Satellite Image Prediction Relying on GAN and LSTM Neural Networks

Zhan Xu*, Jun Du*, Jingjing Wang*, Chunxiao Jiang†‡ and Yong Ren*
*Department of Electronic Engineering, Tsinghua University, Beijing, 100084, China
†Tsinghua Space Center, Tsinghua University, Beijing, 100084, China
‡Key Laboratory of EDA, Research Institute of Tsinghua University in Shenzhen, Shenzhen 518057, China
E-mail: xz17@mails.tsinghua.edu.cn, {blgdujun, chinaeephd, chx.jiang}@gmail.com, reny@tsinghua.edu.cn

***Abstract*—Satellite image is an important resource for weather forecast. It can indicate the evolution of weather systems and is beneficial in terms of guiding people to make accurate weather forecasting. However, the use of satellite images is encountered with the dilemma of such as small data volume and of poor real-time performance. Hence it is important to make accurate prediction for satellite images. The goal of satellite image prediction is to predict the next few images of the image sequence. Essentially, it is a a spatiotemporal sequence prediction problem, where the prediction of satellite images is difficult due to its large-scale observation area. In this paper, we propose a generative adversarial networks-long short-term memory (GAN-LSTM) model for the satellite image prediction by combining the generating ability of the GAN with the forecasting ability of the LSTM network. For evaluation, we conduct our experiments on the FY-2E satellite cloud maps. In addition, we use a score correct rate (CR) to measure the degree of similarity between predictions and ground truth. Experiment results show that the proposed GAN-LSTM network is capable of efficiently capturing the evolution rules of weather systems, which outperforms the traditional autoencoder-LSTM.**

***Index Terms*—Generative adversarial network, long short-term memory, satellite image prediction, machine learning**

I. Introduction

With the development of meteorological satellite technology, meteorological satellite data plays an important role in our daily life. Satellite cloud images characterize the distribution of clouds on the ground, so they can be used to track the evolution of large-scale weather systems. With the help of cloud images, people can make more accurate weather prediction. However, due to the limitation of satellite transmission bandwidth and transmission mode, there exists a large time delay when transmitting satellite images back to the Earth [1], [2]. Thus the satellite image can not meet the need of data analysis, which reduces the timeliness of forecast. Moreover, traditional cloud images are sparse and non-uniform over time domain, which makes the weather forecast more difficult. Deploying new satellites and sensors may solve these problems, but it is very costly [3]. We observe that image sequence prediction provides an effective way to solve the problems mentioned above. Hence in this paper, we investigate the prediction of satellite cloud images.

Essentially, the cloud image prediction is a spatiotemporal sequence forecasting problem, which takes the past cloud image sequence as the input and the output are a number of future cloud images [4]. However, the high dimensionality of the input image lead to the high computation cost, which make the problem more difficult. Moreover, due to the large-scale observation area of the satellite images, the evolutionary rule of image sequence is very complex because of chaotic nature of the atmosphere, which makes it challenging to build an effective prediction model for the satellite images.

With the development of deep learning, these technical issues above can be addressed effectively [5]. Recent advances in deep learning, especially the Long Short-Term Memory (LSTM) network [6] and Generative Adversarial Network (GAN) [7], provide some useful insights on how to solve these problems. LSTM is a very special kind of recurrent neural network (RNN). The RNN might be able to connect previous information to the present task, such as using previous video frames information to understand the present frame. LSTM performs better in learning long-range dependencies. [8] built an LSTM encoder-decoder model to learn the representation of video sequences, which reconstructed the input frames and predicted the future frames simultaneously. [9] put forward the autoencoder-LSTM structure. The model in [9] used an autoencoder to realize the feature learning as well as decrease the dimension, and then used LSTM to learn the temporal information of extracted features. GAN provided us with another thinking of the implicit features and decrease dimension [10]. The structure of GAN is delicate so that it can handle different kinds of image. However, the current research on GAN focuses on model improvement [11], [12] or improving image quality [13], and did not use it to reduce the redundancy.

