Mixture Density Conditional Generative Adversarial Network Models (MD-CGAN)

Jaleh Zand and Stephen Roberts

***Abstract*—Generative Adversarial Networks (GANs) have gained significant attention in recent years, with particularly impressive applications highlighted in computer vision. In this work, we present a Mixture Density Conditional Generative Adversarial Model (MD-CGAN), where the generator is a Gaussian mixture model, with a focus on time series forecasting. Compared to examples in vision, there have been more limited applications of GAN models to time series. We show that our model is capable of estimating a probabilistic posterior distribution over forecasts and that, in comparison to a set of benchmark methods, the MD-CGAN model performs well, particularly in situations where noise is a significant in the time series. Further, by using a Gaussian mixture model that allows for a flexible number of mixture coefficients, the MD-CGAN offers posterior distributions that are non-Gaussian.**

I. Introduction

Generative Adversarial Networks (GANs) have been one of the many breakthroughs in Deep Learning methods in recent years. Several different variations of the model have been introduced since the method was first introduced in [2]. One of the most popular variations of the work are Conditional Generative Adversarial Networks (CGAN) [3] in which the generator and discriminator are both conditioned on some observed information. Within time series forecasting, future values are conditioned on information observed from the past - either from the time series itself, exogenous data or a combination of the two. This makes the CGAN approach particularly useful for time series prediction.

Most applications of (C)GANs have been within computer vision and, to a lesser extent, in natural language processing and there has been some use of GAN models for simulations in a variety of settings, [7], [8], and [9].

The literature on the application of GAN models to problems associated with time series is, to date, limited. However, some work shows the potential usefulness of the method. For example, [4] apply a recurrent GAN to generate realistic synthetic medical data series. In [5] a GAN is used to forecast high-frequency stock datasets, and [6] use GANs to generate missing values for incomplete time series.

In this work, we present a method that expands on the CGAN algorithm. In our model, the generator estimates a multimodal posterior distribution, via a finite mixture of Gaussians. Unlike most variations of GAN models, in which the generator makes a point estimate, the MD-CGAN is capable of estimating a flexible probability distribution. This paper is set out as follows: in Section II we present the structure of the MD-CGAN model. In Section III we test the model on various datasets and discuss the results. Finally, in Section IV, we conclude.

April 9, 2020

The authors are with the Machine Learning Research Group and the Oxford-Man Institute of Quantitative Finance, University of Oxford, Oxford, UK. (e-mail: jz@robots.ox.ac.uk, sjrob@robots.ox.ac.uk).

II. The MD-CGAN Model Framework

We consider a time series, y_t. Our aim is to infer the posterior over some $y_{t'>t}$, conditioned on a set of observations which we denote $\mathbf{x}_t$. In order to form the posterior distribution we model the conditional density $p(y_{t'}|\mathbf{x}_t)$ as an adversarial network. To achieve this we use a Mixture Density Network (MDN) model similar to the one presented in [1] for the generator G. The inputs to the generator network are $\mathbf{x}_t$ and $\mathbf{z}_g$, where $\mathbf{z}_g$ is a collection of samples from a normal distribution, $p(z_n)_g = \mathcal{N}(0, \mathrm{var}_{\mathrm{data}})$. The outputs of $G_{t'}(\mathbf{x}_t, \mathbf{z}_g)$ are the parameters of the Gaussian mixture model, with mixing, s.d. and mean for the i-th component α_i, σ_i, and μ_i. As first proposed in [1], we achieve this by using latent variables $\mathbf{s} = \{\mathbf{s}_\alpha, \mathbf{s}_\sigma, \mathbf{s}_\mu\}$, conditioned on the inputs. The mapping from $[\mathbf{x}_t, \mathbf{z}_g] \mapsto \mathbf{s} \mapsto \{\alpha_i, \sigma_i, \mu_i\}$ is modelled via our network. As the mixings must satisfy $\sum_i \alpha_i = 1$, we map $\mathbf{s}_\alpha$ to α via the *softmax* function. The elements of σ are strictly positive so we adopt, $\sigma_i = \exp(s_{\sigma,i})$. Finally the means can be mapped directly from the latent variables, hence $\mu_i = s_{\mu,i}$.

