DR-TiST: Disentangled Representation for Time Series Translation Across Application Domains

Hiba Arnout
Siemens AG
Technical University of Munich
Munich, Germany
hiba.arnout@siemens.com

Johanna Bronner
Siemens AG
Munich, Germany
johanna.bronner@siemens.com

Johannes Kehrer
Siemens AG
Munich, Germany
kehrer.johannes@siemens.com

Thomas Runkler
Siemens AG
Technical University of Munich
Munich, Germany
thomas.runkler@siemens.com

***Abstract*—In the last few years, a huge progress has been made to achieve image-to-image translation by mapping images from a source domain to a target domain. We exploit the recent progress made in this field to tackle another issue, namely time series translation. This work targets time series translation i.e. maps time series data from a source domain to a target domain. We present our new algorithm DR-TiST, a modified version of DRIT [1], that enables time series translation. We apply DR-TiST to a real word use case where we transfer the time series behavior of a ventilation system to the environmental conditions of a different ventilation system and introduce new evaluation metrics to evaluate its performance. The performance of DR-TiST is compared to CycleGAN-VC [14], a special form of an image-to-image translation algorithm used for voice conversion. We demonstrate that the time series generated by DR-TiST are more realistic than the ones generated by CycleGAN-VC.**

***Index Terms*—Time Series, Deep Learning, Image-to-Image-Translation, DRIT, Generative Adversarial Networks**

I. Introduction

Rooms are equipped with two different ventilation systems from different manufacturers. Each ventilation system starts cooling whenever the room temperature raises above a temperature threshold t_{max} and stops when the temperature drops below a minimal temperature t_{min}. The first room is small and consequently cools down and warms up faster. Hence, the corresponding ventilation system is frequently turned on and off. The second room is bigger. In this case, the ventilation system needs more time to cool down the temperature and the room will not warm up rapidly. The ventilation system of this room will represent slower on/off cycles. The performance of these ventilation systems is only comparable if they are running in the same environment and under the same conditions.

Now assume that we want to use time series data from normal operation to build a condition monitoring system. A straightforward approach is to build individual monitoring systems for each specific machine and environment. It is however often desirable to build only one generalized condition monitoring system that can be applied to any machine in any environment. This approach often leads to better model quality because the model can be based on larger amounts of data from different machines and different operating conditions; and it avoids the effort to manage many individualized models (deployment, maintenance, etc.). For building such generalized models, however, it is necessary to translate time series data between different machines and different environments. For this purpose, this paper proposes a new method called DR-TiST, and compares its performance with CycleGAN-VC.

Recently, the field of image-to-image translation is making rapid progress. In fact, a lot of work has been proposed that aims to determine a mapping function between two visual domains. Some of them [6] require paired-data during the training process to transform a specific image domain to a target domain. Others achieve image-to-image translation without coupled data. By way of example, CycleGAN [4] involves two discriminators and two generators to perform image-to-image translation and relies on a cycle consistency constraint to ensure cyclic reconstruction of the images. Latest works such as UNIT [3] propose a more complex structure that involves coupled GANs. Furthermore, DRIT proposes a new structure that aims to capture the attribute and content specific features of the image.

In this paper we exploit the recent progress in image-to-image translation to perform time series translation i.e. to map time series from one domain to another domain for example for different machines in different environments. Our work contributes as follows:

- We introduce a new method to translate time series from an initial domain A to a target domain B by adapting an existing image-to-image translation algorithm, namely DRIT [1], to time series data.
- Test DR-TiST on a real use case using simulated data where we transfer the behavior of a first machine to the environmental domain of a second machine.
- Evaluate the performance of our algorithm using visual inspection and two new evaluation methods designed for the introduced use case and time series data.
- Compare the performance of our algorithm to CycleGAN-VC [14] a modified version of CycleGAN [4] designed for voice conversion and consequently able to process sequential data.

II. Related Work

Generative Adversarial Networks (GANs) In the last few years, GANs have shown remarkable results in generating

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

realistic images [8], [12]. GANs involve two neural networks, the generator and the discriminator which try to generate new data so that their distribution is the same as the one of the real data. Some authors exploited this technique to generate sequential data such as in C-RNN-GAN [9], RGAN [10] or SeqGAN [18]. Instead of reproducing the distribution of the real data, our work targets another problem, namely time series translation. Our main goal is to transfer a time series of a given machine in a given environment to the domain of a different machine in a different environment.

