IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 28, 2019, accepted April 15, 2019, date of publication May 17, 2019, date of current version May 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2915544

Probabilistic Forecasting of Sensory Data With Generative Adversarial Networks – ForGAN

ALIREZA KOOCHALI[1,2], PETER SCHICHTEL[2], ANDREAS DENGEL[1,3], AND SHERAZ AHMED[3]

[1]Department of Computer Science, University of Kaiserslautern, 67663 Kaiserslautern, Germany
[2]Ingenieurgesellschaft Auto und Verkehr (IAV), 67663 Kaiserslautern, Germany
[3]German Research Center for Artificial Intelligence (DFKI), 67663 Kaiserslautern, Germany

Corresponding author: Alireza Koochali (akoochal@ rhrk.uni-kl.de)

This work was supported by IAV FLaP which is a joint-lab between IAV GmbH and German Research Center for Artificial Intelligence (DFKI GmbH).

ABSTRACT Time series forecasting is one of the challenging problems for humankind. The traditional forecasting methods using mean regression models have severe shortcomings in reflecting real-world fluctuations. While new probabilistic methods rush to rescue, they fight with technical difficulties like quantile crossing or selecting a prior distribution. To meld the different strengths of these fields while avoiding their weaknesses, as well as, to push the boundary of the state-of-the-art, we introduce ForGAN – one step ahead probabilistic forecasting with generative adversarial networks. ForGAN utilizes the power of the conditional generative adversarial network to learn the data generating distribution and compute probabilistic forecasts from it. We argue how to evaluate ForGAN in opposition to regression methods. To investigate probabilistic forecasting of ForGAN, we create a new dataset and demonstrate our method abilities on it. This dataset will be made publicly available for comparison. Furthermore, we test ForGAN on two publicly available datasets, namely Mackey-Glass dataset and Internet traffic dataset (A5M), where the impressive performance of ForGAN demonstrate its high capability in forecasting future values.

INDEX TERMS Time-series, generative adversarial networks, forecasting, probabilistic, prediction.

I. INTRODUCTION

At its core, life is about decision making. Decision making always depends on our perspective of the future. Therefore, the forecast of what might lay before us is one of the most intriguing challenges for humankind. It is no surprise that there is a huge and diverse community concerned with forecasting and decision making. To name, but a few, there is weather and climate prediction [3], [4], flood risk assessment [5], seismic hazard prediction [6], [7], predictions about the availability of (renewable) energy resources [8], [9], economic and financial risk management [10], [11], health care [12]–[14], predictive and preventative medicine [15] and many more.

Since the forecast is the prediction of future values, we can take the predictive view of regression to provide a solution to this problem [16]. The ultimate goal of regression analysis is to obtain information about the conditional distribution of a response given a set of explanatory variables [17]. In the context of forecasting, the ultimate goal is obtaining the predictive probability distribution over future quantities or events of interest. In other words, given the time-dependent observable of interest x, the goal is to acquire $\rho(x_{t+1}|\{x_t, \ldots, x_0\})$. Unfortunately, the ultimate goal is seldom achieved and most of the methods focus on only one designated quantity of the response distribution, namely the mean. The mean regression models have the advantage of being easy to understand and predict however they often lead to incomplete analyses when more complex relationships are presented and also bears the risk of false conclusions about the significance/importance of covariates [18]. In the following sections, we review mean regression forecast methods briefly and then we provide an overview about scientific endeavors on probabilistic forecasting. Note that in this paper, we call the history of events $\{x_t, \ldots, x_0\}$ the condition c and we use x_{t+1} as the notion for the value of the next step i.e. target value.

The associate editor coordinating the review of this manuscript and approving it for publication was Chenping Hou.

A. MEAN REGRESSION FORECASTING

Mean regression forecasting is concerned with predicting $\mu(\rho(x_{t+1}|c))$ most accurately. There is a broad range of mean regression methods available in literature e.g., statistical methods (like ARMA and ARIMA [19] and their variants), machine learning based methods (like

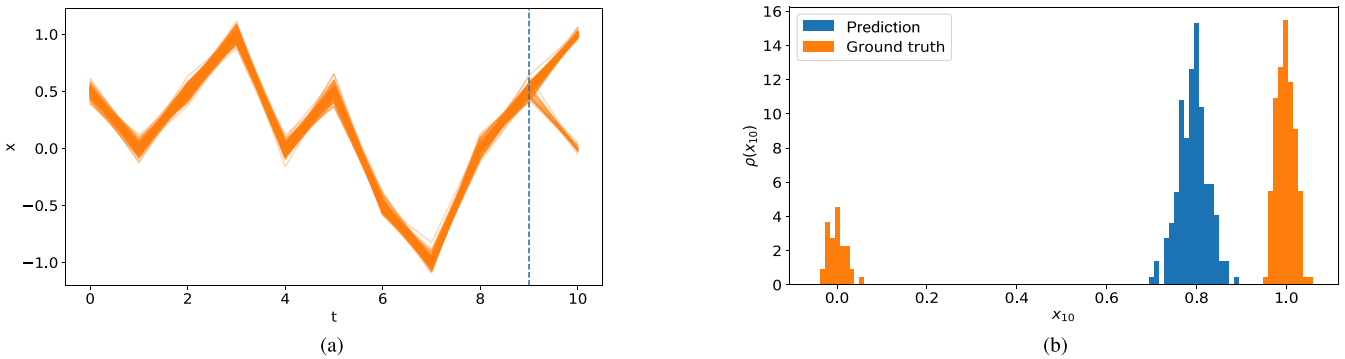

FIGURE 1. (a) Dataset used for training mean regression and (b) visualization of the results.

Support Vector Machines (SVM) [20]–[25], Evolutionary Algorithms (EA) [24]–[30] and Fuzzy Logic Systems (FLS) [29]–[35]), and Artificial Neural Network based methods (ANN) [36]–[39]. These methods use handcrafted features on the data except ANNs which try to automatically extract those features using an end-to-end pipeline. As these methods forecast the future following the principles of mean regression, all of them inherit the main problem/limitation of these principles, i.e. they do not include the fluctuations around the mean value. Hence, their results can be unreliable and misleading in some cases. Fig. 1 presents an example of the problem inherent in all mean regression based methods. It shows a cluster of time series with identical, but noisy, time window $c = \{x_9, \ldots, x_0\}$ and the future value at $t = 10$ (to be found right of the blue dashed line) which can take two distinctive realizations: in 80% of the cases x_{10} yields one while in 20% of the cases it yields zero.[1]

To demonstrate the problem, we train a simple neural network to forecast x_{10}. We present the result in Fig. 1b. It illustrates that the regression model fails to model the data. The best answer we can get from mean regression will converge to 0.8, the weighted average of all possible values for x_{10}. We can observe from Fig. 1b that the values forecasted with mean regression do not have any overlap with ground truth. It indicates mean regression is incapable of predicting any ground truth value precisely and it cannot be improved any further. Note, this does not imply that the mean regression method does not work. The closer the target distribution approximates a Dirac delta distribution, the better and more accurate a mean regression forecast will be. It is upon the researcher to evaluate this constraint carefully.

B. PROBABILISTIC FORECASTING

To solve the shortcomings associated with mean regression, recently many researchers presented solutions which are moving from mean regression to probabilistic forecasting. Probabilistic forecasting serves to quantify the variance in a prediction [16]. Different approaches have been proposed to undertake probabilistic forecasting in various fields [40]–[55]. In the twentieth century, Stigler [56] coined the idea of the transition from point estimation to distribution estimation. However, the shift toward applying probabilistic forecasting on real-world problems did not take place until recent years. Two of the most prominent approaches in these fields are conditional quantile regression and conditional expectile regression. Quantile regression is a statistical technique intended to estimate, and conduct inference about, conditional quantile functions [57]. To estimate the regression coefficients from training data, one uses the asymmetric piecewise linear scoring function, which is consistent for the α-quantile [57], [58]. Expectile regression works similarly but it is based on the asymmetric piecewise quadratic scoring function [59]–[61]. While these methods push regression methods beyond the mean regression, the problem of crossing quantile curves is frequently observed especially when considering a dense set of quantiles or using small dataset [18]. Various methods have been proposed in the literature to overcome this problem [62], [63] but they always require additional efforts and are not always applicable [18]. Furthermore, one can use a collection of point forecasts for a specific quantity or event as an ensemble model for probabilistic forecasting. In this setup, we need some form of statistical post-processing [16]. State-of-the-art techniques for statistical post-processing include the non-homogeneous regression (NR) or ensemble model output statistics (EMOS) technique proposed by Gneiting et al. [41] and the ensemble Bayesian model averaging (BMA) approach developed by Raftery et al. [64]. For an in-depth review of probabilistic forecasting, please refer to [16].

