# Missing Data Imputation for Real Time-series Data in a Steel Industry using Generative Adversarial Networks

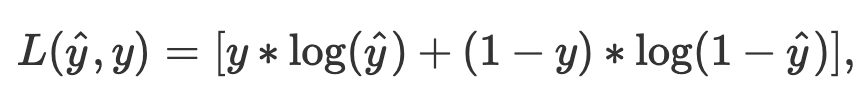
<https://ieeexplore.ieee.org/abstract/document/9589716>

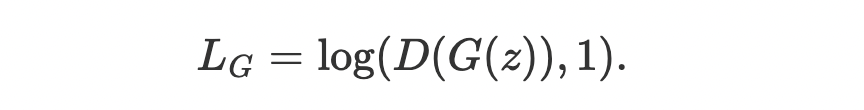
## Methodology

A multivariate time-series data *X* with total *n* samples (*t*0,…,*tn*−1), and *d* features is given as X = (x1 … xt(n-1)

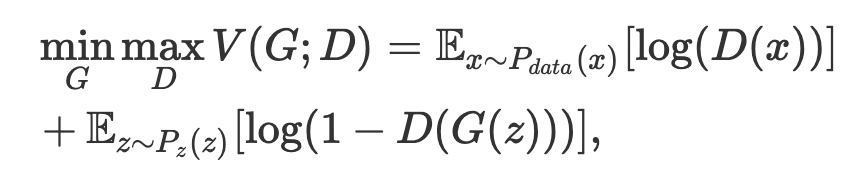
, where ℝ is a set of real numbers and *xti* is a *d*-dimensional observation/sample at *ti.* An incomplete data *Xincomplete* consists of missing values, denoted by *nan. An incomplete data Xincomplete consists of missing values, denoted by nan. In the interest of the missing values, we use the mask matrix M, introduced in to identify the missing values’ location. The aim of the mask matrix is to replace synthetic data over the missing values.*

#### ***Mathematical model***

*Loss function for discriminator - *

*Loss function for generator - *

*Hence, based on the above description, D needs to label original samples x as 1, and generated samples G(z) have to be labeled as 0 i.e., to maximize the D loss function. Whereas in case of G,D(G(z)) has to be labeled as 1 i.e., the generated samples have to be labeled as original samples. Hence, the error function has to be minimized. In other words, D seeks to maximize the average of log probability of the original samples and log of the inverted probabilities of fake samples. Whereas G seeks to minimize the average of the log of inverse probability following two-player min-max game with value loss function V(G; D) as follows:*

**

### ***Hyper-parameters of GAN***

the GAN has various hyper-parameters, such as learning rate, batch size, number of iterations, optimizer, number of layers, activation function, and loss function, which affect the model performance. In this paper, we have mainly considered loss function and optimizer to tune the GAN model, whereas the remaining hyper-parameters are kept fixed. These hyper-parameters are selected as learning rate = 0.001, number of iterations = 6000, number of layers = 3, activation function = Leaky\_ReLU. The batch size is dependent on the number of missing values (here, we have selected batch size = 192). In the following, we briefly discuss the selection of loss function and optimizer.

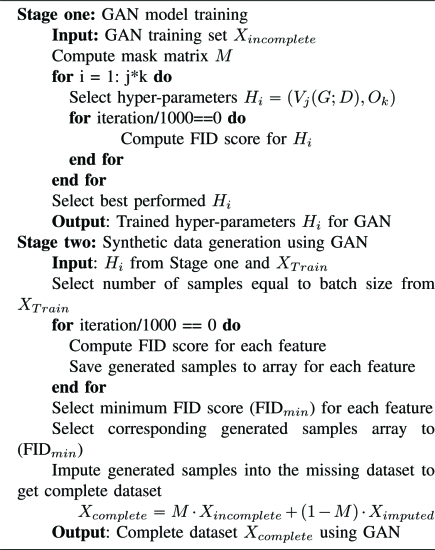
#### **1) Loss function**

The loss function is the main hyper-parameter of the GAN model that may influence the performance majorly. In our case, we have considered seven different loss functions, namely, original GAN, least square, Kullback-Leibler divergence (KLD), Jenson-Shannon divergence (JSD), Pearson *X*2, Square Hellinger, and Reverse KLD.

