Research on Flood Occurrence Prediction Based on Entropy Weighted TOPSIS and Ensemble Machine Learning

Jingchuan Xu[1,a,*], Yuanyuan Ren[1,b], Hao Zhang[1,c], Shuyan Fu[2,d]

[1]School of Computer Science, Xi 'an Shiyou University, Xi'an, China

[2]School of Petroleum Engineering, Xi 'an Shiyou University, Xi'an, China

[a]2282802903@qq.com, [b]2103026720@qq.com, [c]1242017264@qq.com, [d]3231525684@qq.com

*Corresponding author

Keywords: Flood Prediction; Entropy Weight Method; TOPSIS; Ensemble Machine Learning; Stacking Algorithm

Abstract: This paper presents a flood occurrence prediction method based on the entropy weight method and TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) combined with ensemble machine learning. Initially, the flood probability is clustered into three risk levels—low, medium, and high—using the K-means clustering algorithm, and the minimum occurrence probability, maximum occurrence probability, average flood probability, and proportion quantity for each level are calculated. Subsequently, feature extraction of risk indicators is performed through R-type clustering, and similarity measures such as the Pearson correlation coefficient and cosine similarity are utilized to assess the degree of similarity between variables, followed by variable clustering analysis. Furthermore, an early warning evaluation model based on the entropy weight TOPSIS is established, where information entropy is used to quantify indicator weights, reflecting the importance of indicators in the decision-making process. The model calculates information entropy, indicator weights, standardized decision matrix, positive and negative ideal solutions, and comprehensive evaluation index to rank the sample risk levels. Additionally, this paper explores the sensitivity analysis of the model by perturbing the weights of each feature indicator and the flood characteristics, verifying the model's robustness under minor perturbations. Finally, a flood probability prediction model based on different base classifiers is constructed using the Stacking ensemble learning algorithm, and cross-validation methods are employed to enhance the model's generalization capability. The experimental results demonstrate that the Stacking model performs best on evaluation metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Coefficient of Determination (R^2), and Maximum Absolute Error (MaxAE), proving its superiority in predicting the probability of flood occurrence.

1. Introduction

Accurately predicting the probability of flood occurrence is of great significance for flood prevention and mitigation, resource management, and decision support [1]. Traditional flood prediction methods mainly rely on empirical models and statistical analysis, but these methods are often limited by data quality and the subjectivity of experience-based judgment [2]. With the development of machine learning technology, especially the application of ensemble learning methods, new perspectives and powerful tools have been provided for flood prediction. Ensemble learning methods improve the accuracy and robustness of predictions by combining the results of multiple models [3].

This study proposes an ensemble machine learning model based on the entropy weight method and TOPSIS for predicting the probability of flood occurrence [4, 5]. The entropy weight method, as an objective weighting method, can quantify the importance of each indicator in the decision-making process, while the TOPSIS method ranks the schemes by calculating the distance from each scheme to the ideal solution and the negative ideal solution. The combination of these two methods

  DOI: 10.25236/iwmecs.2024.007

can provide a more comprehensive assessment of flood risk. In addition, this study employs the K-means clustering algorithm for sample clustering of flood probability, uses R-type clustering to extract features of risk indicators, and utilizes the Pearson correlation coefficient and cosine similarity to measure the similarity between variables [6, 7]. The application of these methods not only enhances the interpretability of the data but also provides a basis for variable selection and model construction.

The purpose of this study is to construct an accurate and reliable model for predicting the probability of flood occurrence, providing a scientific basis for flood prevention and mitigation. Through model training and evaluation, we expect to verify the effectiveness of the proposed method and explore its application potential under different conditions.

