CLARE-GAN: GENERATION OF CLASS-SPECIFIC TIME SERIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Recently, through numerous works Esteban et al. (2017); Mogren (2016); Yoon et al. (2019) attempts were made to obtain generative models for time series that correctly reproduce the underlying temporal characteristics of a given training data set. However, we prove in this work that the performance of these models is limited on datasets with *high-variability* for example containing different classes. In such setups, it is extremely difficult for a generative model to find the right trade-off between sample fidelity i.e. their similarity to the real time series and sample diversity. Furthermore, it is essential to preserve the original classes and the variation within each class. To tackle this issue, we propose a new *generative class sensitive model*, Class-specific Recurrent GAN (CLaRe-GAN), that conditions the generator on an auxiliary information containing the class-specific and class-independent attributes. Our model relies on *class specific encoders*: a unique encoder for two contradictory functionalities i.e. extracting the inter- and intra-class attributes. To extract the high-level representation of the time series, we make a shared-latent space assumption Liu et al. (2017). At the same time, we use a *class discriminator* that discriminates between the latent vectors to efficiently extract the class-specific attributes. We test our approach on a set of publicly available datasets where the number of classes, the length and the number of available times series for each class varies and evaluate our approach both visually and computationally. We prove that our model outperforms the state-of-the-art generative models and leads to a significant and consistent improvement in the quality of the generated time series while at the same time preserving the classes and the variation of the original dataset.

1 INTRODUCTION

In spite of the great success of GANs with images Karras et al. (2019; 2020); Donahue & Simonyan (2019), their utility and use on time series data is still limited. Indeed, finding a suitable generative model for time series is challenging and a lot of factors must be considered when designing a good generation framework for this type of data. One main requirement is that the time series generation preserves the temporal relationship between the data points. Moreover, as for other types of data, an efficient generative model should be able to find a compromise between *sample fidelity*, i.e. the similarity of the generated data to the real ones and *sample diversity*, i.e. reproducing the variation of the real data. This is particularly challenging for datasets with *high variability* such as datasets with multiple classes, where one or many classes are misrepresented i.e. in imbalanced setups or when the difference between the classes is not obvious and outstanding. A possible solution to this problem would be to train a specific GAN for each class, however this reduces the potential to learn characteristics common to all classes from the full training data set. For such datasets, it is fundamental to produce samples of high-quality that reflect the inter- and intra-class variation. This means that all the classes of the real dataset must be represented in the synthetic samples and that the diversity within each class should be preserved.

A first generative model for continuous sequential data was proposed by Mogren Mogren (2016) namely C-RNN-GAN consisting of a recurrent Generator and Discriminator. A well-known method to improve the performance of GAN is to condition GAN on additional information extracted from the original data Mirza & Osindero (2014). In this context, Cristobal et al. proposed a more efficient model, RCGAN Esteban et al. (2017), i.e. a recurrent GAN augmented with auxiliary conditional information. More recently, Yoon et al. introduced TimeGAN that preserves better the temporal dynamic between the data points and that can be used on mixed datasets i.e. dataset with static and temporal data. In this work, we will prove that these state-of-the-art frameworks, despite generating realistic time series in simple environments, fail to produce acceptable results preserving the inter- and intra-class diversity of the original dataset on multi-class datasets. Driven by the recent success of conditional GAN Mirza & Osindero (2014) and similar to RCGAN, we introduce Class-Specific Recurrent GAN (ClaRe-GAN) a *generative class sensitive framework for time series*. Our approach is based on the simple and intuitive idea: to deal with datasets of *high-variability*, learning their inter- and intra-class variations is the basis and hence our starting point. We assume that time series from various classes within the same dataset share some common information and that this should be contained in the generated data. To extract this information, we use *class-specific encoders*, one encoder per class and make a shared-latent space assumption enforcing the latent vectors to be mapped to a common latent space. At the same time, a major challenge when using GAN on datasets with multiple classes is not only to preserve the classes but also the diversity within each class. To achieve this, we use an extra *class discriminator*, that is trained by applying a class adversarial loss to enable an efficient comparison between the latent codes of the different encoders and a precise extraction of the class-specific attributes. Thanks to this special setup we ensure that the extracted latent codes are class-specific and, at the same time, contain the high-level representations of the original dataset. Moreover, our framework uses a collection of loss functions borrowed from image-to-image translation techniques and generative models for images to improve the diversity of the generated time series. We finally prove that a Generator conditioned on these representations outperforms the existing models by generating time series of much higher-quality i.e. diverse samples that preserve the classes of the original dataset and their properties very well (high fidelity). In order to encourage reproducibility, our code will be made available if the paper is accepted.

