Data augmentation for intelligent manufacturing with generative adversarial framework

Yanxia Wang, Kang Li, Shaojun Gan, Che Cameron, and Min Zheng

***Abstract*—The global economy is greatly shaped by the unprecedented booming of ICT and artificial intelligence technologies. Their applications in manufacturing has led to the advent of intelligent manufacturing and industry 4.0. Data has become a precious asset for modern industry. This paper first introduces an energy monitoring and data acquisition system namely the Point Energy Technology, which has been developed by the team and installed in several industrial partners, including a local bakery. The lack of data always exists due to various reasons, such as measurement or transmission errors at data collection and transmission stage, leading to the loss of varied length of data samples that are key for process monitoring and control. To solve this problem, we introduce a generative adversarial framework which is based on a game theory for data augmentation. This framework consists of two multi-layer perceptron networks, namely generator and discriminator. An improved framework with Q-net that extracts the latent variables from real data is also proposed, in which the Q-net shares the structure with discriminator except for the last layer. In addition, the two optimization methods, namely mini-batch gradient descent and adaptive moment estimation are adopted to tune the parameters. To evaluate the performance of these algorithms, energy consumption data collected from a bakery process is used in the experiment. The experimental results confirm that the latent generative adversarial framework with adaptive moment estimation could generate good quality data samples to compensate the random loss of samples in time series data.**

***Index Terms*—monitoring system, data augmentation, generative adversarial framework, optimization algorithm**

I. INTRODUCTION

Climate change and environment pollution caused by substantive consumption of fossil fuels have become a global concern in the past decade [1]. The UK government has introduced the Climate Change Act 2008 with commitment to reduce its greenhouse gas/carbon dioxide emissions (GHG) by 80% by 2050 (compared to the 1990's level) which is a huge leap from the current emissions level [2]. A new UK government plan proposed in June 2019 will further cut the Greenhouse gas emissions to almost zero by 2050. As a part of commitment to lower GHG emissions, the government has made the reduction of industrial energy consumption a priority [3]. Industrial manufacturing is one of the heavy energy consuming sectors, accounting for 16% of annual usage, and should consider the GHG reduction target as a priority [4]. The bakery industry, which produces staple foods such as breads, cakes and other pastries to meet people's daily dietary demand, consumes a lot of energy from gas and electricity [5]. The UK-based government organization, the Carbon Trust reports that the total energy consumption is 2,000 GWh per year for UK baking industries [6]. Therefore, it is of significant importance to research the status of energy consumption and seek opportunities to improve the energy efficiency of baking processes.

Data analysis for improvement of energy efficiency has created unprecedented opportunities for the industry given sufficient data [7]. While in realities, we need to achieve goals with limited datasets or incomplete datasets, which often occur due to various reasons, such as breakdown of the data sensing and transmission systems unrelated to the experimental process [8]. In these cases, loss of varied length of data leads to poor generalisation performance in data analysis. Techniques have been developed over the years to generate more data from the original dataset to compensate for the data loss [9]. Single Imputation requires a method of creating a predictive distributions of the missing values based on the observed data [10]. There are two generic approaches to generate distribution, explicit modelling which is based on a formal statistical model and implicit modelling which implies an underlying model. An obvious limitation of single imputation approaches is that the standard variance formulas used to generate artificial values may underestimate the systematic variance of estimates [11]. Various estimation methods for missing data could be implemented based on the likelihood function. The maximum likelihood (ML) approaches for treating missing data have been reported in the literature [12]. Given a statistical model of a distribution including an unknown parameter, the method of ML finds the value of parameter that maximize the likelihood function. Intuitively, the selected parameter makes the data most probable. The expectation maximization algorithm is an efficient iterative procedure to compute ML estimation [13], [14]. This algorithm has less conceptuality and computation complexity but may converge to a local minimum [15]. Multiple imputation is a general approach based on Bayesian estimation for incomplete data [16]. The method has three key steps: i) introduce random variation into the process and generate several different datasets; ii) perform analysis on each dataset; iii) combine the results into