Image-to-Image Translation Several works have been presented to perform image-to-image-translation with the purpose of learning the mapping function between two different visual domains. Isola et al. [7] used in their method, Pix2pix, conditional GAN to translate images to a target domain. In this case, paired-data are required during the training process. This technique was later extended by Wang et al. in CycleGAN [4] that relies on a cycle consistency loss to deal with unpaired images. Liu et al. presented unsupervised image-to-image translation networks (UNIT) [3] which are based on coupled GANs that map to a common latent space used as a cycle-consistency constraint. Huang et al. introduced a multimodal version of UNIT, MUNIT [17], that enables a diverse output generation for a source image by decomposing the latent space into a domain-invariant content space and a style space. In contrast to MUNIT [17], BicycleGAN [6] requires paired data during the training process. This technique consists of a Conditional Variational AutoEncoder GAN (cVAE-GAN) and a Conditional Latent Regressor GAN (cLR-GAN) to guarantee a one-to-one mapping between each input image and the corresponding output. DiscoGAN [5] learns to distinguish between the domain and the style of the image by observing the relationships between various visual fields. To enable a multimodal image-to-image translation, DRIT [1], [2] tries to capture the relationship between two different visual domains by dividing each image into a content and attribute. In fact, a disentangled representation is applied to the latent space representation. The content consists of the common information between both visual domains whereas the attribute space preserves the domain-specific information. DRIT [1] involves two content encoders, two attribute encoders, two generators, two domain discriminators and a content discriminator. An input image from a domain $\mathcal{X}$ is projected by the attribute encoder and by the content encoder into the domain specific content space. The content and attribute vectors determined by the encoders are later used by the generator to translate images in the domain $\mathcal{X}$. A domain discriminator learns to distinguish between the real images from domain $\mathcal{X}$ and the images generated by the generator. Finally, a content discriminator is involved to differentiate between the learned content representations of both domains. We refer to [1] for a detailed description of the algorithm. These approaches were mainly designed for images and can not be directly applied to time series data. This is mainly due to the sequential structure of this type of data. To this end, we introduce DR-TiST a modified version of DRIT [1]. CycleGAN-VC [14] proposes a modified version of CycleGAN that combines gated CNNs with an identity-mapping loss to focus on non-parallel voice-conversion. The previously discussed translation and generation techniques are summarized in Table I.

	Image Generation	Time Series Generation
$A \rightarrow A$	GAN [8] DCGAN [12]	C-RNN-GAN [9] SeqGAN [18]
$A \rightarrow B$ and $B \rightarrow A$	CycleGAN [4] UNIT [3] MUNIT [17] DRIT [1], [2]	CycleGAN-VC [14] DR-TiST

TABLE I: Comparison of different GAN generation methods and translation techniques for images and time-oriented data

III. Algorithm Description

In this paper, we propose a new algorithm called DR-TiST that achieves time series translation thanks to the disentangled representation proposed by DRIT. Each time series is divided into a functional behavior highlighting the properties of the time series and a context describing the environmental setup. For example, a time series depicting the behavior of an engine over time can be divided into a functional behavior depicting its behavior in the on/off states and its operating mode i.e. times at which it is off or on. Based on the extracted functional behavior and operating mode, it is possible to translate the functional behavior of this time series to other operating modes or to simulate the behavior of other engines in its operating mode. The encoders, generators and discriminators of DRIT were originally designed with a 2-dimensional convolutional neural network (CNN) in order to process images. This neural network structure is not suitable for time series data due to their sequential structure. To adapt DRIT to time series data we make two major modifications: apply the gated CNN structure and replace the two-dimensional CNN with the one-dimensional CNN that take the temporal relationship between the data points into consideration. Gated temporal convolutions were originally introduced by Dauphin et al. [13] and achieve state-of-the-art results in language- and speech modeling. In contrast to recurrent networks where the output of a layer is computed with the recurrent function $h_i = f\left(h_{i-1}, w_{i-1}\right)$, gated CNN can be employed in a parallel manner. This allows a faster computation. Gated CNN utilizes a Gated Linear Units (GLUs) as an activation function. The output of a layer $l+1$, H_{l+1}, is computed based on the output of the layer H_l and the model parameters W, V, b and c as follows:

$$H_{l+1} = \left(H_l \cdot W + b\right) \otimes \sigma\left(H_l \cdot V + c\right). \tag{1}$$

where $\otimes$ is the element-wise product and σ is the sigmoid function. Fig. 1 shows the gated CNN structure.

The gated structure is tested on the different components of DRIT. Tests show that the best results are obtained when it is only applied on the generator. Thus, the proposed DR-TiST structure applies gated CNN on the residual blocks of the generator. As a last modification, we integrated instance normalization [19], a well-known method for improving the quality of images during the generation process and replace the deconvolution blocks used the generator with a pixel

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

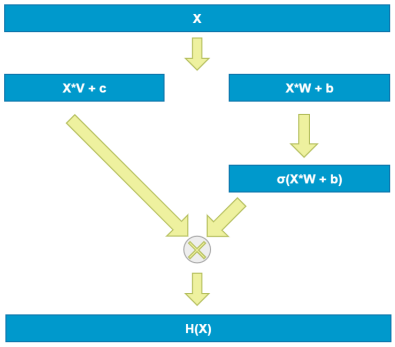

Fig. 1: Illustration of the gated CNN structure. The output of the layer $H(X)$ for an input X is computed by multiplying element-wise $X \cdot V + c$ and $\sigma(X \cdot W + b)$ where $X \cdot V + c$ and $X \cdot W + b$ are the resulting vectors of the convolution on the input X and the sigmoid function is applied on $X \cdot W + b$.

shuffler [20]. The resulting residual blocks used in DR-TiST are illustrated in Fig. 2. The discriminators and the encoders remain unchanged.