In this paper, the satellite image predicting problem is studied from a different perspective, and a new predict model is proposed. The goal of this work is to give precise and timely prediction of cloud image to a certain period of time, we formulate the satellite image prediction as a spatiotemporal sequence forecasting problem that can be solved by machine learning. In order to model the spatiotemporal relationships, we put forward GAN-LSTM which has the image generator of GAN attached to the output of LSTM. Specifically, the use of GAN innovatively solves the problem of high input data dimension and greatly improves the quality and diversity of the desired image, which outperform the traditional autoencoder network. In addition, the LSTM network learn the spatiotemporal relationship without complex atmospheric modeling. For

978-1-5386-8088-9/19/$31.00

evaluation, we conduct our experiments on the Feng-Yun-2E (FY-2E)[1] image data set and use a score to measure the effect of the proposed model. Experiment results show that our model can make effective prediction over large time and space scale and outperforms the traditional autoencoder-LSTM.

This paper is organized as follows: Section II formulates the cloud image predicting problem and introduces the background of our model. Section III describes our prediction model. Section IV describes various evaluation experiments on image data and gives a qualitative discussion of the experimental results. Finally, Section V concludes our paper.

II. Preliminaries

A. Formulation of Cloud Image Predicting Problem

The goal of cloud image prediction is to use previously observed cloud image frames to forecast the next few frames. Suppose our observation area is an $M \times N$ gird consisting of M rows and N columns. Then the cloud image can be represented by a vector $v \in \mathbf{R}^{M \times N}$. We record a set of data observed continuously as $\widetilde{v}_1, \widetilde{v}_2, \widetilde{v}_3, \cdots$. The prediction problem is to predict the next frame with j previous frames obtained by continuous observation:

$$\hat{v}_{n+1} = \arg\max_{v_{n+1}} p(v_{n+1} | \widetilde{v}_{n-j+1}, \widetilde{v}_{n-j+2}, \ldots, \widetilde{v}_n). \quad (1)$$

For further prediction, we use the prediction result to replace the real satellite image:

$$\hat{v}_{n+2} = \arg\max_{v_{n+2}} p(v_{n+2} | \widetilde{v}_{n-j+2}, \ldots, \widetilde{v}_n, \hat{v}_{n+1}). \quad (2)$$

B. Generative Adversarial Networks

In our model, we use the GAN to generate satellite images ($v \in \mathbf{R}^{M \times N}$) from a set of random data. We use $\boldsymbol{z} \sim p_z$ to represent the input random noise, then define a multilayer perceptron $G(\boldsymbol{z}; \theta_g)$ which maps the random noise to data space $\mathbf{R}^{M \times N}$, where $v = G(\boldsymbol{z})$. The distribution of v is totally defined by $\boldsymbol{z}$, so we also called $\boldsymbol{z}$ the hidden variable. Then we define another multilayer perceptron $D(\boldsymbol{x}; \theta_d)$. The input of D is real sample $\boldsymbol{x} \sim p_{data}$ or fake sample generated by G, the output of D is a scalar between 0 and 1. The basic structure of GANs is shown in Fig. 1. The generator is trained to fake the discriminator and the discriminator is trained to make the right judgment. The loss function of GAN is written as:

$$\begin{aligned} \min_G \max_D V(D, G) = {} & \mathbb{E}_{\boldsymbol{x} \sim p_{data}(\boldsymbol{x})} [\log D(\boldsymbol{x})] \\ & + \mathbb{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})} [\log(1 - D(G(\boldsymbol{z})))]. \end{aligned} \quad (3)$$

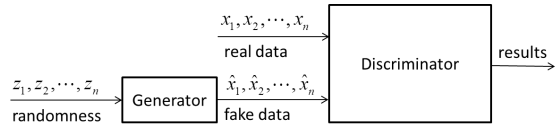


Fig. 1. The basic structure of GAN.

[1]5th flight unit of the FY-2 series. Running on geosynchronous orbit. Mission: operational meteorology.

We use the hidden variables $\boldsymbol{z}$ to replace the $M \times N$ tensor, then the Equation (1) can be written as:

$$\hat{v}_{n+1} = \arg\max_z p(G(\boldsymbol{z}) | \widetilde{v}_{n-j+1}, \widetilde{v}_{n-j+2}, \ldots, \widetilde{v}_n). \quad (4)$$

With the help of GAN, we can greatly decrease the dimension of the desired data. Instead of the $M \times N$ tensor, we only need to know the hidden variables $\boldsymbol{z}$ whose dimension is far less than that of $\boldsymbol{x} \sim p_{data}$.