Yanxia Wang, Kang Li and Shaojun Gan are with the School of Electronics and Electrical Engineering, University of Leeds, Leeds, UK (e-mail: wyxdsky@gmail.com, k.li1@leeds.ac.uk, cqugsj@gmail.com).
Che Cameron is with the School of Electronics, Electrical Engineering and Computer Science, Queends University of Belfast, UK, (ccameron07@qub.ac.uk).
Min Zheng is with the school of Mechatronics and Automation, Shanghai University, China, (zhengmin203@shu.edu.cn).

a single set of parameter estimates, standard errors and test statistics [17]. There is an indeterminacy in the results due to the random samples. Further, many parameters have to be tuned during the implementation, which makes the process quite complicated [18]. Generative adversarial network (GAN) can be used to represent the probability distributions of the observed data via an adversarial process [19]. It is a framework with two models, a generative model that captures the data distribution, and a discriminative model that estimates the probability that a sample comes from the real data rather than from the generator. This method has achieved great success in a number of applications, such as generating realistic images and stabilizing sequence learning methods [20]. By building a large invariance space, the GAN captures the cross-class transformations and move data points to follow the equivalent distribution, thus it can be applied to the data augmentation without known classes [21].

In this paper, a generative adversarial framework that could generate industrial data samples based on the game theory is discussed for data augmentation. We first introduce the point energy monitoring system developed by our research team (www.pointenergy.org), which is used for different industrial partners, including a local bakery company. The system collects voltage, current, power factor and frequency data from one of the core production lines. The generative adversarial framework adopts two multi-layer perceptron networks as the generator and the discriminator respectively. Considering the latent variables, a Q-net is introduced into this framework. The Q-net shares the layers with the discriminator except for the last layer. The mini-batch gradient descent and adaptive moment estimation are both used to optimise the parameters. To evaluate the performance of these algorithms, an industrial data set collected by our monitoring system is analysed. The remainder of this paper is organised as follows. The preliminary/related work is introduced in section 2. In section3, the generative adversarial framework is presented in details. The two optimisation algorithms are described in section 4. Section 5 discusses the experimental procedure and results. Finally, section 6 concludes this paper.

II. Energy Monitoring System

A desire for more detailed knowledge of power consumption, both in terms of increased sample rate and different granularity of use location has driven the development of the Point Energy monitoring system (www.pointenergy.org). Measurements of whole-factory power consumption as well as individual machine units are achieved using a combination of current transformers, interfaces to existing power meters and customised smart meters for different energy and power sources. The system has been field-tested in different industrial sectors including a local bakery company which is eager to know how much energy they use daily and more specifically, how much energy is consumed by each production line and even at the machine level.

The system has the Data Acquisition layer and the Data Analytics layer, bridged by an on-site base station, detailed in Figure 1. The data acquisition layer is composed of a series

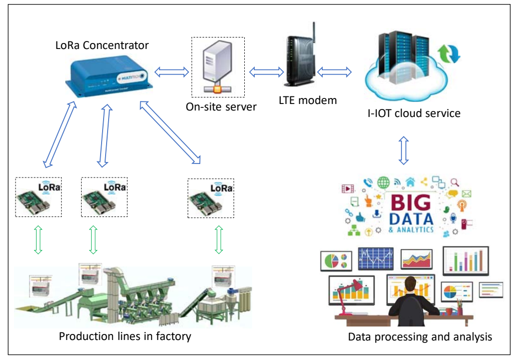

Fig. 1. The Energy monitoring system

of microcontroller nodes that form a wireless sensor network (WSN) using LoRa as the radio interconnect. These nodes are responsible for continuously measuring power, voltage, current and power factor for the three phase systems in the factory [22]. The management of the WSN is done by an on-site base station which is composed of a server, a LoRa concentrator and a 3G/4G internet connection. The LoRa concentrator bridges the local LoRa network to the server which then manages node inventory, concatenates and pre-processes data, and then sends the measurements on to various cloud services using the MQTT protocol. The data analysis layer is made up of a private server and I-IOT platforms which are used to run analytics on the dataset, as well as present a combination of real-time and historical energy usage information in a user-friendly format.

