existing deep learning methods for time series imputation, since they outperform others and make great progress recently. We mainly researched 3 deep learning methods including GRU-D, GRUI-GAN, and E^{2}GAN. All of them based on RNN, and the latter two also adopt GAN for more accurate imputation. We also find the relationships among them: GRUI-GAN is based on the definitions from GRU-D, and E^{2}GAN improves the generator of the GRUI-GAN with auto-encoder. And so far, E^{2}GAN achieves state-of-the-art.

Since the imputation problem is fundamental, we believe with these methods, the filled data would benefit downstream applications in many aspects. And as we observed, most of the techniques in other fields can be adopted in this task since time series data is everywhere. In the future, we would like to see the time information can be utilized properly, and the methods can be more general and accurate so that we would not need to choose the best one from too many methods, and the missing data of the time series would not be a problem.

6. Future Research Opportunities

Based on our observation from surveying the development of time series imputation methodologies, we try to highlight some potential research opportunities in this field. Most existing researches mainly focus on the structure of RNN and try to use bidirectional RNN, Auto-Encoder structure and GAN to enhance the model. With the rapid development in the deep learning society (especially Natural Language Processing (NLP) where time series are also highly concerned), some techniques have reached better performance (e.g., attention models). These models can be considered to enhance the imputation models. Further, most existing methods ignore the missing of timestamps which can also appear obscurely [36]. Therefore, there is still demand for such techniques. Existing methods can be extended to impute missing timestamps. Moreover, query answering without directly imputing missing values is another perspective of dealing with missing values. Under such scenarios, specific values do not need imputation, and consistent queries in inconsistent probabilistic databases should be generated.

6.1. Attention Mechanism Enhanced

In recent years, the attention mechanism has been shown successful in deep learning society, especially in NLP fields. When adopted in RNN, the attention mechanism allocates weights for each hidden state to draw information from the sequence. With such mechanism, the model is improved to capture latent patterns in historical data, thus may benefit time series imputation. Compared to existing RNN models (e.g., LSTM and GRU) which already take long-term dependencies into consideration, the attention mechanism for instance temporal attention enables the model to see features and status globally. However, LSTM and GRU will still lose long-term information due to the forget gate unit.

Recently, pure attention models are proposed without RNN. The Transformer proposed in [47] is one of the popular frameworks. In the proposed Transformer

framework, it only adopts an attention layer called self-attention, which is computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where Q, K, V are queries, keys and values respectively, and d_k is the dimension of the input.

Accepting a single sequence as input, the self-attention mechanism relates different positions of the input and tries to compute a representation of the sequence. Without applying RNN, the Transformer relies entirely on the self-attention layers to former an encoder-decoder structure, which is similar to the auto-encoder introduced in Section 4.1. Such a structure provides the ability to extract high-dimensional features for reconstructing, which benefits tasks like machine translation introduced in [47].

For improving the performance of data imputation, due to the effectiveness of the attention mechanisms, models based on attention mechanisms may also address the time series imputation problems. And two aforementioned categories of the attention mechanisms including temporal attention and self-attention are both potential techniques which may benefit the time series imputation. Moreover, with the idea of removing RNN and leveraging only attention mechanisms, structures like the Transformer may contribute to a new framework for the imputation tasks.

To summary, two categories of attention mechanisms including temporal attention and self-attention may bring future opportunities on time series imputation. And the pure attention frameworks are also new directions to model time series.

6.2. Imputing Missing Timestamps

Missing timestamps often appear obscurely [36], e.g., denoted by `00:00:00`. Most of existing methods mainly focus on the missing values of the time series. However, once timestamps are missing, these methods may fail to capture the information of time and unable to obtain accurate imputation results. Thus, an extension of existing methods to impute missing timestamps is potentially appropriate direction to deal with such scenarios.

6.3. Consistent Query Answering

Following [22], query answering without determining the specific imputation of each missing value is crucial in probabilistic databases [10], when data from many sources can be inconsistent and uncertain. Therefore, consistent query answering (CQA) is needed. Missing values data in CQA problem increase the difficulty of answering the query consistently. Both the inconsistent data from different sources and missing values should be considered. Therefore, a combination of data imputation methods and CQA methods can be a potential approach.

References

[1] E. Acuna and C. Rodriguez. The treatment of missing values and its effect on classifier accuracy. In *Classification, clustering, and data mining applications*, pages 639–647. Springer, 2004.