C. Long Short-Term Memory Networks

The Long Short-Term Memory networks (LSTM) provides some information for the hidden variable of the generator of GAN. LSTM has proven powerful for learning long-range dependencies in sequence modeling. The core of LSTM is the cell state denoted by C_t, which acts as an accumulator of the state information. C_t is accessed and update by two controlling gates. The net uses f_t to decide when to forget C_t, and uses i_t to decide when to keep or override C_t. Whether the latest cell output C_t will be propagated to the final state h_t is further controlled by the output gate o_t. In our model, h_t represents the evolutionary information derived from the sequence of satellite images:

$$h_t = f_{lstm}(\widetilde{v}_{n-j+1}, \widetilde{v}_{n-j+2}, \ldots, \widetilde{v}_n). \quad (5)$$

We can see from above that the LSTM cell uses the memory cell and gates to control the information flow. The learning result is the evolution process of the satellite image.

III. GAN-LSTM Model for Image Training

We now propose our GAN-LSTM model for the cloud image training. The GAN-LSTM model combines the generating ability of GAN with the temporal context usage of an LSTM. LSTM model has the ability to extract the evolutionary information from the image sequence. For the cloud image applications, such evolutionary information contains information about which cloud will present and how these clouds move, so that the motion can be extrapolated. So we call it implicit feature. To extract the implicit feature, we use one LSTM to read the input images, one timestep at a time, to obtain fixed dimensional vector representation, and then we use the vector as the input of generator, so that the generator can extract the output sequence from that vector. The structure of GAN-LSTM is shown in Fig. 2.

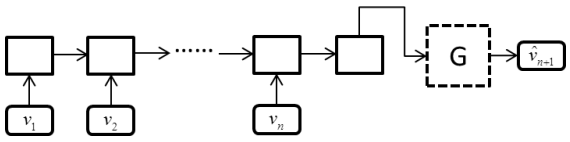


Fig. 2. GAN-LATM model, where G represents generator that cut off from the trained GAN

The training process can be divided into two steps. First, we use the real satellite image data set to train the GAN model. The generator of GAN is supposed to generate new satellite images with restrict input. After the GAN is trained well, the

generator will be cut from the GAN and an LSTM will be attached. In the later training process, the parameters of the generator will be fixed. The data set used in the second step is the same as the first step, but the input of LSTM is time-series satellite images. Besides, the redundant information of the image is very large, so we use two hidden layers to reduce the input dimension. Afterwards, the LSTM network is trained to produce vectors that have the same dimension and restriction as the input of generator. The LSTM network will try to capture the implicit features which contain the evolution information about the clouds. Then the generator unfolds these implicit features to give the final prediction:

$$\begin{aligned}\hat{v}_{n+1} &= \arg\max_{v_{n+1}} p(v_{n+1}|\widetilde{v}_{n-j+1}, \widetilde{v}_{n-j+2}, \ldots, \widetilde{v}_n) \\ &= \arg\max_{z} p(G(\boldsymbol{z})|\widetilde{v}_{n-j+1}, \widetilde{v}_{n-j+2}, \ldots, \widetilde{v}_n) \\ &\approx \arg\max_{z} p(G(\boldsymbol{z})|f_{lstm}(\widetilde{v}_{n-j+1}, \widetilde{v}_{n-j}, \ldots, \widetilde{v}_n)) \\ &\approx G(h_t)\end{aligned} \tag{6}$$

The neuron number of generator is bigger than that of LSTM network, so if we train them together, the generator will play a major role. The target of the training process will be generate more realistic images instead of discover the evolutionary rules of image sequence. If we directly use the image to train the LSTM, the redundance of images will make it hard to find the rules. So we train the generator and LSTM separately.