The above formalism allows us to directly model the predictive likelihood conditioned on an input, and the likelihood of G, conditioned on the observations $\mathbf{x}_t$ and samples $\mathbf{z}_g$ as:

$$\mathcal{L}(G_{t'}(\mathbf{x}_t, \mathbf{z}_g)) = \sum_{i=1}^{m} \alpha_i(\mathbf{x}_t, \mathbf{z}_g)\mathcal{N}_i(y_{t'}|\mu_i(\mathbf{x}_t, \mathbf{z}_g), \sigma_i(\mathbf{x}_t, \mathbf{z}_g))$$

where m is the number of mixture components.

As in the CGAN model, the discriminator, D, is also conditioned on $\mathbf{x}_t$. The input to the discriminator model is, by design, $\mathbf{x}_t\sqrt{2\pi}\sigma_a\mathcal{L}(y_{t'})$, where σ_a is the s.d. of the true y_t in the MDN model and the output is $\mathbf{x}_t$. For true values of $y_{t'}$, $\sqrt{2\pi}\sigma_a\mathcal{L}(y_{t'})$ is maximized. The generator tries to 'fool' the discriminator by generating $G_{t'}$ such that the $\sqrt{2\pi}\sigma_a\mathcal{L}(G_{t'})$ is maximized. The loss function for the generator, L_G in Equation 1, reflects this concept. The discriminator network, on the other hand, tries to differentiate between true $y_{t'}$ values and the pseudo-values created by the generator. The loss function for the discriminator, L_D in Equation 2, reflects this, where the lowest value is achieved when $\sqrt{2\pi}\sigma_a\mathcal{L}(y_{t'})$ is maximal (unity) and $\mathcal{L}(G_{t'}(\mathbf{x}_t, \mathbf{z}_g))$ is minimal (zero).

$$L_G = \mathbb{E}_{z\sim P_z(z)}[-\mathcal{L}(G_{t'}(\mathbf{x}_t, \mathbf{z}_g))] \tag{1}$$

$$L_D = \mathbb{E}_{y\sim P_{\text{data}}(y)}[\|\mathbf{x}_t\sqrt{2\pi}\sigma_a\mathcal{L}(y_{t'}) - \mathbf{x}_t\|^2] + \mathbb{E}_{z\sim P_z(z)}[\|\mathbf{x}_t\sqrt{2\pi}\sigma_a\mathcal{L}(G_{t'}(\mathbf{x}_t, \mathbf{z}_g))\|^2] \quad (2)$$

The algorithm follows the steps in Algorithm 1.

Algorithm 1 MD-CGAN Algorithm

1: **for** number of training iterations **do**
2: **for** j steps do **do**
3: Sample N noise samples, $\{\mathbf{z}^1,...,\mathbf{z}^N\}$ from $p_g(\mathbf{z})$
4: Sample N data points, $\{\mathbf{x}^1,...,\mathbf{x}^N\}$ from $p_{\text{data}}(\mathbf{x})$
5: Update the discriminator by descending its stochastic gradient:

$$\nabla_{\theta_\mathcal{L}} \sum_{n=1}^{N}[\|\mathbf{x}^{(n)}\sqrt{2\pi}\sigma_a\mathcal{L}(y^{(n)}) - \mathbf{x}^{(n)}\|^2 + \|\mathbf{x}^{(n)}\sqrt{2\pi}\sigma_a\mathcal{L}(G(\mathbf{z}^{(n)}, \mathbf{x}^{(n)}))\|^2]$$

6: **end for**
7: Sample N noise samples, $\{\mathbf{z}^1,...,\mathbf{z}^N\}$ from $p_g(\mathbf{z})$
8: Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \sum_{n=1}^{N} -\mathcal{L}(G(\mathbf{z}^{(n)}, \mathbf{x}^{(n)}))$$

9: **end for**

Figure 1 illustrates the structure and the interaction between the generator and the discriminator for the MD-CGAN model.

