Long-Term Rainfall Forecast Model Based on The TabNet and LightGbm Algorithm

Tianyu Xu (835884315@qq.com)
Beijing University of Technology
Yongchuan Yu
Beijing University of Technology
Jianzhuo Yan
Beijing University of Technology
Hongxia Xu
Beijing University of Technology

Research Article

Keywords: AUC, rainfall, forecast

Posted Date: November 24th, 2020

DOI: https://doi.org/10.21203/rs.3.rs-107107/v1

Long-term rainfall forecast model based on the TabNet and LightGbm algorithm

Jianzhuo Yan[1], Tianyu Xu[1]*, Yongchuan Yu[1], Hongxia Xu[1]

[1] Beijing University of Technology, Faculty of Information Technology, Beijing, 100124, China
*Correspondence to [835884315@qq.com]

Abstract

Due to the problems of unbalanced data sets and distribution differences in long-term rainfall prediction, the current rainfall prediction model had poor generalization performance and could not achieve good prediction results in real scenarios. This study uses multiple atmospheric parameters (such as temperature, humidity, atmospheric pressure, etc.) to establish a TabNet-LightGbm rainfall probability prediction model. This research uses feature engineering (such as generating descriptive statistical features, feature fusion) to improve model accuracy, Borderline Smote algorithm to improve data set imbalance, and confrontation verification to improve distribution differences. The experiment uses 5 years of precipitation data from 26 stations in the Beijing-Tianjin-Hebei region of China to verify the proposed rainfall prediction model. The test set is to predict the rainfall of each station in one month. The experimental results shows that the model has good performance with AUC larger than 92%. The method proposed in this study further improves the accuracy of rainfall prediction, and provides a reference for data mining tasks.

Introduction

Rainfall is an important parameter in weather forecasting and flood control. How to obtain precipitation information more quickly and accurately has attracted more and more attention from meteorological researchers[1,2]. Nowadays, meteorological disasters such as droughts and floods frequently occur and cause serious losses. This requires

further improvement of the accuracy of weather forecasts[3]. Rainfall is affected by many key factors, such as hydrology, location, circulation, etc., and is a nonlinear system[4]. Therefore, it is of great significance to establish an accurate and good generalized rainfall prediction model[5, 6].

At present, there are various methods for predicting the probability of rainfall.Suning Liu[7] developed a recursive approach to long-term prediction of monthly precipitation using genetic programming.Hongya Li[8] used Multicellular Gene Expression Programming algorithm for modeling the historical precipitation data series decomposed by Empirical Mode Decomposition.Bo Xiang [9] used the rainfall data from 2011 to 2018 in Chongqing, China, and established a rainfall prediction model based on LightGbm.Guohui Li[10] combined the variational mode decomposition, the improved butterfly optimization algorithm, the least squares support vector machine model predicted the precipitation of two stations in Shaanxi Province

With decades of data accumulation, neural networks stand out among many methods by virtue of their excellent processing capabilities for massive data.Jinle Kang[11] deployed Long Short-Term Memory (LSTM) network models for predicting the precipitation based on meteorological data from 2008 to 2018 in Jingdezhen City. Yongtao Wang[12] innovatively combines the artificial bee colony (ABC) algorithm and the backpropagation neural network into a precipitation prediction model. Yang Liu [13] used the BP-NN algorithm and added the Precipitable water vapor (PWV) feature to establish a high-accuracy short-term rainfall prediction model. The above research results show that the rainfall forecast model based on machine learning is practical and reliable.

Due to the problems of unbalanced data sets and distribution differences in long-term rainfall prediction, the current rainfall prediction model had poor generalization performance and could not achieve good prediction results in real scenarios. This paper has made improvements in the following aspects: (1) Using the method of model fusion to fuse the TabNet network and the LightGbm[27] has obtained better generalization ability (2) Using adversarial verification to improve the distribution difference (3) Generating descriptions Statistical features and the use of feature fusion to improve model accuracy (4) Use Borderline SMOTE algorithm to improve data imbalance.

Theory of Tabnet algorithm

Topological structure of the TabNet algorithm. At present, deep neural networks

have achieved great success in images[23], text[24], and audio[25]. However, for tabular data sets, tree models are still mainly used. In many data mining competitions, xgboost and LightGbm rely on its (1) Fit the hyperplane boundary in tabular data well (2) Good interpretability (3) Fast training speed becomes the first choice among many algorithms. For traditional DNN, blindly stacking network layers can easily lead to model over-parameters, resulting in DNN performance on the tabular data set is not satisfactory. In August 2019, the Tabnet network proposed by SercanÖ. Arık[14], on the basis of retaining the end-to-end and representation learning characteristics of DNN, it also has the advantages of tree model interpretability and sparse feature selection. Gradually become the first choice for tabular data tasks.

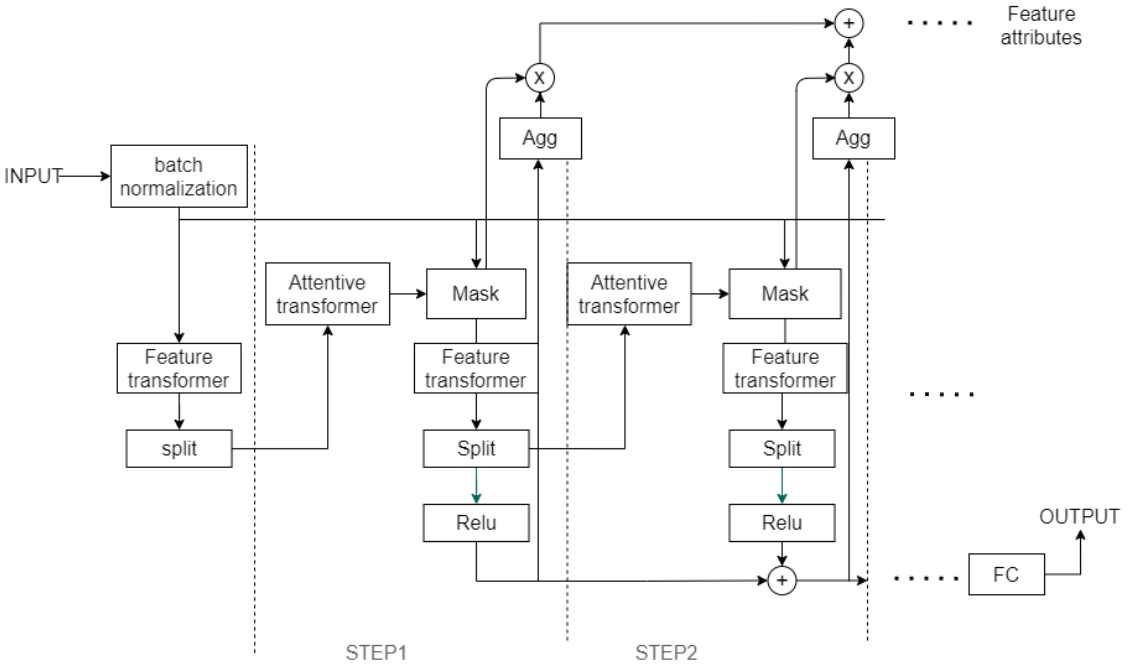

Figure 1. Topological structure of the TabNet algorithm. [Te fgure is plotted by Draw.io].

The input of the model is $B * D$, where B is the batch size and D is the dimension of the feature; and the output of the model is a vector or a number (classification or regression task).

Figure 1 shows that the TabNet network is mainly composed of Feature transformer layer, Split layer, Attentive transformer layer, and mask layer.

(1) Feature transformer layer: Feature calculation, split for the decision step output and information for the subsequent step[14].The structure is shown in figure 2:

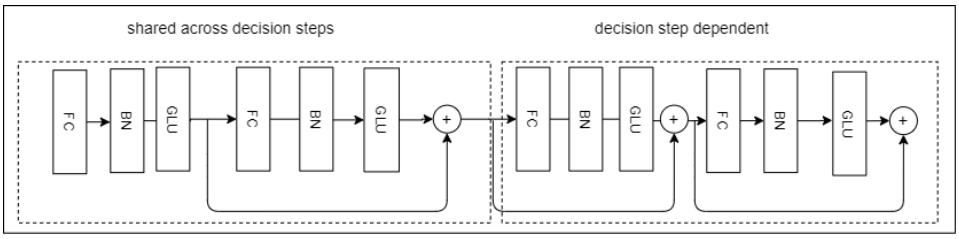

Figure 2. Topological structure of the Feature transformer layer. [Te fgure is plotted by Draw.io]

It can be seen that the Feature transformer layer consists of two parts. The parameters of the first half of the layer are shared, which means that they are jointly trained on all steps; while the second half is not shared, and is trained separately on each step. For each step, the input is the same features, so we can use the same layer to do the common part of feature calculation, and then use different layers to do the feature part of each step.

GLU is a gated linear unit [25], which is based on the original FC layer plus a gating. The residual connection is used in the layer, and it is multiplied by $\sqrt{2}$ to ensure the stability of the network.

The Feature transformer layer realizes the calculation of the features selected in the current step. A decision tree constructs a combination of the size relationship of a single feature, and does not consider more complex situations. Therefore, TabNet uses a more complex Feature transformer layer to perform feature calculations. In some feature combinations, it does better than decision trees.

(2) Attentive transformer layer: feature selection, the function of this layer is to calculate the Mask layer of the current step based on the result of the previous step, The structure is shown in figure 3.

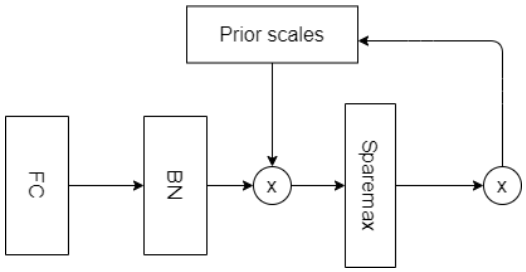

Figure 3. Topological structure of the Attentive transformer layer. [Te fgure is plotted by Draw.io].

We use a learnable mask layer to select salient features, and through the selection of sparse features, the learning of the model is more effective in each step[14]. According to the structure of the Attentive transformer layer, its calculation formula can be written as:

$$\mathrm{M[i]} = \ \mathrm{Sparsemax}\big(\mathrm{P[i-1]} * \ h_i(\mathrm{a[i-1]})\big) \tag{1}$$

Sparseamx is the sparseness of Softmax， encourages sparsity by mapping the Euclidean projection onto the probabilistic simplex[15], $h_i(.)$ represents the FC+BN layer, where P[i−1] is divided by the Split layer in the previous step, and p[i-1] is the Prior scales item, which is used to indicate the application of a certain feature in the previous step degree.

$$\mathrm{P[i]} = \ \prod_{j=1}^{i}(r - M[j]) \tag{2}$$

If a feature has been used many times in the previous step, it should no longer be selected by the model. Therefore, the model uses this Prior scales item to reduce the weight ratio of this type of feature. It can be seen from the formula, If r=1, then each feature can only be used once.

The Attentive transformer layer can obtain the Mask matrix of the current step according to the results of the previous step, and try to make the Mask matrix sparse and non-repetitive. The Mask vector of different samples can be different, which means that TabNet can allow different samples to choose different features instance-wise, and this feature is not available in tree models. For additive models such as XGBoost, a step is a tree. , And the features used in this decision tree are selected on all samples (for example, by calculating information gain), it cannot be instance-wise[28].

(3) Split layer: Cut the vector output from the Feature transformer layer into two parts.

$$\big[\mathrm{d[i]}, \mathrm{a[i]}\big] = f_i(M[i] * f) \tag{3}$$

$\mathrm{d[i]}$ will be used to calculate the final output of the model, and $\mathrm{a[i]}$ will be us3ed to calculate the Mask layer of the next step

Feature attribute outputs the global importance of the feature. The model first sums the output vector of a step to obtain a scalar. This scalar reflects the importance of this step to the final result. Multiplied by the Mask matrix of this step reflects the importance of each feature in this step. Add up the results of all the steps to get the global importance of the feature.

Formula 4 is the contribution of the i-th step to the final result.

$$\varphi_b[i] = \ \sum_{c=1}^{N_d} ReLu(d_{b,c}[i]) \tag{4}$$

The global importance of the normalized feature can be expressed as formula 5.It also proves the interpretability of TabNet

$$M_{agg-b,j} = \sum_{i=1}^{N_{steps}} \varphi_b[i] * M_{b,j} / \sum_{j=1}^{D} \sum_{i=1}^{N_{steps}} \varphi_b[i] * M_{b,j} \quad (5)$$

In general, TabNet uses a sequential multi-step framework to construct a neural network similar to an additive model. The key points in the model are the Attentive transformer layer and the Feature transformer layer.

Self-supervised learning of the TabNet algorithm. TabNet applies a self-supervised learning method to obtain the representation of tabular data through the encoder-decoder framework, which is also helpful for classification and regression tasks.

Different features of the same sample are related, so self-supervised learning is to first mask some features, and then use the encoder-decoder model to predict the masked features. The encoder model trained in this way can effectively characterize the features of the sample and enhance the performance of the model. The encoder model for self-supervised learning is figure 1, and the decoder model is shown in figure 4:

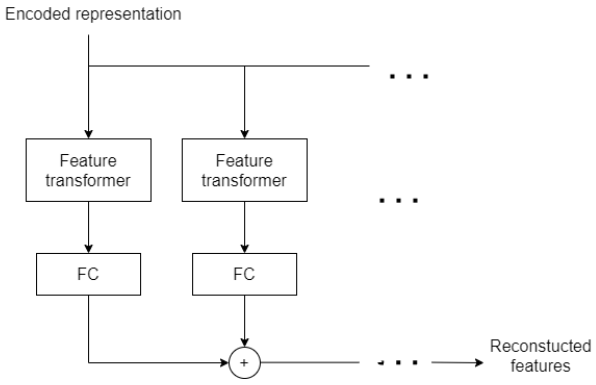

Figure 4. Topological structure of the decoder. [Te fgure is plotted by Draw.io].

The encoded representation is the sum vector of the encoder without the FC layer. The encoded representation is used as the input of the decoder. The decoder uses the Feature transformer layer to reconstruct the representation vector into a feature. After the addition of several steps, we will output the reconstructed freature.

The matrix for masking the feature is $S \in \{0,1\}^{B*D}$,the feature data is f ,then the input of the encoder is $(1 - S) * f$, if the final decoder output is $\hat{f}$, then self-supervised learning is to reduce the true value $S * f$ and The differen{, #1}ce between the reconstruction value $S * \hat{f}$ considering that the magnitude of different features is not necessarily the same, so the regularized MSE is used as the loss.

$$\sum_{b=1}^{B} \sum_{j=1}^{D} |(\hat{f}_{b,j} - f_{b,j}) * S_{b,j} / \sqrt{\sum_{b=1}^{B} (B * \sum_{b=1}^{B} f_{b,j})^2}|^2 \quad (6)$$

In addition, in order for the model to learn the representation method of the entire feature data, in the training process of self-supervised learning, the matrix S will be resampled every round to ensure the overall representation ability of the encoder model.

Feature engineering.

This paper selects meteorological data from 26 stations in the Beijing-Tianjin-Hebei region of China as the research object. The data comes from the Beijing Environmental Planning Center. The data time span is from January 2012 to December 2016. The data is collected once a day. The feature dimension is 30, which consists of geographic features (longitude, latitude, height of the station, prefecture-level city, province, etc.) and meteorological features (evaporation, surface temperature, air pressure, humidity, wind speed, wind direction, etc.). The training set is from January 2012 to November 2016, and the test set is December 2016.

Feature engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work. Commonly used feature engineering methods include feature selection, feature extraction, and feature construction. The main innovation of this article is feature construction.

Descriptive Statistical Features. Figure 5 shows the correlation between each feature and the probability of rainfall

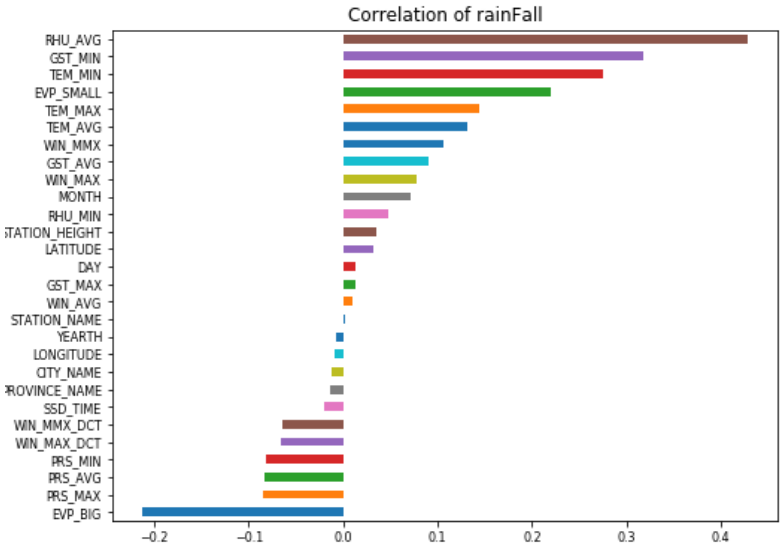

Figure 5. Correlation of each feature with the probability of rainfall [Te fgure is generated by matplotlib].

As shown in Figure 3, the probability of rainfall is determined by many factors, but the probability of rainfall has a relatively high correlation with RHU_AVG(average relative humidity), GST_MIN(minimum surface temperature), TEM_MIN(minimum atmospheric temperature), and EVP_SMALL(small-scale evaporation). The characteristics are combined with the discrete characteristics to generate descriptive statistical characteristics, as shown in Figure 5, which generates the average and standard deviation of the humidity of each station.

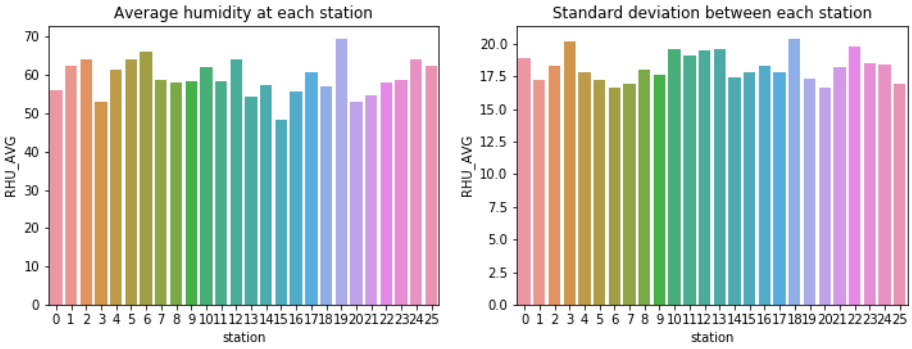

Figure 6. Average and standard deviation of humidity at each station[Te fgure is generated by matplotlib].

Figure 6 shows that the average value and standard deviation of each group(group humidity according to measuring station) have a certain variance. For this type of data, station-humidity-mean (average humidity of each station), station-humidity-std (standard deviation of humidity between stations)) As a data enhancement, it can effectively improve the accuracy of classification. After experiments, we finally selected the city, observation site, and wind direction as the grouping base point, grouped the average relative humidity, the lowest atmospheric temperature, and the small-scale evaporation, and calculated the mean and standard deviation of each group. Increase the data set dimension by 18 dimensions.

Feature fusion. Precipitable water vapor (PWV)[16] is an important meteorological parameter. Abundant water vapor is the basic condition for the formation of rainfall and strong convective weather processes. Therefore, PWV is one of the important data needed for weather forecasting.

zenith total delay (ZTD) occurs as the The Global Navigation Satellite System (GNSS) signal is affected by the atmospheric refraction when it passes through the troposphere, ZTD includes zenith hydrostatic delay (ZHD) and Zenith Wet Delay ZWD[17]. ZHD accounts for approximately 90% of ZTD[18]. ZHD can be calculated by

using the (7):

$$\mathrm{ZHD} = \frac{0.0022768 * p_w}{1 - 0.002266 * \cos(2\phi) - 0.00028 * H} \tag{7}$$

where PW is the surface pressure of the station with a unit of °C, ϕ refers to the latitude of the station with a unit of radian, and H is the geodetic height of the station with a unit of km. Therefore, ZWD can be obtained by extracting ZHD from ZTD, and PWV can be calculated by using the (8):

$$\mathrm{PWV} = \frac{\Pi * ZWD}{P_W} \tag{8}$$

where ρW is the water vapor density, and Π represents the conversion factor:

$$\Pi = [-1 * \mathrm{sgn}(\phi) * 1.7 * 10^{-5} * |\phi|^{H_f} - 0.0001] * \cos\left(\frac{DoY - 28}{365.25} * 2\pi\right) + [0.165 - (1.7 * 10^{-5}) * |\phi|^{1.65}] + (-2.38 * 10^{-6}) * H \tag{9}$$

H_f is an empirical parameter, which is approximately 1.48 in the northern hemisphere.We combine pressure, latitude, height of the station features into ZTD features according to formulas, and combine date, height of the station and ztd into PWV features according to formulas. PWV and ZTD further improve the accuracy of the model.

Model training

Evaluation index. The ROC (Receiver Operating Characteristic) curve is a common indicator in imbalanced data sets [21]. Its horizontal axis is FPR (False Positive Rate) and the vertical axis is TPR (True Positive Rate). The calculation formulas for FPR and TPR are (10),(11).

$$\mathrm{TPR} = \frac{TP}{TP + FN} \tag{10}$$

$$\mathrm{FPR} = \frac{FP}{TN + FP} \tag{11}$$

Among them, TP is a real example, FN is a false negative example, FP is a false positive example, and TN is a true negative example.

AUC is the sum of the area of each part under the ROC curve. This article uses AUC as the evaluation index of the model.

Unbalanced data set. Unbalanced data is common in financial risk control, anti-fraud, advertising recommendations and medical diagnosis. Generally speaking, the proportions of positive and negative samples of unbalanced data are very different. For

models, models built with unbalanced data are more willing to favor the labels of multi-category samples, which has low practical application value. Figure 5 shows the rainfall probability distribution.

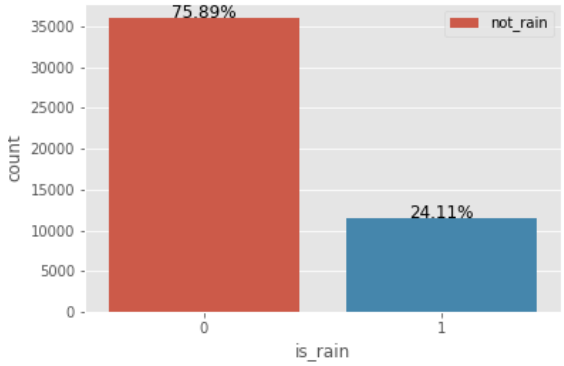

Figure 7. Rainfall probability distribution [Te fgure is generated by matplotlib]

Figure 7 shows the number of days without rain accounted for 75.89%, and the number of days with rain only accounted for 24.11%. The difference between positive and negative samples is huge. The processing methods of unbalanced data sets are mainly divided into oversampling and undersampling

This research uses the Borderline SMOTE[19] algorithm to deal with the problem of data imbalance. Borderline SMOTE is an improved algorithm based on SMOTE (Synthetic Minority Over-sampling Technique), which belongs to oversampling. The basic idea of oversampling is to randomly select samples from a minority class to add new samples.

The basic idea of SMOTE[20] is to use K-Nearest Neighbor to analyze minority samples, and artificially synthesize new samples based on minority samples and add them to the data set.

$$X_{new} = X + B * (X' - X) \quad (12)$$

Where B is a random function, generally a random decimal number from 0 to 1, and X' is a sample randomly selected by the K-Nearest Neighbor.

However, the SMOTE algorithm has an important flaw. It treats minority samples equally and does not consider the category information of neighboring samples. Sample aliasing often occurs, resulting in poor classification performance. In order to improve this problem, this study uses Borderline SMOTE to improve Data imbalance.

Borderline SMOTE is an improved oversampling algorithm based on SMOTE, which uses only a few samples on the border to synthesize new samples.The Borderline SMOTE sampling process divides the minority samples into 3 categories, namely Safe,

Danger and Noise, and only oversamples the minority samples of Danger.

(1) Safe: More than half of the samples are minority samples, point B in figure 8.

(2) Danger: More than half of the samples around are majority samples, which are regarded as samples on the boundary,point C in figure 8.

(3) Noise: The samples are surrounded by most types of samples, which are regarded as noise.,point A in figure 8.

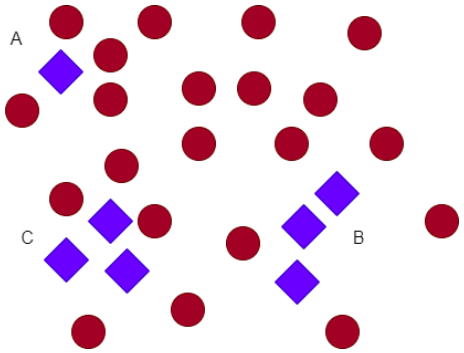

Figure 8.Borderline SMOTE sampling example [Te fgure is generated by Draw.io]

After using the Borderline SMOTE algorithm to process the training set, the ratio of the number of non-rainy days (0) to the number of rainy days (1) is shown in the figure 9, which shows that the problem of data imbalance has been improved.

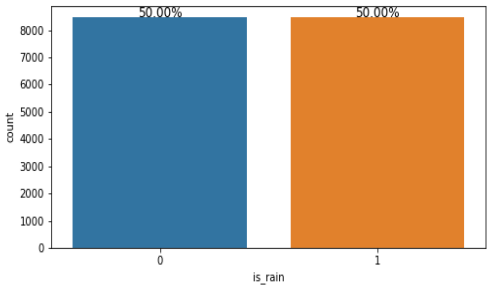

Figure 9. Borderline SMOTE improved data set [Te fgure is generated by matplotlib]

Adversarial verification. Long-term rainfall prediction will face the problem of the difference in the distribution of the data set, which leads to a large variance between the training set and the test set, the model is unstable, and has no practical application value. This study uses the method of confrontation verification to improve this problem.

We add labels to the training set and test set to distinguish the data set as a training set or a test set. For example, 'is_test' = 0 is the training set and 'is_test' = 1 is the test set, and then merge the two data sets , then train a model (Lightgbm in this study) to do classification prediction on 'is_test'. If AUC is larger than 0.7, it means that the distribution of the training set and the test set is quite different. If AUC=0.5, it means

that there is no obvious distribution between the training set and the test set. The difference, the AUC of this study is 0.89, proves that the data set has obvious distribution differences.

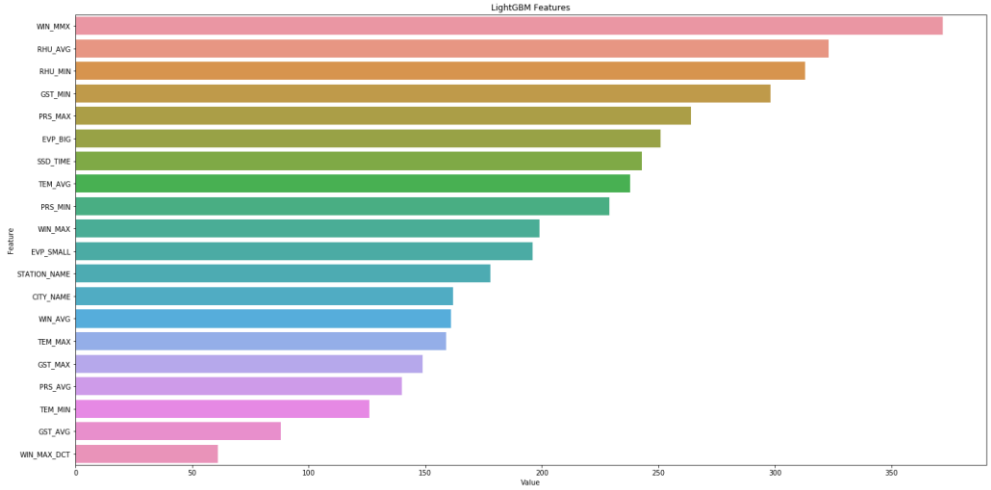

Figure 10. Feature importance distribution [Te fgure is generated by matplotlib]

Figure 10 shows the WIN_MAX (maximum wind speed) contributes the most to the classification of the model, indicating that the WIN_MAX feature has the largest distribution difference between the training set and the test set, so we delete the WIN_MAX feature to reduce the distribution difference between the training set and the test set. Only through the above figure, we found that RHU_AVG (average relative humidity) and RHU_MIN (minimum relative humidity) also have a greater contribution to the model, but if you delete these two features, it will seriously reduce the model's fit in the training set, and the AUC of the test set will also be with this reduction, the confrontation verification is obviously not orthogonal.

Hyperparameter. The TabNet neural network constructed in this research is based on[22]，we use a pre-defned hyperparameter search space.Tabel 1 shows the selected parameters.

Hperparameter	**Value**
n_d(Width of the decision prediction layer)	24
n_a(Width of the attention embedding for each mask)	24
Gamma(This is the coefficient for feature reusage in the masks)	1.3
optimizer	Adam
learning_rate	0.01
n_steps(Number of steps in the architecture)	3

Table 1. TabNet network parameters

n_d, n_a, n_steps are important parameters that determine the capacity of the model. For most data sets, n_steps range from 3 to 10 is a reasonable parameter, and n_d = n_a is a reasonable choice[14], in order to obtain high performance of the model, To reduce over-fitting, it is necessary to appropriately adjust the capacity of the model. Reducing n_d, n_a, n_steps is an effective way to improve the generalization ability of the model without significantly reducing the accuracy.

gamma determines the selection strength of sparse features, A value close to 1 will make mask selection least correlated between layers. Values range from 1.0 to 2.0.

Model fusion. This research also trained the LightGbm model for model fusion with TabNet，LightGbm is a fast, distributed, high-performance gradient Boosting algorithm based on decision trees. Neural networks and decision tree models often have diversity and differences.The LightGbm model and TabNet model are merged to make the final model better promote the unknown data, reduce over-fitting, and increase generalization ability. The model fusion method used in this research is to avarge the output of two models.Figure 11 is a flowchart of the entire experiment.

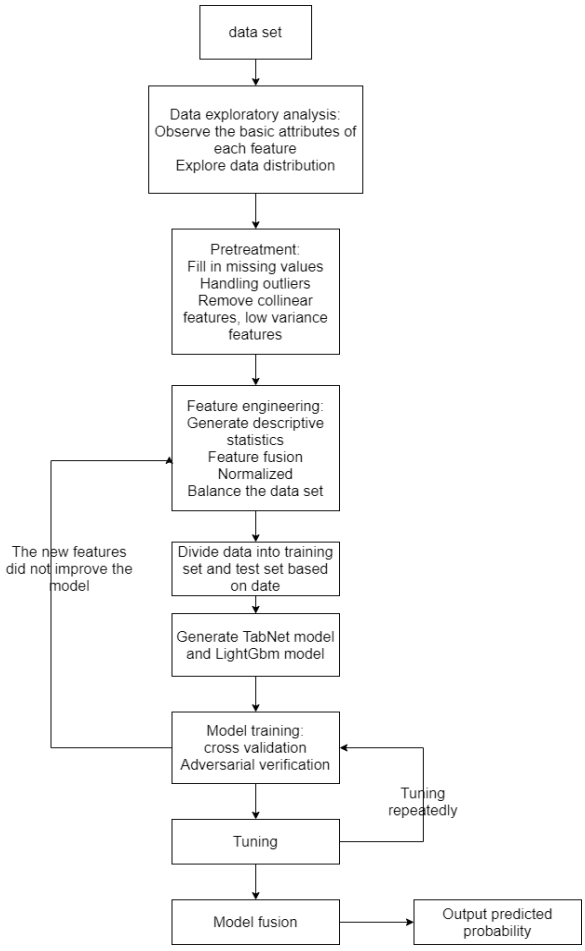

Figure 11. Experimental flowchart [Te fgure is generated by draw.io]

Experimental results. This study uses BP-NN[29], LSTM[30], LightGbm as comparative experiments. Table 2 shows the experimental results which are the mean values of three-fold cross-validation.

Algorithm	Training set	Validation set	Test set
BP-NN	89.5273	88.1298	86.7482
LSTM	90.3298	89.3726	87.8812
LightGbm	94.0712	93.0056	91.0313
TabNet	95.2145	94.1147	91.8567
TabNet-LightGbm	95.4513	94.2271	92.7817

Table 2. Comparative Experiment

Table 2 shows that the LightGbm model and TabNet model used in this experiment obtained 91.0313% and 91.8567% AUC on the test set, respectively, which is about 5% higher than the BP-NN and LSTM models used in the comparative experiment. Similarly, after the model fusion of TabNet and LightGbm, the AUC of the model has been slightly improved, especially in the test set, which reduces overfitting and proves the feasibility of model fusion. These results proves that using the fusion model of LightGbm and TabNet to predict the probability of rainfall can achieve good results.

Conclusion

Rainfall is affected by a variety of meteorological factors and is a complex nonlinear system. Therefore, this paper proposes a prediction model that uses a combination of TabNet neural network and LightGbm decision tree, and uses feature engineering to generate descriptive statistical features to improve the model's performance Accuracy, using feature fusion, mining the potential value of each feature to improve the upper limit of the model, using the Borderline SMOTE algorithm to improve the imbalance of the data set. In the training phase, adversarial verification is used to improve the distribution difference between the training set and the test set. Finally, the prediction results of LightGbm and TabNet are averaged to reduce the impact of overfitting. The performance of the model is 0.9278 AUC. This result proves the reliability of using the hybrid model of TabNet and LightGbm to predict precipitation, and provides a new method for data mining tasks. In future research, more data, better parameters, and more reasonable feature engineering methods should be used to increase the generalization ability of the model.

Reference

[1] Jiang, T., Su, B. & Hartmann, H. Temporal and spatial trends of precipitation and river flow in the Yangtze River Basin, 1961–2000. *Geomorphology* **85**, 143-154 (2007).

[2] Xingchuang, X. U., Xuezhen, Z., Erfu, D. & Wei, S. Research of trend variability of precipitation intensity and their contribution to precipitation in China from 1961 to 2010. *Geographical Research* **33**, 1335-1347 (2014).

[3] Pranatha, M. D. A., Pramaita, N., Sudarma, M. & Widyantara, I. M. O. Filtering Outlier Data Using Box Whisker Plot Method for Fuzzy Time Series Rainfall Forecasting. *2018 4th International Conference on Wireless and Telematics (ICWT)*(2018).

[4] Maheswaran, R. & Khosa, R. A Wavelet-Based Second Order Nonlinear Model for Forecasting Monthly Rainfall. *Water Resources Management* **28**, 5411-5431 (2014).

[5] Qiu, J., Shen, Z., Wei, G., Wang, G. & Lv, G. A systematic assessment of watershed-scale nonpoint source pollution during rainfall-runoff events in the Miyun Reservoir watershed. *Environmental ence & Pollution Research International* **25**, 6514 (2018).

[6] Chau, K. W. & Wu, C. L. A hybrid model coupled with singular spectrum analysis for daily rainfall prediction. *Journal of Hydroinformatics* **12**, 458-473 (2010).

[7] Liu, S. & Shi, H. A Recursive Approach to Long-Term Prediction of Monthly Precipitation Using Genetic Programming. *Water Resources Management* **33**, 1103-1121 (2019).

[8] Li, H., Peng, Y., Deng, C., Pan, Y. & Zhang, H. *Multicellular Gene Expression Programming-Based Hybrid Model for Precipitation Prediction Coupled with EMD*. (Springer, Cham, 2018).

[9] Xiang, B., Zeng, C., Dong, X. & Wang, J. The Application of a Decision Tree and Stochastic Forest Model in Summer Precipitation Prediction in Chongqing. *Atmosphere* **11**, 508 (2020).

[10] Li, G., Chang, W. & Yang, H. A Novel Combined Prediction Model for Monthly Mean Precipitation with Error Correction Strategy. *IEEE Access*, 1-1 (2020).

[11] Kang, J., Wang, H., Yuan, F., Wang, Z. & Qiu, T. Prediction of Precipitation

Based on Recurrent Neural Networks in Jingdezhen, Jiangxi Province, China. *Atmosphere* **11**, 246 (2020).

[12] B, Y. W. A., A, J. L., A, R. L., A, X. S. & A, E. L. Precipitation forecast of the Wujiang River Basin based on artificial bee colony algorithm and backpropagation neural network. *Alexandria Engineering Journal* **59**, 1473-1483 (2020).

[13] Liu, Y., Zhao, Q., Yao, W. *et al.* Short-term rainfall forecast model based on the improved BP–NN algorithm. *Sci Rep* **9,** 19751 (2019).

[14] Arik, S. O. & Pfister, T. TabNet: Attentive Interpretable Tabular Learning. arXiv:1908.07442 https://ui.adsabs.harvard.edu/abs/2019arXiv190807442A (2019).

[15] Martins, A. F. T. & Fernandez Astudillo, R. From Softmax to Sparsemax: A Sparse Model of Attention and Multi-Label Classification. arXiv:1602.02068 https://ui.adsabs.harvard.edu/abs/2016arXiv160202068M (2016).

[16] Shilpa, M., Hui, L. Y., Song, M. Y., Feng, Y. & Teong, O. J. GPS-Derived PWV for Rainfall Nowcasting in Tropical Region. *IEEE Transactions on Geoence and Remote Sensing* **56**, 4835-4844 (2018).

[17] Li, P ., Wang, X., Chen, Y . & Lai, S. Use of GPS Signal Delay for Real-time Atmospheric Water Vapour Estimation and Rainfall Nowcast in Hong Kong. T*he First International Symposium on Cloud-prone & Rainy Areas Remote Sensing, Chinese University of Hong Kong*. 6–8 (2005).

[18] Saastamoinen, J. Atmospheric correction for the troposphere and stratosphere in radio ranging satellites. The use of artificial satellites for geodesy. **15**, 247–251 (1972)

[19] Han, H., Wang, W. Y. & Mao, B. H. Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. *Proceedings of the 2005 international conference on Advances in Intelligent Computing - Volume Part I* (2015).

[20] Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* **16**, 321-357 (2002).

[21] Bradley, P. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition* **30**, 1145-1159 (1997).

[22] PyTorch implementation of TabNet.https://github.com/dreamquark-ai/tabnet

[23] He, K., Zhang, X., Ren, S. & Sun, J. Deep Residual Learning for Image Recognition.arXiv:1512.03385https://ui.adsabs.harvard.edu/abs/2015arXiv151203385H (2015).

[24] Conneau, A., Schwenk, H., Barrault, L. & Lecun, Y. Very Deep Convolutional

Networks for Text Classification. arXiv:1606.01781 https://ui.adsabs.harvard.edu/abs/2016arXiv160601781C (2016).

[25] Amodei, D. *et al.* Deep Speech 2: End-to-End Speech Recognition in English and Mandarin. arXiv:1512.02595 https://ui.adsabs.harvard.edu/abs/2015arXiv151202595A (2015).

[26] Dauphin, Y. N., Fan, A., Auli, M. & Grangier, D. Language Modeling with Gated Convolutional Networks. arXiv:1612.08083 https://ui.adsabs.harvard.edu/abs/2016arXiv161208083D (2016)

[27] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma. Lightgbm: A highly efficient gradient boosting decision tree , *Advances in neural information processing systems*, 3146-3154 (2017).

[28] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. INV ASE: Instance-wise Variable Selection using Neural Networks. *ICLR* (2019).

[29] Liu, X. S., Deng, Z. & Wang, T. L. Real estate appraisal system based on GIS and BP neural network. *Transactions of Nonferrous Metals Society of China* **21**, s626-s630 (2011).

[30] Hochreiter, S. & Schmidhuber, J. Long Short-Term Memory. *Neural Computation* **9**, 1735-1780 (1997).

Acknowledgements

Te authors of the article would like to thank Beijing Environmental Planning Center for providing the meteorological data.

Autor contributions

YAN and XU participated in the design of this research. YAN was mainly responsible for the preparation of the manuscript, XU was mainly responsible for the realization of the research, and YU and XU were mainly responsible for the investigation of background and related knowledge. All authors read and approved the fnal manuscript.

Competing interests

Te authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to T.XU.

Figures

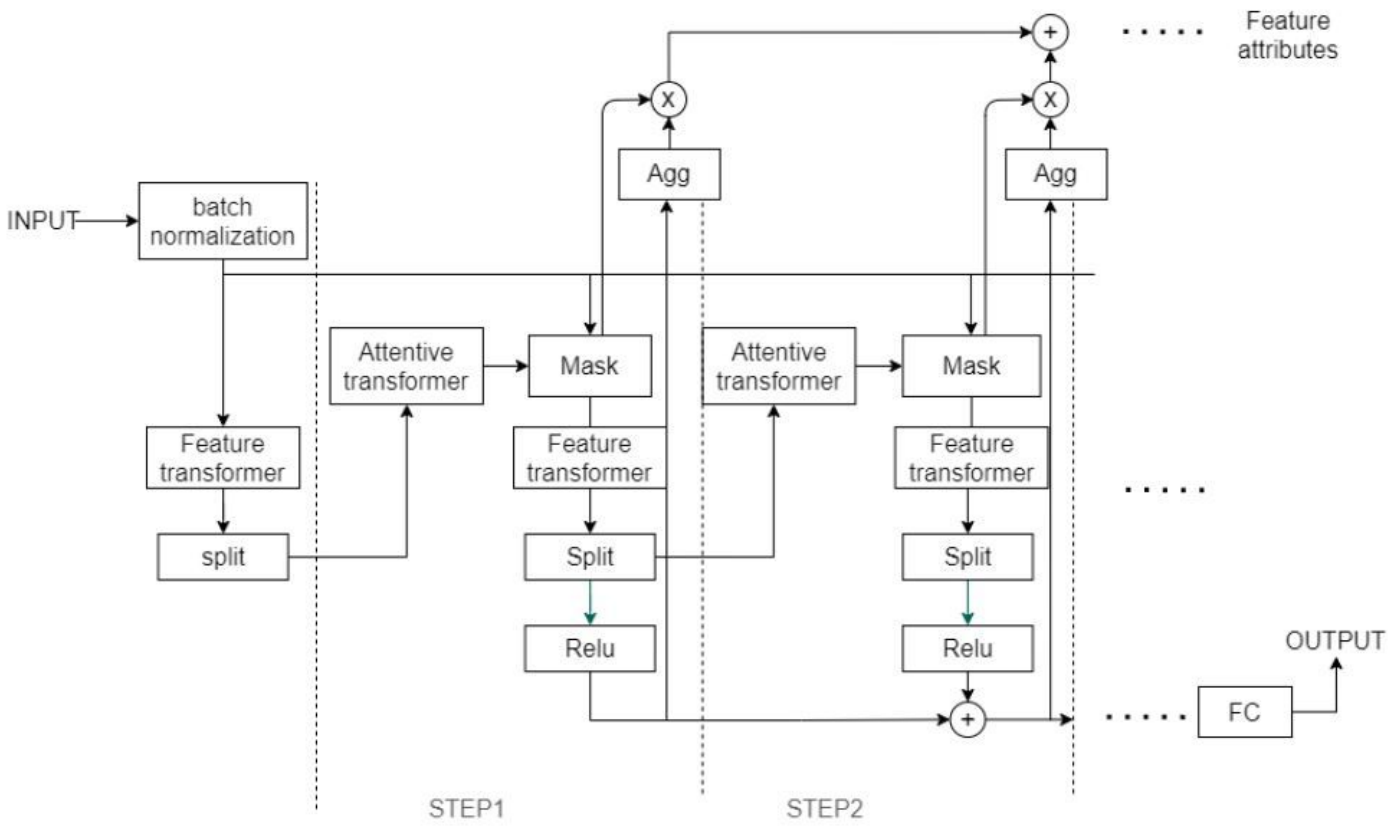

Figure 1

Topological structure of the TabNet algorithm. [Te fgure is plotted by Draw.io].

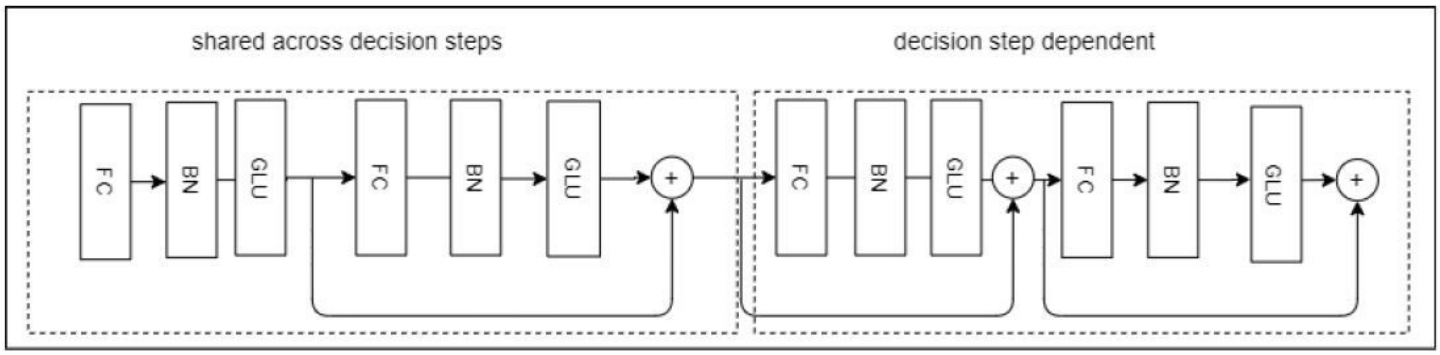

Figure 2

Topological structure of the Feature transformer layer. [Te fgure is plotted by Draw.io]

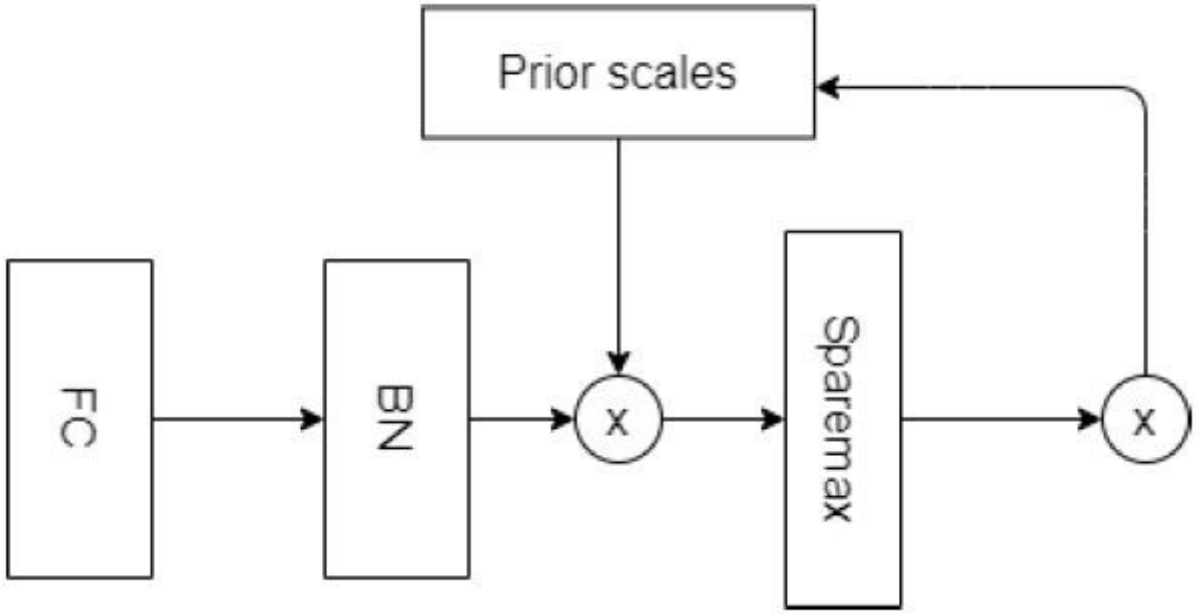

Figure 3

Topological structure of the Attentive transformer layer. [Te fgure is plotted by Draw.io].

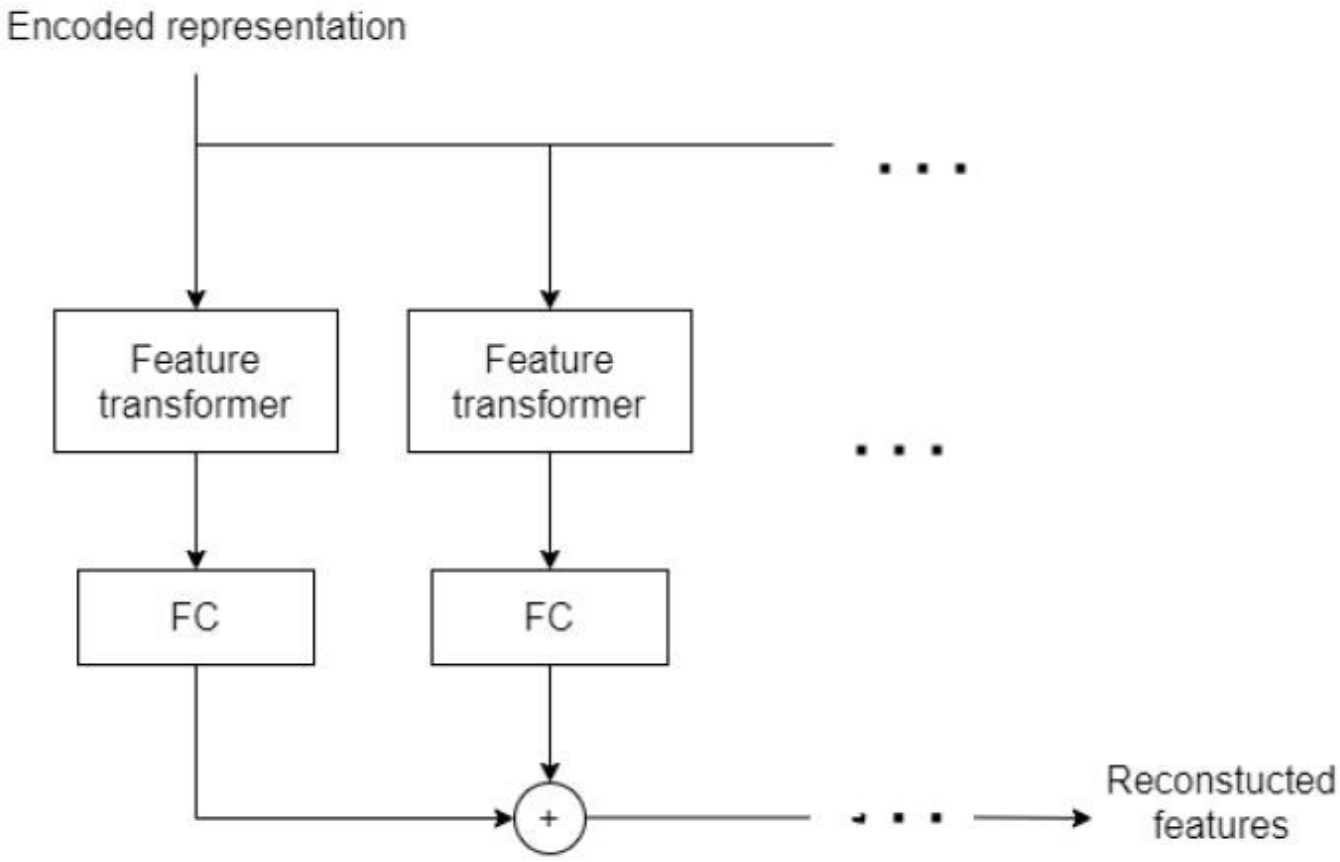

Figure 4

Topological structure of the decoder. [Te fgure is plotted by Draw.io].

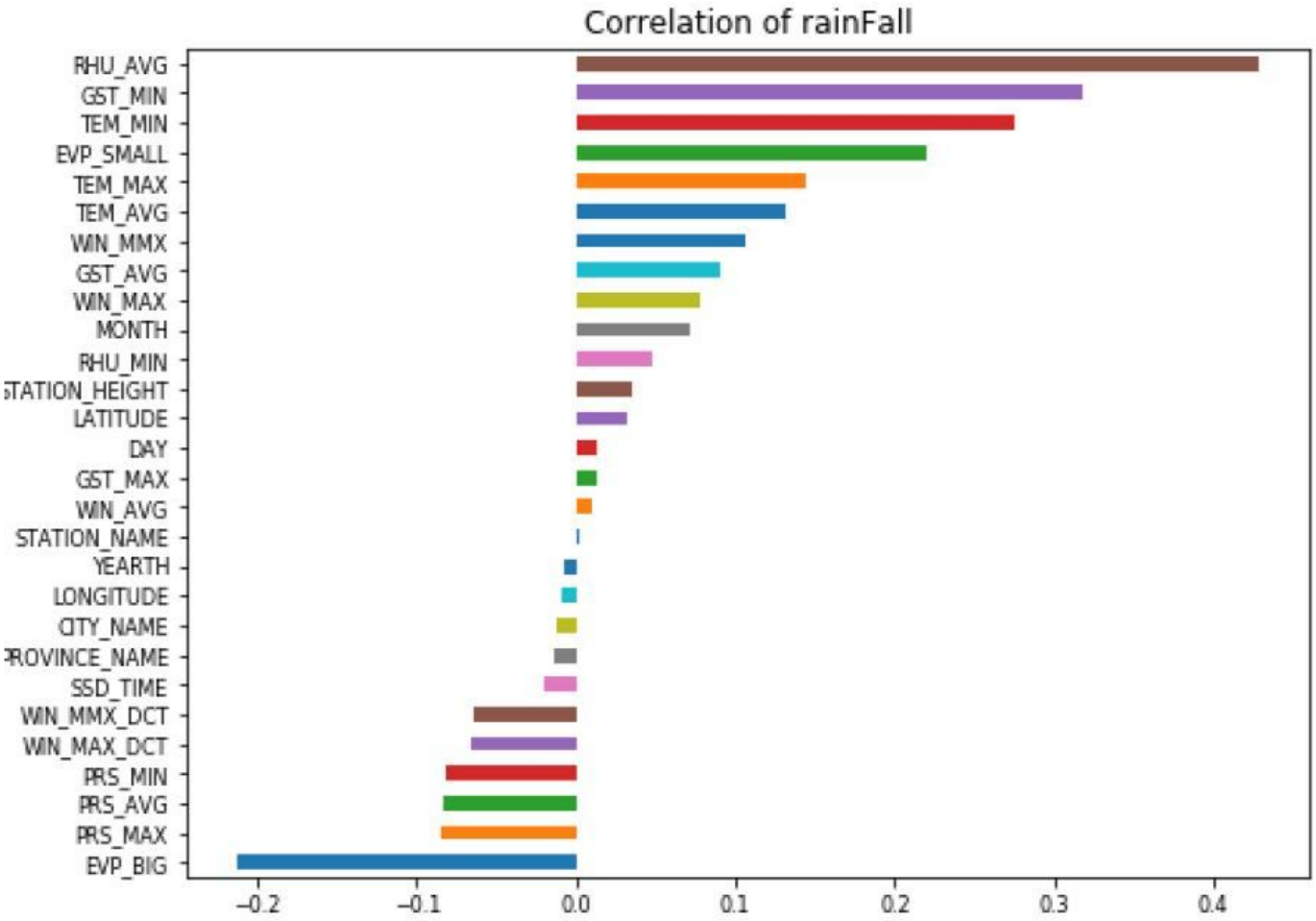

Figure 5

Correlation of each feature with the probability of rainfall [Te fgure is generated by matplotlib].

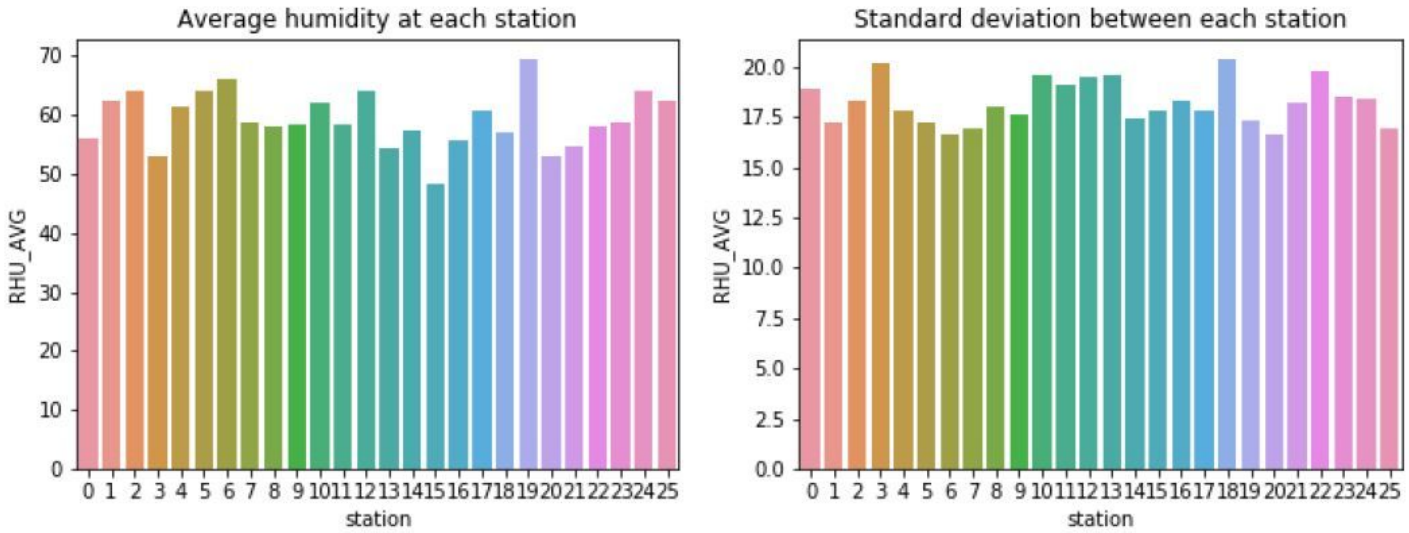

Figure 6

Average and standard deviation of humidity at each station[Te fgure is generated by matplotlib].

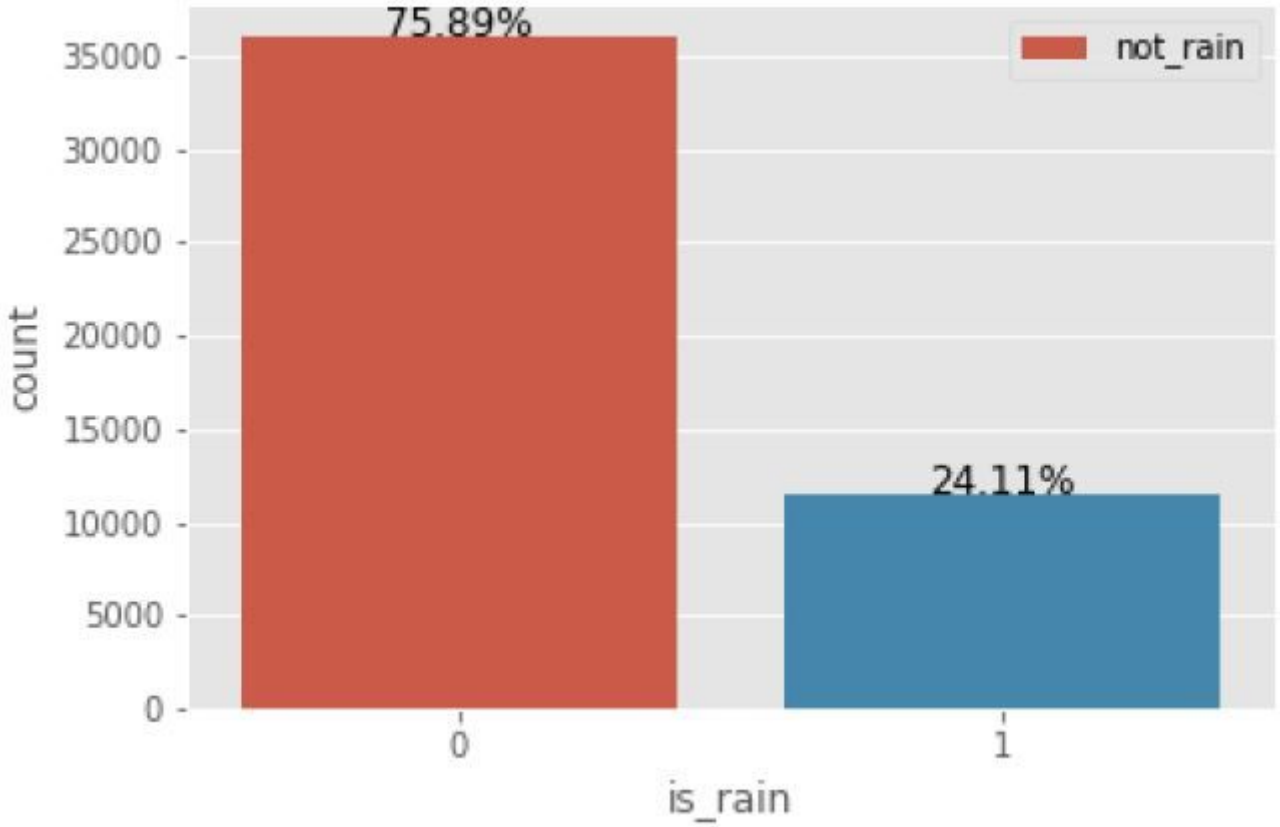

Figure 7

Rainfall probability distribution [Te fgure is generated by matplotlib]

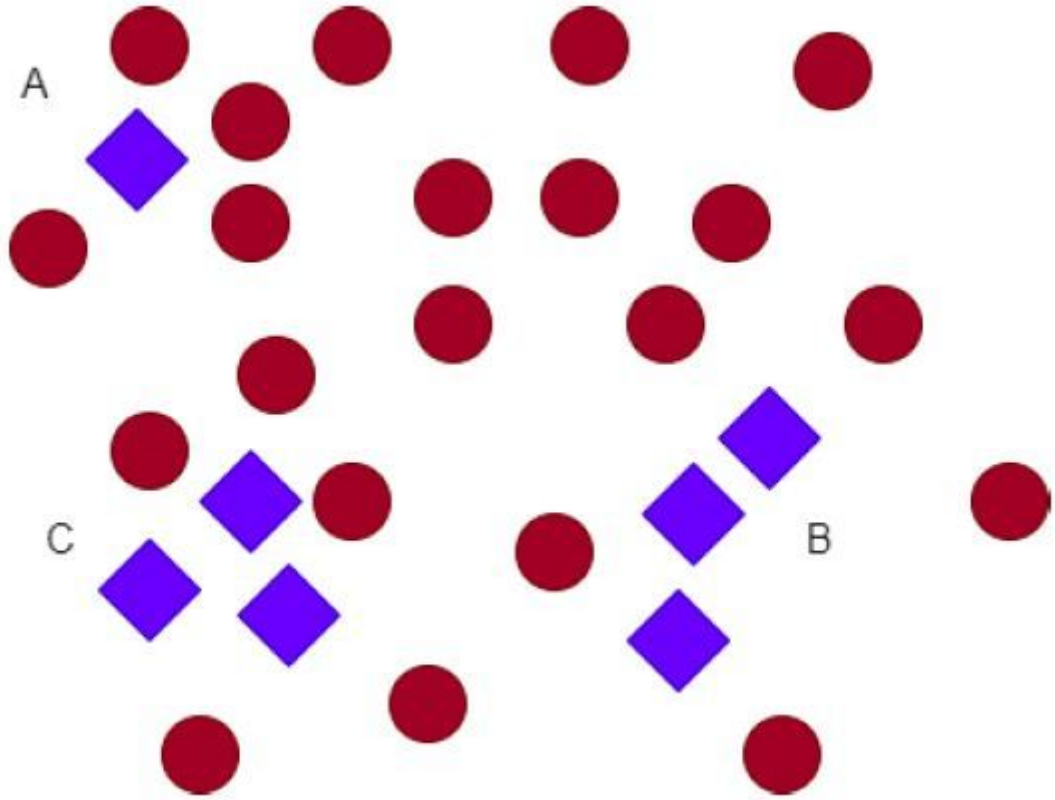

Figure 8

Borderline SMOTE sampling example [Te fgure is generated by Draw.io]

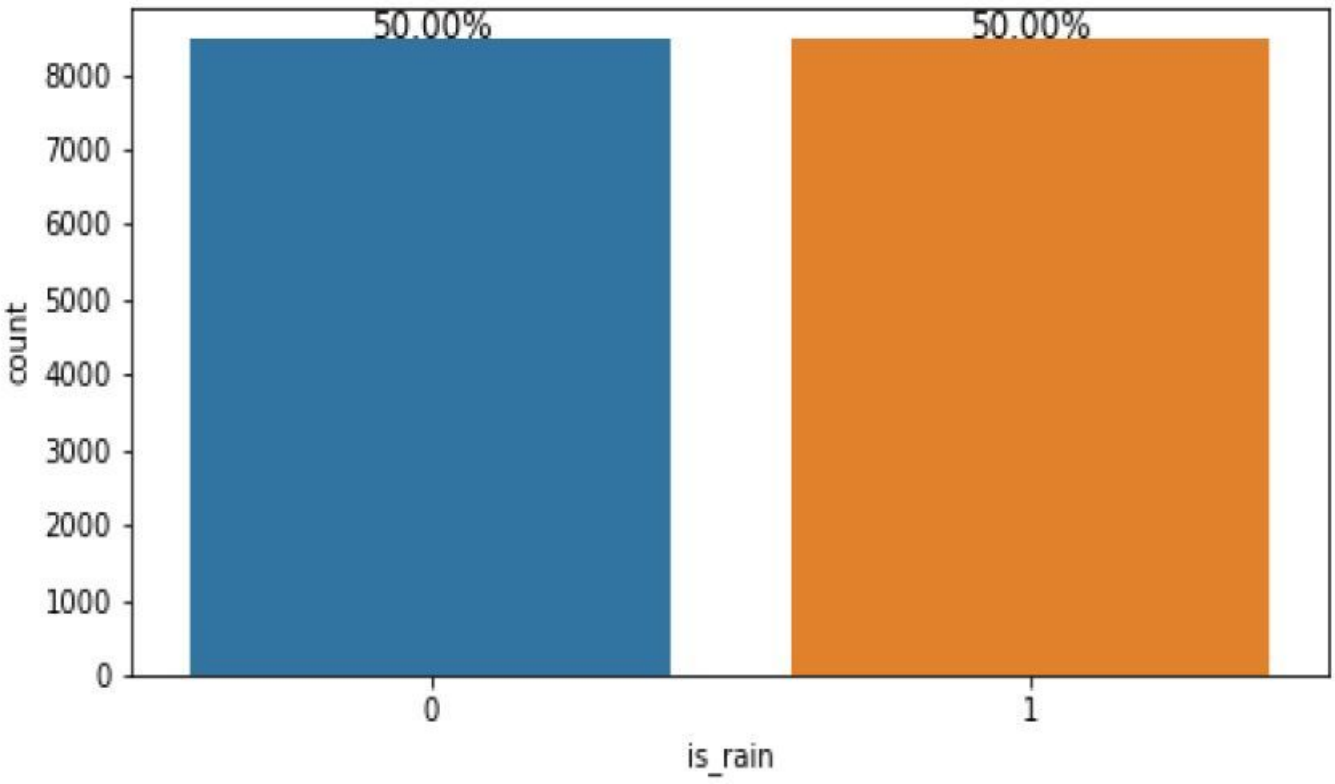

Figure 9

Borderline SMOTE improved data set [Te fgure is generated by matplotlib]

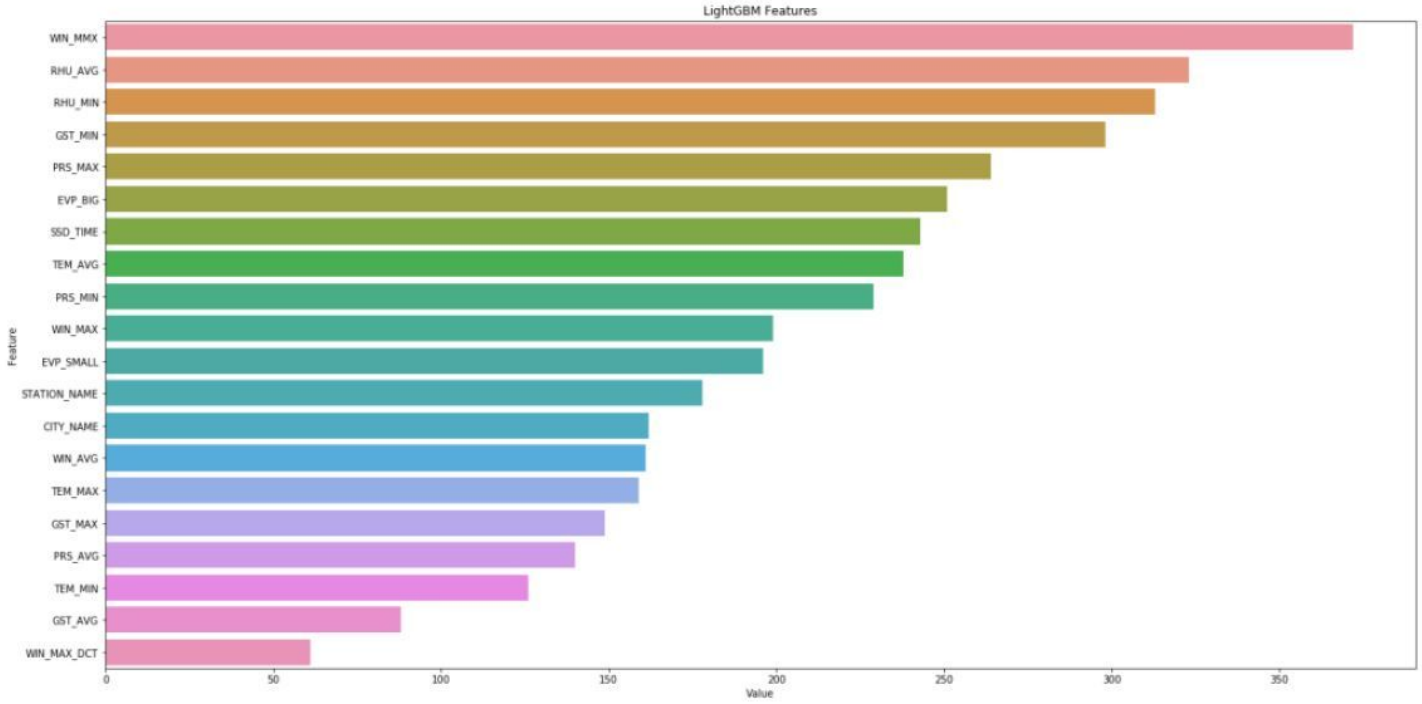

Figure 10

Feature importance distribution [Te fgure is generated by matplotlib]

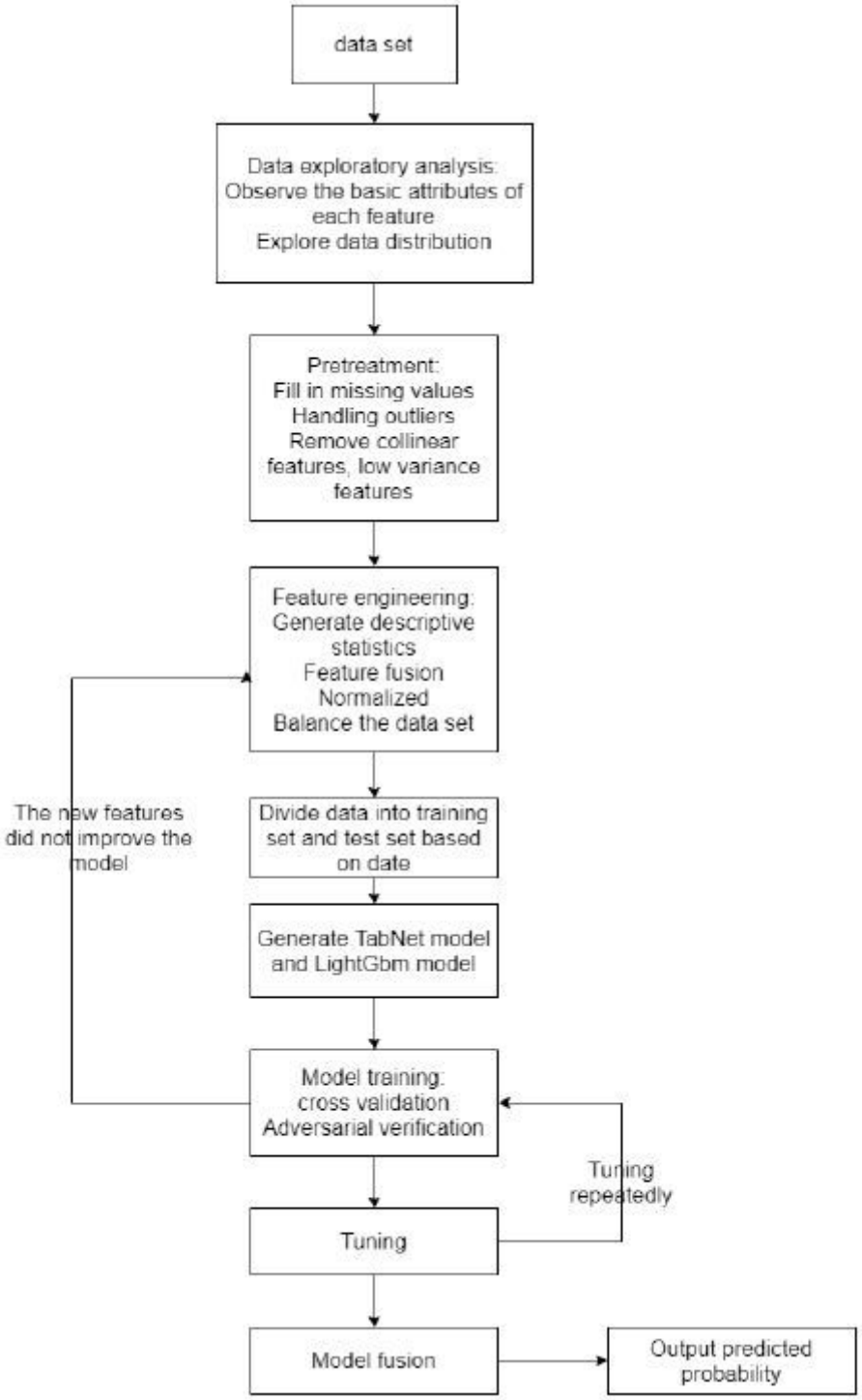

Figure 11

Experimental flowchart [Te fgure is generated by draw.io]