Besides these methods, researchers employ Bayesian probability theory to provide approaches for probabilistic forecasting. Bayesian probability theory offers mathematically grounded tools to reason about model uncertainty, but these usually come with a prohibitive computational cost [65]. This computation complexity stems from marginalizing, a computationally intensive integration required by Bayesian models which makes the computation for complex models impossible. To solve this problem, many approximate integration algorithms have been developed, including Markov chain Monte Carlo (MCMC) methods, variational

[1] If you like you may imagine this as an experiment on a chaotic system.

approximations, expectation propagation, and sequential Monte Carlo [66]–[69]. However, this method still suffers from a prohibitive computational cost, rapid growth in the number of parameters and time-intensive convergence [65]. Furthermore, the success of Bayesian model heavily relies on selecting a prior distribution. Selecting a suitable prior distribution is a delicate task which requires insight into the data. Recently, Gal et al. [65] use dropout [70] layers for probabilistic machine learning. While dropout is used in many models in deep learning as a way to avoid over-fitting, Gal has shown in his paper that a neural network with arbitrary depth and non-linearities, with dropout applied before every weight layer, is mathematically equivalent to an approximation to the probabilistic deep Gaussian process [71]. For an in-depth review of Bayesian probabilistic machine learning, please refer to [72], [73].

C. GENERATIVE ADVERSARIAL NETWORKS

Generative Adversarial Network (GAN) [74] is a new type of neural networks which enables us to learn an unknown probability distribution from samples of the distribution. GAN can learn the probability distribution of a given dataset and generate synthetic data which follows the same distribution. As a result, they are capable of synthesizing artificial data which looks realistic. While GANs were originally proposed to solve the problem of data scarcity, its promising results have drawn a lot of attention in the research community and many interesting derivations, extensions, and applications have been proposed for GANs [75]–[79]. Unfortunately, despite their remarkable performance, evaluating and comparing GANs is notoriously hard. Thus, the application of GANs is limited to the domains where the results are intuitively assessable like image generation [75], music generation [80], voice generation [81], and text generation [82].

Diverse approaches have been proposed for probabilistic forecasting. However, each of these methods has limitations which prevent them from becoming canonical approach in the industry. Hence, mean regression methods are widely employed by industry section with their critical shortcomings. As mentioned before, GANs are a powerful method for learning probability distributions. In this paper, we exploit the potentials of GANs to learn full probability distribution of future values without the restrictions of the aforementioned methods. We introduced ForGAN, a conditional GAN [79] for probabilistic forecasting. The main contributions of this paper are as follows:

- We propose ForGAN, a novel approach to employ a conditional GAN for forecasting future value. Our method can learn the full conditional probability distribution of future values even in complex situations without facing conventional problems of probabilistic forecasting methods such as quantile crossing or dependency on the chosen prior distribution.
- We conduct various experiments to investigate the predictive capabilities of our method and compare it with the state-of-the-art methods as well as a conventional regression neural network model with a similar structure to ForGAN. Our method outperforms its counterparts on various metrics.
- We introduce a new dataset for later reference and comparison.

II. RELATED WORK

Lately, GANs have been applied to various problems in the sequential data domain and achieved remarkable results. In this section, we give a brief overview of studies related to our work.

Most research regarding applying GAN on sequential data is concerned with discrete problems, e.g. text generation task. Since the discrete space of words cannot be differentiated in mathematics, modifying a GAN to work with discrete data is a challenging task. Many papers have been published to address this problem and they have reported remarkable results [82]–[85]. However, we are interested in the (quasi) continues regime. Therefore, these techniques are not directly applicable here.

In the continuous regime, we find GANs being utilized to generate auditory data. C-RNN-GAN [80] works on music waveforms as continuous sequential data to generate polyphonic music. This GAN uses Bidirectional LSTM in the structure of the generator and discriminator. Moreover, there are many other studies on auditory data which work on audio spectrograms and consider them as 2D images. For instance, Donahue et al. [86] as well as Michelsanti, Tan et al. [87] employ GAN on audio spectrograms for speech enhancement. Fan et al. [88] propose a GAN for separating the singing voice from background music. Donahue et al. [89] propose a GAN for synthesizing raw-waveform audio and Gao et al. [81] employ GAN for synthesizing of impersonated voices. However, contrary to our work these studies are first not concerned with forecasting and second, the results are intuitive. The latter point is important as analogous to the image domain, music can be judged by listening to it.

Since there is no consensus on a process for evaluating GANs, application of GANs beyond text and auditory data is a very challenging task. We found a few attempts on the application of GANs beyond these data types. Hyland and Esteban [90] propose RGAN and RCGAN to produce realistic real-valued multi-dimensional medical time series. Both of these GANs employ LSTM in their generator and discriminator while RCGAN uses Conditional GAN instead of Vanilla GAN to incorporate a condition in the process of data generation. They also describe novel evaluation methods for GANs, where they generate a synthetic labeled training dataset and train a model using this set. Then, they test this model using real data. They repeat the same process using a real train set and synthetic labeled test set. GAN-AD [91] is proposed to model time-series for anomaly detection in Cyber-Physical Systems (CPSs). This GAN uses LSTM in both generator and discriminator, too. Zhang et al. [92] propose a conditional GAN for generating synthetic time-series

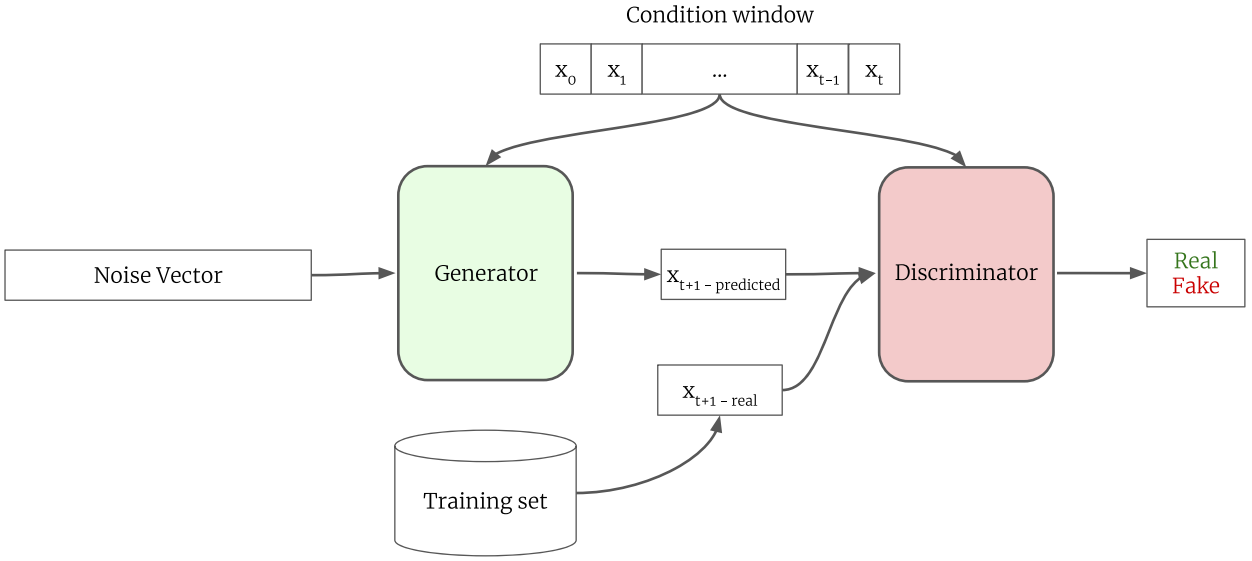

FIGURE 2. Overview of proposed ForGAN architecture. The condition *c* is handed to generator *G* and discriminator *D*.

in smart-grids. Unlike previous work, this GAN employs CNN to construct generator and discriminator.

To the best of our knowledge, this is for the first time that a GAN is employed for the forecasting task. Our work is analogous to RCGAN [90], however, we pursue a different goal. As a result, we need to take a different approach to train and evaluate the performance of ForGAN.

III. METHODOLOGY

A. GENERATIVE ADVERSARIAL NETWORK (GAN)

Generative Adversarial Networks (GANs) [74] are a class of algorithms for modeling a probability distribution given a set of samples from the data probability distribution ρ_{data}. A GAN consists of two neural networks namely **generator** G and **discriminator** D. These components are trained simultaneously in an adversarial process. First, a noise vector z is sampled from a known probability distribution $\rho_{\text{noise}}(z)$ (normally a Gaussian distribution). G takes the noise vector z as an input and trains to generate a sample whose distribution follows ρ_{data}. On the other hand, D is optimized to distinguish between generated data and real data. In other words, D and G play the following two-player minimax game with value function $V(G, D)$:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x\sim\rho_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim\rho_{\text{noise}}(z)}[\log(1 - D(G(z)))]\,.$$

While training the GAN, generator G learns to transform a known probability distribution ρ_z to the generators distribution ρ_G which resembles ρ_{data}.