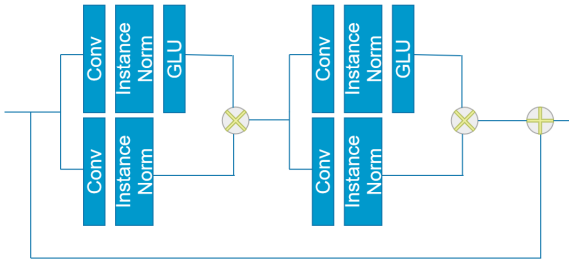

Fig. 2: Illustration of the residual block used in the generator of DR-TiST. Conv denotes a convolution, instance norm denotes instance normalization and GLU denotes the activation function.

IV. Use Case

A. Use Case Description

In this use case, we are taking into consideration two ventilation systems, system 1 and system 2, placed in two different rooms which constitute the environmental setup. The first room, room A, is huge and has a cold environment contrary to the second one, room B, which is warmer and smaller. The variation of the temperature is different from one room to another. In fact, due to the size of room A its temperature takes time to raise and fall. Hence, its ventilation system will not be frequently switching on and off. This leads to slow on/off cycles i.e. times at which the machine switches its operating mode. In room B on the other hand, the on/off cycles are changing faster because the temperature raises and falls quickly. Generally condition monitoring systems are designed for individual ventilation systems and operating conditions leading to a very specific solution for exactly one setup. When AI based condition monitoring systems are designed using only the available data of both machines in their particular environment, very single models will arise. There is a strong interest to design condition monitoring systems, that generalize, hence can be applied to any machine in any environment. In reality the amount of data is hard to collect and it is only feasible if they are available. We work around this problem by generating a more heterogeneous training data set, consisting of "mixed" time series data, where system 1 controls room B and system 2 room A, which are a non-realistic conditions.

B. Use Case Simulation

To the best of our knowledge, this is the first study with the goal of aligning time series. Thus, the evaluation of DR-TiST on a real sensor dataset where the behavior may be unexpected will be complicated. To be able to exactly evaluate the performance of DR-TiST we design a synthetic dataset where the success or failure of translating the behavior of one system into the operating mode of another system is obvious and quantifiable. To this end, we simulate the behavior of two different engines of two different ventilation systems, engine 1 and engine 2. The engines are turned on and off in different time slots and are operating differently i.e. engine 1 is frequently turned on and off while engine 2 shows a more stable behavior.

Following discussions with domain experts we decided to model the machines' behavior using the standard exponential behavior. Hence, the machine's behavior in the on and off states is computed as follows:

$$y_{on}(t) = MRS \cdot \left(1 - \exp\left(-\frac{t - \tau_1}{\tau}\right)\right) + n \tag{2}$$

$$y_{off}(t) = \exp\left(-\frac{t - \tau_0}{\tau}\right) + n \tag{3}$$

where τ and MRS characterize the engines and $n \sim N(0, 0.01)$ is a Gaussian noise.

Engine 1, driving the ventilation system in room A, is running with a low Maximal Rotational Speed (MRS) equal to 1. Whereas a more efficient engine 2 is placed in the hotter environment, room B. Its MRS is equal to 1.5. Moreover, we assume that engine 1 is older and therefore slower in reaching the MRS value or the minimal value when it is started or stopped. Thus, it has a larger value of $\tau = 5$, compared to the newer engine 2 with a value of $\tau = 2$. Table II summarizes the characteristics of engine 1 and 2 used in our experiments. The initially collected data depicts the properties of engine 1 and 2 in the conditions of room A i.e. slow on/off cycles and room B i.e. fast on/off cycles respectively. We consider the task of generating the functional behavior of engine 1, characterized by $MRS = 1$ and $\tau = 5$, in the operating mode of engine 2 characterized by fast on/off cycles and vice versa. Fig. 4 illustrates the expected time series translation scheme.

C. Experimental Setup

Given initial data depicting the properties and the operating mode of engine 1 and engine 2, our main goal is to make sure that DR-TiST is able to generate new data that depicts

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

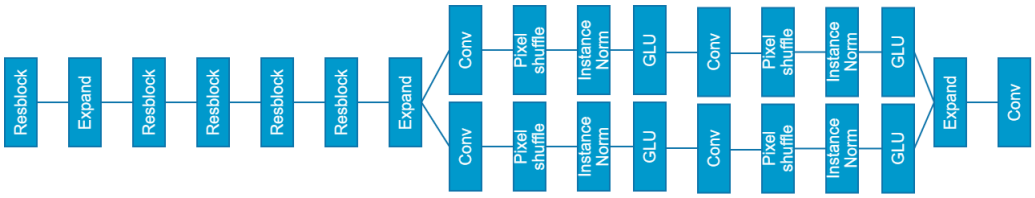

Fig. 3: Illustration of the generator's architecture used in DR-TiST.