The performance of generator directly determines the quality of the predicted image. The generated images are supposed to be not only low noise, but also diverse. This poses a challenge to the training of GAN. The GAN model put forward by [7] is just a general framework. The framework is relatively coarse. Based on the classic GAN structure, many variants have been developed. Among them, Wasserstein generative adversarial network (WGAN) [12], deep convolutional generative adversarial network (DCGAN) [11] are outstanding. These GANs use different loss functions for training, which improves the performance of GAN to some extent. We can try them and choose the one that performs best.

IV. Experiments

In this section, we design the experiments to accomplish the following objectives:

- Compare the capability of generating satellite images of different variants of GAN, WGAN and DCGAN.
- Get a qualitative understanding of what the GAN-LSTM learns to do.

A. Dataset Description

Our experiments are conducted on the FY-2E image dataset. FY-2E is a geostationary satellite and can observe an area continuously. The image we use is the China land area cloud cover image whose visual field cover the whole territory of China and surrounding area. These data can be downloaded directly from the website of the National Satellite Meteorological Centre (NSMC). These images have a resolution of 552×552 and were sampled at least every three hours.

B. Image Preprocessing

We preprocess the images before sending into the network. Because the FY-2E appears motionless to ground observers, so the range of images it take is constant. Besides, the images download from the website are already labeled with the coast line, which eliminates the hassle of uniform size and alignment position. In order to reduce the computing consumption, the images are read in grayscale. Then the pixels are normalized to be in the range $[0, 1]$. The reason for the above trick is that the cost surface will be steep in some directions and shallow in others if the input is too large, which means that the convergence will be very slow.

C. Compare of Several Kinds of GAN

Training GANs is well known for being delicate and unstable for reasons theoretically investigated in [14]. The same GANs preform different in different dataset. In this section, we test the classic GAN and two variants of the GAN model mentioned in section III. Among all the three networks, we set the patch size as 32 and each image sample is of the size 128×128. The dimension of hidden variable $\boldsymbol{z}$ is 64, which is far less than the sample size.

The training process of GAN is described in Algorithm 1, which makes some changes on the basis of [7]. The loss function of generator and discriminator is stated in Table I [15] and the training parameters are stated in Table II.

Algorithm 1 GAN Training Algorithm

Require: α, the learning rate. m, the batch size. n_D, the number of iterations of the discriminator per generator iteration.

Require: θ_{d0}, initial discriminator's parameters. θ_{g0}, initial generator's parameters.

for number of training iterations **do**

  for $t = 0, \ldots, n_D$ **do**

    Sample $\left\{\boldsymbol{x}^{(i)}\right\}_{i=1}^{m} \sim p_{data}$ a batch from the real data.

    Sample $\left\{\boldsymbol{z}^{(i)}\right\}_{i=1}^{m} \sim p_z$ a batch from the noise prior.

    $g_{\theta_d} \leftarrow \nabla_{\theta_d}\left[\frac{1}{m}\sum_{i=1}^{m}\mathcal{L}_D(\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(i)}; \theta_d, \theta_g)\right]$

    $\theta_d \leftarrow \theta_d - \alpha \cdot Optimizer\left(\theta_d, g_{\theta_d}\right)$

  end for

  Sample $\left\{\boldsymbol{z}^{(i)}\right\}_{i=1}^{m} \sim p_z$ a batch from the noise prior.

  $g_{\theta_g} \leftarrow \nabla_{\theta_g}\left[\frac{1}{m}\sum_{i=1}^{m}\mathcal{L}_G(\boldsymbol{z}^{(i)}; \theta_d, \theta_g)\right]$

  $\theta_g \leftarrow \theta_g - \alpha \cdot Optimizer\left(\theta_d, g_{\theta_g}\right)$