B. CONDITIONAL GAN (CGAN)

Conditional GAN (CGAN) [79] is an extension of GAN which enables us to condition the model on some extra information y. This could be any kind of auxiliary information, such as class labels or data from other modalities. We can perform the conditioning by feeding y into both the discriminator and generator as additional input layer. The new value function $V(G, D)$ for this setting is:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x\sim\rho_{\text{data}}(x)}[\log D(x|y)] + \mathbb{E}_{z\sim\rho_z(z)}[\log(1 - D(G(z|y)))]\,.$$

C. PROBABILISTIC FORECASTING WITH CGAN

In this paper we aim to model the probability distribution of one step ahead value x_{t+1} given the historical data $c = \{x_0, .., x_t\}$, i.e. $\rho(x_{t+1}|c)$. We employ CGAN to model $\rho(x_{t+1}|c)$. Figure 2 presents an overview of ForGAN. The historical data is provided to generator and discriminator as condition. The generator takes the noise vector which is sampled from a Gaussian distribution with mean 0 and standard deviation 1 and forecasts x_{t+1} with regard to the condition window c. The discriminator takes the x_{t+1} and inspects whether it is a valid value to follow c or not. Hence, the ForGAN value function is:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x_{t+1}\sim\rho_{\text{data}}(x_{t+1})}[\log D(x_{t+1}|c)] + \mathbb{E}_{z\sim\rho_z(z)}[\log(1 - D(G(z|c)))]$$

By training this model, the optimal generator models the full probability distribution of x_{t+1} for a given condition window. With having the full probability distribution in hand, we can extract information regarding any possible outcome and the probability of their occurrence by sampling.

D. ARCHITECTURE

We use one of the members of RNN family as the main component of both generator and discriminator. We select between LSTM or GRU using the procedure described in section IV-C. The generator (Fig. 3(a)) takes the condition window and passes the condition through an RNN layer to

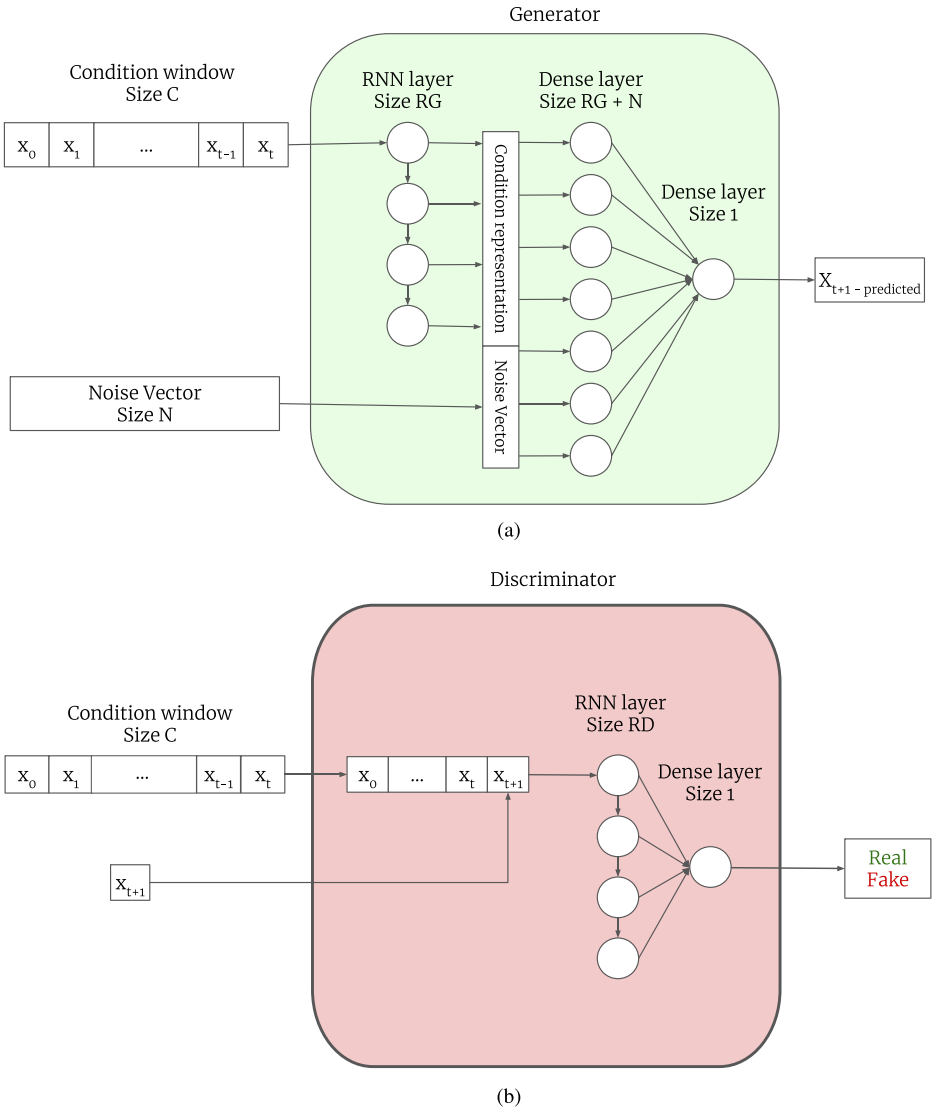

FIGURE 3. (a): The architecture of the generator in detail. The generator takes noise vector and a time windows and forecasts the value of next step (x_{t+1}). (b): The architecture of the discriminator in detail. The discriminator receives x_{t+1} and time window and determines if x_{t+1} is valid.

construct its representation. Then, it concatenates the condition representation with the noise vector and passes them through two dense layers which result in the predicted x_{t+1} value. The discriminator (Fig. 3(b)) takes x_{t+1} either from the generator or the dataset alongside the corresponding condition window and concatenates x_{t+1} at the end of the condition window to obtain $\{x_0, .., x_{t+1}\}$. The rest of the network tries to check the validity of this time window. For this purpose, it passes the obtained time window through an LSRM/GRU layer followed by a dense layer to acquire a single value which specifies the validity of the aforementioned time window.

E. G-REGRESSION MODEL

Normally, forecasting models are trained by optimizing a point-wise error metric as loss function however we employ adversarial training to train neural network for forecasting. To study the effectiveness of adversarial training in comparison to conventional training, we construct the G-regression model, a model with identical structure to generator G. To follow the conventional way of training neural networks for forecasting, we train this model by optimizing RMSE as the loss function and compare its results with ForGAN.

IV. EXPERIMENTS

To investigate the performance of ForGAN, we test our method with three experiments and later on, if applicable, compare results with the state-of-the-art methods. Let us first introduce the used datasets. Next, we elaborate on common evaluation methods for forecasting task and how we assess ForGAN to produce correct and meaningful results. We then demonstrate how we chose the particular hyperparameters for each dataset. Last but not least we elaborate on the setup of our experiments.

TABLE 1. Initial values y_0 and relative composition of our Lorenz dataset.

Index	y_0	Relative Occurrence
0	1.0001	5.5 %
1	1.000001	22 %
2	1.00000001	42 %
3	1.0000000001	24 %
4	1.000000000001	6.5 %

A. DATASETS

1) LORENZ DATASET

In the first experiment, we create a complex dataset to inspect the probabilistic forecasting capability of our method. We form a dataset which contains multiple time window clusters. Each cluster consists of similar and complex time windows generated using the Lorenz system. The Lorenz equations describe the atmospheric convection x, the horizontal temperature variation y, and the vertical temperature z as a function of time t. Using a dot for temporal derivative the system of coupled differential equations is given by

$$\begin{aligned} \dot{x} &= \sigma(y - x)\,, \\ \dot{y} &= x(\rho - z)\,, \\ \dot{z} &= y\,x - \beta\,z\,, \end{aligned} \tag{1}$$

where σ is proportional to the Prandtl number [93], ρ is proportional to the Rayleigh number [94] and β is connected to physical dimensions of the atmospheric layer of interest [95]. One of the most interesting features of the Lorenz equations is the emergence of chaotic behavior [95], [96] for certain choices of the parameters σ, ρ, and β. In the following we fix $\sigma = 16$, $\rho = 45.92$, and $\beta = 4$. Furthermore, we fix the initial conditions $x_0 = 1$ and $z_0 = 1$. To construct the dataset, first, we select five y_0 to serve as the seeds for our clusters and specify the relative occurrence of clusters as presented in Tab. 1. Then we generate 100000 data samples with the length of 26 seconds and the resolution of 0.02s using these y_0. The result is presented in Fig. 4(a). We add a Gaussian noise with mean 0 and standard deviation of 7.2 to create unique time windows while preserving similarity inside each cluster. From Fig. 4(a), we locate the bifurcation region and select the region between 12 and 17 seconds as the condition time window for training. The dataset of condition time windows is plotted in Fig. 4(b). Finally, for the target values x_{t+1}, we sample randomly from $t \in (20, 22, 25)$ which forms the probability distributions as they are presented in Fig. 4(c). Fig. 4(d) presents the full probability distribution of the x_{t+1} for the entire dataset.

2) MACKEY-GLASS DATASET

The time delay differential equation suggested by Mackey and Glass [1] has been used widely as a standard benchmark model to generate chaotic time-series for the forecasting task.

$$\dot{x} = \frac{a\,x(t-\tau)}{(1 + 10 \cdot (t-\tau)) - b\,x(t)}\,. \tag{2}$$

To make our result comparable with state-of-the-art [97], we set $a = 0.1$, $b = 0.2$ and $\tau = 17$. We generate a dataset with length 20000 using Eq.(2) for our second experiment.

3) INTERNET TRAFFIC DATASET

For our last experiment, we apply our method to a real-world problem, forecasting internet traffic. We use a dataset which belongs to a private ISP with centers in eleven European cities (which is commonly known as A5M) [2]. It contains data corresponding to a transatlantic link and was collected in 2005 from 06:57 on 7th of June to 11:17 on 29th of July.