#### 

#### **2) Optimizer**

The optimization algorithm is an important hyper-parameter of GAN on which the speed and reliability of the model is dependent. We have considered RMSProp optimizer and Adam optimizer in our tuning case in which Adam (Adaptive moment estimation), a first-order gradient-based optimization of stochastic objective function based on adaptive lower-order moments, works faster and more reliable when the value function is tuning

**

## *Results And Discussions*

#### ***1) Mode collapse***

*The generator aims to produce a wide variety of output similar to real samples. However, it happens to limit the variety of samples to the same kind of samples (e.g., nan), and the generator is stuck in further iterations. This kind of issue is known to be a mode collapse. In our case, we have obtained the mode collapse problem for two loss functions, JSD and KLD. In contrast, the best performance of GAN has been obtained for the least square loss function. Hence, in the final model of GAN,* ***the least square loss function is used***

*In this paper, we have presented an efficient data imputation technique based on GANs. Several experimental validations have been performed to show the effectiveness of the proposed method. The GAN based data imputation outperforms the state-of-the-art data imputation methods.*

*To extend the applications of GAN to Industrial data, generating the fault samples identifying the distribution of the fault event could be one future direction. Hence, the problem of imbalanced data can be resolved.*

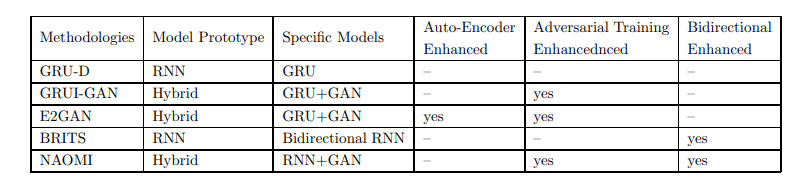
**Time Series Data Imputation:**

**Source-**

[**https://arxiv.org/pdf/2011.11347.pdf**](https://arxiv.org/pdf/2011.11347.pdf)

* **Deep Learning Based Methods for Time series Imputation-**

The main deep learning methods for time series imputation are GRU-D , GRUI-GAN , E2GAN , BRITS and NAOMI

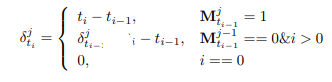
** **Definitions**

1. **Multivariable Time Series-** timestamp lists **T** = (t0, t1, ..., tn−1), and the time series **X** = {xt0 , xt1 , ..., xtn−1 } **T** as a sequence of n observations. The i-th observation of **X is xti** , which consists of d attributes {x 0 ti , x1 ti , ..., xd ti }.

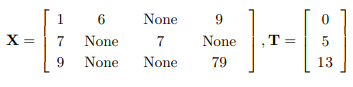
2.**Mask Matrix**-Mask Matrix M represents the missing values in X

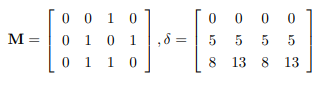


3**.Time Lag**- We use δ ∈ R n×d to record the time lag, and we calculate it in an iterative way as follows