end for

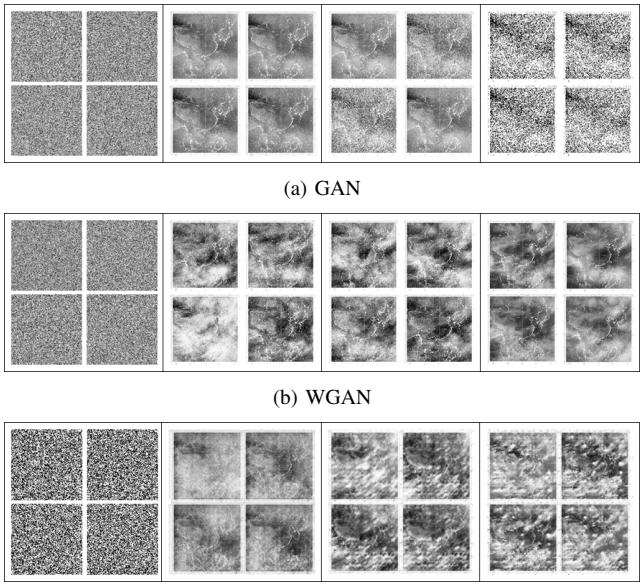
TABLE II
Training Parameters

Variant	α[1]	m[2]	n_D^3	$Optimizer$
GAN	0.001	32	1	Adam
WGAN	0.001	32	5	RMSProp
DCGAN	0.001	32	1	Adam

[1]the learning rate.
[2]the batch size.
[3]the number of iterations of the discriminator per generator iteration.

TABLE I
GENERATOR AND DISCRIMINATOR LOSS FUNCTIONS

GAN	Discriminator Loss $\mathcal{L}_D(\boldsymbol{x}, \boldsymbol{z}; \theta_d, \theta_g)$	Generator Loss $\mathcal{L}_G(\boldsymbol{z}; \theta_d, \theta_g)$
GAN	$-\mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$	$\mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$
WGAN	$-\mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[D(G(\boldsymbol{z}))]$	$-\mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[D(G(\boldsymbol{z}))]$
DCGAN	$-\mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log D_{conv}(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D_{conv}(G_{deconv}(\boldsymbol{z})))]$	$\mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D_{conv}(G_{deconv}(\boldsymbol{z})))]$



(a) GAN

(b) WGAN

(c) DCGAN

Fig. 3. Samples generated by different variants. The moments that the images are generated are, the 0th iteration, the 4000th iteration, the 8000th iteration, the 10000th iteration.

Fig. 3 shows parts of our experiment results. we chose 4 moments in the training process and generated 4 different samples at each time. We can see the evolution of our models. In the beginning, all the three networks could not generate any significant image. With the increase of training iterations, all the three GANs gradually learn some features of satellite images. Specifically, all of them learn the coarse estimation of map shape. Then we compare the outcomes of the three models, we can see that the WGAN has many superiorities. 1) The four images generated at the same iteration is very different to each other, which means the generalization capability of model is good. 2) The quality of the generated image is stable, and there is little interference noise in the pictures. 3) The contrast of pictures are much higher. The white area in the picture represents the cloud cluster and the black area for the land, so the high contrast of picture help us figure out the shape of the cloud cluster. The original GAN, on the country, shows worse generalization capability and worse image quality. Besides, with the increase of iteration, the generated images back to meaningless noise picture. In the generated samples of DCGAN, we can only figure small scale cloud cluster. While in real cloud images, the cloud usually gathered into big cluster or elongated shape. The convolutional structures are good at extract the profile and local features of objects. However, the cloud image doesn't show clear outline of cloud cluster, so the DCGAN can't learn the cloud shape in the images. In summary, the WGAN has the best generating ability in our experiment.

D. Qualitative Analysis of GAN-LSTM Model

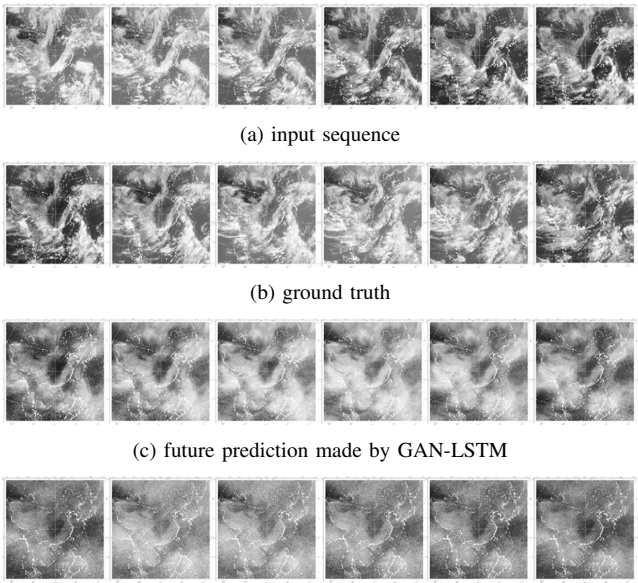
The aim of experiments in this part is to visualize the properties of proposed model. After the WGAN model is trained well, the generator will be cut from the original model and attached to the output terminal of LSTM network.