III. Generative Adversarial Framework

In this section, a specific generative adversarial framework of two competitive multi-layer networks is proposed, aiming to learn the data distribution from a set of samples implicitly and then generate new samples from the learned distribution.

A. Generative adversarial framework

Assume the real data distribution is $P_{real}(x)$. The generative adversarial framework is an approach for generating new samples based on random noise $P_{noise}(z)$. As depicted in figure 2, this framework consists of two multi-layer networks as generator and discriminator respectively. The generator takes the form as follows:

$$V = f(z) \tag{1}$$

where the function f is implemented by a multi-layer network; V is the vector data produced, which should match the real data in distribution; z is statistical noise having a continuous uniform distribution with lower and upper endpoints specified by 0 and 1. The generated and original data are input to the discriminator which outputs the probability of real samples.

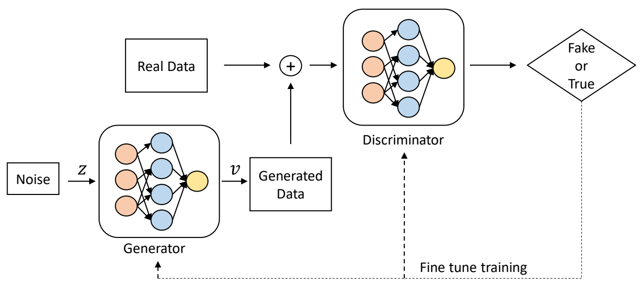

Fig. 2. The architecture of generative adversarial framework

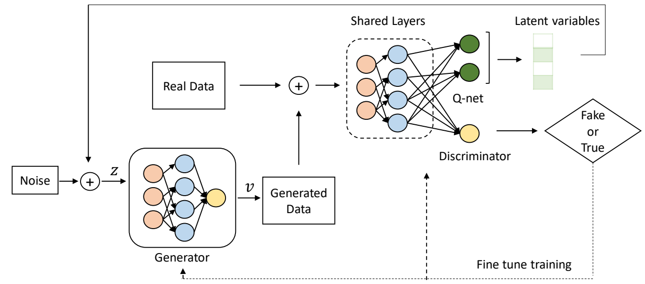

Fig. 3. The structure of latent generative adversarial framework

The two multi-layer perceptron networks are trained simultaneously. They learn and train their structures by minimizing the distribution discrepancy between the true data and generated data. The mixmax game could be expressed as follows:

$$\min_{G}\max_{D} V(D,G) = E[logD(x)] + E[1 - logG(z)] \quad (2)$$

During the training process, the generator could improve its ability to synthesize more realistic data. At the same time, the discriminator could improve its ability to distinguish the real from the generated data. Hence, this adversarial training could be thought as a kind of game theory.

B. Latent generative adversarial framework

As shown in figure 3, there are three multi-layer networks in the latent generative adversarial framework, namely the generator, discriminator and Q-net. Assume the discriminator has a L-layer architecture, then the Q-net shares the layers $\{1, 2, \cdots, L-1\}$ with the discriminator except for the last layer. In the last layer, the activation function is softmax for Q-net while sigmoid for discriminator. The latent variables are inferred automatically by Q-net and then are put into the generator, which could be regarded as prior information. Thus, in the latent generative adversarial framework, instead of using a single noise vector, the input of generator could be decomposed into two parts: i) noise vector z, which brings the variation to the new generations; ii) the latent variable b, which is related to the distribution features.

While training the latent generative adversarial framework, in addition to generator $G(z, b)$ and discriminator $D(x)$ which are similar to that of generative adversarial framework, Q-net $Q(b|x)$ is also trained with the mutual information. Considering the mutual information $I(b; G(z, b))$ between the latent variables b and the real samples x, the overall loss of latent GA framework is then given as follows [23]:

$$\begin{aligned}\min_{G}\max_{D} V_I(D,G) =& E[logD(x)] + E[1 - logG(z,b)] - \\ & \lambda I(b; G(z,b))\end{aligned} \quad (3)$$

where λ is an extra hyper parameter.