B. EVALUATION METRICS

Commonly, in forecasting tasks, point-wise error metrics are used. To be able to compare to the state-of-the-art we report RMSE, MAE and MAPE which are related to each other by

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_i (x_i - \hat{x}_i)^2}\,, \tag{3}$$

$$\text{MAE} = \frac{1}{N}\sum_i \left|x_i - \hat{x}_i\right|\,, \tag{4}$$

$$\text{MAPE} = \frac{1}{N}\sum_i \left|10^2 \times \frac{x_i - \hat{x}_i}{x_i}\right|\,. \tag{5}$$

Here N is the number of data samples x_i, and $\hat{x}_i$ are the actual predictions. However, point-wise error metrics are not suitable for assessing distributions similarities. Since ForGAN models the full probability distribution of x_{t+1}, we are interested in measuring how accurate we managed to reproduce the data distribution. Therefore, we select the Kullback-Leibler divergence (KLD) [98] to report the performance of our method. KLD measures the divergence between two probability distributions P and Q. Since we have finite data samples and ForGAN by nature samples, we select the discrete version of KLD which is defined as:

$$\text{KL}\,(P|Q) = \sum_i P_i \log \frac{P_i}{Q_i}\,. \tag{6}$$

Note, P denotes data distribution and Q indicates prediction probability distribution. Hence, due to the appearance of Q in the denominator, if predictions distribution does not cover data distribution correctly KLD is not defined. To determine the optimal number of bins for the histogram of distribution, we follow the method suggested in [99] which aims for the optimum between shape information and noise. To evaluate our method and compare our results to the state-of-the-art in one step ahead forecasting, we train ForGAN alongside G-regression and report RMSE, MAE, MAPE, and (if possible) KLD. To compute KLD for ForGAN, we sample 100 forecast of x_{t+1} for any condition in the test set. Then, we form ForGAN's prediction probability distribution for the entire test set and calculate KLD between this distribution

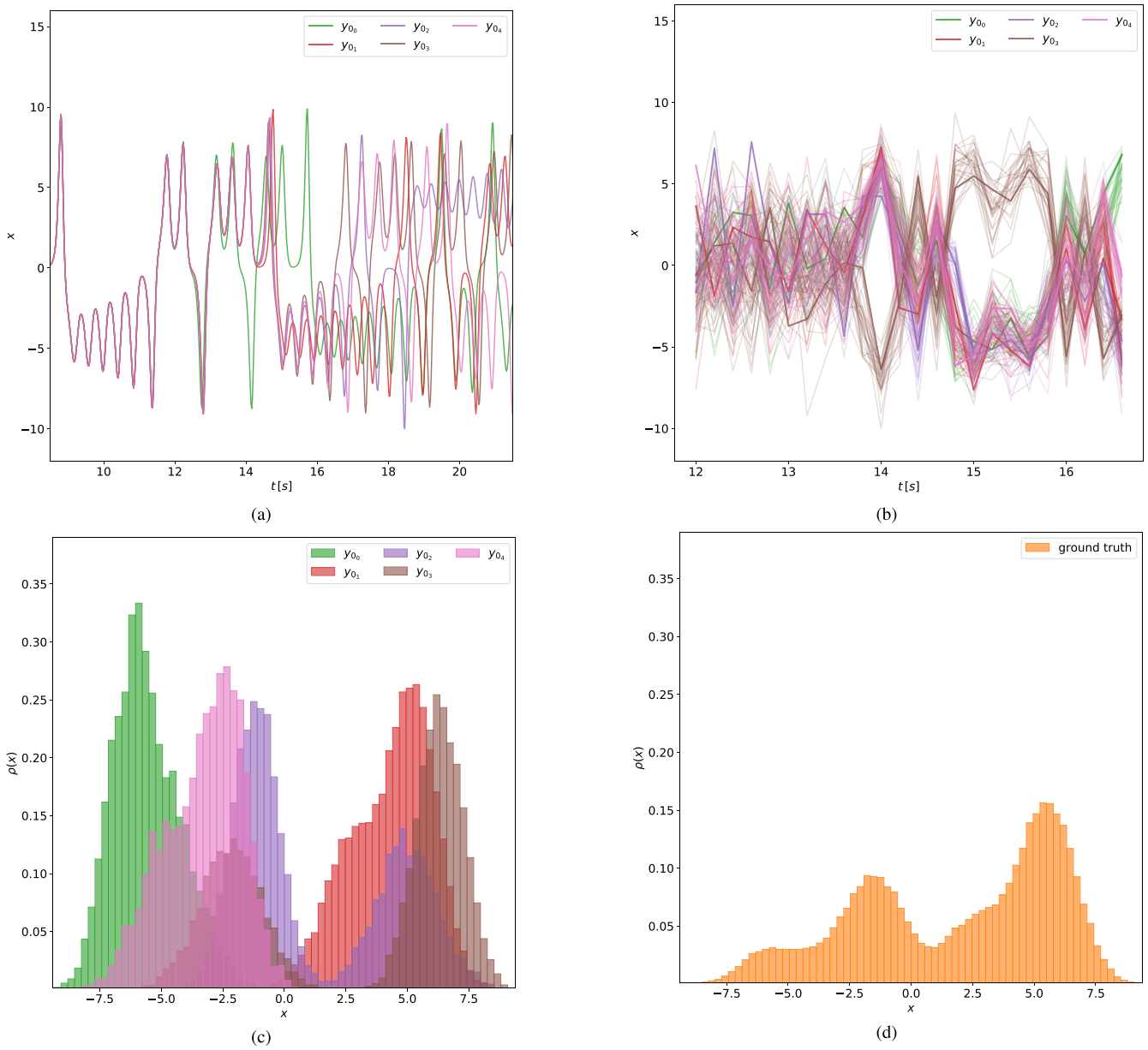

FIGURE 4. (a): Solution to the Lorenz system for different initial values y_0. (b): The bifurcation region after the data augmentation steps described in the text. (c): Possible values x_{t+1} distinguished by initial value y_0. (d): Full probability distribution of x_{t+1}.

and test set data distribution. Therefore, the G-regression model does not output probability distribution. Thus, we use the histogram of G-regression predictions to calculate KLD. To calculate point-wise error metrics for ForGAN, we run it 100 times over the test set and report the mean and the standard deviation of these metrics as the result. With this information, we present a complete and clear image of ForGAN performance. The KLD value shows how accurate our method has learned the data distribution and the point-wise error metrics specifies how well it has considered the condition to forecast x_{t+1}. Furthermore, we have the possibility to compare ForGAN with other methods based on various criteria.

C. HYPERPARAMETER TUNING

The ForGAN structure has a set of hyperparameters which we tune for each experiment separately. Tab. 2 provides the list of hyperparameters alongside the allowed values. During training, at each epoch, we train the discriminator several times but our generator is only trained once per epoch. The number of training iterations for discriminator in one epoch is one of the hyperparameters (D_{Iter} in Tab. 2).

To tune these hyperparameters, we use a genetic algorithm. Genetic algorithms [100] are a class of methods for optimization task which are inspired by the evolution in nature. These methods provide an alternative to traditional optimization techniques by using directed random searches

TABLE 2. The List of ForGAN hyperparameters alongside the range of allowed values.

Hyperparameters	Abbreviation	Values
The type of cells	T	GRU, LSTM
The number of cells in generator	RG	1, 2, 4, 8, 16, 32, 64, 128, 256
The number of cells in discriminator	RD	1, 2, 4, 8, 16, 32, 64, 128, 256
The size of noise vector	N	1, 2, 4, 8, 16, 32
The size of look-back window (Condition)	C	1, 2, 4, 8, 16, 32, 64, 128, 256
Number of training iteration for discriminator	D_{Iter}	1, 2, 3, 4, 5, 6, 7

TABLE 3. The optimal hyperparameters used to construct ForGAN for different experiments.

		Lorenz Experiment	Mackey-Glass Experiment	Internet Traffic Data Experiment
Hyperparameters	T	GRU	LSTM	GRU
	RG	8	64	8
	RD	64	256	128
	N	32	4	16
	C	24	32	32
	D_{Iter}	2	6	3

to locate optimal solutions in complex landscapes [101]. The hyperparameters are encoded in a vector which is called a gene. The algorithm starts with a set of randomly initialized genes to form a gene pool and tries to find the most optimized gene through iterative progress. In each iteration, the genes in the gene pool are evaluated using a fitness function and those with low scores are eliminated. Then, the remaining genes are used to create offsprings. After multiple iterations, the algorithm converges to a gene with the most optimized combination of values. For further detailed information on genetic algorithms, we refer to this comprehensive survey [101].

Our genetic algorithm has a gene pool of size 8 and we run it for 8 iterations. At each iteration, we use 4 of the genes with the best scores to create offsprings. 4 new genes are created using crossover while 4 other genes are created using mutation. Using the values of a gene, we construct a ForGAN and train it on a train set while we monitor KLD on a validation set.