	Engine 1	Engine 2
$MRS[Hz]$	1	1.5
$\tau[min]$	5	2

TABLE II: Characteristics of machine 1 and machine 2 in the on and off states

	μ_1	μ_2
$Test\ 1$	20	20
$Test\ 2$	30	40
$Test\ 3$	35	45
$Test\ 4$	40	50

TABLE III: Mean of the Bernoulli distribution of the on/off times in the different experiments

the behavior of engine 1 with the on/off cycles of engine 2 and vice versa. To this end, we conduct four different tests where we change the distribution of the on/off times for each engine. The mean of the Bernoulli distribution is varied for the different tests. Table III illustrates the mean of the on/off times for the different tests where μ_1 and μ_2 denote the mean of the Bernoulli distribution for the on/off times of engine 1 and engine 2 respectively. For test 1, the same Bernoulli distribution is used for engine 1 and engine 2 i.e. $\mu_1 = \mu_2 = 20$. In contrast to test 1, μ_1 and μ_2 are in the other tests different. The initial data are computed based on the equations 2 and 3, the characteristics of the machines presented in Table II and the on/off cycles sampled from the Bernoulli distribution. In the training phase we use 1000 time series for each engine with 508 data points. Additionally, in order to assess the impact of the amount of data on the performance of the framework we repeat test 2 with different number of data points per time series, namely 208 and 416. We evaluate the performance of DR-TiST by generating 100 time series in the test phase. We expect that DR-TiST is able to generate new time series of engine 1 in the time domain of engine 2 and vice versa. Finally, we compare the performance of DR-TiST to CycleGAN-VC, a modified version of CycleGAN designed for voice conversion purposes. As CycleGAN-VC is computationally expensive, we reduce the length of the data points per time series to 208 and rerun test 2 and 3.

D. Evaluation Methods

In our experiments we evaluate the performance of DR-TiST with three different methods: visual inspection and two additional metrics that rely on ground truth data.

Visual inspection The evaluation of GANs and image-to-image translation techniques is still an open research problem. Current evaluation methods involve humans and rely on a visual inspection to assess the quality of the generated data [21]. Various work [7], [15], [16] rely on user studies to evaluate the realism of the generated images. Inspired by previous works as an initial check, we perform a visual investigation to assess how realistic a time series appears to a domain expert and we visually inspect the quality of the generated time series. Beyond visual inspection, we evaluate the performance of DR-TiST with two additional methods that assess the quality of the generated time series by comparing them to ground truth data.

Error in on/off time prediction Our main goal is to simulate the behavior of a first machine with the on/off times of a second machine i.e. we want to transfer the behavior of a machine to the time domain of a second one. During the simulation of the behavior of each engine we save its on/off times, these will correspond to the expected on/off time for the other engine. The expected behavior for each machine Y is then computed based on its expected on/off times and on the equations 2 and 3. The root mean square error (RMSE) between Y and the time series generated by DR-TiST, namely $\widehat{Y}$ can be calculated as follows:

$$RMSE = \sqrt{\frac{1}{N} \cdot \sum_{i=1}^{N} \left(Y_i - \widehat{Y}_i\right)^2} \tag{4}$$

where N denotes the number of data points per time series.

Moreover, we compute the mean of the point-wise difference between the expected and obtained time series Y and $\widehat{Y}$:

$$D = \frac{\sum_{i=1}^{N} \mid Y_i - \widehat{Y}_i \mid}{\sum_{i=1}^{N} \mid Y_i \mid} \tag{5}$$

V. Experiments

	Engine 2 Generated		Engine 1 Generated	
	$D2_{cyc}$	$D2_{DR}$	$D1_{cyc}$	$D2_{DR}$
$Test\ 2$	0.41	0.36	0.31	0.2
$Test\ 3$	1.77	0.43	1.66	0.35

TABLE IV: D values for generating time series of engine 1 and engine 2 by CycleGAN-VC and DR-TiST for test 2,3 and 4. D_{cyc} and D_{DR} denote the values of D for the time series of CycleGAN-VC and DR-TiST respectively.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