[2] M. Amiri and R. Jensen. Missing data imputation using fuzzy-rough methods. *Neurocomputing*, 205:152–164, 2016.

[3] G. E. Batista, M. C. Monard, et al. A study of k-nearest neighbour as an imputation method. *HIS*, 87(251-260):48, 2002.

[4] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1171–1179, 2015.

[5] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[6] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li. BRITS: bidirectional recurrent imputation for time series. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 6776–6786, 2018.

[7] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):6085, 2018.

[8] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun. Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine Learning for Healthcare Conference*, pages 301–318, 2016.

[9] W. S. Cleveland and C. Loader. *Smoothing by Local Regression: Principles and Methods*, pages 10–49. Physica-Verlag HD, Heidelberg, 1996.

[10] N. N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. *VLDB J.*, 16(4):523–544, 2007.

[11] C. Fang, S. Song, Z. Chen, and A. Gui. Fine-grained fuel consumption prediction. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 2783–2791, 2019.

[12] P. J. García-Laencina, P. H. Abreu, M. H. Abreu, and N. Afonoso. Missing data imputation on the 5-year survival prediction of breast cancer patients with unknown discrete values. *Comp. in Bio. and Med.*, 59:125–133, 2015.

[13] P. J. García-Laencina, J. Sancho-Gómez, and A. R. Figueiras-Vidal. Pattern classification with missing data: a review. *Neural Computing and Applications*, 19(2):263–282, 2010.

[14] Z. Ghahramani and M. I. Jordan. Supervised learning from incomplete data via an EM approach. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6, [7th NIPS Conference, Denver, Colorado, USA, 1993]*, pages 120–127. Morgan Kaufmann, 1993.

[15] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680, 2014.

[16] J. W. Graham. Missing data analysis: Making it work in the real world. *Annual review of psychology*, 60:549–576, 2009.

[17] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6):602–610, 2005.

[18] T. Hsieh, H. Hsiao, and W. Yeh. Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm. *Appl. Soft Comput.*, 11(2):2510–2525, 2011.

[19] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[20] M. Kantardzic. *Data mining: concepts, models, methods, and algorithms*. John Wiley & Sons, 2011.

[21] Y. Li, Y. Wang, Z. Zhang, Y. Wang, D. Ma, and J. Huang. A novel fast and memory efficient parallel MLCS algorithm for long and large-scale sequences alignments. In *32nd IEEE International Conference on Data Engineering, ICDE 2016, Helsinki, Finland, May 16-20, 2016*, pages 1170–1181. IEEE Computer Society, 2016.

[22] X. Lian, L. Chen, and S. Song. Consistent query answers in inconsistent probabilistic databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2010, Indianapolis, Indiana, USA, June 6-10, 2010*, pages 303–314, 2010.

[23] Z. C. Lipton, D. Kale, and R. Wetzel. Directly modeling missing data in sequences with rnns: Improved classification of clinical time series. In *Machine Learning for Healthcare Conference*, pages 253–270, 2016.

[24] R. J. Little and D. B. Rubin. *Statistical analysis with missing data*, volume 793. John Wiley & Sons, 2019.

[25] Y. Liu, R. Yu, S. Zheng, E. Zhan, and Y. Yue. NAOMI: non-autoregressive multiresolution sequence imputation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 11236–11246, 2019.

[26] X. Luo, M. Zhou, H. Leung, Y. Xia, Q. Zhu, Z. You, and S. Li. An incremental-and-static-combined scheme for matrix-factorization-based collaborative filtering. *IEEE Transactions on Automation Science and Engineering*, 13(1):333–343, 2014.

[27] Y. Luo, X. Cai, Y. Zhang, J. Xu, and X. Yuan. Multivariate time series imputation with generative adversarial networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 1603–1614, 2018.

[28] Y. Luo, Y. Zhang, X. Cai, and X. Yuan. 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 2019, Macao, China, August 10-16, 2019*, pages 3094–3100. ijcai.org, 2019.

[29] P. E. McKnight, K. M. McKnight, S. Sidani, and A. J. Figueredo. *Missing data: A gentle introduction*. Guilford Press, 2007.

[30] J. Mei, Y. de Castro, Y. Goude, and G. Hébrail. Nonnegative matrix factorization for time series recovery from a few temporal aggregates. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 2382–2390. PMLR, 2017.