D. SETUP

For each experiment, the dataset is divided into three subsets. 50% of the dataset is used as the train set, 10% as the validation set and 40% as the test set. We code the ForGAN using TensorFlow [102] and run it on a DGX-1 machine.

V. RESULTS AND DISCUSSION

In Tab. 3 we present the set of optimal hyperparameters which are found by the genetic algorithm for each experiment. The numerical results achieved by ForGAN are summarized in Tab. 4 alongside the state-of-the-art results on the Mackey-Glass dataset [97] and Internet traffic dataset (A5M) [2]. Furthermore, we report the results obtained from G-regression model.

In the Lorenz experiment, the G-regression method performs better than GAN based on RMSE, MAE and MAPE values. However, we can perceive from Fig. 5 how misleading these metrics can be. Fig. 5 presents the probability distribution learned by ForGAN alongside the histogram of the G-regression predictions and the data distribution for each cluster on the test set as well as the entire test set. These plots indicate that ForGAN learns the probability distribution of the dataset precisely with respect to the corresponding cluster. Contrary, G-regression predictions are completely inaccurate While it obtained better scores on the point-wise metrics in comparison to ForGAN. The G-regression method has converged to the mean value of x_{t+1} distribution for each cluster and as a result, the predictions do not represent the ground truth at all. Since the histogram of G-regression predictions does not cover the range of ground truth values, it is not possible to calculate KLD for G-regression method in this experiment.

Furthermore, we expect ForGAN to forecast all possible outcomes for a given time window. To investigate the validity of this assumption, we select two random time windows from the test set and forecast x_{t+1} 100 times using ForGAN. Fig. 6 portrays the distribution of sampled x_{t+1} alongside the probability distribution of their cluster. We can perceive from this figure that ForGAN can model the full probability distribution of x_{t+1} for a given time window condition accurately.

In the Mackey-Glass experiment, ForGAN outperforms both state-of-the-art [97] and G-regression model based on point-wise error metrics as well as KLD. The G-regression model has the same structure as ForGAN and it is optimized directly on RMSE, yet ForGAN performs significantly better than G-regression. We find this observation to be the evidence for the effectiveness of adversarial training for forecasting in comparison to standard training methods.

Finally, in our last experiment on Internet traffic dataset, G-regression method outperforms state-of-the-art and ForGAN based on the MAPE value. On the other hand, ForGAN performs almost two times better than G-regression method based on KLD. Furthermore, the KLD for the state-of-the-art method is not available. Due to inconsistency between point-wise error metrics and divergence measure, selecting the best method with certainty is not possible. However, in the Lorenz experiment, we witness that it is possible to have a mean regression algorithm with a small point-wise error which is completely imprecise in forecasting future values. In any case, the performance of ForGAN on Internet traffic dataset is quite impressive. It outperforms the G-regression

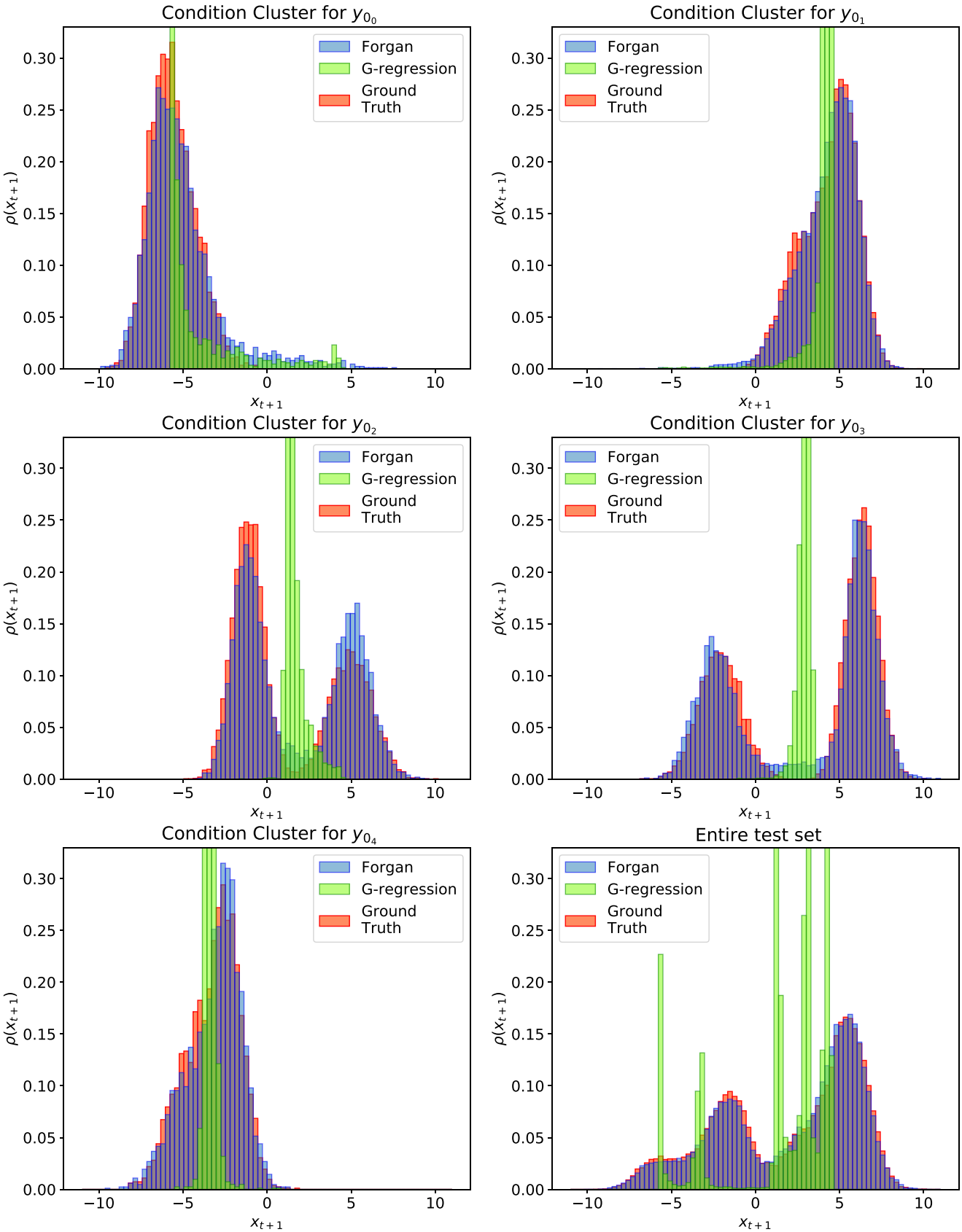

FIGURE 5. The prediction of x_{t+1} produced by ForGAN (blue), G-regression (green) alongside the ground truth distribution (orange) for each time window cluster $c \in [y_{0_0}, \ldots, y_{0_4}]$ and for the entire dataset on the Lorenz dataset.

TABLE 4. The results achieved by ForGAN alongside the results from G-regression model and state-of-the-art on Mackey-Glass dataset [97] and Internet traffic dataset [2]. The numbers in the parenthesis indicate the one standard deviation of results.

		state-of-the-art	G-regression	ForGAN [**Our Method**]
Lorenz dataset	RMSE	-	**2.91**	4.06 (0.01)
	MAE	-	**2.39**	2.94 (0.01)
	MAPE	-	**2.25** %	3.35 (0.24) %
	KLD	-	Nan	$\mathbf{1.67 \times 10^{-2}}$
Mackey-Glass dataset	RMSE	4.38×10^{-4}	5.63×10^{-4}	$\mathbf{3.82\ (0.02) \times 10^{-4}}$
	MAE	-	4.92×10^{-4}	$\mathbf{2.93\ (0.01) \times 10^{-4}}$
	MAPE	-	6.29×10^{-2} %	$\mathbf{3.46\ (0.02) \times 10^{-2}}$ %
	KLD	-	8.00×10^{-3}	$\mathbf{3.18 \times 10^{-3}}$
Internet traffic dataset (A5M)	RMSE	-	$\mathbf{1.27 \times 10^{8}}$	$1.31\ (0.00) \times 10^{8}$
	MAE	-	$\mathbf{9.01 \times 10^{7}}$	$9.29\ (0.03) \times 10^{7}$
	MAPE	2.91 %	**2.85** %	2.94 (0.01) %
	KLD	-	5.31×10^{-11}	$\mathbf{2.84 \times 10^{-11}}$