We test our approach on a collection of datasets from the UEA & UCR Time Series Classification Repository Bagnall et al. (2018) which vary in time series' length and number of classes and compare its performance to the state-of-the-art generative models for time series. The performance of our model is evaluated against different criteria namely *diversity*, *fidelity* and *usefulness*. In order to enable a fair comparison, we use the same evaluation metrics of Yoon et al. (2019) i.e. Discriminative and Predictive Score. While the Discriminative Score assess the *fidelity* of the generated time series through a classifier meant to distinguish between the real and the generated datasets, the Predictive Score evaluates the *usefulness* of the generated data on prediction tasks. Furthermore, we visualize the real and the synthetic time series in full and reduced dimensions using PCA Bryant & Yarnold (1995) and t-SNE Maaten & Hinton (2008) analysis such as in Yoon et al. (2019) to allow a direct comparison and to asses the *diversity* of the generated time series. In the conducted experiments, we show that our method is scalable with the length of the time series and the number of classes and conclude that our framework produces time series of high-quality. It outperforms the existing state-of-the-art methods visually and computationally enabling significant progress in the research area of generative models for time series.

2 Related Work

Recently, several generative models for time series data have been proposed. Mogren proposed C-RNN-GAN Mogren (2016), illustrated in Fig. 1 (d), that consists of a recurrent Generator and Discriminator and uses techniques from Salimans et al. (2016) to stabilize the GAN's training. The proposed framework was tested on classical music. Later, Cristobal et al. proposed RCGAN Esteban et al. (2017) (Fig. 1 (c)) a conditional version of recurrent GAN that conditions the input of the generator and the discriminator on an additional information. A more sophisticated architecture was presented by Yoon et al. Yoon et al. (2019) in TimeGAN (Fig. 1 (d)) consisting of an embedding and recovery functions and a recurrent Generator and Discriminator. The main goal behind TimeGAN was to design a good generative model that preserves the temporal dynamic for mixed datasets with temporal and static data. Unlike these models, our approach presents an efficient architecture to extract the conditional information i.e. the class-specific and class-independent attributes of the dataset. This is achieved by means of a special architecture for the class-specific encoders and an additional *class* discriminator that discriminate between the latent representations by applying a class adversarial loss. A comparison between the different models architecture is illustrated in Fig. 1.

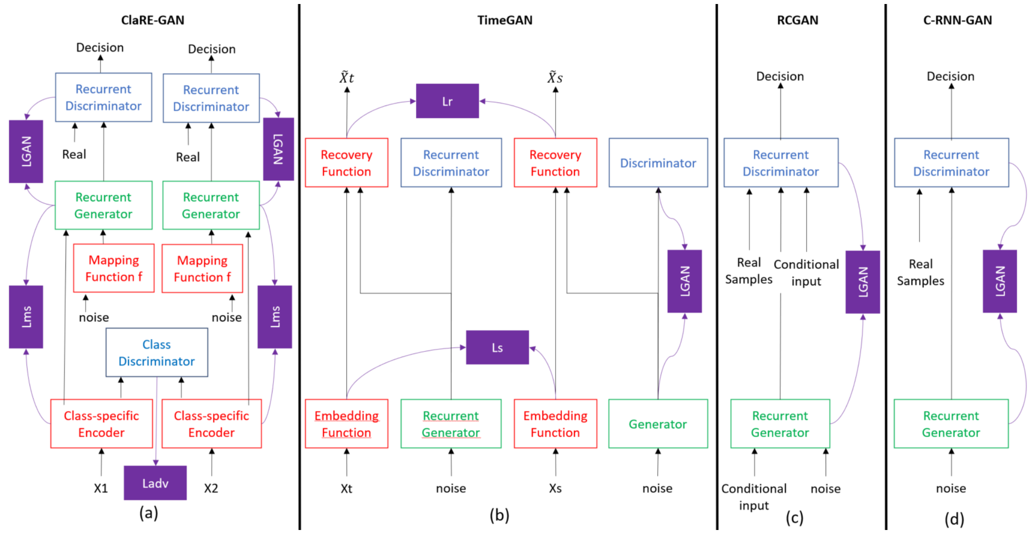

Figure 1: Comparison of the architecture of the different generative models for time series data. The used loss functions are depicted in lilacs X_s, X^s and X_t, X^t denotes the static and temporal data respectively. X_1 and X_2 are time series from two different classes.