[31] M. Mirza and S. Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014.

[32] F. V. Nelwamondo, S. Mohamed, and T. Marwala. Missing data: A comparison of neural network and expectation maximization techniques. *Current Science*, pages 1514–1521, 2007.

[33] S. Nordbotten. Neural network imputation applied to the norwegian 1990 population census data. *JOURNAL OF OFFICIAL STATISTICS-STOCKHOLM-*, 12:385–402, 1996.

[34] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[35] P. K. Sharpe and R. J. Solly. Dealing with missing values in neural network-based diagnostic systems. *Neural Computing and Applications*, 3(2):73–77, 1995.

[36] S. Song, Y. Cao, and J. Wang. Cleaning timestamps with temporal constraints. *PVLDB*, 9(10):708–719, 2016.

[37] S. Song and L. Chen. Differential dependencies: Reasoning and discovery. *ACM Trans. Database Syst.*, 36(3):16:1–16:41, 2011.

[38] S. Song, L. Chen, and H. Cheng. Efficient determination of distance thresholds for differential dependencies. *IEEE Trans. Knowl. Data Eng.*, 26(9):2179–2192, 2014.

[39] S. Song, L. Chen, and P. S. Yu. On data dependencies in dataspaces. In *Proceedings of the 27th International Conference on Data Engineering, ICDE 2011, April 11-16, 2011, Hannover, Germany*, pages 470–481, 2011.

[40] S. Song, L. Chen, and P. S. Yu. Comparable dependencies over heterogeneous data. *VLDB J.*, 22(2):253–274, 2013.

[41] S. Song, C. Li, and X. Zhang. Turn waste into wealth: On simultaneous clustering and cleaning over dirty data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, pages 1115–1124, 2015.

[42] S. Song, Y. Sun, A. Zhang, L. Chen, and J. Wang. Enriching data imputation under similarity rule constraints. *IEEE Trans. Knowl. Data Eng.*, 32(2):275–287, 2020.

[43] S. Song, A. Zhang, L. Chen, and J. Wang. Enriching data imputation with extensive similarity neighbors. *PVLDB*, 8(11):1286–1297, 2015.

[44] S. Song, A. Zhang, J. Wang, and P. S. Yu. SCREEN: stream data cleaning under speed constraints. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31 - June 4, 2015*, pages 827–841, 2015.

[45] S. Song, H. Zhu, and L. Chen. Probabilistic correlation-based similarity measure on text records. *Inf. Sci.*, 289:8–24, 2014.

[46] Y. Sun, S. Song, C. Wang, and J. Wang. Swapping repair for misplaced attribute values. In *36th IEEE International Conference on Data Engineering, ICDE 2020*. IEEE, 2020.

[47] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

[48] J. Wang, S. Song, X. Lin, X. Zhu, and J. Pei. Cleaning structured event logs: A graph repair approach. In *31st IEEE International Conference on Data Engineering, ICDE 2015, Seoul, South Korea, April 13-17, 2015*, pages 30–41, 2015.

[49] J. Wang, S. Song, X. Zhu, and X. Lin. Efficient recovery of missing events. *PVLDB*, 6(10):841–852, 2013.

[50] J. Wang, S. Song, X. Zhu, X. Lin, and J. Sun. Efficient recovery of missing events. *IEEE Trans. Knowl. Data Eng.*, 28(11):2943–2957, 2016.

[51] W. Wothke. Longitudinal and multigroup modeling with missing data. 2000.

[52] H. Yu, N. Rao, and I. S. Dhillon. Temporal regularized matrix factorization for high-dimensional time series prediction. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 847–855, 2016.

[53] A. Zhang, S. Song, Y. Sun, and J. Wang. Learning individual models for imputation. In *35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8-11, 2019*, pages 160–171. IEEE, 2019.

[54] A. Zhang, S. Song, and J. Wang. Sequential data cleaning: A statistical approach. In *Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, pages 909–924. ACM, 2016.

[55] A. Zhang, S. Song, J. Wang, and P. S. Yu. Time series data cleaning: From anomaly detection to anomaly repairing. *PVLDB*, 10(10):1046–1057, 2017.

[56] G. P. Zhang. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, 50:159–175, 2003.

[57] X. Zhu, S. Song, X. Lian, J. Wang, and L. Zou. Matching heterogeneous event data. In *International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014*, pages 1211–1222, 2014.