$I(b; G(z, b))$ represents the amount of information learned from $G(z, b)$ about latent variables b, which is difficult to optimize without knowing the distribution of $P(b|x)$. According to [24], the loss function could be rewritten as:

$$\begin{aligned}\min_{G}\max_{D} V_I(D,G) =& E[logD(x)] + E[1 - logG(z,b)] - \\ & \lambda L_I(G,Q)\end{aligned} \quad (4)$$

IV. Optimization algorithm

A. Mini-batch gradient descent

Mini-batch gradient decent combines the advantages of both the batch gradient descent [25] and the stochastic gradient descent [26], performing an update for every mini-batch as follows:

$$\theta_j = \theta_{j-1} - \eta \cdot \triangledown_{\theta_j} \cdot J(\theta_j) \quad (5)$$

where θ is the parameter to be updated; η is the learning rate; $\triangledown_{\theta_j} \cdot J(\theta_j)$ represents the gradient of current mini-batch (i.e. the vector of partial derivatives of los function J).

Assume there are m samples in the data set and n samples in every mini-batch, thus there are m/n mini-batches. Let ω_k represent the k-th mini-batch and Ω_j being the collection of all mini-batches in j-th iteration (Num is the maximum number of iteration), then:

$$\Omega_j = \{\omega_k : k = 1, 2, \cdots, m/n\} \quad (6)$$

The flow of mini-batch gradient descent algorithm could be described as follows:

Algorithm: mini-batch gradient descent
For $j = 1 : Num$
While each ω_k in Ω_j do:
Update parameters with equation (5)
End while
End for

Mini-batch gradient descent is a typical algorithm to optimise neural networks. It could reduce the variance of parameter updates to achieve more stable convergence. The mini-batch is very efficient to compute the gradient compared with the batch gradient descent. While it also faces a challenge that it is difficult to choose a proper learning rate. Additionally, adjusting learning rate to adapt different parameters' features is another barrier.

B. Adaptive moment estimation

The adaptive moment estimation is an advanced method that computes adaptive learning rate for different parameter. This algorithm updates exponential moving averages of gradient m_j (also named the first moment estimation) and squared gradients v_j (also named the second moment estimation) respectively as follows:

$$f(x) = \begin{cases} m_j = \beta_1 m_{j-1} + (1-\beta_1) \cdot g_j \\ v_j = \beta_2 v_{j-1} + (1-\beta_2) \cdot g_j^2 \\ g_j = \nabla_{\theta_j} \cdot J(\theta_j) \end{cases} \tag{7}$$

where β_1 and β_2 both belong to [0,1), controlling the exponential decay of these moving averages; m_j and v_j are initialized with zeros. When the decay rates are small (i.e. βs close to 1), these moving averages are biased towards zeros. To counteract the biases, the bias-corrected first and second moment estimates could be computed:

$$f(x) = \begin{cases} \hat{m_j} = m_j/(1-\beta_1^j) \\ \hat{v_j} = v_j/(1-\beta_2^j) \end{cases} \tag{8}$$

Hence, the parameters could be updated as follows:

$$\theta_{j+1} = \theta_j - \frac{\eta}{\sqrt{\hat{v_j}} + \varepsilon} \cdot \hat{m_j} \tag{9}$$

where η represents the learning rate; ε is a very small number (e.g. 10^{-8}) to avoid any division by zero in the implementation.