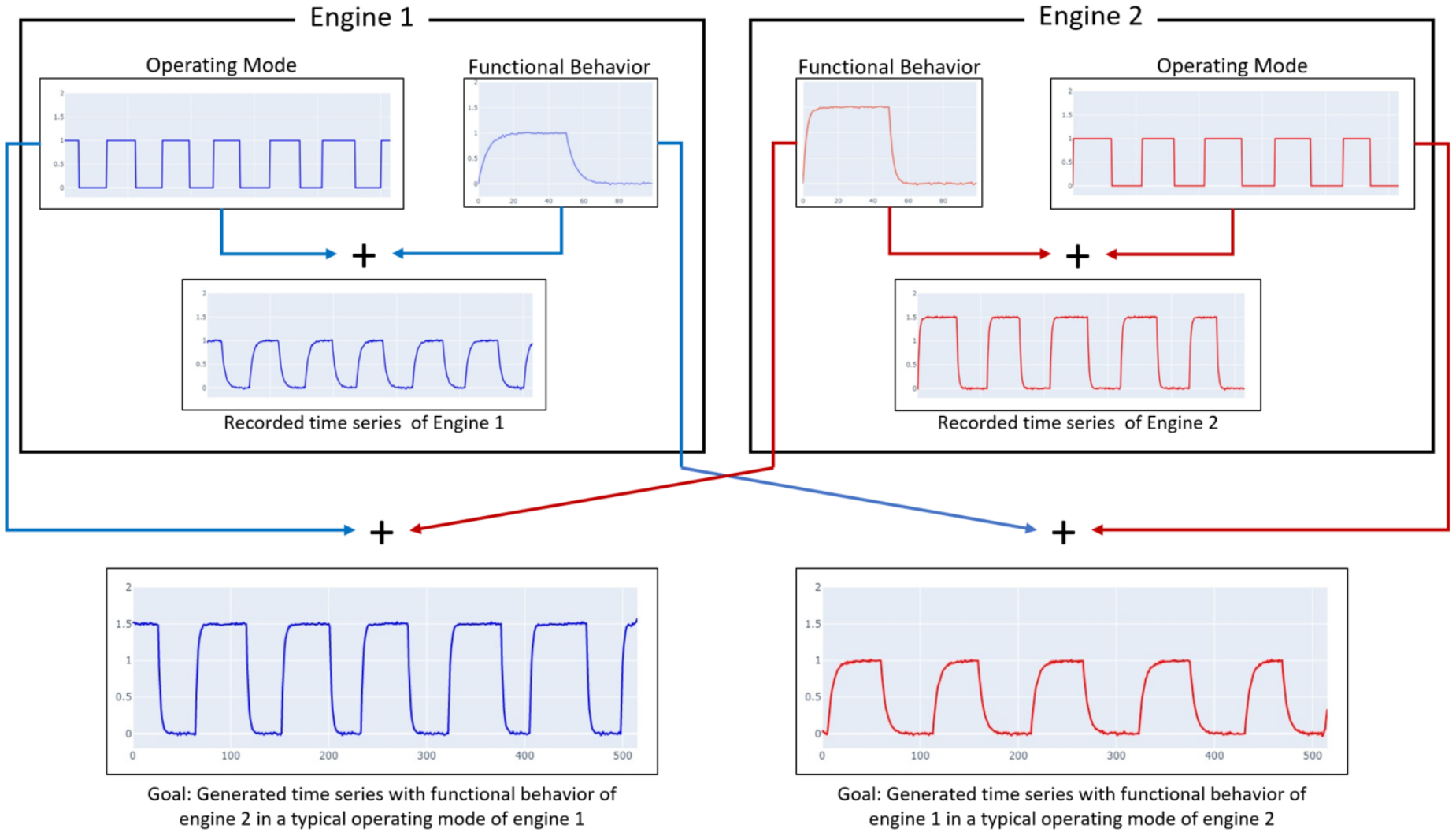

Fig. 4: **Time series translation** The proposed algorithm divides a given time series into operating mode and functional behavior. This learned representation allows to map the functional behavior of engine 1 into the operating mode of engine 2 and vice versa and hence help to simulate the behavior of different engines in different environmental setups.

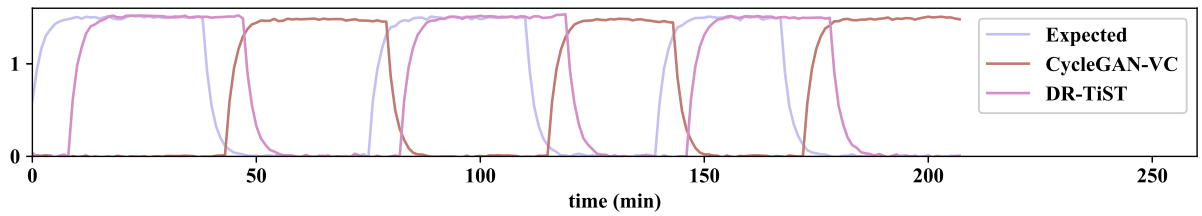

Fig. 5: **Comparison of the time series of engine 2 generated with the operating mode of engine 1 in** $Test\ 3$ **where** $\mu_1 = 35$ **and** $\mu_2 = 45$**.** Time series generated by CycleGAN-VC, depicted in red, has a completely different behavior than the expected time series, depicted in green. The time series produced by DR-TiST, depicted in blue, is more realistic.

	Engine 2 Generated		Engine 1 Generated	
	$RMSE2_{cyc}$	$RMSE2_{DR}$	$RMSE1_{cyc}$	$RMSE1_{DR}$
$Test\ 2$	0.565	0.51	0.25	0.18
$Test\ 3$	1.37	0.59	0.86	0.28

TABLE V: RMSE values for generating time series of engine 1 and engine 2 by CycleGAN-VC and DR-TiST for test 2,3 and 4. $RMSE_{cyc}$ and $RMSE_{DR}$ denote the values of $RMSE$ for the time series of CycleGAN-VC and DR-TiST respectively.

Fig. 6 and Fig. 7 illustrate the time series generated by the DR-TiST for engine 2 and engine 1 respectively. We see in Fig. 6 the recorded time series of engine 1 depicted in blue. The time series produced by DR-TiST, depicted in orange, have the functional behavior of engine 2 and the same on/off cycles as the ones of engine 1. Fig. 7 demonstrates the originally recorded time series of engine 2, in blue, i.e. with a maximal value of 1.5 and the corresponding time series generated, in orange, by the algorithm with the same on/off times and a different functional behavior. For the different examples, we clearly see that the trained model was able to simulate the behavior of engine 1 with the on/off times of engine 2 and vice versa. This corresponds to the desired behavior. Since we want to show that this is not only an exemplary time series, for each test, we compute our quantifiable metrics on a set of 100 generated time series. Table VI summarizes the results of $RMSE$ and D values respectively when generating time series of engine 1 and engine 2 in the different experiments. The obtained results demonstrate that test 1 has the lowest $RMSE$ and D values i.e. $D_{eng1} = 0.0652$ and $RMSE_{eng2} = 0.26$. It is to notice that in test 1 $\mu_1 = \mu_2$. In test 2, $RMSE_{eng1}$ and $RMSE_{eng2}$ are higher when DR-TiST is trained with 208 data points instead of 416 data points. Hence, the amount of data has an impact on the results. Test 3 and test 4 are characterized by higher $RMSE$ and D values. By way of example, D_{eng2} is equal to 0.5 and 0.16 for test 3 and test 4 respectively.