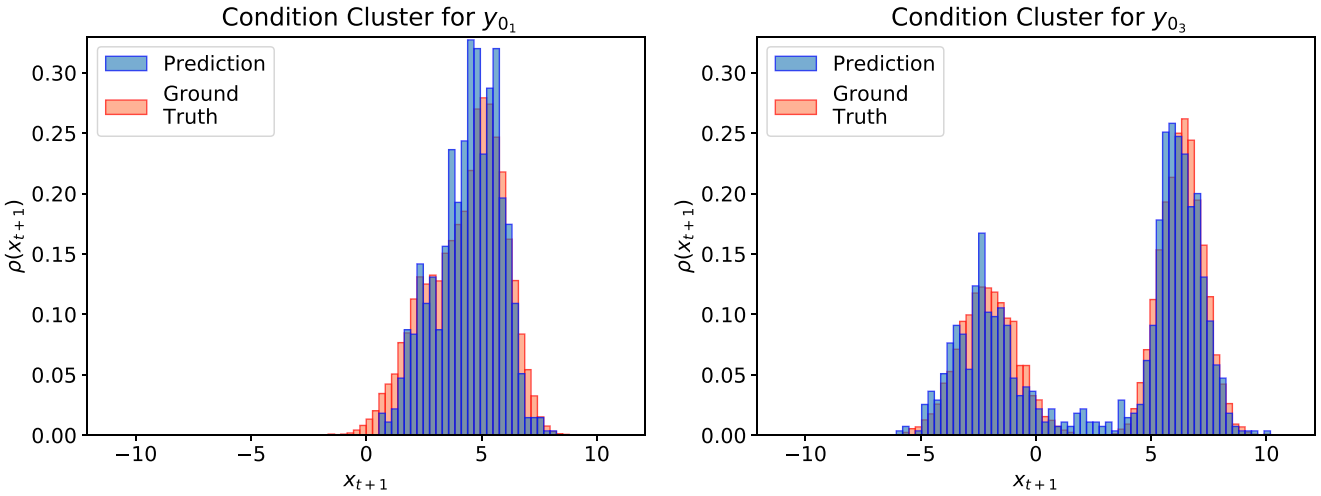

FIGURE 6. The probability distribution of x_{t+1} learned by ForGAN for two randomly selected time windows c and the data distribution of the time window cluster they origin from on Lorenz dataset.

based on KLD and it falls behind other methods based on point-wise error metrics only with a narrow margin.

VI. CONCLUSION AND FUTURE WORK

We present ForGAN, a neural network for one step ahead probabilistic forecasting. Our method is trained using adversarial training to learn the conditional probability distribution of future values.

We test our method with three experiments. In the first experiment, ForGAN demonstrates its high capability of learning probability distributions while taking the input time window into account. In the next two experiments, ForGAN demonstrates impressive performance on two public datasets, showing the effectiveness of adversarial training for forecasting tasks.

We compare ForGAN to G-regression, where the generator architecture is kept, but RMSE loss is optimized. We demonstrate that while G-regression performs better than ForGAN based on some point-wise error metrics, it does not accurately model the real data distribution and ForGAN outperforms G-regression considering distribution divergence measure. Our experiments show that point-wise error metrics are not a precise indicator for the performance of forecasting methods. Furthermore, ForGAN demonstrates its high capability in forecasting full probability distribution of future values which makes it superior to conventional mean regression methods. Adversarial training enables us to train a model for probabilistic forecasting easily without facing any technical problems like quantile crossing nor any dependency on the chosen prior.

Our experiments reveal that in the presence of strong noise, the effectiveness of ForGAN is more prominent as we illustrate in Lorenz experiments. The performance of mean regression methods is close to ForGAN when the noise is weak. Since ForGAN can model data distributions with any level of noise, it is more reliable and a robust choice for forecasting in comparison to mean regression methods.

For future reference and comparison, we introduce the Lorenz dataset for probabilistic forecasting. It is based on a chaotic differential equation. The Lorenz dataset can be downloaded from https://cloud.dfki.de/owncloud/index.php/s/KGJm5iNKrCnAwEg.

A. FUTURE WORK

With the promising results from ForGAN, there are many possibilities to pursue this line of research further. One possible direction is to investigate the capability of ForGAN to forecast multiple-step ahead values. It would be interesting to find out how far we can push the horizon with ForGAN and compare to the state-of-the-art in multi-step prediction. Another direction is improving the architecture of ForGAN. In this paper, we limit the ForGAN structure to one layer of LSTM/GRU in generator and discriminator however one can study the performance of ForGAN with different architectures e.g. CNNs or different loss functions like Wasserstein [76]. As in other works of studying GAN, we found some issues in evaluating and comparing the methods. Hence, further research in that direction would be beneficial, too.

REFERENCES

[1] M. C. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, no. 4300, pp. 287–289, 1977.
[2] P. Cortez, M. Rio, M. Rocha, and P. Sousa, "Multi-scale Internet traffic forecasting using neural networks and time series methods," *Expert Syst.*, vol. 29, no. 2, pp. 143–155, May 2012.
[3] E. Racah, C. Beckham, T. Maharaj, S. E. Kahou, M. Prabhat, and C. Pal, "Extremeweather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3402–3413.
[4] E. R. Rodrigues, I. Oliveira, R. Cunha, and M. Netto, "DeepDownscale: A deep learning strategy for high-resolution weather forecast," in *Proc. IEEE 14th Int. Conf. e-Sci. (e-Sci.)*, Oct./Nov. 2018, pp. 415–422.
[5] S. Nevo *et al.*, "ML for flood forecasting at scale," 2019, *arXiv:1901.09583*. [Online]. Available: https://arxiv.org/abs/1901.09583
[6] S. M. Mousavi, W. Zhu, Y. Sheng, and G. C. Beroza, "CRED: A deep residual network of convolutional and recurrent units for earthquake signal detection," Oct. 2018, *arXiv:1810.01965*. [Online]. Available: https://arxiv.org/abs/1810.01965
[7] Z. E. Ross, Y. Yue, M.-A. Meier, E. Hauksson, and T. H. Heaton, "PhaseLink: A deep learning approach to seismic phase association," *J. Geophys. Res., Solid Earth*, vol. 124, no. 1, pp. 856–869, 2019. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018JB016674
[8] A. Gensler and B. Sick, "A multi-scheme ensemble using coopetitive soft-gating with application to power forecasting for renewable energy generation," 2018, *arXiv:1803.06344*. [Online]. Available: https://arxiv.org/abs/1803.06344
[9] Y. Chen, Y. Wang, D. Kirschen, and B. Zhang, "Model-free renewable scenario generation using generative adversarial networks," *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 3265–3275, 2018.
[10] W. Ronny Huang and M. A. Perez, "Accurate, data-efficient learning from noisy, choice-based labels for inherent risk scoring," Nov. 2018, *arXiv:1811.10791*. [Online]. Available: http://arxiv.org/abs/1811.10791
[11] Q. Zhang, R. Luo, Y. Yang, and Y. Liu, "Benchmarking deep sequential models on volatility predictions for financial time series," Nov. 2018, *arXiv:1811.03711*. [Online]. Available: http://arxiv.org/abs/1811.03711
[12] A. Avati *et al.*, "Predicting inpatient discharge prioritization with electronic health records," Dec. 2018, *arXiv:1812.00371*. [Online]. Available: http://arxiv.org/abs/1812.00371
[13] T. Janssoone, C. Bic, D. Kanoun, P. Hornus, and P. Rinder, "Machine learning on electronic health records: Models and features usages to predict medication non-adherence," Nov. 2018, *arXiv:1811.12234*. [Online]. Available: http://arxiv.org/abs/1811.12234
[14] A. Avati, K. Jung, S. Harman, L. Downing, A. Ng, and N. H. Shah, "Improving palliative care with deep learning," Nov. 2017, *arXiv:1711.06402*. [Online]. Available: http://arxiv.org/abs/1711.06402
[15] M. Zitnik, F. Nguyen, B. Wang, J. Leskovec, A. Goldenberg, and M. M. Hoffman, "Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities," *Inf. Fusion*, vol. 50, pp. 71–91, Oct. 2019.
[16] T. Gneiting and M. Katzfuss, "Probabilistic forecasting," *Annu. Rev. Statist. Appl.*, vol. 1, pp. 125–151, Jan. 2014.
[17] T. Hothorn, T. Kneib, and P. Bühlmann, "Conditional transformation models," *J. Roy. Stat. Soc., B Stat. Methodol.*, vol. 76, no. 1, pp. 3–27, 2014.
[18] T. Kneib, "Beyond mean regression," *Stat. Model.*, vol. 13, no. 4, pp. 275–303, 2013.
[19] G. E. P. Box and G. Jenkins, *Time Series Analysis, Forecasting and Control*. San Francisco, CA, USA: Holden-Day, 1990.
[20] X. Yan and N. A. Chowdhury, "A comparison between SVM and LSSVM in mid-term electricity market clearing price forecasting," in *Proc. 26th IEEE Can. Conf. Elect. Comput. Eng. (CCECE)*, May 2013, pp. 1–4.
[21] X. Yan and N. A. Chowdhury, "Mid-term electricity market clearing price forecasting using multiple least squares support vector machines," *IET Gener., Transmiss. Distrib.*, vol. 8, no. 9, pp. 1572–1582, Sep. 2014.
[22] G. Rubio, H. Pomares, I. Rojas, and L. J. Herrera, "A heuristic method for parameter selection in LS-SVM: Application to time series prediction," *Int. J. Forecasting*, vol. 27, no. 3, pp. 725–739, 2011.
[23] V. Vapnik, *Statistical Learning Theory*, vol. 3. New York, NY, USA: Wiley, 1998.
[24] H. Frohlich, O. Chapelle, and B. Scholkopf, "Feature selection for support vector machines by means of genetic algorithm," in *Proc. 15th IEEE Int. Conf. Tools Artif. Intell.*, Nov. 2003, pp. 142–148.
[25] C.-L. Huang and C.-J. Wang, "A GA-based feature selection and parameters optimizationfor support vector machines," *Expert Syst. Appl.*, vol. 31, no. 2, pp. 231–240, 2006.
[26] P. Cortez, M. Rocha, and J. Neves, "Genetic and evolutionary algorithms for time series forecasting," in *Engineering of Intelligent Systems*, L. Monostori, J. Váncza, and M. Ali, Eds. Berlin, Germany: Springer, 2001, pp. 393–402.
[27] M. Gan, H. Peng, and X.-P. Dong, "A hybrid algorithm to optimize RBF network architecture and parameters for nonlinear time series prediction," *Appl. Math. Model.*, vol. 36, no. 7, pp. 2911–2919, 2012.
[28] K.-J. Kim and I. Han, "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index," *Expert Syst. Appl.*, vol. 19, no. 2, pp. 125–132, Aug. 2000.
[29] Q. Cai, D. Zhang, B. Wu, and S. C. Leung, "A novel stock forecasting model based on fuzzy time series and genetic algorithm," in *Proc. Int. Conf. Comput. Sci.*, vol. 18, 2013, pp. 1155–1162. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050913004249
[30] E. Bas, V. R. Uslu, U. Yolcu, and E. Egrioglu, "A modified genetic algorithm for forecasting fuzzy time series," *Appl. Intell.*, vol. 41, no. 2, pp. 453–463, 2014.
[31] T.-C. Chu, C.-T. Tsao, and Y.-R. Shiue, "Application of fuzzy multiple attribute decision making on company analysis for stock selection," in *Proc. Soft Comput. Intell. Syst. Inf. Process.*, Dec. 1996, pp. 509–514.
[32] Q. Song, R. P. Leland, and B. S. Chissom, "A new fuzzy time-series model of fuzzy number observations," *Fuzzy Sets Syst.*, vol. 73, no. 3, pp. 341–348, 1995.
[33] E. Egrioglu, C. H. Aladag, and U. Yolcu, "Fuzzy time series forecasting with a novel hybrid approach combining fuzzy c-means and neural networks," *Expert Syst. Appl.*, vol. 40, no. 3, pp. 854–857, 2013.
[34] M. Shah, "Fuzzy based trend mapping and forecasting for time series data," *Expert Syst. Appl.*, vol. 39, no. 7, pp. 6351–6358, 2012.
[35] C. H. Aladag, U. Yolcu, E. Egrioglu, and A. Z. Dalar, "A new time invariant fuzzy time series forecasting method based on particle swarm optimization," *Appl. Soft Comput.*, vol. 12, no. 10, pp. 3291–3299, 2012.
[36] M. Assaad, R. Boné, and H. Cardot, "A new boosting algorithm for improved time-series forecasting with recurrent neural networks," *Inf. Fusion*, vol. 9, no. 1, pp. 41–55, 2008.
[37] O. Ogunmolu, X. Gu, S. Jiang, and N. Gans, "Nonlinear systems identification using deep dynamic neural networks," 2016, *arXiv:1610.01439*. [Online]. Available: https://arxiv.org/abs/1610.01439