Similarly, let ω_k represent the k-th mini-batch and Ω_j is the collection of all mini-batches in j-th iteration. Then based on the equation (6), the flow of adaptive moment estimation method could be described as follows:

Algorithm: adaptive moment estimation
For $j = 1 : Num$
While each ω_k in Ω_j do:
Calculate the gradient of the k-th mini-batch g_j
Update m_j and v_j with equation (7)
Compute $\hat{m_j}$ and $\hat{v_j}$ with equation (8)
Update parameter θ with equation (9)
End while
End for

V. Experiment

A. Industrial data set

In the factory, a large quantity of electricity, about 30-35% is consumed in the baking process. This paper documents the initial energy consumption data set of the baking process working with three-phase 415V AC power over a randomly selected period. The following features are monitored at a ten-minute interval across all three phases: voltage, current, power, power factor, frequency and temperature. Each of these dimensions in the collected data represents different signals with different scales. Thus, normalisation is required so that all the inputs are at a comparable range [27]. In the experiment, the data set is collected from baking process covering ten-day time, beginning from the Thursday 10th May. 2018. As is shown in figure 4, the normalised data set consists of 2246 samples, and the number of dimension is 14 for each sample.

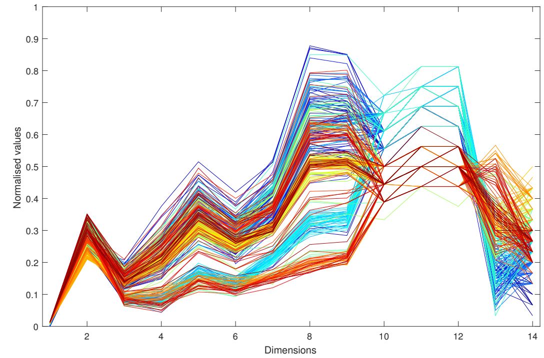

Fig. 4. The samples of original baking data

B. Experimental setup

There are four algorithms conducted in the experiment, namely the generative adversarial framework combined with mini-batch gradient descent (GAF-MNGD), the generative adversarial framework combined with adaptive moment estimation (GAF-Adam), the latent generative adversarial framework combined with mini-batch gradient descent (LGAF-MNGD) and the generative adversarial framework combined with adaptive moment estimation (LGAF-Adam). To evaluate the performance of these four approaches, the parameters are set as follows.

For GAF-MNGD and GAF-Adam, the architectures of generator and discriminator are same, both with three layers. The number of nodes in each layer of generator is [100, 512, 14], while being [14, 512, 1] for discriminator. The activation functions in hidden and last layers are sigmoid function and Relu function respectively. For LGAF-MNGD and LGAF-Adam, the generator and discriminator also both have a three-layer structure. The number of nodes in each layer of generator is [100+N_l, 512, 14] (where N_l means the number of latent variables obtained by Q-net), while being [14, 512, 1] for discriminator. The activation functions in the second and third layers are sigmoid function and Rule function respectively. The Q-net shares the same structure with discriminator except the last layer. The number of nodes in last layer is N_l and set to be five. For mini-batch gradient descent and adaptive moment estimation, there are 100 samples in each mini-batch. The number of iteration is 100 and the learning rate is 0.001. In the experiment, 100 samples would be randomly selected from the original data set as the real data. Monte Carlo simulation is a computational technique based on constructing a random process for a problem, which could understand the impact of risk and uncertainty due to the randomness [28]. Therefore, the experiment will be repeated for 200 times using the Monte Carlo method, and then one of them will be picked out as the final result which is closest to the average value of outputs.

The expert assessment is calculated as the total root mean squared error (RMSE), an effective measure of the deviations in distances between the real and produced point coordinates. The formula for RMSE is shown as follows:

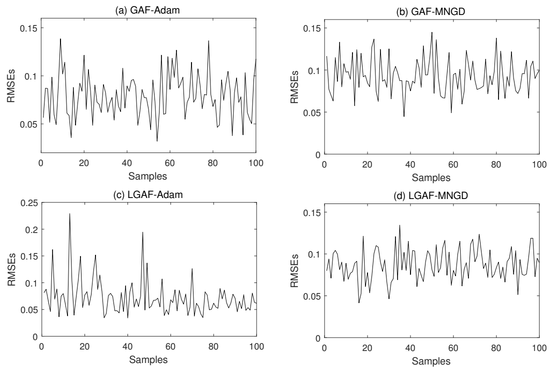

Fig. 5. The experimental result of state 1

$$\text{RMSE} = \sqrt{\frac{1}{R}\sum_{i=1}^{R}(y_i - \hat{y}_i)^2} \tag{10}$$

where given R dimensions to the output, y_i is the i-th dimension of the real data and $\hat{y}_i$ is the i-th dimension of the generated data.

C. Results and discussion

In view of the reality of data collection during industrial application, we define random loss of data samples (i.e. non-time series missing data) as state 1. To simulate the actual situations, the real data set contains 100 samples which are selected completely randomly.

According to the Monte Carlo method, the result of state 1 is obtained from the 97th Monte Carlo simulation and illustrated in figure 5. The four subplots (a)-(d) display the RMSEs of 100 samples associated with GAF-Adam, GAF-MBGD, LGAF-Adam and LGAF-MBGD respectively. The mean values of RMSEs of these four methods are 0.080, 0.093, 0.072, and 0.087 respectively. The average RMSE of LGAF-Adam is 10.7%, 29.2% and 20.8% smaller than that of the other three algorithms. Therefore, the Latent Generative Adversarial Framework with Adam algorithm is the best for data augmentation.

Furthermore, we use the generated data samples produced by LGAF-Adam to fill in the missing data for State 1, and then calculate the energy consumption based on the produced data, which is compared with the real energy usage during the chosen ten days. In figure 6, the hourly energy consumptions during the missing data period based on the generated data and real data are illustrated for State 1. It is obvious that although there is a minor gap between real data and generated data with regard to the hourly energy usage for the baking process, the maximum difference is less than 1kwh hourly, which is acceptable in local bakery.

VI. Conclusion

In this paper, the energy monitoring system developed by Point Energy Technology (www.pointenergy.org) is introduced, which monitors and records the operating conditions

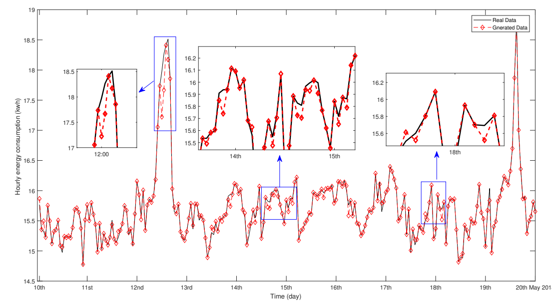

Fig. 6. The comparison between real and generated data for state 1

of industrial machinery. During the data collection procedure, we found there is always missing data due to various reasons. Hence, a Generative Adversarial Framework based on Game Theory is proposed for data augmentation. There are two competitive multi-layer perceptron networks, namely the Generator and the Discriminator. They are not updated directly with data samples, but rather use back propagation to obtain gradients for parameter learning. To extract the distribution features of the industrial data, the framework is upgraded with a Q-net which shares its structure with the Discriminator except for the last layer. The sigmoid cross entropy is computed as the cost function. In addition, Mini-Batch Gradient Descent and Adaptive Moment Estimation are adopted to optimise the parameters. To compare the effectiveness of the frameworks and optimisation methods, baking data over a randomly selected ten-day period in May 2018 is used in the experiment. The type of missing data, missing completely at random, is discussed in details.

The experimental results are summarised as follows: i) the effectiveness of the four algorithms yields no significant difference for data augmentation on the missing data. ii) The Latent Generative Adversarial Framework with Adaptive Moment Estimation optimisation has better performance than the other three algorithms. iii) With the proposed data augmentation technique, we could conduct data analysis sufficiently reliable rigorous to make professional recommendations for industrial partners.

Acknowledgment

This research is financially supported by the UK Engineering and Physical Sciences Research Council (EPSRC) under grant EP/P004636/1 'Optimising Energy Management in Industry - OPTEMIN'.

References

[1] S. Meyers, B. Schmitt, M. Chester-Jones, and B. Sturm, "Energy efficiency, carbon emissions, and measures towards their improvement in the food and beverage sector for six european countries," *Energy*, vol. 104, pp. 266–283, 2016.