III. Experimental Results

A. Comparison with other Learning Models

We compare the MD-CGAN model to the Mixture Density Network model (MDN) [1], the CGAN model [3] and a standard neural network (SNN).

We perform experiments on four datasets, the Mackey-Glass chaotic dataset, sunspot dataset ([11]), US initial jobless claims (USIJC, weekly intervals, [12]), and the EURUSD foreign exchange daily rates (EURUSD FX rate). For consistency, the generator in the MD-CGAN, has the same structure as the MDN, the generator of the CGAN and the SNN for our experiments. For each dataset, the series is split into training and out-of-sample test sets. The training data sets in all our experiments comprise 2000 samples (save for the sunspot dataset which is 1000 data points) and test sets consist of 400 data points post the training set. All algorithms have as input the last k data points, set to $k = 5$ for the purpose of our experiments; this value is not optimized, however, and is chosen to allow simple comparisons across methods. All data sets are normalized, to the [0,1] interval, again to allow for simpler comparison across data and methods. We further note that both CGAN and SNN make *point estimate* predictions, whilst MD-CGAN and MDN estimate posterior *distributions*. To enable a simple comparison, we therefore report the mean-square error (MSE) for all methods. The number of mixture components, m, is set to unity (we vary this in Section III-D), to further enable comparison. The most-likely value (which for $m = 1$ is the mean) of the predictive distribution is taken as the forecast value for both the MDN and MD-CGAN models.

B. Data & one-step forecasts

Mackey-Glass and Sunspot time series: We start our experiments looking at one-step ahead forecasts on two well known data sets, the Mackey-Glass chaotic time series [10] and the Sunspot data set [11]. One-step forecast error comparisons are indicated in Table I. We note that both GAN models (CGAN and MD-CGAN) our outperformed, in terms of MSE, for these datasets. We attribute this to the high signal to noise associated with these data sets and thus the low levels of observed noise in the test data. In these circumstances, the 'adversarial advantage' [3] of GAN approaches, which offers robustness to input perturbations (in this case noise), is less important.

In the next experiments we thus add (30% by amplitude) normally distributed noise to the test data (from a GAN perspective, these input perturbations are, in effect, treated as adversarial attacks). We note that no noise is added to the training dataset. MSE errors are presented in Table I, and referenced as "Mackey-Glass with Noise" and "Sunspot with Noise" respectively. We note that, under this noisy-observation paradigm, GAN models perform best, confirming adversarial robustness of the GAN models.

US initial jobless claims (USIJC) and Euro-Dollar foreign exchange (EURUSD FX) daily rate: Financial times series are highly stochastic, and we expect GAN approaches to be well-suited to forecasting in these circumstances. We consider two financial time series, namely the USIJC and EURUSD FX datasets. Test set forecasts are shown in Figure 2 and the associated MSEs in Table I. We note that, for these data sets, the GAN approaches perform well and our method, MD-CGAN has lowest errors.

C. Financial forecasts over longer-horizons

One-step forecasts were presented in Subsection III-A. Here we extend analysis of the financial data over longer horizons. All models were used to make estimates over a horizon of ten weeks for an extended set of financial time series; the USIJC, EURUSD FX rate (as previous), WTI crude oil spot prices (WTI, [13]), Henry Hub Natural Gas spot prices (Nat Gas, [13]), and the CBOE Volatility Index (VIX [14]). A ten-week horizon represents a 50 step forecast for the daily datasets (FX, WTI, Nat Gas & VIX) and 10 steps for the weekly USIJC dataset. Further we perform comparisons against standard econometric linear models, namely a 5-th order autoregressive, AR(5), model and the martingale, or AR(0) model, in which the forecast is the last observed datum. Taking the martingale model as a baseline, we present in Table II the mean-square errors as the ratio to the martingale model error. We note that the MD-CGAN approach delivers ratios below unity and provides the lowest error of all models in this scenario.