Learning efficient representation of time series have been exhaustively investigated to perform different Machine Learning (ML) tasks i.e. for clustering purposes Ma et al. (2019), to perform classification tasks in semi-supervised Jawed et al. (2020) or unsupervised Srivastava et al. (2015) settings and to perform forecasting Lyu et al. (2018) or prediction Dai & Le (2015) or to achieve disentanglement Li & Mandt (2018). In this context, Denton et al. (2017) applied an adversarial loss to split the video frames into time-dependent and time-independent parts. This method was later adopted by Lee et al. (2018; 2020) in order to disentangle each image in content and attribute to perform image-to-image translation tasks. In a similar fashion, the *class discriminator* is trained via an adversarial loss to distinguish between the representations of the different classes. Moreover, representation learning was successfully applied in several application domains such as for medical purposes Jones et al. (2016) to model different medical concepts Choi et al. (2016) or to predict the patient phenotype Suresh et al..

Different attempts in conditioning GAN based on an auxiliary information have been proposed in the literature. Several works focused on the generator and discriminator architectures and the methods used to concatenate the label with the data Mirza & Osindero (2014); Reed et al. (2016). Others proposed a projected based discriminator Miyato & Koyama (2018) or a class-aware generator Zhou et al. (2020). Conditional GAN was also used for other purposes than image generation such as image-to-image translation Kim et al. (2017) or text-to-image translation Zhang et al. (2017). Rather than conditioning the generator on the label or proposing a class-aware architecture for the generator Zhou et al. (2020), we propose a novel approach that conditions the generator on an auxiliary information that efficiently regroup the common characteristics between the different classes and the class-specific representation.

3 Class-specific Recurrent GAN

In Conditional GAN (cGAN), labeled setups were used to condition the input of the generator leading to a significant improvement in the quality of generated data. In a similar fashion, we exploit the supervised setup to find a good generative model for time series stemming from different classes. Our main goal is to learn the inter- and intra-class variations of the dataset and to exploit this information to improve the quality and the diversity of the generated time series. Our approach combines *class-specific encoders* and a *class discriminator* to enable an efficient extraction of the class-dependent and class-independent attributes. Our model is composed of *class-specific encoders*, one encoder per class, a *class discriminator* and recurrent generators and discriminators, as illustrated in Fig. 1. *Class-specific encoders* are trained in a parallel manner using the real time series belonging to the different classes and the *class discriminator*. Our model can therefore be used for any number of classes.

We consider a set $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2 \cup \ldots \cup \mathcal{X}_N$ of labeled time series $\{x_t\}_{t=1}^{T}$ from N different classes following a distribution $\mathbb{P}_r$. Let $S_r^{tr} = \left\{x_{1,1:T}^{tr}, x_{2,1:T}^{tr}, \ldots, x_{k,1:T}^{tr}\right\}$ be a set of training samples uniformly sampled from $\mathbb{P}_r$. For each class $n \in \{1, \ldots, N\}$, we use a GAN consisting of a Generator G_n and a Discriminator D_n to learn a distribution $\mathbb{P}_g$ that approximates $\mathbb{P}_r$. The Discriminator D_n is a binary classifier that tries to distinguish between the real time series and the ones synthesized by the Generator G_n. The minimax game between both components is summarized as follows:

$$\mathcal{L}_{GAN}\left(G_n, D_n\right) = \mathbb{E}_{x \sim p(x)}\left[log\left(D_n\left(x\right)\right)\right] + \mathbb{E}_{z \sim p_z(z)}\left[log\left(1 - D_n\left(G_n\left(z\right)\right)\right)\right] \tag{1}$$

where z is a latent vector and $x \in \mathcal{X}_n$.

A *class-specific encoder* $E_n : \mathcal{X}_n \rightarrow \mathcal{C}$ learns a representation for each time series $x \in \mathcal{X}_n$ belonging to the same class n by mapping them into a vector of predetermined length. This *class-specific encoder* is used to gather class-specific and class-independent attributes by learning the factor of variation of each real time series:

$$c_x^n = E_n\left(x\right) \tag{2}$$

We assume that all time series are mapped to the same latent space $\mathcal{C}$. This is achieved by sharing the weights Liu et al. (2017) of the last 2 layers of the encoders. This assumption guarantees a common extraction of the high-level representation of the time series i.e. we extract the class-independent properties. At the same time, the *class-specific encoders* are trained adversarially by means of an additional *class-specific discriminator*. It discriminates between the representation of the different time series to enable an efficient extraction of the class-specific features. The extracted latent codes are later concatenated with an input noise vector in a sophisticated manner to generate time series of high-quality. Furthermore, we make no restriction on the architecture of these encoders in our case we used Convolutional Neural Networks, but architectures such as fully connected layers or Recurrent neural networks can replace those as well.