2. Sample Clustering

2.1. Clustering of Flood Probability with K-means

K-means clustering is a commonly used unsupervised learning algorithm for dividing data points into K clusters. The basic steps can be summarized as follows: Initially, K data points are randomly selected as the initial cluster centers; then, each data point is assigned to the nearest cluster center to form K clusters; next, the center of each cluster is recalculated, typically by taking the mean of all points within the cluster; afterwards, the process of reassignment and center updating is repeated until the cluster centers no longer change significantly or a preset number of iterations is reached; finally, a stable cluster division is obtained. This process iteratively optimizes to ensure that points within a cluster are as similar as possible, while points between clusters are as different as possible. The final cluster centers are shown in Table 1.

Table 1 Results of the K-means clustering.

	Minimum Occurrence Probability	Maximum Occurrence Probability	Average Flood Probability	Proportion Quantity
Low Risk	0.285	0.475	1.2527	30.40%
Medium Risk	0.48	0.535	1.1433	43.40%
High Risk	0.54	0.725	0.04981	20.60%

The table above categorizes the data into three risk levels: low, medium, and high. Approximately 30.40% of the samples fall into the low-risk category, 43.40% into the medium-risk category, and 20.60% into the high-risk category.

2.2. Feature Extraction of Risk Indicators via R-Type Clustering

K-means clustering is a commonly used unsupervised learning algorithm for dividing data points into K clusters. The basic steps can be summarized as follows.

2.2.1. Variable Similarity Measurement

Before conducting cluster analysis on variables, it is essential to first determine the measure of similarity between variables. Similarity measures are used to assess the degree of similarity between two variables, with common methods including the Pearson correlation coefficient and cosine similarity. Suppose there are indicator variables (x_1, x_2, ..., xp), and their observation matrix on n sample points is recorded as $A = \left(a_{ij}\right)_{n\times p}$, where a_{ij} represents the observation value of the j-th variable for the i-th sample. Since the Pearson correlation coefficient and cosine similarity are not affected by the variables themselves, there is no need for data standardization in this case.

Pearson correlation coefficient is used to measure the linear correlation between two continuous variables. The formula is as follows:

$$r_{xy} = \frac{\sum_{i-1}^{n}(x_i,x)(y_i\bar{y})}{\sqrt{\sum_{i-1}^{n}(x_i\bar{x})^2\sum_{i-1}^{n}(y_i\bar{y})^2}} \quad (1)$$

In the formula, x_i and y_i represent the observed values of variables x and y for the i-th sample, respectively, while $\bar{x}$ and $\bar{y}$ are the means of variables x and y. n is the number of samples, and p is the number of variables. This formula uses the data from the observation matrix A of the samples to calculate the correlation coefficient, reflecting the linear correlation between variables x and y.

Cosine Similarity is used to measure the similarity in direction between two vectors, commonly used in fields such as text mining. Typically, each row is considered a vector ai, where $a_i = (a_{i1}, a_{i2}, \ldots, a_{ip})$ is the i-th row of the matrix. The cosine similarity between the i-th row and the j-th row aj in the matrix is given by the following formula:

$$\text{similarity}(a_i, a_j) = \frac{a_i \cdot a_j}{|a_i||a_j|} \qquad (2)$$

In the formula, $a_i \cdot a_j$ represents the dot product (inner product) of vectors a_i and a_j, and $|a_i|$ and $|a_j|$ denote the norms (magnitudes) of vectors a_i and a_j, respectively. In this way, by using the rows of the observation matrix A, we can calculate the cosine similarity between the rows, thereby measuring their similarity or correlation.

2.2.2. Variable clustering

Variable clustering is a statistical method used for the analysis of multivariate data, aimed at identifying and organizing patterns of correlation among multiple variables. Similar to the shortest distance method and longest distance method commonly used in sample cluster analysis, variable clustering methods adopt similar ideas and steps. When dealing with variable clustering problems, the longest distance method and shortest distance method are also commonly used.

Longest Distance Method: The longest distance between two sets A and B is defined as the maximum distance between any two points within them. Commonly, this can be the Euclidean distance, Manhattan distance, or other distance metrics.

Shortest Distance Method: Suppose there are two sets A and B, their shortest distance is defined as the minimum distance between any two points within them.