D. Multi-modal posterior predictions

Finally, we compare the performance of MD-CGAN over varying numbers of mixture components. In all the previous experiments we set $m = 1$ (hence the model produced a single predictive Gaussian posterior). Here we briefly present the

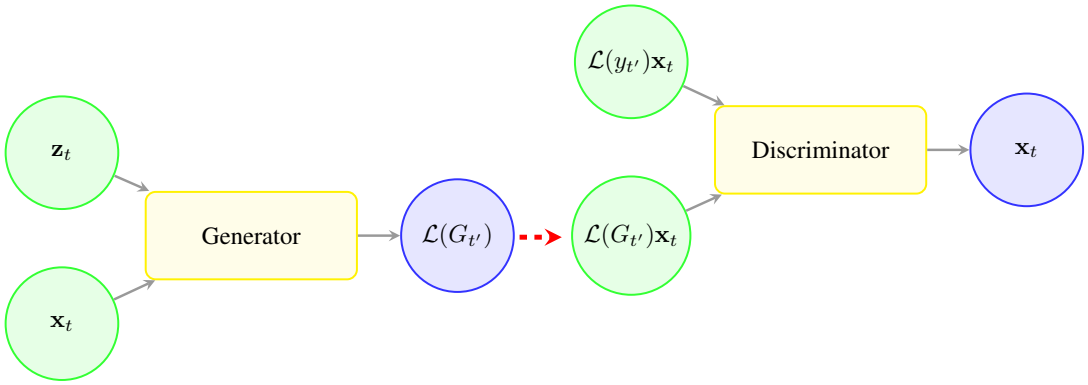

Fig. 1. Schematic of the MD-CGAN model showing Generator and Discriminator components and associated variables.