Inspired by Lee et al. (2018; 2020), we impose a *class discriminator* $D_{\mathcal{X}}^{Cl}$ to discriminate between the learn class representations of the encoders E_n allowing a more precise extraction of the different class features. The *class discriminator* is trained by applying a class adversarial loss to improve the quality of the variation learned by the encoders. For a dataset with 2 classes the class adversarial loss can be expressed as follows:

$$\begin{aligned}\mathcal{L}_{adv}\left(E_1, E_2, D_{\mathcal{X}}^{Cl}\right) = \mathbb{E}_{x1}\left[0,5 \cdot \log\left(D_{\mathcal{X}}^{Cl}\left(E_1\left(x_1\right)\right)\right) + 0,5 \cdot \log\left(1 - D_{\mathcal{X}}^{Cl}\left(E_1\left(x_1\right)\right)\right)\right] + \\ \mathbb{E}_{x2}\left[0,5 \cdot \log\left(D_{\mathcal{X}}^{Cl}\left(E_2\left(x_2\right)\right)\right) + 0,5 \cdot \log\left(1 - D_{\mathcal{X}}^{Cl}\left(E_2\left(x_2\right)\right)\right)\right]\end{aligned} \tag{3}$$

As in Karras et al. (2019; 2020), our generator $G_n : \{\mathcal{C}, z\} \rightarrow \mathcal{X}$ is equipped with a mapping function $f : \mathbb{R}^m \rightarrow \mathcal{W}$ consisting in 3 fully connected layers. However, our mapping function f is used on the noise vector z instead of the latent vector c_x^n i.e. $f\left(z\right) = w$ where the noise vector z is sampled randomly from a pre-defined distribution. In our case, we used the Gaussian distribution. The obtained vector w is later concatenated with the latent code c_x^n and fed to a Recurrent Neural Networks (RNNs). Like all other generative models designed for time series we opt for the usage of RNNs due to their well-known ability in modeling sequential data. We used in our case Long short-term memory (LSTM) but we make no restriction on the recurrent architecture.

Our discriminator $D_n : \mathcal{X} \rightarrow [0, 1]$ minimizes the original discriminator loss function as proposed by Goodfellow when introducing GAN i.e. Eq. 1. We use multi-scale discriminators Wang et al. (2018); Zhang et al. (2018) originally designed for images. In this case, many discriminators are used and trained with different image resolutions. In our experiment, we found that this multi-scaling techniques i.e. feeding the same input time series under different levels of compression to a multitude of discriminators eases the training process and improves the quality of the synthesized time series. Additionally to the class adversarial loss, we apply further loss functions that facilitate the training and improve the quality of the generated time series.

Mode Seeking regularization To improve the diversity of the generated data and to prevent mode collapse, we apply the mode seeking regularization term Mao et al. (2019) that helps to capture the different modes present in the real dataset by maximizing the ratio of the distance between two generated time series $G(x, z_1)$ and $G(x, z_2)$ given an input time series x, and two latent noise vectors z_1 and z_2,

$$\mathcal{L}_{ms} = \max_{G_n} \left(\frac{d_x \left(G_n \left(z_1, x \right), G_n \left(z_2, x \right) \right)}{d_z \left(z_1, z_2 \right)} \right) \tag{4}$$

where d_* denotes the mean absolute error.

The full objective function of our framework can then be written as:

$$\lambda_{GAN} \mathcal{L}_{GAN} \left(G_n, D_n \right) + \lambda^c_{adv} \mathcal{L}_{adv} \left(E_1, E_2, D^{Cl}_{\mathcal{X}} \right) + \lambda_{ms} \mathcal{L}_{ms} + \lambda_c \parallel c^n_x \parallel^2 \tag{5}$$

where $\parallel c^n_x \parallel^2$ is a $L2$ regularization term applied to prevent overfitting and λ_{GAN}, λ^c_{adv} and λ_{ms} are the model parameters. The used loss function are summarized in Fig. 1. In our experiments, we used $\lambda_{GAN} = 1$, $\lambda^c_{adv} = 1$ $\lambda_c = 0.01$ and $\lambda_{ms} = 1e-5$.

4 Evaluation Methods

In order to enable a fair comparison between the different frameworks we use as part of our evaluation the evaluation methods proposed in Yoon et al. (2019) when introducing TimeGAN. It is to be noted that TimeGAN was also compared to the previous existing frameworks namely RCGAN and C-RNN-GAN. As in Yoon et al. (2019), we focus on three important criteria when assessing the quality of the generated data namely *diversity*, *fidelity* and *usefulness*:

- **Diversity** the generated data should reflect the variation present in the real dataset
- **Fidelity** the generated data must be indistinguishable from the real ones
- **Usefulness** asses the applicability and the utility of the generated data i.e. the synthetic samples should be as useful as the real data when performing prediction tasks

We perform a computational and visual evaluation by computing the Predictive and Discriminative scores and by visualizing the synthetic and real samples:

- **Discriminative Score** Yoon et al. (2019) A 2-layer LSTM is trained to discriminate between the real and synthetic samples. During the training process, each real time series is labeled as real and each generated time series is labeled as fake. The Discriminative score denotes classification error of a test set consisting of a mix between real and generated samples. This score measures the similarity between both datasets and hence checks whether the generated samples are indistinguishable from the real data i.e. fulfill the fidelity criterion.
- **Predictive Score** In order to assess the usefulness of the generated time series, a 2-layer LSTM is trained to predict the next coming value for each sequence of the generated time series. The trained model is later tested on the real time series. The Predictive score represents the mean absolute error between the predicted and the real values.
- **Visualization** Yoon et al. Yoon et al. (2019) applied t-SNE Maaten & Hinton (2008) and PCA Bryant & Yarnold (1995) on the real and generated time series to enable an efficient comparison of the two distributions in a 2-dimensional space. These visualizations aim to compare the diversity of the real and synthetic samples. However, we will show in the following that the evaluation of time series similarity in a reduced dimension space alone is not sufficient to ensure high quality time series. To convince the reader about the generated time series, we opt for an additional visualization method where we plot all the real time series and the generated time series side-by-side to enable a direct comparison.

5 Experiments

Dataset Description We evaluate the performance of our approach on a collection of publicly available datasets from the UEA & UCR Time Series Classification Repository Bagnall et al. (2018; 2017) with time series of different properties namely ItalyPowerDemand, TwoLeadECG Goldberger et al. (2000), Yoga, DistalPhalanxTW Davis (2013) and FreezerRegularTrain. The datasets used in

Table 1: Summary of the characteristics of the used datasets. The datasets are publicly available in the UEA & UCR Time Series Classification Repository Bagnall et al. (2018) and differ in the length of the time series the number of classes and the ratio of data per class.

Dataset	***Length***	***Number of classes***	***Ratio of data available per class***
ItalyPowerDemand	24	2	50%-50%
TwoLeadECG	82	2	50%-50%
FreezerRegularTrain	301	2	50%-50%
Yoga	425	2	50%-50%
DistalPhalanxTW	80	6	34,18%-34,18% -3,48%-5,76% -16,08%-% 6,32

our experiments vary in the length of the real time series ranging from 24 to 425, the number of class present in the real dataset ranging from 2 to 6, the number of time series per class: balanced and imbalanced datasets, and the characteristics of the times series such as the class properties or the level of noise. Table 1 summarizes the characteristics of the used datasets. The experiments are conducted on t2.large AWS EC2 instances with 8 GiB of system memory and 2 vCPUs. For a detailed description of the implementation, we refer to the appendix.

Baseline We compare our method to the state-of-the-art generative models for time series data namely TimeGAN Yoon et al. (2019), RCGAN Esteban et al. (2017) and C-RNN-GAN Mogren (2016).

Results The results of the PCA analysis are illustrated in Fig. 2. We clearly see that C-RNN-GAN shows a limited performance in terms of samples diversity. A better performance was noticed for RCGAN with the ItalyPowerDemand, TwoLeadECG and Yoga datasets. However, the distribution of the samples generated by RCGAN differs from the distribution of the real samples. Furthermore, the samples generated by TimeGAN are not as diverse as the real dataset for the TwoLeadECG dataset and TimeGAN fails in capturing the distribution of the real samples for the other datasets. The PCA and t-SNE (available in appendix) plots show that there is a significant improvement in the diversity of the samples generated by ClaRe-GAN in comparison with the other methods. We clearly see that the distribution of the samples generated by ClaRe-GAN is the closest to the distribution of the real samples and that the performance of CLaRe-GAN scales well with the number of class and the ratio of data available per class (DistalPhalanxTW dataset) and the length of the time series (Yoga and FreezerRegularTrain datasets). ClaRe-GAN was able to capture the distribution of the real data independently of the dataset and its properties.

To allow a direct comparison of the time series, we visualize the real and generated time series by each model. The results for the yoga dataset are depicted in Fig. 3. Further examples are available in Appendix. We clearly see that visualizing both datasets in a reduced 2-dimensional space is not enough. In fact, Fig. 3 shows that the time series generated by RCGAN are noisy and the main class properties are ignored during the generation process. Moreover, according to Fig. 2, TimeGAN is showing a good performance and the difference between the distribution of the generated and the real samples is not as important as for RCGAN. However, we see in Fig. 3 that the time series generated by TimeGAN don't reflect the properties of the real dataset and are smoother than the real ones. In contrast to that, ClaRe-GAN captures properly the properties of the real dataset and its class characteristics. While the state-of-the-art methods presented a poor or limited performance, our method was able to learn the inter- and intra-class variations of the original dataset and to reflect these properties in the generated dataset.

The fidelity and usefulness of the generated time series are assessed with the Discriminative and Predictive Score respectively. The obtained results are summarized in Tables 2 and 3. They show that the fidelity and usefulness of the time series synthesized by C-RNN-GAN is limited. A better performance is noticed for RCGAN and TimeGAN. For all the datasets and independently from their characteristics, ClaRe-GAN achieves the lowest Predictive and Discriminative Scores which proves that the time series generated by this framework are of high-fidelity and are as useful as the real time series. For example, a great improvement was noticed in terms of Discriminative Score (half of the best Discriminative Score achieved by the other methods) for the yoga dataset. Moreover, for the

DistalPhalanxTW dataset, an imbalanced dataset with 6 classes, the ClaRe-GAN's Predictive Score is equal to 0,13. It is to be noted that ClaRe-GAN is also the most efficient in terms of computation.