Examples of time series generated by CycleGAN-VC and DR-TiST in test 2 and 3 are presented in Fig. 8 and 5 respectively. The point-wise differences between the time series generated by CycleGAN-VC and the expected time series are higher than the one generated with DR-TiST. Moreover, the time series of DR-TiST are more realistic and fit better to the target time domain than the time series generated by CycleGAN-VC. Fig. 5 shows that for test 3 the time series of CycleGAN-VC present a completely wrong on/off cycles. Table V and IV show the $RMSE$ and D values of DR-TiST compared to CycleGAN-VC. In test 2 and 3 DR-TiST outperforms CycleGAN-VC. The D and $RMSE$ values of DR-TiST are for both experiments lower than the one of CycleGAN-VC i.e. in test 4 $D2_{cyc} = 1.77$ and $D2_{DR} = 0.43$. It is to notice that DR-TiST is faster and more efficient than CycleGAN-VC in terms of time and computation.

	D_{eng1}	D_{eng2}	$RMSE_{eng2}$	$RMSE_{eng1}$
$Test\ 1$	0.066	0.1693	0.053	0.266
$Test\ 2_{lg208}$	0.202	0.36	0.18	0.51
$Test\ 2_{lg416}$	0.11	0.21	0.1105	0.347
$Test\ 3$	0.309	0.5	0.66	0.26
$Test\ 4$	0.13	0.164	0.122	0.29

TABLE VI: Computed D and $RMSE$ values for the different tests. D_{eng1} and D_{eng2}, $RMSE_{eng2}$ and $RMSE_{eng1}$ denote the computed D and $RMSE$ values during the test phase when generating time of engine 1 and engine 2 respectively.

VI. Conclusion

This work tackles a special problem, namely time series translation by decomposing them into a functional behavior and an operating mode. The presented framework applies the gated CNN structure on DRIT an algorithm originally designed for image-to-image translation purposes. The utility of the proposed method is tested on a real world use case where we transfer the behavior of a ventilation system operating in a small room with slow on/off cycles to another environment characterized by a faster on/off cycles. We conduct four different tests where we vary the on/off times of the ventilation systems. Results show that we were able to transfer the behavior of a system to the time domain of the other system and that DR-TiST outperforms CycleGAN-VC both in quality of the generated time series as well as in runtime. The evaluation of translation techniques and GANs remains a challenge and relies in most of the cases on a human judge. In the future, we plan to investigate new evaluation methods that are more suitable for time series data. Moreover, we plan to consider other domain adaptation algorithms and appply it to real data.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

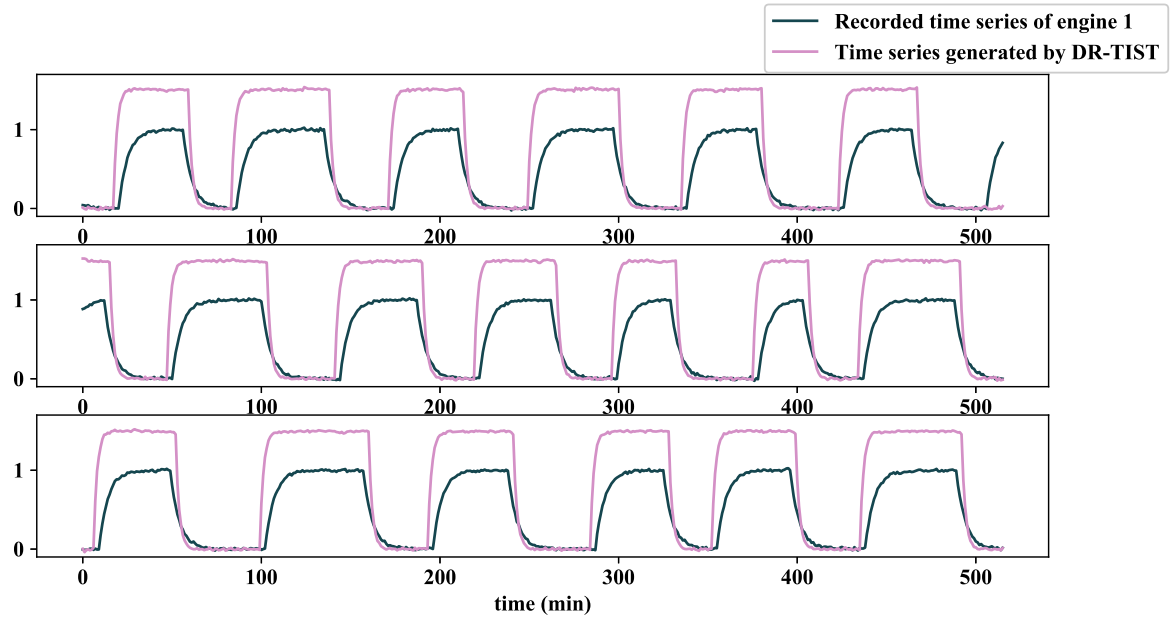

Fig. 6: **Examples of time series of engine 2 in operating mode of engine 1 generated in** $test\ 4$ **where** $\mu_1 = 40$ **and** $\mu_2 = 50$**.** The originally recorded time series of engine 1 are depicted in blue. DR-TiST was able to map the functional behavior of engine 2, characterized by a higher amplitude, in the time domain of engine 1. The resulting time series are depicted in orange.