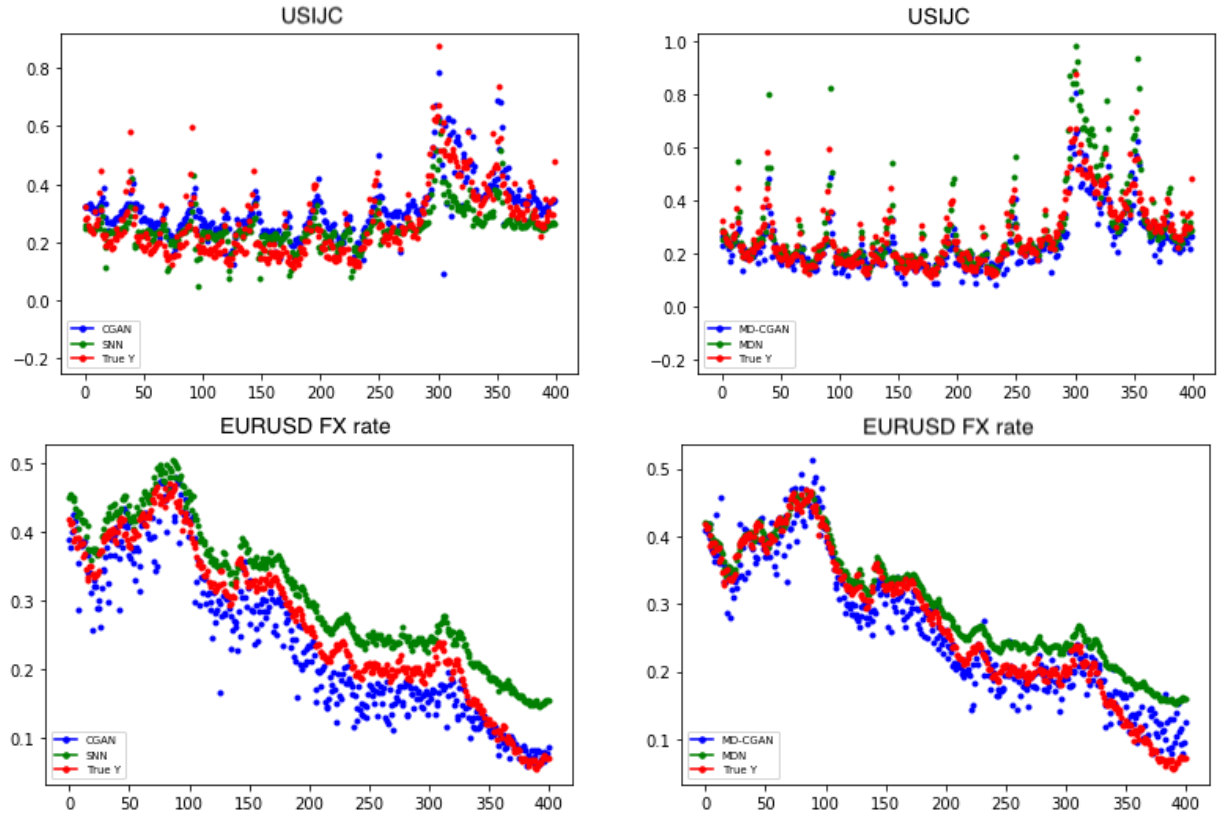

Fig. 2. Comparative forecasts over USIJC and EURUSD FX time series. Left plots: Comparison of CGAN (blue) & SNN (green) predictive means and true data (red). Right plots: Comparison of MD-CGAN (blue) & MDN (green) predictive means and true data (red).

results for the five finance datasets with $m \in \{1, 2, 3\}$. We report (negative) log-likelihood measures (as we do not compare against point-value models in this section) and consider one-step forecasts on all the data sets. Table III presents the performance across data sets for varying numbers of mixture components in the posterior prediction. We note performance improvement for some datasets for $m > 1$. We note that we do not attempt to infer m, though choosing its value based on performance on a set of cross-validation data would be an option, as would enforcing regularization over the mixture model posterior, through more extensive use of Bayesian inference. We leave these extensions for future research.

IV. Conclusion

In this paper we presented the MD-CGAN model an extension of the CGAN [3] method. In the experiments considered, we find the MD-CGAN outperforms all other models on the noisy Mackey-Glass and Sunspot datasets as well as all financial time series, for both short and long term forecast horizons. As a GAN model, our approach retains adversarial robustness, most notable when noise is extensively present in data, making the approach particularly well suited to dealing with financial data. Furthermore, our MD-CGAN model can effectively estimate a flexible posterior distribution, in contrast to standard GAN models. Exploiting the rich, multi-model, posterior distribution is not reported in detail here but

TABLE I
MSEs FOR ALL EXPERIMENTS, WITH STANDARD DEVIATIONS IN BRACKETS.

	Mackey-Glass	**Sunspot**	**Mackey-Glass with Noise**	**Sunspot with Noise**	**USIJC**	**EURUSD FX rate**
SNN	0.0014 *(0.0020)*	0.0154 *(0.0260)*	0.1640 *(0.1939)*	0.0536 *(0.0777)*	0.0070 *(0.0154)*	0.0022 *(0.0017)*
CGAN	0.0036 *(0.0048)*	0.0196 *(0.0358)*	0.0360 *(0.0525)*	0.0266 *(0.0380)*	0.0074 *(0.0157)*	0.0018 *(0.0026)*
MDN	**0.0002** ***(0.0003)***	**0.0105** ***(0.0239)***	0.1402 *(0.1597)*	0.0758 *(0.1144)*	0.0080 *(0.0238)*	0.0016 *(0.0020)*
MD-CGAN	0.0026 *(0.0030)*	0.0172 *(0.0293)*	**0.0264** ***(0.0392)***	**0.0203** ***(0.0359)***	**0.0041** ***(0.0089)***	**0.0008** ***(0.0011)***

TABLE II
RATIO OF MODEL MSE TO MARTINGALE BASELINE MODEL OVER 10-WEEK FORECAST HORIZON.