[38] G. Dorffner, "Neural networks for time series processing," *Neural Netw. World*, vol. 6, pp. 447–468, 1996.

[39] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, *Long Short Term Memory Networks for Anomaly Detection in Time Series*. Neuve, Belgium: Presses Univ. de Louvain, 2015, p. 89.

[40] M. Collins, "Ensembles and probabilities: A new era in the prediction of climate change," *Philos. Trans. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 365, pp. 1957–1970, Jun. 2007.

[41] T. Gneiting and A. E. Raftery, "Weather forecasting with ensemble methods," *Science*, vol. 310, no. 5746, pp. 248–249, 2005.

[42] T. N. Palmer, "The economic value of ensemble forecasts as a tool for risk assessment: From days to decades," *Quart. J. Roy. Meteorolog. Soc.*, vol. 128, no. 581, pp. 747–774, 2002.

[43] T. Palmer, "Towards the probabilistic earth-system simulator: A vision for the future of climate and weather prediction," *Quart. J. Roy. Meteorolog. Soc.*, vol. 138, no. 665, pp. 841–861, 2012.

[44] H. L. Cloke and F. Pappenberger, "Ensemble flood forecasting: A review," *J. Hydrol.*, vol. 375, nos. 3–4, pp. 613–626, 2009.

[45] R. Krzysztofowicz, "The case for probabilistic forecasting in hydrology," *J. Hydrol.*, vol. 249, nos. 1–4, pp. 2–9, 2001.

[46] T. Jordan *et al.*, "Operational earthquake forecasting. State of knowledge and guidelines for utilization," *Ann. Geophys.*, vol. 54, no. 4, 2011. [Online]. Available: https://www.annalsofgeophysics.eu/index.php/annals/article/view/5350

[47] P. Pinson, "Wind energy: Forecasting challenges for its operational management," *Statist. Sci*, vol. 28, no. 4, pp. 564–585, Nov. 2013.

[48] X. Zhu and M. G. Genton, "Short-term wind speed forecasting for power system operations," *Int. Statist. Rev.*, vol. 80, no. 1, pp. 2–23, 2012.

[49] J. J. J. Groen, R. Paap, and F. Ravazzolo, "Real-time inflation forecasting in a changing world," *J. Bus. Econ. Statist.*, vol. 31, no. 1, pp. 29–44, 2013.

[50] A. Timmermann, "Density forecasting in economics and finance," *J. Forecasting*, vol. 19, no. 4, pp. 231–234, 2000.

[51] J. M. Montgomery, F. M. Hollenbach, and M. D. Ward, "Ensemble predictions of the 2012 US presidential election," *PS, Political Sci. Politics*, vol. 45, no. 4, pp. 651–654, 2012.

[52] L. Alkema, A. E. Raftery, and S. J. Clark, "Probabilistic projections of HIV prevalence using Bayesian melding," *Ann. Appl. Statist.*, vol. 1, no. 1, pp. 229–248, 2007.

[53] A. E. Raftery, N. Li, H. Ševčíková, P. Gerland, and G. K. Heilig, "Bayesian probabilistic population projections for all countries," *Proc. Nat. Acad. Sci. USA*, vol. 109, no. 35, pp. 13915–13921, 2012.

[54] H. E. Jones and D. J. Spiegelhalter, "Improved probabilistic prediction of healthcare performance indicators using bidirectional smoothing models," *J. Roy. Stat. Soc., A Statist. Soc.*, vol. 175, no. 3, pp. 729–747, 2012.

[55] L. Hood, J. R. Heath, M. E. Phelps, and B. Lin, "Systems biology and new technologies enable predictive and preventative medicine," *Science*, vol. 306, no. 5696, pp. 640–643, 2004.

[56] S. M. Stigler, "The transition from point to distribution estimation," *Bull. Int. Stat. Inst.*, vol. 46, pp. 332–340, Feb. 1975.

[57] R. Koenker, *Quantile Regression* (Econometric Society Monographs). Cambridge, U.K.: Cambridge Univ. Press, 2005.

[58] R. Koenker and G. Bassett, Jr., "Regression quantiles," *Econometrica, J. Econ. Soc.*, vol. 46, no. 1, pp. 33–50, 1978.

[59] B. Efron, "Regression percentiles using asymmetric squared error loss," *Stat. Sinica*, vol. 1, no. 1, pp. 93–125, Jan. 1991.

[60] W. K. Newey and J. L. Powell, "Asymmetric least squares estimation and testing," *Econometrica, J. Econ. Soc.*, vol. 55, no. 4, pp. 819–847, 1987.

[61] F. Sobotka and T. Kneib, "Geoadditive expectile regression," *Comput. Statist. Data Anal.*, vol. 56, no. 4, pp. 755–767, 2012.

[62] H. Dette and S. Volgushev, "Non-crossing non-parametric estimates of quantile curves," *J. Roy. Stat. Soc., B Stat. Methodol.*, vol. 70, no. 3, pp. 609–627, 2008.

[63] S. K. Schnabel and P. H. Eilers, "Simultaneous estimation of quantile curves using quantile sheets," *AStA Adv. Stat. Anal.*, vol. 97, no. 1, pp. 77–87, 2013.

[64] A. E. Raftery, T. Gneiting, F. Balabdaoui, and M. Polakowski, "Using Bayesian model averaging to calibrate forecast ensembles," *Monthly Weather Rev.*, vol. 133, no. 5, pp. 1155–1174, 2005.

[65] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.

[66] R. M. Neal, "Probabilistic inference using Markov chain Monte Carlo methods," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Tech. Rep. CRG-TR-93-1, 1993.

[67] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Mach. Learn.*, vol. 37, no. 2, pp. 183–233, Nov. 1999.

[68] A. Doucet, N. De Freitas, and N. Gordon, "An introduction to sequential Monte Carlo methods," in *Sequential Monte Carlo Methods in Practice*. New York, NY, USA: Springer, 2001, pp. 3–14.

[69] T. P. Minka, "Expectation propagation for approximate Bayesian inference," in *Proc. 17th Conf. Uncertainty Artif. Intell.* San Mateo, CA, USA: Morgan Kaufmann, 2001, pp. 362–369.

