On Deep Learning Models for Detection of Thunderstorm Gale

Yan Li[1], Haifeng Li[2], Xutao Li[2], Xian Li[2], Pengfei Xie[2]

[1] School of Artificial Intelligence, Shenzhen Polytechnic, China
[2] Shenzhen Key Laboratory of Internet Information Collaboration, Harbin Institute of Technology, Shenzhen, China
liyan@szpt.edu.cn; 253154285@qq.com; lixutao@hit.edu.cn; 2696693557@qq.com; 645992705@qq.com

Abstract

The purpose of this paper is to perform a comprehensive study on the performance of different deep learning models for detection of thunderstorm gale. We construct a benchmark dataset from the radar echo images in Guangdong province of China. Each radar image is partially labeled according to the wind velocities recorded by meteorological observation stations. We design four deep learning models to address the thunderstorm gale detection problem, including a simple convolution neural network (CNN), a recurrent neural network (S-RCNN), a time context recurrent convolutional neural network (T-RCNN), and a spatio-temporal recurrent convolutional neural network (ST-RCNN). Ten traditional machine learning algorithms are selected as comparison baselines. Experimental results demonstrate that four deep learning models can achieved better detection performance than traditional machine learning algorithms.

Keywords: Radar echo images, Thunderstorm gale detection, Machine learning, Deep learning

1 Introduction

Accurate prediction the meteorological disaster is a very important and meaningful task. Thunderstorm gale is typical disaster that often produces very badly destructive damages to agricultures, industries and economics. However, the thunderstorm prediction is very challenging, compared to mesoscale meteorological phenomenon because of its small scale, rapid birth and growth. In the literature, many studies and methods have been conducted and developed by meteorological scientists to address the problem.

A typical line of studies is to leverage the priori knowledge and experiences of meteorologist for the detection. For instance, Johns et al. [1] found that the bow-like echo map features play a distinct and important role in the thunderstorm gale detection. Based on the extensive correlation analysis between thunderstorm gale and echo top, Darrah el al. [2] found an important phenomenon, namely the thunderstorm gale has a lower top than that of echo top of hail, which can be leveraged its detection. In [3], Yu et al. developed a method to detect thunderstorm gale with a combination of the mid-altitude radial convergence (MARC) and bow-like echo features. Dong et al. [4] found that vertical integrated liquid (VIL) is a distinguish factor for thunderstorm detection. If the VIL is larger than 30kg/m2, the likelihood that thunderstorm gale may appear is high. Once the VIL is larger than 40kg/m2, the confidence becomes very strong. In [5], Yan et al. showed that if the VIL and center of a storm decrease rapidly and simultaneously, a thunderstorm gale is quite likely to appear. In [6], the correlation and connection between squall line and thunderstorm gale were studied. In [7], Wang et al. showed two important features for thunderstorm gale detection: (1) the thunderstorm gale often appears in isolated storms; (2) the position of appear at the bow-like echo region or the tail part of a jet inflow. In [8], Zhou et al. showed that four features are important to detect thunderstorm gale, which are a high radar reflectivity factor, a high VIL, a fast-moving speed, a decrease of echo top and MARC. Liao et al. [9] developed a detection method based on the shape features, VIL and wind fields. In [10-11], a fuzzy logic model was developed with six heuristic features for thunderstorm gale detection. We can see that all the existing methods for thunderstorm gale detection are quite simple, heuristic and experience based, which cannot deliver robust prediction performance. Moreover, all the methods require users to construct features manually.

With the rapid development and great success of machine learning and deep learning techniques, it is interesting to study whether it is possible to leverage such techniques for thunderstorm detection. Different from the existing experience-based models, machine learning and deep learning models are able to automatically optimize their parameters based on historical observations, namely training samples. Moreover, the deep learning models can learn without a feature engineering procedure.

In this paper, we focus on the investigation of deep learning methods to the thunderstorm gale detection problem. We consider the investigation based on a

*Corresponding Author: Xutao Li; E-mail: lixutao@hit.edu.cn
DOI: 10.3966/160792642020072104001

benchmark data set from Guangdong province. As for a comparison, we also introduce ten conventional machine learning approaches as baselines. The ten approaches include Decision Tree Regressor (DT), Linear Regression (LR), Ridge regression (Ridge), Lasso regression (Lasso), Random Forest Regressor (RFR), K-nearest Neighbor Regressor (KNNR), Bayesian Ridge Regressor (BR), Adaboost Regressor (AR), Support Vector Regressor (SVR), and Gradient Boosting Regressor (GBR). To apply the ten models, ten important manual features are also constructed.

Different from conventional machine learning appraoches, which needs manual features as a prequiresite, deep learning methods are able to make predictions without a feature engineering step. Hence, we design four deep learning architectures to address the thunderstrom gale detection problem, including a simple convolution neural network architecture (CNN), a recurrent neural network architecture (S-RCNN), a time context recurrent convolutional neural network(T-RCNN), and a spatio-temporal recurrent convolutional neural network architecture (ST-RCNN). In the models, CNN simply fuses the spatial and temporal contexts for thunderstorm gale detection; S-RCNN and T-RCNN recursively exploit the spatial context and temporal context, respectively; ST-RCNN recursively leverages both the spatial and temporal contexts.

The four deep learning models are compared with the aforementioned ten traditional machine learning methods that use manual extraction features. Experimental results show that machine learning approaches can effectively identify thunderstorm gale from radar images, and the deep learning methods perform better. We note that the paper is an extension to our conference paper in [12]. The key difference is that in our conference paper we mainly focus on the comparison of conventional machine learning approaches for thunderstorm gale detection, while in this paper we introduce and design some deep learning models to address the problem.

2 Methodology

2.1 Problem Statement

The thunderstorm gale refers to the wind whose velocity is larger than 17m/s. The thunderstorm gale detection task is to find where a thunderstorm gale appears based on the radar images. In reality, the radar images are switched by the local images captured by multiple Doppler radars. For Guangdong, the switched radar image covers 700 km × 900km, which is composed of local images captured by seven radars. Each radar scans every six-minute, and the scanned height is from 500m to 19500m, where the scanned height interval is 1000m. In the radar images, each pixel represents a 1 km×1km region. Hence, every six-minute, we obtain 20 radar images of 700×900. A lot of automatic meteorological stations are distributed in Guangdong province, which can record the wind velocity, rainfall, humidity, temperature, and air pressure. Each record can be corresponded to a pixel in radar image. Based on such correspondence, we can label the wind velocity of 1 km×1km regions and thus construct the training set. Our aim is to predict the wind velocity of each region based on its radar pixel features. This is essentially a machine learning problem, i.e., regress a real value from the radar pixel features to denote the wind velocity. Specifically, we divide the radar image into $M \times N$ patches, and select K height level, and T time steps. By doing so, each sample can be represented as a tensor $X \in R^{M \times N \times K \times T}$. If we let W denote the corresponding wind velocity, the thunderstorm detection then aims to find a function $W = f(X)$. With the prediction function, we are able to calculate the wind velocity and the detection is conducted by thresholding the velocity with 17m/s.

2.2 Feature Engineering

Next, we elaborate the feature engineering procedure for machine learning approaches. In this paper, ten important features are constructed by referring the conventional experiences from meteorologists.

Radar echo intensity：This is original pixel value in radar images, which is usually ranged in [0, 80] and the unit is dBZ.

Radar reflectivity factor: The factor is proportional to the spectral diameter of raindrop. The connection of the feature to radar echo intensity is:

$$Z = 10^{\frac{dbz}{10}}$$

Radar combined reflectivity: As there are K height level radar images, we can calculate an image based on them. Each pixel in the image denotes the maximum radar reflectivity factor value of the pixels located in the same position in the K radar images. The calculated image is radar combined reflectivity.

Vertical integrated liquid (VIL): The feature denotes the integration of possible precipitation in unit column volume, which is computed as follows:

$$q_{Max} = \sum_{i=1}^{K-1} 3.44 \times 10^{-6} \times (\frac{Z_i + Z_{i+1}}{2}) \Delta h_i$$

where Z_i denotes the radar reflectivity factor of the i-th height level, Δh_i is the height differences between the i-th and (i+1)-th height level, K is the number of levels.

Radar echo top: This denotes the height that the maximal radar reflectivity factor lies in.

The change of radar echo intensity w.r.t. time: The feature is calculated as:

$$\Delta dbz = dbz_i - dbz_{i-1}$$

where dbz_i and dbz_{i-1} represents the radar echo intensities of current and previous scans, respectively. The time difference is six minutes.

The change of radar reflectivity factor w.r.t. time: The feature is formally expressed as:

$$\Delta Z = Z_i - Z_{i-1}$$

where Z_i and Z_{i-1} indicates the radar reflectivity factors of current and previous scans, respectively.

The change of radar combined reflectivity:

$$\Delta R = Z_{Max_i} - Z_{Max_{i-1}}$$

where Z_{Max_i} denotes the radar combined reflectivity of current and previous scans, respectively.

The change of VIL w.r.t. time:

$$\Delta q = q_{Max_i} - q_{Max_{i-1}}$$

Here Z_{Max_i} and $Z_{Max_{i-1}}$ denotes the maximal VILs of current and previous scans, respectively.

The decrease of echo top w.r.t. time:

$$\Delta h = h_{Max_i} - h_{Max_{i-1}}$$

Here h_{Max_i} and $h_{Max_{i-1}}$ denote the echo tops of the current and previous scans.

2.3 Traditional Machine Learning Approaches

By utilizing the above features, we can conduct experiments and examine the performance of the traditional machine learning approaches. In this paper, we adopt ten regression methods, which employ the radar features as input the corresponding wind velocity as output. Next, we give a brief introduction to the ten methods.

Decision Tree Regressor (DTR). It is a decision tree based regressor. Different from decision tree classifier, the tree grows by optimizing the mean square error of leaf nodes. In this paper, we leverage the ten types of features above to build the DTR.

Linear Regression (LR). LR is a linear regression model, which leverage the linear combination of features to predict wind velocity. Similarly, the ten types of features are employed.

Ridge Regression (Ridge). This is indeed the mean square error model with l-2 norm regularization on the parameters. By introducing the regularization term, we can avoid the non-singularity problem of the mean square error model and the results can be more stable.

Lasso Regression (Lasso). Similar to Ridge regression, Lasso regression is the mean square error model with l-1 norm regularization.

Random Forest Regressor (RFR). RFR is an ensemble-based regression model. In this model, multiple trees based regressors are built based on randomly sampled subsets (from both instance and feature views) of training samples and final prediction is obtained by averaging the outputs of the multiple regressors.

K-Nearest Neighbor Regressor (KNNR). Given a test sample, the model finds its K-nearest neighbors in the training set and utilizes their average as the final prediction.

Bayesian Ridge Regressor (BR). This is a Bayesian based ridge regression model, which can incorporate and automatically optimize the priors on feature correlations and importance.

Adaboost Regressor (AR). This is indeed an ensemble model, where multiple predictors are constructed sequentially. The (i+1)-th predictor aims to correct the mistakes made by the i-th predictor, which is achieved by sampling instances according to their derivations to the ground-truth in the i-th round, and then utilizing them to build the (i+1)-th predictor.

Support Vector Regressor (SVR). It is a regression model based on the principle of support vector machine (SVM), which is a binary classification method. Different from SVM, SVR addresses the regression problem by designing a similar objective function to SVM.

Gradient Boosting Regressor (GBR). GBR is also an ensemble learning method, which can learn multiple predictors from mistakes. The predictor is often decision trees.

2.4 Deep Learning Approaches

In this subsection, we introduce the developed deep learning approaches. Specifically, four deep learning architectures are designed, including a simple convolution neural network architecture (CNN), a recurrent neural network architecture (S-RCNN), a time context recurrent convolutional neural network(T-RCNN), and a spatio-temporal recurrent convolutional neural network architecture (SP-T-CNN). Next, we elaborate the four models, respectively.

2.4.1 Overall Procedure

Before going to the details of the models, we first explain the overall procedure of the deep learning detection models, which is shown as in Figure 1. Given a radar echo image at a time point, we first fetch the two images from two time points. As each image is composed of seven channels, putting all of them together we obtain a nine-layer radar echo image. Since our detection is pixel by pixel, we can filter out pixels that are obviously not thunderstorm gale to accelerate the detection speed. Hence, two filtering steps are conducted, namely checking whether the 30% of the radar echo intensity around the pixel is higher than zero and whether the pixel is inside a squall line region. After the filtering steps, a normalization step is performed and then the deep learning model is carried out based on the patch centered at the pixel to predict whether the pixel belongs to thunderstorm gale.

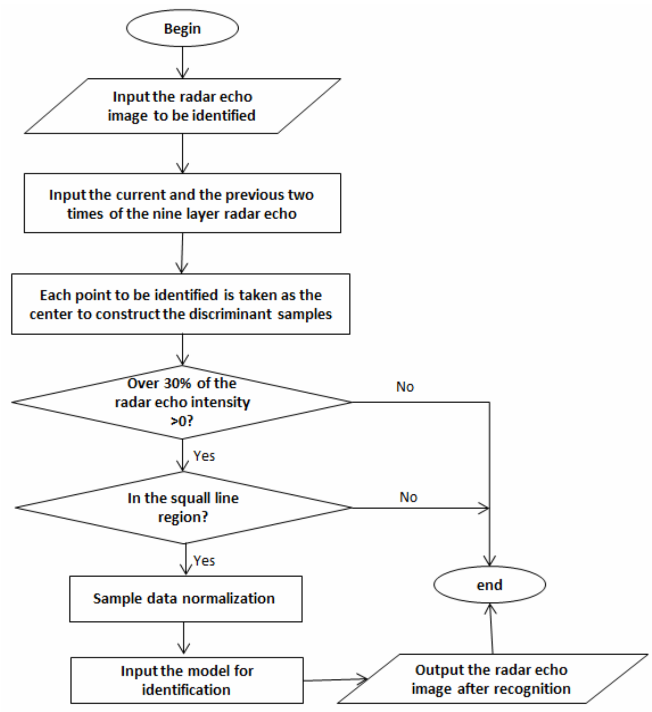


Figure 1. The procedure of deep learning-based thunderstorm gale detection

2.4.2 Convolutional Neural Network (CNN)

This model belongs to the deep learning techniques, which is composed of eight layers and the architecture is shown as in Figure 2. In CNN, we utilize 9 height levels of radar echo images and 3 time points. Hence, we have a 700-by-900-by-27 image. Then, we extract a 13-by-13-by-27 patch to predict the wind velocity at its center position. In each convolution layer, the size of convolution filter is 3x3 and the number of filters is 30. The stride number is 1. To avoid overfitting, we utilize the dropout strategy in full connection layer and the dropout ratio is 0.5. Adam optimizer is leveraged to train the network and batch size is 500. The epoch is set to be 1000. Note that different from conventional machine learning approaches, the CNN can be directly applied to radar images and does not need the handcrafted features above.

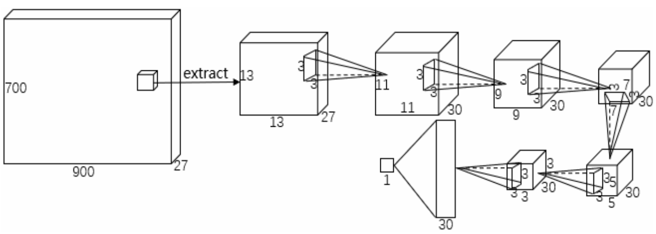


Figure 2. The architecture of CNN model

2.4.3 Spatial Context Recurrent Convolutional Neural Network (S-RCNN)

Thunderstorm gale detection systems identify the wind speed at the most central point. However, as the wind field are often continuous in space, it is important to consider the spatial context in the detection. However, it is also challenging to incorporate the context. As the radar pixels contribute differently to the center pixels in terms of their locations, we must be very careful to select the spatial range. If the range of selected image size is too large, the surrounding noise information will be introduced to interfere with the detection of thunderstorm and gale at the center position. On the contrary, if the selection range is too small, it will easily cause the loss of surrounding context information and lead to insufficient recognition accuracy. In order to make better use of the spatial context information of radar image, a spatial context recurrent convolution neural network model (S-RCNN) is designed to make full use of the surrounding information and reduce the influence of noise.

S-RCNN has the structure of recurrent convolution neural network in spatial. In particular, the convolution kernels are shrunk at multiscale to convolute and extract features at different levels. During the procedure, the original radar echo image is also scaled and concatenated at each level to exploit the features.

The architecture of S-RCNN is shown as in Figure 3. In S-RCNN, we also have a 700-by-900-by-27 image. Then, we extract a 13-by-13-by-27 patch to predict the wind velocity at its center position. In each convolution layer, the size of convolution filter is 3x3 and the number of filters is 30. The ReLU activation function is used between each convolution layer. The stride number is 1. Therefore, the feature maps generated by each convolution layer are 11-by-11-by-30, 9-by-9-by-30 and 7-by-7-by-30. For spatial recurrent structure, the patch of the original radar echo data before convolution is taken as its center 7-by-7-by-9, and then combined with the last convolution layer by concatenation operator, and then three convolution layers are connected. The size of convolution kernel in each layer is 3-by-3, and the number of filters in each layer is 30. After that, the size of the feature map generated by each convolution layer is 5-by-5-by-30, 3-by-3-by-30 and 1-by-1-by-30. The last layer is transformed into a one-dimensional vector, and the full connection layer is appended for the regression. To avoid overfitting, we utilize the dropout strategy in full connection layer and the dropout ratio is 0.5. Adam optimizer is leveraged to train the network and batch size is 500. The epoch is set to be 1000. Again, we observe that S-RCNN can be directly applied to radar images and does not need the handcrafted features using in traditional machine learning models.

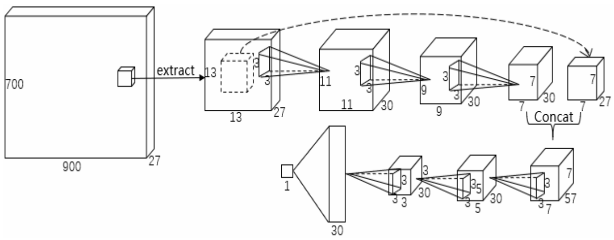


Figure 3. The architecture of S-RCNN model

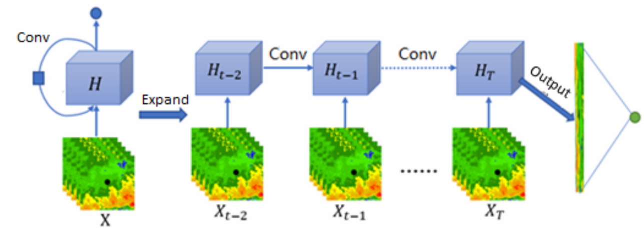


Figure 4. The Design sketch of T-RCNN

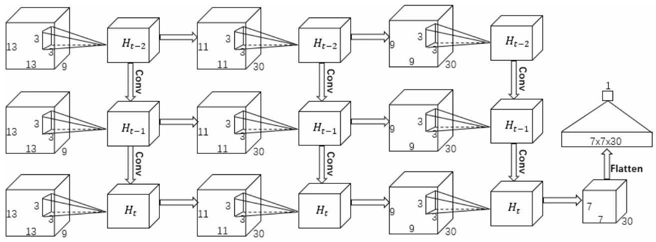


Figure 5. The architecture of T-RCNN model

2.4.4 Time Context Recurrent Convolutional Neural Network (T-RCNN)

The S-RCNN model can make full use of the spatial scale information, but in the model the radar time series context is used as channels for convolution and cannot be modeled and exploited well. For general time series problems, the commonly used deep learning models are Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). However, when dealing with two-dimensional spatiotemporal data, these models take the image as a vector for full connection directly, which is not appropriate. It will bring a lot of redundancy for spatial data, and cannot describe the local characteristics of the image. Therefore, this paper designs a time context recurrent neural network model, which considers the two-dimensional information of radar echo images as well as time series.

In T-RCNN, M × N is used to represent the height and width of each sample, K denotes the number of radar height layers, and T represents the time point. The features of thunderstorm gale samples can be expressed by the tensor $X \in R^{M\times N\times K\times T}$. The wind speed at the center position is identified by the patch of radar echo images, and then the thunderstorm gale detection problem is the wind speed detection problem by radar spatiotemporal information. Instead of treating the temporal images as channels in X, we break them into X_{t-2}, X_{t-1} and X_t as in Figure 4, and then perform RNN-like convolution to extract features as the time evolves. The detailed architecture of T-RCNN is shown as in Figure 5.

2.4.5 Spatio-Temporal Recurrent Convolutional Neural Network (ST-RCNN)

ST-RCNN regards the thunderstorm gale detection problem as an end-to-end regression problem. The input is radar echo images of multiple altitudes, multiple spatial scales and multiple time serieses, and the output is the wind speed of the location. ST-RCNN extends S-RCNN and T-RCNN by combining both strengths, namely appropriately modeling the spatial and temporal contexts at the same time.

ST-RCNN can effectively consider a wider range of spatial context through spatial recurrent convolutions, and utilize the temporal information of radar echo nice. The input data of model design is shown in Figure 6.

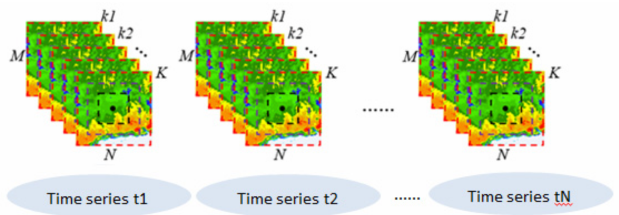


Figure 6. The organization of input data for ST-RCNN

ST-RCNN combines the design concepts of S-RCNN and T-RCNN. Figure 7 presents its design sketch, and Figure 8 gives its detailed architecture.

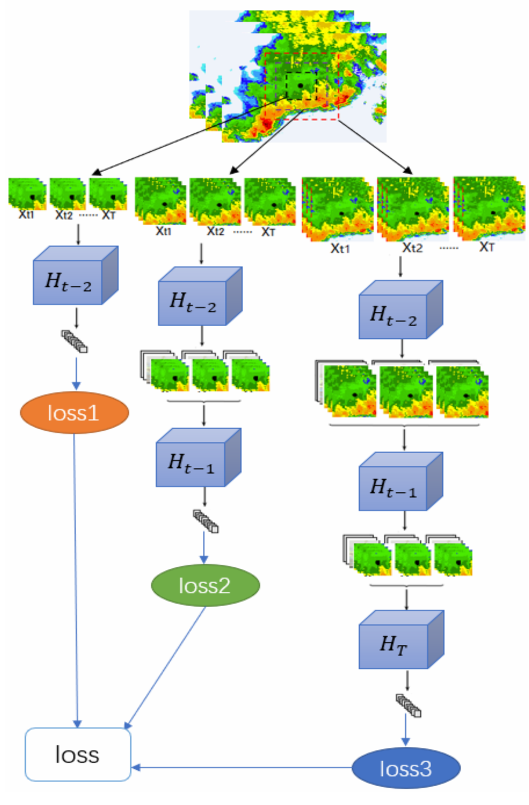


Figure 7. The Design sketch of ST-RCNN model

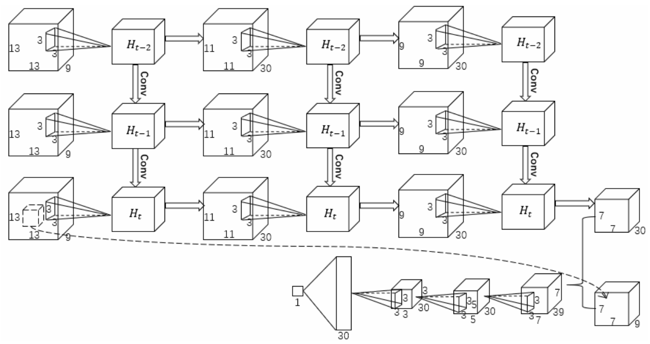


Figure 8. The detailed architecture of ST-RCNN model

In summary, this section presents four deep learning thunderstorm gale detection methods. CNN is a simple convolutional neural network model and adopts time series and height information of radar echo as channels. S-RCNN considers the spatial context of radar echo by spatial circulation. T-RCNN generates models for time series, but can't take into account the spatial context factors which are important for the detection of thunderstorms gales. ST-RCNN is a network model with cycle and convolution structure in both time series and spatial scale, which can obtain more sufficient space-time information and effectively reduce noise in spatial context. These models can not only deal with the detection of thunderstorm gale, but also can be used for classification or regression based on the characteristics of spatiotemporal series.

3 Experiment

3.1 Experimental Setup

In this paper, we utilize the radar images and data from automatic meteorological stations to construct our thunderstorm gale detection data set. For evaluation, three metrics are adopted, which are Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and R2-score. The smaller the MAE and RMSE are, the better the performance is. Larger R2-score indicates better performance. All the algorithms run on a computer with Intel Core i5-7500(3.40 GHz * 4), GeForce GTX 1080Ti GPU (12 GB) and 32 GB RAM.

3.2 Data Construction

The thunderstorm detection data set is collected by Shenzhen Meteorological Bureau, which record the radar images and data from automatic meteorological stations in Guangdong Province from 2015-2017. The resolution of each radar image is 700×900, and the spatial resolution is $0.01^{\circ}\times0.01^{\circ}$, namely each pixel denotes $1\text{km}\times1\text{km}$. From the height of 500m to 19500m, a radar image is recorded for every 1000m. Hence, we have totally 20 height level of radar images. Those images are updated every 6-minute. In automatic meteorological stations, we record the wind velocity, rainfall, temperature, air pressure, humidity. As our main aim is to identify the thunderstorm gale, we only keep the records where the wind velocity is larger than 5m/s. Then, for each automatic meteorological station, we extract a 13×13×9×3 patch from the radar images. Here the patch is centered at the location of the station, 9 indicates that we select 9 height levels and 3 means we utilize the radar images from three time points, which are current, 6-minute ago, and 12-minute ago. Hence, each sample is denoted as $13\times13\times9\times3$ tensor, as introduced in the problem statement in Section 2.1. The wind velocity of each patch (sample) is denoted by the value recorded in the corresponding station. The details of experimental data are shown in Table 1.

Table 1. Experiment data

Gale	Training Data	Validation Data	Testing Data	Total
Yes	4000	1000	1000	6000
No	16000	4000	4000	24000
Total	20000	5000	5000	30000

According to the structure of the neural network framework, four deep learning models are compared in the following experiments. Their processes are consistent, but the neural network structure and parameter settings are different, the details are shown in Table 2.

Table 2. Parameters of the four models

Model	Channels	Recurrent Time Context	Recurrent Space Context
CNN	27	No	No
S-RCNN	27	Yes	No
T-RCNN	9	No	Yes
ST-RCNN	9	Yes	Yes

3.3 Experimental Results

We conduct experiments on the constructed datasets to compare the performance of ten traditional models, namely DTR, LR, Ridge, Lasso, RFR, KNNR, BR, AR, SVR and GBR and four deep learning models, namely CNN, S-RCNN, T-RCNN and ST-RCNN. Except for deep learning models, the ten type of features constructed above are utilized. For deep learning models, they are end-to-end methods and do not need a feature engineering procedure. We record the average accuracy, precision, recall and F1-Score on the 5000 test samples of those models as final results. The results are shown in Table 3.

3.4 Experimental Analysis

From the results, we find that ST-RCNN outperforms the other models in thunderstorm gale detection. It achieves the best results in accuracy,

Table 3. The comparison results

Methods	Accuracy	Precision	Recall	F1-Score
DTR	75.2%	60.5%	63.5%	0.59
LR	77.5%	73.2%	74.3%	0.67
Ridge	77.3%	72.3%	73.6%	0.68
Lasso	78.9%	72.8%	72.4%	0.70
RFR	79.9%	76.8%	78.2%	0.72
KNNR	76.3%	74.1%	74.6%	0.69
BR	75.8%	70.2%	72.3%	0.67
AR	77.4%	77.8%	72.5%	0.69
SVR	79.2%	78.8%	76.3%	0.71
GBR	**81.3%**	**79.5%**	**78.3%**	**0.75**
CNN	76.90%	79.70%	78.10%	0.74
S-RCNN	81.80%	80.70%	80.40%	0.77
T-RCNN	82.10%	81.40%	81.50%	0.78
ST-RCNN	**83.50%**	**82.80%**	**83.40%**	**0.82**

precision, recall and F1-Score at the same time. S-RCNN and T-RCNN are also better than the ten traditional machine learning methods. GBR deliver promising performance among the ten traditional machine learning methods.

The reasons are discussed as follows: Firstly, the Recurrent Spatio-Temporal Convolutional Neural Network model can effectively integrate the temporal factors and spatial context, and can make better use of the spatiotemporal information in radar echo data. Secondly, the deep learning models learn the local features of two-dimensional images, and then abstract and combine the features level by level through local connection and weight sharing. The methodology is better than manual feature extraction utilized in traditional machine learning methods. Finally, it is found that ensemble models such as GBR and RFR perform well in thunderstorm gale detection tasks due to their good generalization ability. If the number of training samples is small, the proposed deep learning models will be slightly lower than the traditional ensemble models by manual feature extraction. When there are enough training samples, the effect of deep learning models is obviously better than the traditional machine learning methods.

3.5 Thunderstorm Gale Detection System

Based on our comparison study, we implement a thunderstorm gale detection system. The system will be deployed to meteorological bureau to help meteorological reporters. In this system, we utilize ST-RCNN as a default model, because its performance is the best according to our comparative study. The system is quite easy to use, which mainly includes three steps.

We can choose a radar image by specifying the file path of database and the filename of the image (shown as in Figure 9). After specifying the file path and filename, the system will load the corresponding radar image and visualize it. As there are 20 height levels and here, we visualize the radar image of 2500m, which is important and usefully according to meteorologists' experiences. Note that here the image is 700×900, not a small patch.

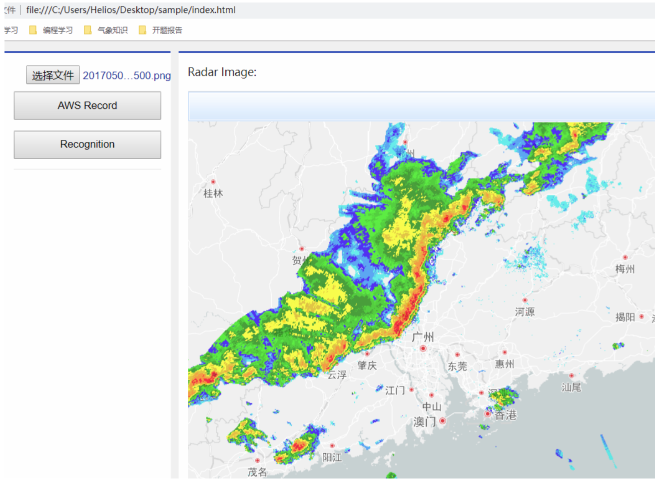


Figure 9. Choose a radar image

We can press the "Recognition" button and then the system will output the regions that have thunderstorm gale based on our CNN prediction model (shown as in Figure 10). The identified regions are marked with black colors. The procedure carried out is as follows. For each pixel, we extract a 13×13×9×3 patch centered on it, as the instructions in Section 3.2. Then, we apply the CNN model to predict the wind velocity of the patch. If the prediction is larger than 17m/s, then we label it as a thunderstorm gale; otherwise, it is not a thunderstorm gale.

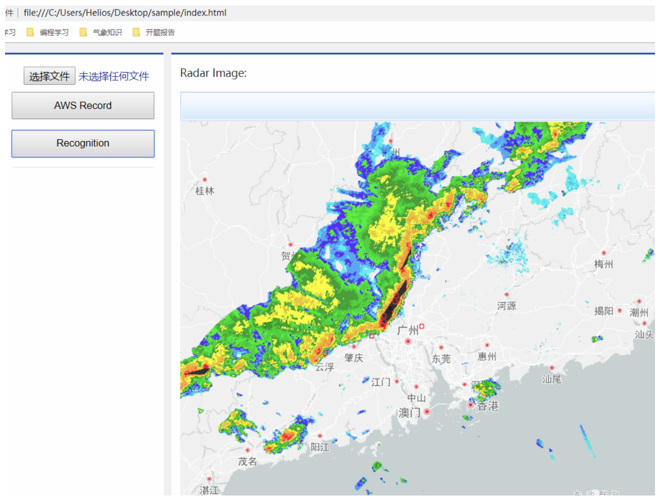


Figure 10. Thunderstorm gale detection

We can utilize the "AWS record" button to show the wind velocity records in the meteorological stations. Here the black square denotes the location where a meteorological station really records a thunderstorm gale. The red dots indicate the locations where the recorded wind velocity is larger than 10 m/s but smaller than 17 m/s. By doing so, we can visually validate whether the identified thunderstorm gale appears in reality. We note that our identified region (in Figure 10) could be larger than the ground-truth region (in Figure 11). There are two reasons for this: (i) a thunderstorm gale really appears in the identified region, but due to lack of meteorological stations, we have no ground truth for the region; in real system, we make a prediction for each pixel instead of only for the ones with automatic meteorological stations in the above comparative study. (ii) a false alarm is reported due to the wrong prediction made by our model.

4 Conclusion

In this paper, we make a comparative study by utilizing ten traditional machine learning approaches and four deep learning methods to address the thunderstorm gale detection problem. To this end, a benchmark data set is constructed by using the radar images and data from automatic meteorological stations in Guangdong from 2015 to 2017. Ten important features are extracted to apply traditional

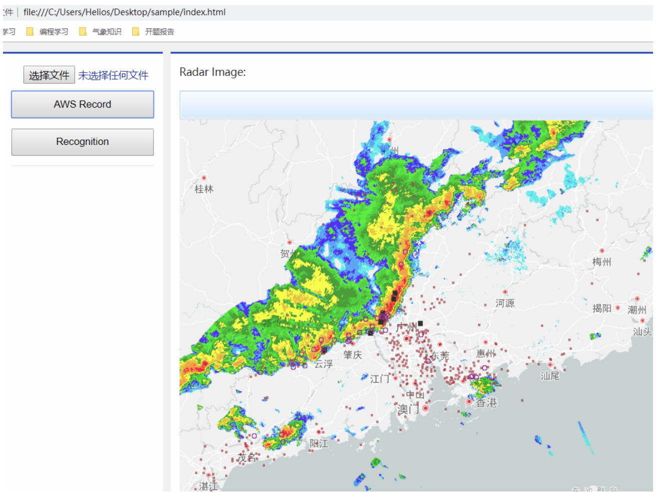


Figure 11. Visual validation

machine learning approaches. Experimental results show that the deep learning models deliver the promising performance, and ST-RCNN performs the best. Based on the comparative study, a deep learning-based thunderstorm gale detection system is developed, which is very easy to use. In the future, we are interested in designing more sophisticated deep learning models that can better exploit the spatiotemporal information for thunderstorm detection.

Acknowledgments

The research was supported by the Shenzhen Science and Technology Program under Grant JCYJ20170811160212033. In this paper, Haifeng Li is co-first author.

References

[1] R. H. Johns, W. D. Hirt, Derechos: Widespread Convectively Induced Windstorms, *Weather Forecasting*, Vol. 2, No. 1, pp. 32-49, March, 1987.

[2] R. P. Darrah, On the Relationship of Severe Weather to Radar Tops, *Monthly Weather Review*, Vol. 106, No. 9, pp. 1332-1339, July, 2009.

[3] X. Yu, X. Zhou, X. Wang, The Progress of Forecasting Technology for Thunderstorms and Strong Convections Weather, *Journal of Meteorology*, Vol. 70, No. 3, pp. 311-337, June, 2012.

[4] G. Dong, T. Wu, Application of Vertically Integrated Liquid (VIL) Water in Disastrous Wind Nowcasting, *Meteorological Science and Technology*, Vol. 35, No. 6, pp. 877-881, June, 2007.

[5] X. Yan, X. Zhang, J. Zhu, Application of CINRAD / SA Radar Storm Trend Products in Hail and Strong Wind Warning, *Meteorological technology*, Vol. 37, No. 2, pp. 230-233, February, 2009.

[6] F. Wang, S. Wu, B. Zheng, The Application of Doppler Radar data in the Prediction of Thunderstorm Gale, *Shandong Meteorology*, Vol. 26, No. 4, pp. 15-16, December, 2006.

[7] J. Wang, J. Zhang, Y. Wang, Echo Characteristics of Thunderstorm Gale Doppler Weather Radar in Eastern Hubei Province, *Rainstorm Disaster*, Vol. 28, No. 2, pp. 143-146, April, 2009.

[8] J. Zhou, M. Wei, T. Wu, Research on Doppler Radar Data Identification Method for Convective Windy Weather, *28th Annual Meeting of the Chinese Meteorological Society*, Beijing, 2011.

[9] Y. Liao, Z. Pan, Q. Guo, A Strong Convective Weather Forecast Warning Method Based on Single Doppler Weather Radar Products, *Meteorological science*, Vol. 26, No. 5, pp. 564-571, October, 2006.

[10] G. Li, L. Liu, Z. Lian, Using Radar Echo Three-dimensional Puzzle Data to Identify Thunderstorm Gale Statistics Research, *Journal of Meteorology*, Vol. 72, No. 1, pp. 168-181, February, 2011.

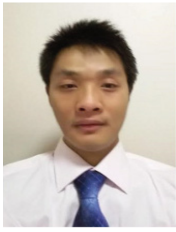
[11] G. Li, L. Liu, B. Zhang, Automatic Identification of Convective Ground Winds Based on 3D Radar Network Data, *Journal of Meteorology*, Vol. 71, No. 6, pp. 1160-1171, December, 2013.

[12] H. Li, Y. Li, X. Li, Y. Li, Y. Ye, X. Li, Pengfei Xie, A Comparative Study on Machine Learning Approaches to Thunderstorm Gale Identification, *11th International Conference on Machine Learning and Computing*, Zhuhai, 2019, pp. 12-16.

Biographies

Yan Li received the Ph.D. degree in the Department of Computing, Hongkong Polytechnic. She is currently an Associate Professor in Shenzhen Polytechnic. Her research interests include data mining and machine learning.

Haifeng Li received the Bachelor degree from Tianjin Polytechnic University, Tianjin, China in 2016. He is currently pursuing the Master degree in the Shenzhen Graduate School, Harbin Institute of Technology. His research interests is image processing.



Xutao Li is now an Associate Professor in the Shenzhen Graduate School, Harbin Institute of Technology. He received the Ph.D. and Master degrees in Computer Science from Harbin Institute of Technology in 2013 and 2009, and the Bachelor from Lanzhou University of Technology in 2007. His research interests include data mining, machine learning, graph mining and social network analysis, especially tensor based learning and mining algorithms.

Xian Li received the Bachelor degree fromHenan University, Kaifeng, China in 2018. He is currently pursuing the Master degree in the Shenzhen Graduate School, Harbin Institute of Technology. His research interests include data mining, computer vision and remote sensing.

Pengfei Xie received the Master degree from HIT, ShenZhen, China in 2020. He currently works as an algorithm engineer in Alibaba group. His research interests include data mining, machine learning.