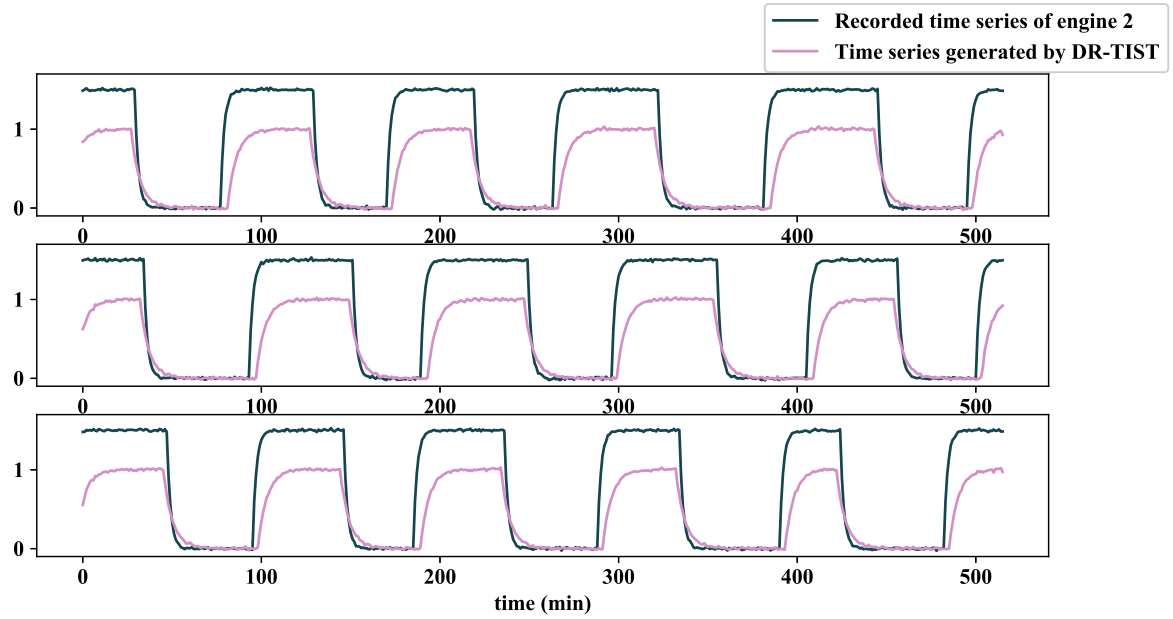

Fig. 7: **Examples of time series of engine 1 in operating mode of engine 2 generated in** $test\ 4$ **where** $\mu_1 = 40$ **and** $\mu_2 = 50$**.** The originally recorded time series of engine 2 are depicted in blue. DR-TiST was able to map the functional behavior of engine 1, characterized by a lower amplitude, in the time domain of engine 2. The resulting time series are depicted in orange.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