[2] C. H. Dyer, G. P. Hammond, C. I. Jones, and R. C. McKenna, "Enabling technologies for industrial energy demand management," *Energy Policy*, vol. 36, no. 12, pp. 4434–4443, 2008.

[3] P. W. Griffin, G. P. Hammond, and J. B. Norman, "Industrial energy use and carbon emissions reduction: a uk perspective," *Wiley Interdisciplinary Reviews: Energy and Environment*, vol. 5, no. 6, pp. 684–714, 2016.

[4] P. Agnese, M. Rizzo, and G. A. Vento, "Smes finance and bankruptcies: The role of credit guarantee schemes in the uk," *Journal of Applied Finance and Banking*, vol. 8, no. 3, pp. 1–16, 2018.

[5] O. Ekechukwu, A. Madu, S. Nwanya, and J. Agunwamba, "Optimization of energy and manpower requirements in nigerian bakeries," *Energy conversion and management*, vol. 52, no. 1, pp. 564–568, 2011.

[6] T. J. Foxon, *Introducing Innovation for a Low-carbon Future: Drivers, Barriers and Policies: a Report for the Carbon Trust*. Carbon Trust, 2003.

[7] Z. Ma, J. Xie, H. Li, Q. Sun, Z. Si, J. Zhang, and J. Guo, "The role of data analysis in the development of intelligent energy networks," *IEEE Network*, vol. 31, no. 5, pp. 88–95, 2017.

[8] R. J. Little and D. B. Rubin, *Statistical analysis with missing data*, vol. 333. John Wiley & Sons, 2014.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[10] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 1, pp. 97–107, 2014.

[11] A. R. T. Donders, G. J. Van Der Heijden, T. Stijnen, and K. G. Moons, "A gentle introduction to imputation of missing values," *Journal of clinical epidemiology*, vol. 59, no. 10, pp. 1087–1091, 2006.

[12] C. K. Enders and D. L. Bandalos, "The relative performance of full information maximum likelihood estimation for missing data in structural equation models," *Structural equation modeling*, vol. 8, no. 3, pp. 430–457, 2001.

[13] C. K. Enders, "A primer on maximum likelihood algorithms available for use with missing data," *Structural Equation Modeling*, vol. 8, no. 1, pp. 128–141, 2001.

[14] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image segmentation using expectation-maximization and its application to image querying," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1026–1038, 2002.

[15] X. Sheng and Y.-H. Hu, "Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 53, no. 1, pp. 44–53, 2005.

[16] P. Royston *et al.*, "Multiple imputation of missing values," *Stata journal*, vol. 4, no. 3, pp. 227–41, 2004.

[17] P. Royston *et al.*, "Multiple imputation of missing values: update of ice," *Stata Journal*, vol. 5, no. 4, p. 527, 2005.

[18] P. D. Allison, "Handling missing data by maximum likelihood," in *SAS global forum*, vol. 2012. Statistical Horizons, Havenford, PA, 2012.

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[20] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems*, pp. 5767–5777, 2017.

[21] A. Antoniou, A. Storkey, and H. Edwards, "Data augmentation generative adversarial networks," *arXiv preprint arXiv:1711.04340*, 2017.

[22] J. Kline and C. Kline, "Power modeling for an industrial installation," in *Cement Industry Technical Conference, 2017 IEEE-IAS/PCA*, pp. 1–10. IEEE, 2017.

[23] A. Spurr, E. Aksan, and O. Hilliges, "Guiding infogan with semi-supervision," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 119–134. Springer, 2017.

[24] D. B. F. Agakov, "The im algorithm: a variational approach to information maximization," *Advances in Neural Information Processing Systems*, vol. 16, p. 201, 2004.

[25] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[26] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, pp. 177–186. Springer, 2010.

[27] Y. Liu, J. Cao, B. Li, and J. Lu, "Normalization and solvability of dynamic-algebraic boolean networks," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 7, 2018.

[28] R. Mesleh, S. S. Ikki, and H. M. Aggoune, "Quadrature spatial modulation," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 6, pp. 2738–2742, 2015.