	USIJC	**EURUSD FX rate**	**WTI**	**Nat Gas**	**VIX index**
AR(5)	0.78	1.91	0.85	1.01	0.71
SNN	0.79	1.25	0.89	0.94	0.71
CGAN	0.77	0.85	1.53	1.07	0.91
MDN	0.84	3.48	1.48	1.13	0.77
MD-CGAN	**0.73**	**0.76**	**0.80**	**0.82**	**0.66**

TABLE III
NEGATIVE LOG-LIKELIHOOD VARIATION WITH m.

	USIJC	**EURUSD FX rate**	**WTI**	**Nat Gas**	**VIX index**
$m = 1$	-1.01	**-1.79**	**-1.75**	-1.28	**-1.50**
$m = 2$	-1.05	-0.98	-1.24	**-1.33**	-1.39
$m = 3$	**-1.09**	-0.67	-1.33	-1.25	-1.48

will feature in follow-up work. In summary, the MD-CGAN model combines the advantageous features of both probabilistic forecasting and GAN methods. We see this as a particularly useful approach for dealing with time series in which noise is significant (such as in financial data) and for providing robust, long-term forecasts beyond simple point estimates.

REFERENCES

[1] Bishop, Christopher M. (2006) Pattern Recognition and Machine Learning, *Springer-Verlag*, ISBN 0387310738.

[2] Goodfellow, Ian and Pouget-Abadie, Jean and Mirza, Mehdi and Xu, Bing and Warde-Farley, David and Ozair, Sherjil and Courville, Aaron and Bengio, Yoshua (2014) Generative adversarial nets. *Advances in neural information processing systems*, 2672-2680.

[3] Mirza, Mehdi and Osindero, Simon (2014) Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*

[4] Esteban, Cristóbal and Hyland, Stephanie L and Rätsch, Gunnar (2017) Real-valued (medical) time series generation with recurrent conditional GANs. *arXiv preprint arXiv:1706.02633*

[5] Zhou, Xingyu and Pan, Zhisong and Hu, Guyu and Tang, Siqi and Zhao, Cheng (2018) Stock Market Prediction on High-Frequency Data Using Generative Adversarial Nets. *Mathematical Problems in Engineering*, 10.1155/2018/4907423

[6] Luo, Yonghong and Cai, Xiangrui and Zhang, Ying and Xu, Jun and Xiaojie, Yuan (2018) Multivariate Time Series Imputation with Generative Adversarial Networks. *Advances in Neural Information Processing Systems*, 1596-1607.

[7] Wu, Haimeng and Gu, Bowen and Wang, Xiang and Pickert, Volker and Ji, Bing (2019) Design and control of a bidirectional wireless charging system using GAN devices. *2019 IEEE Applied Power Electronics Conference and Exposition (APEC).*

[8] Hodge, John A and Mishra, Kumar Vijay and Zaghloul, Amir I (2019) Joint multi-layer GAN-based design of tensorial RF metasurfaces. *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)* 1-6.

[9] Barth, Christopher B and Assem, Pourya and Foulkes, Thomas and Chung, Won Ho and Modeer, Tomas and Lei, Yutian and Pilawa-Podgurski, Robert CN (2019) Design and control of a GaN-based, 13-level, flying capacitor multilevel inverter. *IEEE Journal of Emerging and Selected Topics in Power Electronics*.

[10] Mackey, M. C. and Glass, L. (1977). Oscillation and chaos in physiological control systems. *Science*, 197(4300):287-289.

[11] Clette, Frédéric & others (2015) WDC-SILSO. *http://www.sidc.be/silso/*.

[12] Bureau of Labor Statistics (2018) United States Department of Labor. *https://www.bls.gov/*.

[13] U.S. Energy Information Administration (2020) *https://www.eia.gov/*.

[14] CBOE Volatility Index (2020), Historical Data, *Yahoo!Finance. https://finance.yahoo.com/quote/%5EVIX/history?p=%5EVIX*.