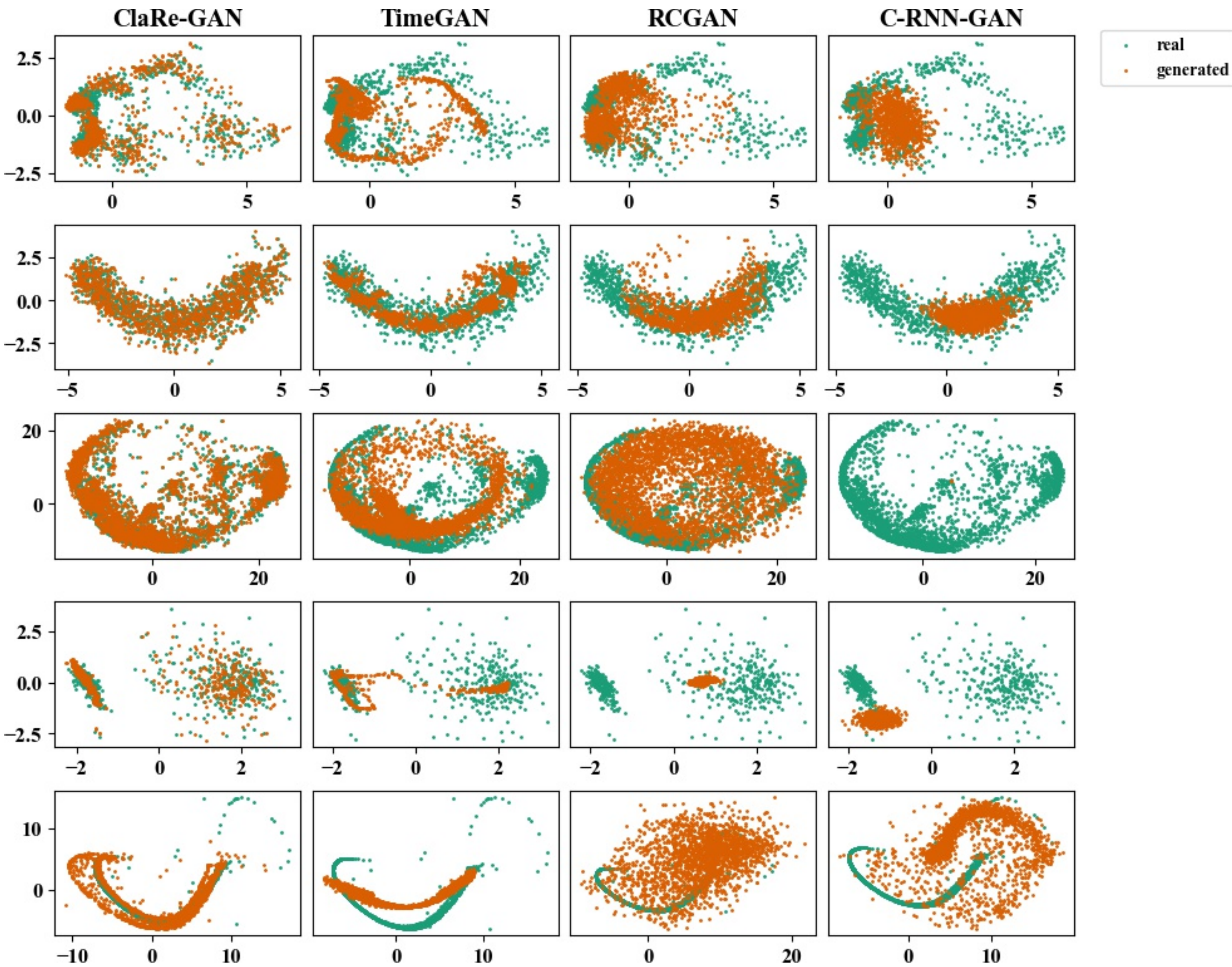

Figure 2: Comparison of the real (depicted in green) and generated (depicted in orange) dataset with PCA. Each row corresponds to a specific dataset and each column to a method. The results are presented in the following order (top to bottom): ItalyPowerDemand, TwoLeadECG, Yoga, DistalPhalanxTW and FreezerRegularTrain.

6 CONCLUSION

We introduced a new generative model for time series with class information which efficiently simulates novel times series capturing both the class association and variability of the training data set. Our approach relies on *class conditional encoders* and a *class discriminator* to extract simultaneously class-specific and class-independent features. We compared our model to different state-of-the-art generative models for time series and prove that our model extracts effectively the inter- and intra-class features leading to a drastic improvement in the quality of the synthesized time series even in challenging setups such as imbalanced datasets. In the future, we plan to investigate the utility of our framework for other purposes such as translation or domain adaptation tasks. Furthermore, the generalization to multivariate time series data and the proper evaluation of these results is a planned next step.

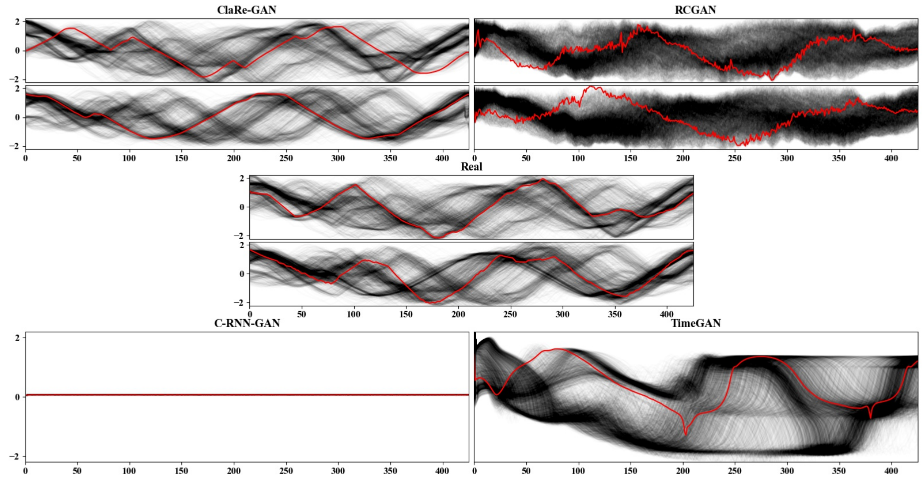

Figure 3: Illustration of the real and generated time series by ClaRe-GAN, RCGAN, C-RNN-GAN and TimeGAN for the Yoga dataset. The time series are depicted in black. The red line presents an example time-series for each subplot. For the conditional GANs, ClaRe-GAN and RCGAN, and the real dataset we visualize the time series of each class separately.

Table 2: Discriminative Scores computed on the time series generated by the different frameworks (ClaRe-GAN TimeGAN, RCGAN and C-RNN-GAN) for the different datasets namely TwoLeadECG, Yoga and DistalPhalanxTW. A lower Discriminative Score denotes a high-fidelity to the real datasets.

Dataset	***ClaRe-GAN***	***TimeGAN***	***RCGAN***	***C-RNN-GAN***
TwoLeadECG	0.224	0.3985	0.2633	0.4498
Yoga	0.08	0.2	0.17	0.4998
DistalPhalanxTW	0.4273	0.496	0.447	0.4981

Table 3: Predictive Scores computed on the time series generated by the different frameworks (ClaRe-GAN TimeGAN, RCGAN and C-RNN-GAN) for the different datasets namely TwoLeadECG, Yoga and DistalPhalanxTW. A lower Predictive Score denotes a better usefulness of the generated time series.

Dataset	***ClaRe-GAN***	***TimeGAN***	***RCGAN***	***C-RNN-GAN***
TwoLeadECG	0.117	0.1246	0.127	0.5965
Yoga	0.156	0.157	0.16	0.5349
DistalPhalanxTW	0.1349	0.1749	0.2164	0.4784

REFERENCES

A. Bagnall, J. Linesand A. Bostrom, J. Large, and E. Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 31(3):606–660, 2017.

A. Bagnall, J. Lines, W. Vickers, and E. Keogh. The uea & ucr time series classification repository. *URL http://www. timeseriesclassification. com*, 2018.

Fred B Bryant and Paul R Yarnold. Principal-components analysis and exploratory and confirmatory factor analysis. 1995.

Edward Choi, Mohammad Taha Bahadori, Elizabeth Searles, Catherine Coffey, Michael Thompson, James Bost, Javier Tejedor-Sojo, and Jimeng Sun. Multi-layer representation learning for medical concepts. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1495–1504, 2016.

Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In *Advances in neural information processing systems*, pp. 3079–3087, 2015.

Luke M Davis. *Predictive modelling of bone ageing*. PhD thesis, University of East Anglia, 2013.

Emily L Denton et al. Unsupervised learning of disentangled representations from video. In *Advances in neural information processing systems*, pp. 4414–4423, 2017.

J. Donahue and K. Simonyan. Large scale adversarial representation learning. In *Advances in Neural Information Processing Systems*, pp. 10542–10552, 2019.

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

A. L. Goldberger, , L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, and H. E. Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23): e215–e220, 2000.

Shayan Jawed, Josif Grabocka, and Lars Schmidt-Thieme. Self-supervised learning for semi-supervised time series classification. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 499–511. Springer, 2020.

Corinne L Jones, Sham M Kakade, Lucas W Thornblade, David R Flum, and Abraham D Flaxman. Canonical correlation analysis for analyzing sequences of medical billing codes. *arXiv preprint arXiv:1612.00516*, 2016.

T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8110–8119, 2020.

Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. *arXiv preprint arXiv:1703.05192*, 2017.

Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 35–51, 2018.

Hsin-Ying Lee, Hung-Yu Tseng, Qi Mao, Jia-Bin Huang, Yu-Ding Lu, Maneesh Singh, and Ming-Hsuan Yang. Drit++: Diverse image-to-image translation via disentangled representations. *International Journal of Computer Vision*, pp. 1–16, 2020.

Yingzhen Li and Stephan Mandt. Disentangled sequential autoencoder. *arXiv preprint arXiv:1803.02991*, 2018.

Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in neural information processing systems*, pp. 700–708, 2017.

Xinrui Lyu, Matthias Hueser, Stephanie L Hyland, George Zerveas, and Gunnar Rätsch. Improving clinical predictions through unsupervised time series representation learning. *arXiv preprint arXiv:1812.00490*, 2018.

Qianli Ma, Jiawei Zheng, Sen Li, and Gary W Cottrell. Learning representations for time series clustering. In *Advances in neural information processing systems*, pp. 3781–3791, 2019.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Qi Mao, Hsin-Ying Lee, Hung-Yu Tseng, Siwei Ma, and Ming-Hsuan Yang. Mode seeking generative adversarial networks for diverse image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1429–1437, 2019.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Takeru Miyato and Masanori Koyama. cgans with projection discriminator. *arXiv preprint arXiv:1802.05637*, 2018.

O. Mogren. C-rnn-gan: Continuous recurrent neural networks with adversarial training. *arXiv preprint arXiv:1611.09904*, 2016.

Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pp. 2234–2242, 2016.

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pp. 843–852, 2015.

H Suresh, P Szolovits, and M Ghassemi. The use of autoencoders for discovering patient phenotypes. arxiv [cs. lg] 2017.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8798–8807, 2018.

J. Yoon, D. Jarrett, and M. van der Schaar. Time-series generative adversarial networks. In *Advances in Neural Information Processing Systems*, pp. 5508–5518, 2019.

Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 5907–5915, 2017.

Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1947–1962, 2018.

Peng Zhou, Lingxi Xie, Xiaopeng Zhang, Bingbing Ni, and Qi Tian. Searching towards class-aware generators for conditional generative adversarial networks. *arXiv preprint arXiv:2006.14208*, 2020.