The dataset is split into 80% for training, 10% for validation and 10% for testing. The model is trained on the training set, and the network with the lowest worse on the validation set is chosen for a final evaluation on the test dataset. To find out the evolution process of clouds, the images is taken every 3 hours. If the time slot is too short, the difference between the two adjacent pictures will be too small to be noticed. The model take 8 frames as input and predicts the next 1 frame. We use mean squared error (MSE) loss function. RMSProp gives much better performance than stochastic gradient descent with momentum. The training process of GAN-LSTM is described in Algorithm 2.

Algorithm 2 GAN-LSTM Training Algorithm

Require: α, the learning rate. m, the batch size.
Require: θ_{f0}, initial LSTM's parameters.
  Train GAN with Algorithm 1.
  Cut the generator (G) of trained GAN, attach it to the LSTM (f_{lstm}).
  for number of training iterations **do**
    Sample $\left\{\boldsymbol{v}^{(i)}\right\}_{i=1}^{m} \sim p_{data}$ a batch from the real data. Where $\boldsymbol{v}^{(i)} = \widetilde{v}_{i-j+1}, \widetilde{v}_{i-j+2}, \ldots, \widetilde{v}_i$.
    $g_{\theta_f} \leftarrow \nabla_{\theta_f}\left[\frac{1}{m}\sum_{i=1}^{m} MSE(G(f_{lstm}(\boldsymbol{v}^{(i)}), \widetilde{v}_{i+1})\right]$
    $\theta_f \leftarrow \theta_f - \alpha \cdot Optimizer\left(\theta_f, g_{\theta_f}\right)$
  end for

Also, we test the traditional autoencoder-LSTM model: the encoder compresses the picture into a low-dimensional vector, and the decoder restores the picture from the low-dimensional vector. After the encoder and decoder is trained well, they will be attached into the LSTM. Then the composite model will be trained to produce desired output.



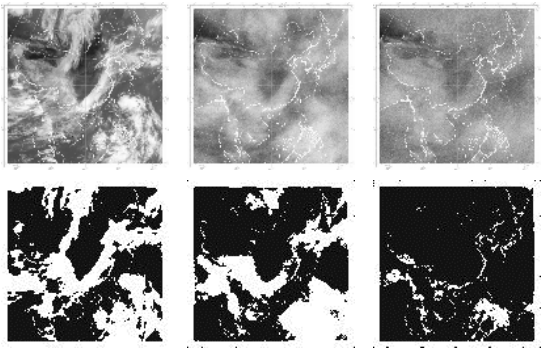
(a) input sequence

(b) ground truth

(c) future prediction made by GAN-LSTM

(d) future prediction made by autoencoder-LSTM

Fig. 4. Examples for the satellite cloud image prediction. All the predictions and ground truth are sampled every three hours.



(a) (b) (c)

Fig. 5. The satellite cloud images and their cloud segmentation image. From left to right: (a) the ground truth, (b) the prediction of GAN-LSTM, (c) the prediction of autoencoder-LSTM

Fig. 4 shows the example of running this model. The first row show the true sequences. The model takes eight frames as inputs, and only the last six frames of the input sequence are shown here. The second row is the ground truth images. The third row shows the future prediction from the GAN-LSTM model, and the last row shows the prediction results made by the autoencoder-LSTM. Every image in the last two rows is the predict result based on eight real earlier images. To evaluate the effect of the model, we use the fuzzy c-means (FCM) algorithm [16] to segment the image, which is broadly used in cloud image segmentation. The cloud image will be extracted into a 0/1 image, the white part (0) represents the cloud cluster and the black part (1) represents the area not covered by the cloud. So we call it the cloud segmentation (CS) image. We use the FCM algorithm to segment the ground truth and the predictions made by GAN-LSTM and encoder-decoder LSTM. Results are shown in Fig. 5. Then we use a commonly used