[70] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[71] A. Damianou and N. Lawrence, "Deep Gaussian processes," in *Proc. 16th Int. Conf. Artif. Intell. Statist.* (Proceedings of Machine Learning Research), vol. 31, C. M. Carvalho and P. Ravikumar, Eds. Scottsdale, AZ, USA: PMLR, May 2013, pp. 207–215. [Online]. Available: http://proceedings.mlr.press/v31/damianou13a.html

[72] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, no. 7553, pp. 452–459, 2015.

[73] Y. Gal, "Uncertainty in deep learning," Ph.D. dissertation, Univ. Cambridge, Cambridge, U.K., 2016.

[74] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[75] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," Nov. 2015, *arXiv:1511.06434*. [Online]. Available: http://arxiv.org/abs/1511.06434

[76] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*. [Online]. Available: https://arxiv.org/abs/1701.07875

[77] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," May 2016, *arXiv:1605.09782*. [Online]. Available: http://arxiv.org/abs/1605.09782

[78] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," Jun. 2016, *arXiv:1606.03657*. [Online]. Available: http://arxiv.org/abs/1606.03657

[79] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*. [Online]. Available: https://arxiv.org/abs/1411.1784

[80] O. Mogren, "C-RNN-GAN: Continuous recurrent neural networks with adversarial training," 2016, *arXiv:1611.09904*. [Online]. Available: https://arxiv.org/abs/1611.09904

[81] Y. Gao, R. Singh, and B. Raj, "Voice impersonation using generative adversarial networks," 2018, *arXiv:1802.06840*. [Online]. Available: https://arxiv.org/abs/1802.06840

[82] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient," in *Proc. AAAI*, 2017, pp. 2852–2858.

[83] O. Press, A. Bar, B. Bogin, J. Berant, and L. Wolf, "Language generation with recurrent generative adversarial networks without pre-training," 2017, *arXiv:1706.01399*. [Online]. Available: https://arxiv.org/abs/1706.01399

[84] J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, and D. Jurafsky, "Adversarial learning for neural dialogue generation," 2017, *arXiv:1701.06547*. [Online]. Available: https://arxiv.org/abs/1701.06547

[85] Y. Zhang, Z. Gan, and L. Carin, "Generating text via adversarial training," in *Proc. NIPS Workshop Adversarial Training*, vol. 21, 2016, pp. 1–6.

[86] C. Donahue, B. Li, and R. Prabhavalkar, "Exploring speech enhancement with generative adversarial networks for robust speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 5024–5028.

[87] D. Michelsanti and Z.-H. Tan, "Conditional generative adversarial networks for speech enhancement and noise-robust speaker verification," 2017, *arXiv:1709.01703*. [Online]. Available: https://arxiv.org/abs/1709.01703

[88] Z.-C. Fan, Y.-L. Lai, and J.-S. R. Jang, "Svsgan: Singing voice separation via generative adversarial network," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 726–730.

[89] C. Donahue, J. McAuley, and M. Puckette, "Adversarial audio synthesis," Feb. 2018, *arXiv:1802.04208*. [Online]. Available: http://arxiv.org/abs/1802.04208
[90] C. Esteban, S. L. Hyland, and G. Rätsch, "Real-valued (medical) time series generation with recurrent conditional gans," 2017, *arXiv:1706.02633*. [Online]. Available: https://arxiv.org/abs/1706.02633
[91] D. Li, D. Chen, J. Goh, and S.-K. Ng, "Anomaly detection with generative adversarial networks for multivariate time series," 2018, *arXiv:1809.04758*. [Online]. Available: https://arxiv.org/abs/1809.04758
[92] C. Zhang, S. R. Kuppannagari, R. Kannan, and V. K. Prasanna, "Generative adversarial network for synthetic time series data generation in smart grids," in *Proc. IEEE Int. Conf. Commun., Control, Comput. Technol. Smart Grids*, Oct. 2018, pp. 1–6.
[93] F. M. White and I. Corfield, *Viscous Fluid Flow*, vol. 3. New York, NY, USA: McGraw-Hill, 2006.
[94] S. Chandrasekhar, *Hydrodynamic and Hydromagnetic Stability*. Chelmsford, MA, USA: Courier Corporation, 2013.
[95] C. Sparrow, *The Lorenz Equations: Bifurcations, Chaos, and Strange Attractors*, vol. 41. New York, NY, USA: Springer, 2012.
[96] S. H. Kellert, *In the Wake of Chaos: Unpredictable Order in Dynamical Systems*. Chicago, IL, USA: Univ. of Chicago Press, 1993.
[97] E. Méndez, O. Lugo, and P. Melin, "A competitive modular neural network for long-term time series forecasting," in *Nature-Inspired Design of Hybrid Intelligent Systems*. Cham, Switzerland: Springer, 2017, pp. 243–254.
[98] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, Mar. 1951. doi: 10.1214/aoms/1177729694.
[99] K. H. Knuth, "Optimal data-based binning for histograms," 2006, *arXiv:physics/0605197*. [Online]. Available: https://arxiv.org/abs/physics/0605197
[100] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.
[101] M. Srinivas and L. M. Patnaik, "Genetic algorithms: A survey," *Computer*, vol. 27, no. 6, pp. 17–26, Jun. 1994.
[102] M. Abadi *et al.* (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. [Online]. Available: https://www.tensorflow.org/

ALIREZA KOOCHALI received the master's degree in computer science from the University of Kaiserslautern, Germany. His master's thesis was on Multimodal Sentiment Analysis with Deep Neural Networks. He is currently pursuing the Ph.D. degree in computer science at learning from the Test Data Research Lab (FLaP) - a joint-lab between IAV GmbH and German Research Center for Artificial Intelligence (DFKI GmbH), under the supervision of Prof. Dr. Prof. h.c. A. Dengel. His research interests include probabilistic machine learning using generative adversarial networks, probabilistic machine learning, generative adversarial networks, deep neural networks, and time series analysis.

PETER SCHICHTEL received the Diploma degree in physics and the Dr. rer. nat. degree from the University of Heidelberg, under the supervision of Prof. Dr. T. Plehn. His thesis topic was More Jets in more LHC Searches. He is currently a Research Engineer at Learning with the Test Data Research Lab (FLaP) - a joint lab between IAV GmbH and German Research Center for Artificial Intelligence (DFKI GmbH), where he is most interested in time series analysis and deep neural network architecture. He was a member of the International Max Planck Research School for Precision Tests of Fundamental Symmetries, Heidelberg, Germany, and the Theoretical Advanced Study Institute of Elementary Particle Physics, University of Boulder, Boulder, CO, USA. He was appointed European Marie Curie Early Stage Researcher with the University of Durham, U.K., and a member of the Monte-Carlo-Network, as well as, Visiting Scientist at CERN and Fermilab, IL, USA. He has over 20 publications in the fields of particle physics, statistical analysis, data mining, and time series analysis.

ANDREAS DENGEL received the Diploma degree in computer science from the University of Kaiserslautern, and the Ph.D. degree from the University of Stuttgart. He was with IBM, Siemens, and Xerox Parc. In 1993, he became a Professor with the Computer Science Department, University of Kaiserslautern, where he holds the Chair Knowledge-Based Systems. Since 2009, he has been a Professor (Kyakuin) with the Department of Computer Science and Information Systems, Osaka Prefecture University. He is currently the Scientific Director of the German Research Center for Artificial Intelligence (DFKI GmbH), Kaiserslautern. His main scientific emphases are in the areas of pattern recognition, document understanding, information retrieval, multimedia mining, semantic technologies, and social media. He is a member of several international advisory boards, has chaired major international conferences, and has founded several successful start-up companies. Moreover, he is a co-editor of the international computer science journals and has written or edited 12 books. He is the author of over 300 peer-reviewed scientific publications and has supervised over 170 Ph.D. and master theses. He is an IAPR Fellow and has received prominent international awards.

SHERAZ AHMED received the master's degree from the University of Kaiserslautern, Germany, in computer science, and the Ph.D. degree from the University of Kaiserslautern, Germany, under the supervision of Prof. Dr. Prof. h.c. A. Dengel and Prof. Dr. habil. M. Liwicki. His Ph.D. topic is Generic Methods for Information Segmentation in Document Images. From 2012 to 2013, he visited Osaka Prefecture University, Osaka, Japan, as a Research Fellow, supported by the Japanese Society for the Promotion of Science. In 2014, he visited the University of Western Australia, Perth, Australia, as a Research Fellow, supported by the DAAD, Germany, and Go - 8, Australia. He is currently a Senior Researcher with the German Research Center for Artificial Intelligence (DFKI GmbH), Kaiserslautern, where he is leading the area of Time Series Analysis. Over the last few years, he has primarily worked for the development of various systems for information segmentation in document images. His research interests include document understanding, generic segmentation framework for documents, gesture recognition, pattern recognition, data mining, anomaly detection, and natural language processing. He has over 30 publications on the said and related topics, including three journal papers and two book chapters. He is a frequent Reviewer of various journals and conferences, including the *Patter Recognition Letters*, the *Neural Computing and Applications*, IJDAR, ICDAR, ICFHR, DAS, and so on.