**Example**

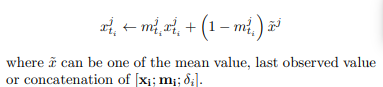


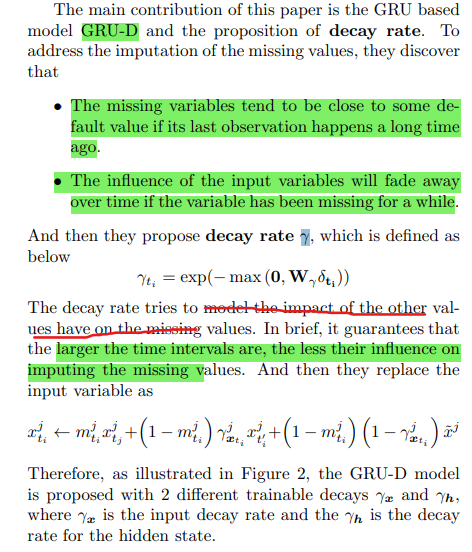


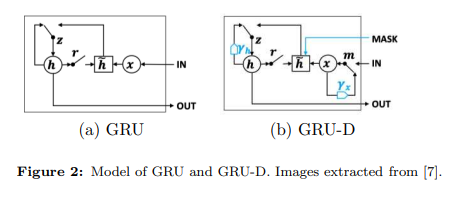
**Methods**

1. **GRU-D (Gated Recurrent Unit)-** GRUs are improved versions of standard recurrent neural networks.

input is replaced with a combination of the existing values and statistical values, element-wise multiplied with M and 1 − M respectively.





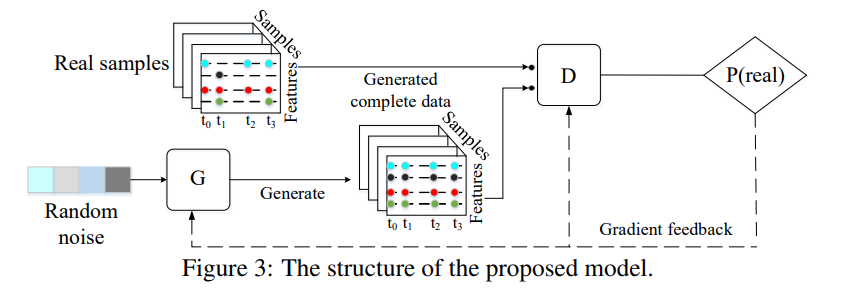


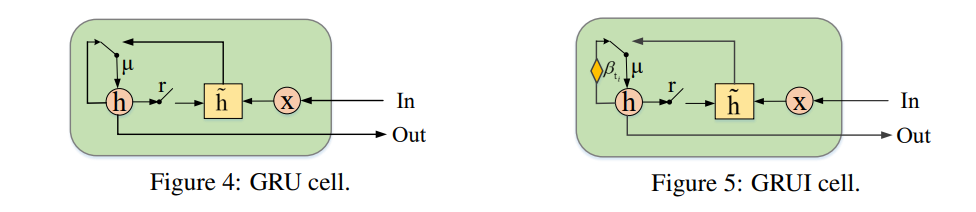
<https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be>

**2.GRUI-GAN-** Gated Recurrent Unit for data Imputation (GRUI) cell to model the temporal irregularity (not regularly spaced) of the incomplete time series.

. In the first phase, by adopting the GRUI in the discriminator and generator in GAN, the well trained adversarial model can learn the distribution of the whole dataset, the implicit relationships between observations and the temporal information of the dataset. In the second phase, we train the input “noise” of the generator of the GAN so that the generated time series is as close as possible to the original incomplete time series and the generated data’s probability of being real is the biggest.

**Method** [**-https://proceedings.neurips.cc/paper/2018/file/96b9bff013acedfb1d140579e2fbeb63-Paper.pdf**](https://proceedings.neurips.cc/paper/2018/file/96b9bff013acedfb1d140579e2fbeb63-Paper.pdf)

****

****

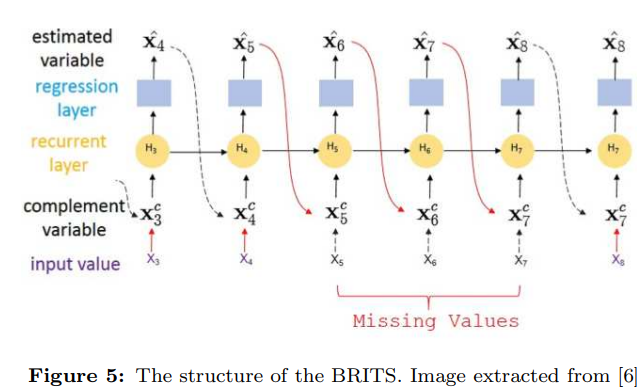
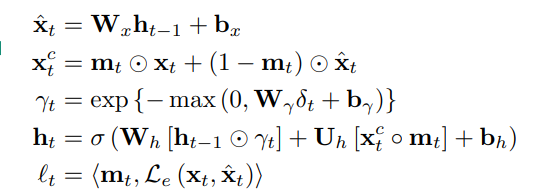
**Beta is the time decay vector** to control the influence of the past observation.