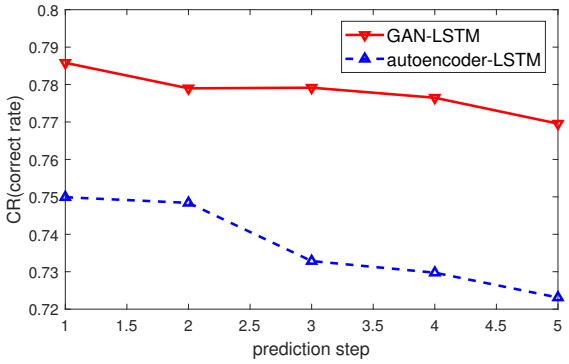
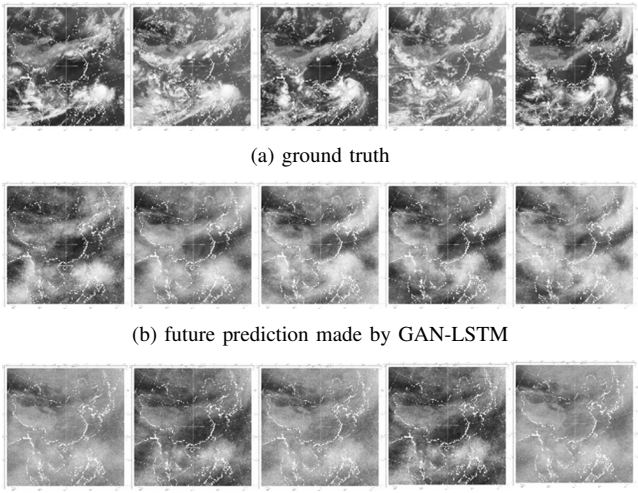


Fig. 6. Comparison of the GAN-LSTM and autoencoder-LSTM in CR (correct rate). The solid line corresponds to autoencoder-LSTM, and the dotted line corresponds to the GAN-LSTM. Each point is 400 samples average

nowcasting metric, correct rate (CR), to evaluate the similarity between the CS images of the ground truth and predictions. The score CR is defined as $CR = \frac{n_{hit}}{n_{miss}+n_{hit}}$, where n_{hit} indicates the number of pixels that have the same pixel value at the same position in the image. We average 400 results of the two models, CR of the GAN-LSTM is 0.7858 and CR of encoder-decoder LSTM is 0.7499. We know that the GAN-LSTM model performs better than the encoder-decoder LSTM.

In order to display the models capability in learning the evolutionary rule of cloud, we use the forecasting results to replace the real satellite images and investigate how the models work. In order to observe the accuracy of the model for the next few hours' prediction, we operate each model to predict the next five satellite cloud images (the interval between adjacent images is three hours). To compare the similarity between the predicted results and the ground truth, we calculate the value of CR for each step of the prediction, and each value of CR is the average of 400 samples. Result are shown in Fig. 6. We can notice that with the prediction duration increasing, the prediction accuracy of both models decreases, but the accuracy of the GAN-LSTM model is always higher than the autoencoder-LSTM model.

We display the prediction results in Fig. 7. We can see that first, the model learns the evolution of clouds. That can be seen clearly in the activity of the cloud in the lower right area of images. At the beginning, the cloud became bigger and bigger and gradually moving toward the coastline of China. Later, the cloud stops to move towards inland, and gradually dissipates in China's coastal areas. Although these results are not strict consistent with the situation at the time, the trend is correct. We know that typhoon will weaken and dissipate after landing, the model learns this rule and give the prediction. Second, the GAN-LSTM is able to handle the boundary conditions well. In real-life forecasting, there are many cases when a sudden cloud appears at the boundary, which indicates that some clouds are coming from the outside. If the network has seen similar patterns during training, it can discover this type of sudden changes in the network and give reasonable predictions in the network. Typical situation can be seen in the upper left corner.