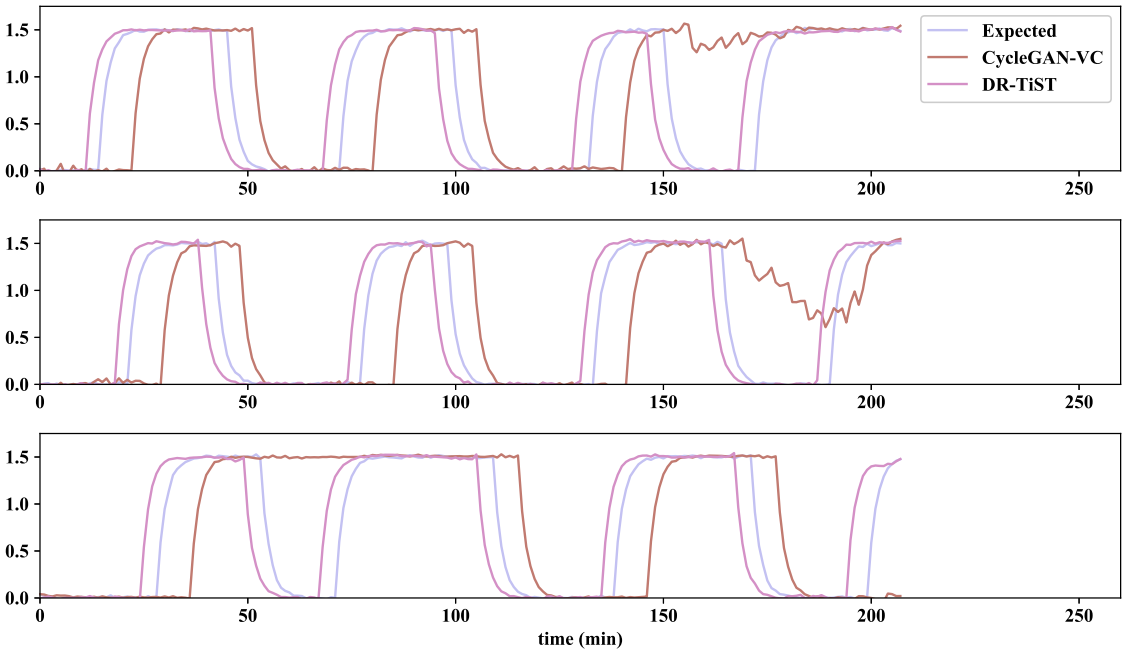

Fig. 8: **Comparison of time series of engine 2 generated with the operating mode of engine 1 in** $test\ 2$ **where** $\mu_1 = 30$ **and** $\mu_2 = 40$**.** Time series generated by CycleGAN-VC are depicted in red while the time series produced by DR-TiST are depicted in blue. The target time series representing the behavior of engine 2 in the operating mode of a selected engine 1 are depicted in green.

REFERENCES

[1] H. Lee and H. Tseng and Q. Mao and J. Huang and Y. Lu and M. Singh and M. Yang, "DRIT++: Diverse Image-to-Image Translation via Disentangled Representations," arXiv preprint arXiv:1905.01270, 2019.

[2] H. Lee and H. Tseng and J. Huang and M. Singh and M. Yang, "Diverse Image-to-Image Translation via Disentangled Representations," In European Conference on Computer Vision (ECCV), pp. 36–52, 2018.

[3] M. Liu and T. Breuel and J. Kautz, "Unsupervised Image-to-Image Translation Networks," In Advances in Neural Information Processing Systems 30, Curran Associates, Inc., pp. 700–708, 2017.

[4] J. Zhu and T. Park and P. Isola and A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 2223-2232, 2017.

[5] J. Zhu and T. Park and P. Isola and A. Efros, "Learning to Discover Cross-Domain Relations with Generative Adversarial Networks," In Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 1857—1865, Sydney, NSW, Australia, 2017.

[6] J. Zhu and R. Zhang and D. Pathak and T. Darrell and A. A Efros and O. Wang and E. Shechtman, "Toward Multimodal Image-to-Image Translation," In Advances in Neural Information Processing Systems 30, Curran Associates, Inc., pp. 465–476, Sydney, NSW, Australia, 2017.

[7] P. Isola and J. Zhu and T. Zhou and A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1125-1134, 2017.

[8] I. Goodfellow and J. Pouget-Abadie and M. Mirza and B. Xu and D. Warde-Farley and S. Ozair and A. Courville and Y. Bengio, "Generative Adversarial Nets," In Advances in Neural Information Processing Systems 27, Curran Associates, Inc., pp. 2672–2680, 2014.

[9] M. Olof, "C-RNN-GAN: A continuous recurrent neural network with adversarial training," Constructive Machine Learning Workshop (CML) at NIPS 2016, pp. 1, 2016.

[10] C. Esteban and S. L. Hyland and G. Rätsch, "Real-valued (medical) time series generation with recurrent conditional gans," arXiv preprint arXiv:1706.02633, 2017.

[11] L. Theis and A. van den Oord and M. Bethge, "A note on the evaluation of generative models," arXiv preprint arXiv:1511.01844, 2016.

[12] A. Radford and L. Metz and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," arXiv preprint arXiv:1511.06434, 2015.

[13] Y. N. Dauphin and A. Fan and M. Auli and D. Grangier, "Language Modeling with Gated Convolutional Networks," In Proceedings of the 34th International Conference on Machine Learning, JMLR.org, vol. 70, pp. 933—941, Sydney, NSW, Australia, 2017.

[14] T. Kaneko and H. Kameoka, "CycleGAN-VC: Non-parallel Voice Conversion Using Cycle-Consistent Adversarial Networks," In 2018 26th European Signal Processing Conference (EUSIPCO), pp. 2100–2104, Sydney, 2018.

[15] R. Zhang and P. Isola and A. A. Efros, "Colorful Image Colorization." In European Conference on Computer Vision (ECCV), Springer, Cham., pp. 649–666, 2016.

[16] R. Zhang and J. Zhu and P. Isola and X. Geng and A. S. Lin and T. Yu and A. A. Efros, "Real-Time User-Guided Image Colorization with Learned Deep Priors," arXiv preprint arXiv:1705.02999, 2017.

[17] X. Huang and M. Liu and S. Belongie and J. Kautz, "Multimodal Unsupervised Image-to-image Translation," In Proceedings of the European Conference on Computer Vision (ECCV), pp. 172–189, 2018.

[18] L. Yu and W. Zhang and J. Wang and Y. Yu, "SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient," In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI Press, pp. 2852—2858, San Francisco, California, USA, 2017.

[19] D. Ulyanov and A. Vedaldi and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," arXiv preprint arXiv:1607.08022, 2016.

[20] W. Shi and J. Caballero and F. Huszar and J. Totz and A. P. Aitken and R. Bishop and D. Rueckert and Z. Wang, "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1874-1883, 2016.

[21] H. Arnout and J. Kehrer and J. Bronner and T. Runkler, "Visual Evaluation of Generative Adversarial Networks for Time Series Data," arXiv preprint arXiv:2001.00062, 2019.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE