Modeling Thunderstorm Based on Paralleled and Improved Naive Bayes

Chengdong Zhou1,2, Leixiao Li1,2,*, Hui Wang1,2 and Shuang Liu1,2
1College of Data science and Application, Inner Mongolia University of Technology, Hohhot, China
2Inner Mongolia Autonomous Region Engineering & Technology Research Center of Big Data Based Software Service, Hohhot, China

*Corresponding author email: 20070000036@imut.edu.cn

Abstract. Aiming at improving the low efficiency and inaccuracy of the traditional thunderstorm prediction model, a paralleled and improved naive Bayesian thunderstorm prediction model SPNBC is proposed. The native Bayesian classifier (NBC) model assumes that the attribute data set is independent, which leads to inaccurate prediction. The principal component analysis algorithm is used to optimize the Bayesian algorithm to build PNBC. First, a new attribute data set is constructed by the principal component analysis (PCA) algorithm to eliminate the dependency between attributes, and a naive Bayesian classifier is constructed on the new attribute data set. Secondly, SPNBC is obtained by parallel design of PNBC based on spark framework to improve the time efficiency. Finally, taking the thunderstorm prediction in Hohhot as the application background, the accuracy rate, false alarm rate, speedup ratio and scalability ratio were used to analyze the experiment. The experimental results show that the SPNBC of this paper is better than the traditional naive Bayes algorithm and neural network algorithm, and the prediction accuracy rate and empty alarm rate are larger when dealing with massive data Performance advantages.

1. Introduction
Lightning observation work has been started for a long time in China. Many meteorological departments have accumulated a large number of lightning meteorological data, but most of these data have not been well used. How to analyze the meteorological data of lightning scientifically and predict effectively is an important research topic. Through the study of thunderstorms, decision support can be provided for vegetation protection, urban power protection, and site selection of high lightning sensitive areas in cities [1,2,3], which has important application value.
The traditional thunderstorm prediction methods are mainly divided into two categories. One is the thunderstorm movement trend warning and the other is the thunderstorm potential prediction. For the early warning of thunderstorm movement trend, a lightning proximity early warning model based on the options clustering algorithm is proposed in [4]. The location data are analyzed by clustering, the sparse points of lightning are eliminated, and the lightning falling area is predicted according to the movement trend of thunderstorm cloud. According to the lightning data collected by lightning location system, an improved DBSCAN thunderstorm clustering model is proposed in [5] to find the thunderstorm activity law. In [6], based on the existing lightning prediction and early warning, an improved density clustering method with active prediction and forward-looking is proposed, which provides important support for improving the accuracy of lightning early warning and reducing the false alarm rate. However, these early warning researches on thunderstorm movement trends require
that thunderstorms occur before further warnings can be made based on thunderstorm movement trends, but the potential of thunderstorms cannot be predicted. About the thunderstorm potential prediction, Lin et al. [7] established the BP neural network thunderstorm potential prediction model based on the 4-year data of Shenyang, and selected 9 factors of local thunderstorm potential prediction, with high accuracy. In reference [8], NCEP data and lightning location data are used to predict thunderstorm on the Hadoop platform with the parallel naive Bayesian algorithm. The results show that the model has a good effect on thunderstorm prediction and provides a new idea for massive thunderstorm data prediction. In [9], the CNN-LSTM deep neural network is employed to establish a 6-hour live data and ECMWF data for the past 30 years in Beijing, and to set up a thunderstorm forecast model for the next 6 hours, with a lower false alarm rate and higher accuracy. However, the above research is only suitable for small batch of meteorological data, and the features are also hand-crafted. With the advent of the era of big data, meteorological data is growing exponentially, and traditional calculation methods have been unable to meet the increasing data processing requirements. The advantage of parallel computing is to deal with big data. Hadoop provides a distributed file system HDFS and a parallel computing interface MapReduce [10-11], capable of processing large amounts of meteorological data. In [12], aiming at mining the characteristics of large amount of meteorological data, Hadoop MapReduce is used for parallelization to improve the time efficiency of the algorithm. In [13], a weather prediction method based on genetic neural network algorithm is constructed on Hadoop platform, which can effectively improve the time efficiency, and it has high prediction accuracy and good scalability. However, MapReduce consumes a lot of time in the communication of disk I/O, which leads to the re Spark overcomes the above problems, which not only has the advantages of traditional MapReduce, but also has the vantages to seamless integration with Hadoop. The duction of the execution efficiency of Hadoop's MapReduce Architecture [14]. However, Spark overcomes the above problems, which not only has the advantages of traditional MapReduce, but also has the vantages to seamless integration with Hadoop. The efficiency of spark framework based on memory computing is greatly improved compared with MapReduce framework [15]. In reference [16], using the combination of Hadoop platform and spark framework, the Hadoop distributed file system provides stable data storage, while spark provides efficient memory computing, and realizes the distributed data mining algorithm. The experimental results show that the platform has great performance advantages.

To sum up, in face of the explosive growth of meteorological data, this paper preprocesses the lightning location data and NCEP data in Hohhot area, and stores the meteorological data using the HDFS distributed file system based on Hadoop cluster. For the conditional independence assumption of naive Bayesian algorithm, the principal component analysis algorithm is used to select the prediction factors and construct the prediction factor genus. Finally, in order to improve the time efficiency, PNBC model is designed in parallel based on spark framework. Through the comparison of prediction accuracy, false alarm rate, speedup ratio and scalability ratio, the experimental results are analyzed.

2. Related Work

2.1. Construction of PNBC Thunderstorm Prediction Model

Naive Bayes algorithm [8] is widely used in the field of weather predict, and the predict results are good, so this paper uses naive Bayes algorithm to predict thunderstorm. When the relationship between attributes of data set is relatively independent, naive Bayesian classification algorithm will have a better accuracy. Although naive Bayesian algorithm comes from mathematical theory and has relatively stable accuracy, but the independence of data set attributes is difficult to meet in many cases. If there is correlation between attributes of data set, it will greatly reduce the effect of the model. So in this paper, PCA algorithm [17] is used to improve naive Bayes classification algorithm, establish PNBC thunderstorm prediction model, and improve the accuracy of naive Bayes model. Firstly, the attribute selection technology PCA is used to select the attributes of the original lightning prediction factor data set, construct a new lightning prediction factor data set, making it can approximately meet the conditional independence assumption. And then it is used to construct a naive
Bayesian classifier on the new lightning prediction factor attribute data set. The principal component analysis is based on the variance contribution. The new data set of thunderstorm prediction attributes can not only represent most of the information provided by the original thunderstorm data, but also obtain the prediction attributes satisfying the conditional independence assumption. The main steps of PNBC algorithm are as follows.

Step 1: The original thunderstorm prediction data set is preprocessed and divided into training set and test set.

Step 2: The principal component analysis of the original data set is carried out, and a new set of training and testing attribute data set is constructed. The two new attribute data sets approximately meet the conditional independence assumption.

Step 3: Based on the new training attributes set training thunderstorm prediction model, the probability of each category is calculated separately, and then the conditional probability of all partitions is calculated for each attribute feature to get the naive Bayesian classification model.

Step 4: The naive Bayesian classification model is used to classify and predict the samples in the new attribute test set.

2.2. Parallel Design of SPNBC Thunderstorm Prediction Model

In order to further improve the time efficiency, the parallel design of PNBC thunderstorm prediction model is carried out. The main idea of parallel design of SPNBC thunderstorm prediction model is to design the parallel scheme of PCA and NBC algorithm. Both of them parallel algorithm based on spark uses RDD (Resilient Distributed Data Set), a distributed elastic data set, to automatically realize the parallel data distribution. The parallel process of them based on spark platform is roughly divided into three parts. First, parallel PCA is used to construct new attribute data set selection. Second, NBC classification and prediction model is built based on new attribute data set in parallel. Finally, NBC classification and prediction model is used for final classification and prediction, as shown in Figure 1.

![Figure 1. Parallel design of SPNBC thunderstorm prediction model.](image)

It can be seen from Figure 1 that the parallel design of the thunderstorm prediction algorithm is mainly divided into three parts. In the first part, a new thunderstorm prediction data set is established by using PCA of spark ml database. Then the parallel NBC algorithm of the training set is used to train the thunderstorm prediction model, and finally the trained thunderstorm prediction model is used for classification. The algorithm pseudo code is as follows.

(1) **Parallelization of PCA to construct independent attribute data set based on Spark**

Input: the original thunderstorm prediction data set

Begin

Step 1. The original thunderstorm prediction data set in the matrix form as dataMatrix.

Step 2. Decentralization of local thunderstorm samples \( x_i \leftarrow x_i - \frac{1}{m} \sum_{j=1}^{m} x_j \)
Step 3. Calculation of covariance matrix of all thunderstorm samples $XX^T$

Step 4. Characteristic decomposition of covariance matrix $XX^T$

Step 5. The selection of eigenvalues $w_1, w_2, \ldots, w_k$ corresponding to the largest $K$ feature value

Output: New attribute data set of Thunderstorm prediction $W = (w_1, w_2, \ldots, w_k)$

After constructing the thunderstorm prediction data set with independent attributes, the thunderstorm prediction data set is divided into training set and test set, and then the thunderstorm prediction model is trained based on the training set in parallel.

(2) Parallel construction of thunderstorm prediction model based on new attribute data set

Input: the preprocessed thunderstorm train set

Begin

Step 1. Define ZeroCombiner [class, (ID, probability)] for map data, the ID consists of time, longitude and latitude, and initialize it.

Step 2. Calculate the sum of total number and feature vector for local thunderstorm prediction training data samples

for each class $i$ do

Calculate the global thunderstorm samples

end for $i$

Step 3. Calculate the denominator of thunderstorm prediction prior probability $P$.

for each class $i$ do

Obtain the number of thunderstorm samples of each class $i$, and calculate thunderstorm prediction class prior probability

$P(i) = \log(N + I \times \lambda)$

for each thunderstorm prediction feature vector $v$ of each class do

Calculate the denominator of thunderstorm prediction class conditional probability $P(v | i)$ and take the logarithm, $DenomTheta = \log(Sum(featuresSum) + numFeatures \times \lambda)$.

Calculate the thunderstorm prediction class conditional probability, $\theta(i)(v) = \log(featuresSum(v) + \lambda) - DenomTheta$, and store it in $\theta$.

end for $v$

end for $i$

End

Output: The thunderstorm prediction training model made up of matrix $\theta$ and vector $P$

After training, a thunderstorm prediction model is obtained, which is represented by sparse matrix of class prior vector and class conditional probability. Next, the test set of thunderstorm prediction will be predicted.

(3) Parallelization of thunderstorm predicting process based on Spark

Input: the thunderstorm test set

Begin

Step 1. Present the thunderstorm test set in the matrix form as $dataMatrix$.

Step 2. Calculate the probability of Thunderstorm prediction.

$P.add(\theta.mmul(dataMatrix))$

Step 3. Take the maximum probability of thunderstorm prediction category as the output value.

End

Output: Results of thunderstorm prediction

3. Experiment and Result Analysis

In this paper, the thunderstorm predict in Hohhot area is studied based on spark platform. Firstly, the lightning location data and NCEP historical reanalysis data are preprocessed, then the lightning in this area is predicted by SPNBC thunderstorm prediction model, and the algorithm is analyzed by accuracy, empty alarm rate, speedup ratio and scalability. Finally, the experimental results are analyzed.
3.1. Algorithm Accuracy Analysis

In order to verify the accuracy of SPNBC thunderstorm prediction model, the accuracy and empty alarm rate of thunderstorm prediction are used to evaluate SPNBC thunderstorm prediction model. The calculation Equations of accuracy and empty alarm rate are shown in Equation (6) and (7).

\[
P = \frac{TP}{TP + FP} \tag{6}
\]

\[
F = \frac{FN}{TN + FN} \tag{7}
\]

Among them, \(P\) represents the accuracy of thunderstorm prediction, \(TP\) represents the number of thunderstorms correctly predicted, \(FP\) represents the number of thunderstorms without thunderstorm live predict, \(T\) represents the empty alarm rate of thunderstorm prediction, \(TN\) represents the number of thunderstorms predicted but not actually occurred, \(FN\) refers to the number of thunderstorms without thunderstorm live predict.

Experimental results show that the SPNBC model adopts cross validation to select \(K\) value of PCA algorithm, as shown in figures 2 and 3. Experimental results show that the SPNBC model adopts cross validation to select \(K\) value of PCA algorithm, as shown in figures 2 and 3.

![Figure 2](image1.png)  ![Figure 3](image2.png)

**Figure 2.** Relationship between \(K\) value and accuracy rate.  
**Figure 3.** Relationship between \(K\) value and empty alarm rate.

It can be seen from figures 2 and 3 that the change of \(K\) value is closely related to the accuracy of Thunderstorm prediction and the empty alarm rate. Because the occurrence of thunderstorm is a small probability event, the accuracy and empty alarm rate should be considered to select \(K\) value.

Considering the accuracy and empty alarm rate of Thunderstorm prediction, \(K\) is set to 13 in this paper. Next, we will use the SPNBC thunderstorm prediction model, BP neural network model and traditional NBC model to predict thunderstorms. The data from January 1, 2015 to December 31, 2018 is used as the training set, and the data from January 1, 2019 to September 27, 2019 is used as the test set. The dataset contains a total of 41526 pieces of data. The \(K\) value of the thunderstorm prediction model in this paper is set to 13. The experimental results are shown in Table 1.

| Algorithm | \(P\) (%) | \(F\) (%) |
|-----------|-----------|-----------|
| NBC       | 90.56     | 37.04     |
| BP        | 35.46     | 12.81     |
| SPNBC     | 81.89     | 24.59     |

It can be seen from table 2 that although the accuracy of SPNBC thunderstorm prediction model proposed in this paper is reduced by 9% compared with the traditional NBC model, the empty alarm rate is reduced by about 13%. The empty alarm rate is obviously superior to the traditional NBC.
model, and thunderstorm prediction belongs to the event with small probability, so the overall accuracy is greatly improved. Compared with BP neural network, SPNBC thunderstorm prediction model has a greater advantage in accuracy, which is increased by about 46%. Although BP neural network model has a greater advantage in the empty alarm rate, the prediction accuracy is too low, which has no great practical significance for thunderstorm prediction in real life. Compared with the neural network model and the traditional NBC model, the SPNBC thunderstorm prediction model proposed in this paper has greater advantages, which can meet the needs of thunderstorm prediction in real life.

3.2. Algorithm Speedup Analysis
In order to verify the parallel performance of SPNBC thunderstorm prediction model, 4 thunderstorm prediction datasets are randomly generated by thunderstorm prediction dataset in this paper, and the size of the dataset is shown in Table 2.

| Data   | Size   |
|--------|--------|
| Data1  | 6.49M  |
| Data2  | 25.9M  |
| Data3  | 103M   |
| Data4  | 519M   |

The speedup ratio is an important index to detect the performance of parallel algorithm. The larger the speedup ratio, the less the relative time is required for parallel execution of the algorithm, that is to say, the higher the efficiency of parallel execution. It enhances the overall performance of the algorithm at the cost of reduced time by parallelization, and its calculation equation is shown in Equation (8).

\[ E_r = \frac{T_s}{T_r} \] (8)

In Equation (8), \( T_s \) represents the time of running the program serially in a single node, and \( T_r \) represents the parallel execution time in the environment of \( r \) nodes. In order to verify the speedup ratio of SPNBC thunderstorm prediction model proposed in this paper under different data sets and different nodes, the four datasets in Table 2 are used to implement the SPNBC thunderstorm prediction model in turn. The ratio of training set and test set is 8:2. The speedup of different data sets at different nodes is shown in Figure 4.

![Figure 4. Speedup ratio of different nodes.](image-url)
It can be seen from Figure 4 that with the increase of calculation nodes, the speedup ratio of data 1 of thunderstorm prediction data set tends to be gentle until about 1.7, which shows that the speedup ratio of small data set in cluster environment is not obvious. However, with the increase of data volume, the speedup curves of Thunderstorm Predict data sets data2, data3 and Data4 increase obviously. The peak speedup ratio of Data4 is about 3.4, and the speedup ratio curve increases with the number of nodes increasing when the data volume is the same. The experimental results show that the thunderstorm prediction model based on Hadoop and spark cluster has better speedup ratio.

3.3 Algorithm Scalability Analysis
The performance analysis of parallel algorithm can not just analyze the speedup of the algorithm, because the speedup of parallel algorithm can not always increase. With the increase of nodes, the speedup ratio will not fully reflect the utilization ratio of the cluster, so we need to introduce the scalability ratio for the performance analysis of parallel algorithm[18]. The calculation Equation of scalability is shown in Equation (9).

$$J = \frac{E_r}{r}$$  \hspace{1cm} (9)

In Equation(9), $E_r$ represents the speedup ratio of the algorithm, $r$ represents the number of computing nodes, the larger the scalability is, the higher the overall utilization rate of the cluster is, and the better the parallel effect of the algorithm can be demonstrated.

In order to test the scalability of thunderstorm prediction model, the scalability ratio of different scale datasets under different cluster nodes is different, and the thunderstorm prediction model proposed in this paper is implemented in parallel under spark cluster based on Hadoop platform. The data set uses the four thunderstorm prediction data sets constructed in the speedup ratio experiment, and the scalability data statistics obtained by Equation (9) are shown in Figure 5.

![Figure 5. Scalability ratio of different nodes.](image)

Figure 5 shows the scalability ratio of four thunderstorm prediction data sets in parallel execution of thunderstorm prediction model based on Hadoop platform spark cluster. It can be seen from the figure that with the increase of nodes and data, the decline speed of the scalability ratio gradually decreases and tends to be stable. Among them, the scalability of thunderstorm prediction data set data1 is the fastest to decline than the curve, while the scalability of thunderstorm prediction data set data2, data3 and Data4 is more and more slow with the increase of data set. The experimental results show that the thunderstorm prediction model proposed in this paper has good scalability in the case of large data sets, but the scalability of small data sets is poor, which meets the scalability requirements of general parallel algorithm.
4. Conclusion
In the face of massive lightning weather data, this paper designs a lightning weather data storage scheme based on the Hadoop platform, and proposes a lightning storm prediction model SPNBC based on spark parallel improved naive Bayes. Through experiments, it is verified that the thunderstorm prediction model in this paper has better prediction accuracy and empty alarm rate than the traditional NBC model and BP neural network, and meets the needs of thunderstorm prediction in real life. Moreover, the speed of SPNBC thunderstorm prediction model is greatly improved compared with the traditional single computer operation, and it has obvious performance advantages in dealing with massive thunderstorm weather data. In conclusion, compared with other models, this study has greater advantages and is expected to obtain good social and economic benefits. At the same time, the method presented here can also be extended to other meteorological data. Our future works will use the big data technology to predict rainfall and other meteorological phenomenon.

Acknowledgments
This research was financially supported by Inner Mongolia Autonomous Region Science and Technology Major Project of China (development and application of private cloud operating system based on OpenStack), Inner Mongolia Autonomous Region Natural Science Fund Project of China (2019MS06027), Inner Mongolia Autonomous Region key technology tackling plan project of China (Research and application of big data storage and analysis mining platform for intelligent transportation).

References
[1] Shevchenko Y, Nazim Y, Tanasoglo A, et al. Refinement of Wind Loads on Lattice Support Structures of the Intersystem Overhead Power Transmission Lines 750 kV [J]. Procedia Engineering, 2015, 117(1):1033-1040.
[2] Veraverbeke S, Rogers B M, Goulden M L, et al. Lightning as a major driver of recent large fire years in North American boreal forests[J]. Nature Climate Change, 2017, 7(7):529-534.
[3] Liu F J, He Q Y, Tang Y, et al. Application and Research of Meteorological Data in Lightning Protection Technical Service of Oil Depot[J].Meteorological and Environmental Research, 2019, 10(01):26-30.
[4] Hou R T, Lu Y, Wang Q, Yuan C S, et al. application of options algorithm in lightning proximity prediction [J]. Computer application, 2014, 34 (01): 297-301
[5] Gao P, Tian H, Li J, et al. Thunderstorm mining and research based on improved DBSCAN algorithm [J]. High voltage apparatus, 2019, 55 (04): 169-177
[6] Huang L Z, Su S, Yang X, et al. Near prediction of thunderstorm movement trend based on LLS [J]. Electric porcelain arrester, 2019, 287 (01): 82-89
[7] Lin Z G, Luan J, et al. Thunderstorm potential prediction model in Shenyang based on double hidden layer BP network [J]. Journal of Southwest University (Natural Science Edition), 2017, 39 (02): 84-91.
[8] Xue S J, Ji F, Xu X L. Preliminary study on Thunderstorm prediction based on Naive Bayes in cloud environment [J]. Journal of Wuhan University of technology, 2014,36 (11): 130-135.
[9] Ni Z, Wen T. A weather prediction model based on CNN and RNN deep neural network -- a case study of 6-hour Thunderstorm Predict in Beijing area [J]. Numerical calculation and computer application, 2018,39 (04): 299-309.
[10] Ghazi M R, Gangodkar D. Hadoop, MapReduce and HDFS: A Developers Perspective[J]. Procedia Computer Science, 2015, 48(04):45-50.
[11] Ramakrishnan Ramanathan, B. Latha. Towards optimal resource provisioning for Hadoop-MapReduce jobs using scale-out strategy and its performance analysis in private cloud environment[J]. Cluster Computing, 2018, 22(2):1-11.
[12] Chao Liu, Wen Jin, Yuting Yu, et al. A Discretization Algorithm for Meteorological Data and its Parallelization Based on Hadoop[J]. Journal of Physics Conference, 2017, 910(1):12-21.
[13] Gou Z J, Ren J L, et al. Application of Hadoop based GA-BP Algorithm in precipitation prediction [J]. Computer system application, 2019, 28 (09): 140-146.
[14] Chaudhary R, Aujla G S, Kumar N, et al. Optimized Big Data Management across Multi-Cloud Data Centers: Software-Defined-Network-Based Analysis[J]. IEEE Communications Magazine, 2018, 56(2):118-126.

[15] Liu P, Zhao H H, Teng J Y, et al. Parallel naive Bayes algorithm for large-scale Chinese text classification based on spark[J]. Journal of Central South University, 2019, 26(1):1-12.

[16] Shah S A, Seker D Z, Rathore M M, et al. Towards Disaster Resilient Smart Cities: Can Internet of Things and Big Data Analytics be the Game Changers?[J]. IEEE Access, 2019, 7(1):91885-91903.

[17] Xie X R, Lei X R, Zhao Y. Application of MI and improved PCA in stock price prediction [J / OL]. Computer engineering and application: 1-9 [2020-02-25]. Http://kns.cnki.net/kcms/detail/11.2127.tp.20191213.1248.008.html

[18] SUN X, ROVER D. Scalability of parallel algorithm machine combinations [J]. IEEE Trans Parallel and Distributed System, 1994, 5(6): 599-613.