(a) ground truth

(b) future prediction made by GAN-LSTM

(c) future prediction made by encoder-decoder LSTM

Fig. 7. Ground truth and predictions: a) Ground truth. b) Predictions obtained by GAN-LSTM. c) Predictions obtained by autoencoder-LSTM.

The input images show no clear cloud in the upper left corner at first, but the model predicts the cloud's appearance and its increasing shape. This is consistent with the actual situation. In other words, the model can figure out the evolution rule of the cloud. The auto encoder-LSTM, however, predict the same results for different steps, which seriously inconsistent with the ground truth.

E. Discussion

In our experiments, the GAN performs as a feature extraction machine. We put the latent variables into the generator and map these variables into a satellite image according to some rules, so the latent variables are the features of the satellite image. We excavate these rules through adversarial training. Results in Fig. 3 indicate that the generated images clearly show the complete outline of the Tibetan Plateau. This situation seldom appears in the real data set. This means that the generated images are not simple copies of existing data. The GAN can greatly eliminate redundant information of satellite image, which is helpful to the LSTM training process.

According to the results of typhoon cloud shown in Fig. 7, the LSTM network can learn the more complex rules of cloud evolution. As revealed by the precious studies, the moving track of typhoon is mainly related to three reasons: the underlying surface, temperature, and subtropical high-pressure belt. Specifically, the temperature can be view as constant within a certain time. In addition, the subtropical high-pressure belt retains unchanged for a relatively long time. Moreover, the underlying surface is stable. Therefore, the short-term prediction of the model can achieve a high accuracy. We can see from above that the model learns the impact of underlying surface on the typhoon.

V. Conclusions

In this paper, we have proposed a GAN-LSTM model that can learn the evolution rules of satellite images and evaluate the model on the FY-2E cloud image dataset. The use of GAN can efficiently decrease the dimension of the output data and improve the quality of the desired images. The LSTM network was able to discover the complex cloud evolution rules. Through qualitative analysis, we showed that the GAN-LSTM model outperformed the traditional autoencoder-LSTM network.

Acknowledgements

This work is supported by the China Postdoctoral Science Foundation under Grant 2018M640130, the Pre-research fund of Equipments of Ministry of Education of China (6141A02022615), the National Natural Science Foundation of China (91338203), the Young Elite Scientist Sponsorship Program by CAST (2016QNRC001), and the new strategic industries development projects of Shenzhen City under Grant JCYJ20170816151922176.

References

[1] J. Du, C. Jiang, J. Wang, Y. Ren, S. Yu, and Z. Han, "Resource allocation in space multiaccess systems," *IEEE transactions on aerospace and electronic systems*, vol. 53, no. 2, pp. 598–618, 2017.

[2] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 98–105, 2017.

[3] J. Yang, Z. Zhang, C. Wei, F. Lu, and Q. Guo, "Introducing the new generation of chinese geostationary weather satellites, fengyun-4," *Bulletin of the American Meteorological Society*, vol. 98, no. 8, pp. 1637–1658, 2017.

[4] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, 2015, pp. 802–810.

[5] J. Wang, C. Jiang, H. Zhang, Y. Ren, K.-C. Chen, and L. Hanzo, "Thirty years of machine learning: The road to pareto-optimal next-generation wireless networks," *arXiv preprint arXiv:1902.01946*, 2019.

[6] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," 1999.

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

[8] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in *International conference on machine learning*, 2015, pp. 843–852.

[9] A. Gensler, J. Henze, B. Sick, and N. Raabe, "Deep learning for solar power forecastingłan approach using autoencoder and lstm neural networks," in *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on*. IEEE, 2016, pp. 002 858–002 865.

[10] S. Feizi, C. Suh, F. Xia, and D. Tse, "Understanding gans: the lqg setting," *arXiv preprint arXiv:1710.10793*, 2017.

[11] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[12] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[13] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network." in *CVPR*, vol. 2, no. 3, 2017, p. 4.

[14] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," *arXiv preprint arXiv:1701.04862*, 2017.

[15] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, "Are gans created equal? a large-scale study," *arXiv preprint arXiv:1711.10337*, 2017.

[16] J. C. Bezdek, R. Ehrlich, and W. Full, "Fcm: The fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.