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.

**3.BRITS-**

BRITS train 2 models in forward direction and backward direction .In BRITS, time lags are still adopted to deal with irregular time series.

They propose a temporal decay factor γt = exp (−max (0,Wγδt + bγ)). 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:



**4. E^2GAN-**

[**https://www.ijcai.org/proceedings/2019/0429.pdf**](https://www.ijcai.org/proceedings/2019/0429.pdf)

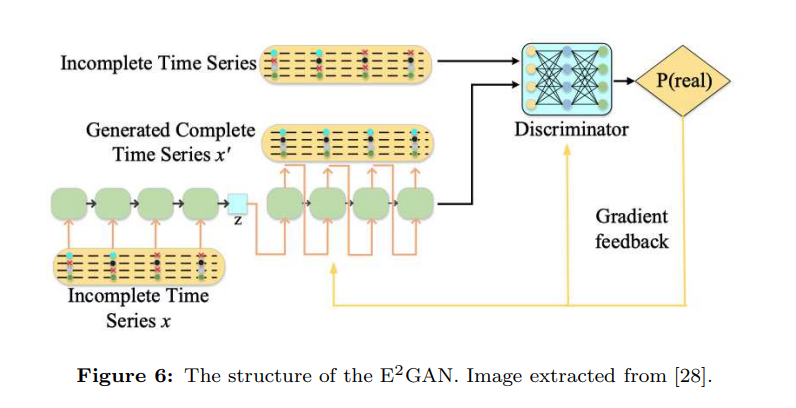
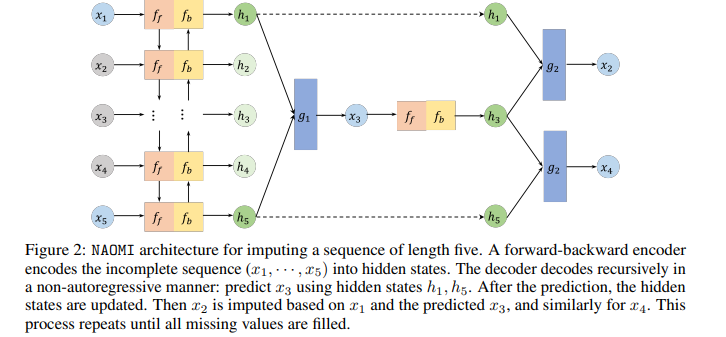


Figure :The structure of the proposed method. The generator is a denoising auto-encoder which is mainly composed of the GRUI cell. The discriminator is another encoder that produces truth probability.

input incomplete time series x is compressed into a low-dimensional vector z with the help of recurrent neural networks. Then we use this vector to reconstruct a complete time series x 0 to fool the discriminator. Meanwhile, we also force the x 0 as close as possible to x by using a squared error loss function. The discriminator of our method attempts to distinguish actual incomplete time series x and the fake but complete sample x 0 through the adoption of recursive neural network

GRUI (GRU for Imputation)is adopted to process the incomplete time series.

**5.NAOMI**

[**https://proceedin****gs.neurips.cc/paper\_files/paper/2019/file/50c1f44e426560f3f2cdcb3e19e39903-Paper.pdf**](https://proceedings.neurips.cc/paper_files/paper/2019/file/50c1f44e426560f3f2cdcb3e19e39903-Paper.pdf)

in the imputation tasks, future values and historical values are both observed,

**c**