By comparing the R-type clustering results of the flood indicators, we found that the clustering effects using the Complete and Chebyshev methods are relatively good. The Complete linkage method defines the distance between two categories as the maximum distance between all members of these two categories, suitable for forming compact and spherical clusters, and particularly suitable for handling data with isotropic distribution. The Chebyshev linkage method, on the other hand, defines the distance between two categories as the maximum difference in all dimensions, typically used for situations where there is a significant difference in data across different dimensions.

3. Establishment of the Entropy Weight TOPSIS Early Warning Evaluation Model

3.1. Evaluation Model

Information entropy, a concept in information theory, is used to measure the uncertainty or disorder of information. In decision analysis, information entropy is incorporated into the calculation of indicator weights to reflect the importance of indicators in the decision-making process. The greater the information entropy, the higher the uncertainty of the indicator, and vice versa. TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) is a multi-attribute decision analysis method designed to help decision-makers choose the best option from multiple candidates. It combines the theory of the ideal solution with the idea of distance measurement, ranking each candidate by calculating the similarity to the ideal solution.

Let the comprehensive evaluation problem contain n evaluation objects and m indicators, the basic steps are as follows:

Step 1: Information Entropy (Ej): The formula for calculating the information entropy of indicator (j) is:

$$E_j = \sum_{i=1}^{m} \frac{p_{ij}}{\ln(m)} \ln\left(\frac{p_{ij}}{\ln(m)}\right) \quad (3)$$

In the formula, P_{ij} is the relative weight of indicator (j) on the (i)th alternative, and m is the number of alternatives.

Step 2: Indicator Weight (wj): The weight of indicator (j) is calculated as:

$$w_j = \frac{1E_j}{n\sum_{j=1}^{n} E_j} \quad (4)$$

In the formula, n is the number of indicators.

Step 3: Standardization of the Decision Matrix (Z): The decision matrix (X) is standardized to (Z) using the following formula:

$$z_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}} \quad (5)$$

Step 4: Positive Ideal Solution (A^+) and Negative Ideal Solution (A^-):

The Positive Ideal Solution (A^+) is calculated as: $A_j^+ = \max(z_{ij}), \forall j = 1, \cdots, n$. The Negative Ideal Solution (A^-) is calculated as: $A_j = \min(z_{ij}), \forall j = 1, \cdots, n$.

Step 5: Comprehensive Evaluation Index (Ci): The comprehensive evaluation index for the candidate scheme (i) is calculated as:

$$C_i = \frac{\sqrt{\sum_{j=1}^{n} w_j\left(A_j^+ z_{ij}\right)^2}}{\sqrt{\sum_{j=1}^{n} w_j (A_j^+ A_j^2)}} \quad (6)$$

The sample risk level sorting results are shown in Table 2.

Table 2 Results of the sample risk level sorting.

id	Flood Probability	Risk Level	Risk Ranking (The smaller the value, the higher the risk)
37	0.57	High Risk	114758
38	0.55	High Risk	104966
44	0.555	High Risk	85104
45	0.54	High Risk	270451
58	0.6	High Risk	179741
62	0.55	High Risk	96530
71	0.575	High Risk	35438
74	0.57	High Risk	218097
77	0.565	High Risk	90953
79	0.56	High Risk	82358
82	0.555	High Risk	93103

3.2. Sensitivity Analysis

Considering the weight w of each feature indicator; the j-th feature of the i-th sample is perturbed as follows:

$$\begin{cases} w_i = w_i + \delta_i \\ x_{ij} = x_{ij} + \Delta\theta_i \end{cases} \quad (7)$$

δ_i and $\Delta\theta_i$ represent the perturbation values of the weights and the flood characteristics of the sample, respectively. δ_i is taken to follow a Gaussian distribution with a mean of 0 and a standard deviation of 0.01, that is $\delta_i \sim N(0, 0.01^2)$. $\Delta\theta_i$ is taken to follow a normal distribution with a mean of 0 and a standard deviation of 0.1, that is $\triangle\ \theta_i \sim N(0, 0.1^2)$. The risk level sorting results after perturbation of the sample are shown in Table 3.

Table 3 Results of the sample risk level sorting after perturbation of the sample .

id	Flood Probability	Risk Level	Risk Ranking (Before Perturbation)	Risk Ranking (After Perturbation)
37	0.57	High Risk	114758	114324
38	0.55	High Risk	104966	1041245
44	0.555	High Risk	85104	85091
45	0.54	High Risk	270451	270041
58	0.6	High Risk	179741	179770
62	0.55	High Risk	96530	96120
71	0.575	High Risk	35438	35411
74	0.57	High Risk	218097	212578
77	0.565	High Risk	90953	91578
79	0.56	High Risk	82358	81257
82	0.555	High Risk	93103	93058

From the new TOPSIS sorting results we obtained, it can be seen that the sorting changes within a very small range after adding minor perturbations, indicating that the model has a certain degree of robustness.

4. Stacking-Based Flood Probability Prediction Model

Stacking is a parallel ensemble learning algorithm that uses heterogeneous models as base classifiers. It integrates the prediction results of the base classifiers by adding a meta-classifier, and finally, the final ensemble result is obtained through retraining with the meta-learner. Unlike Bagging and Boosting algorithms, which combine the prediction results of base classifiers through different strategies, Stacking utilizes a meta-learner to achieve the integration of weak learners.

The Stacking ensemble learning algorithm can be divided into two parts: The first part consists of multiple weak learners that learn directly on the original dataset; the second part is composed of a meta-learner that integrates the prediction results of the weak learners from the first part. Directly using the outputs of the weak learners from the first part as inputs for the meta-learner may lead to overfitting issues, hence it is often necessary to use K-fold cross-validation methods to ensure the model's generalization ability.

4.1. Data Preprocessing

We perform data cleaning on the original data, dealing with missing values, outliers, and duplicate data to ensure the consistency and completeness of the data. Then, we carry out data transformations, including normalization, standardization, and feature scaling. Finally, we split the data, dividing 80% and 20% of the dataset into the training set and the test set, respectively, to evaluate the model's performance.

4.2. Model Training

This paper employs the first layer models (base models) as Random Forest, XGBoost, SVR (Support Vector Regression), and GBDT (Gradient Boosting Decision Tree). When training the second layer meta-learner, the predictions generated by the base learners during the cross-validation process are used as input features. The main purpose of this approach is to observe the original input dataset from multiple angles through different base learners, extract information, and compress the original dataset for use by the second layer meta-learner.

4.3. Results

Table 4 presents the performance results of each model on evaluation metrics. MSE (Mean Squared Error): The average of the squared differences between the model's predictions and the actual values. The smaller the value, the more accurate the model's predictions. RMSE (Root Mean Squared Error): The square root of MSE, used to measure the standard deviation of prediction errors. Compared to MSE, it is easier to interpret and compare. MAE (Mean Absolute Error): The average

of the absolute values of prediction errors, reflecting the overall accuracy of the model's predictions. R^2 (Coefficient of Determination): Also known as the goodness of fit, it indicates the degree to which the model fits the observed data. The value ranges from 0 to 1, with values closer to 1 indicating a better model fit. MaxAE (Maximum Absolute Error): The maximum value of the prediction error, reflecting the maximum deviation that may occur in a single prediction by the model.

Table 4 Evaluation results of various models.

Model	MSE	RMSE	MAE	R^2	MaxAE
SVM	0.0012	0.0353	0.0293	0.5438	0.1030
Random Forest	0.0012	0.0350	0.0289	0.5496	0.1139
GBDT	0.0010	0.0324	0.0268	0.6151	0.1099
XGBoost	0.0008	0.0278	0.0224	0.7173	0.0980
Stacking	0.0006	0.0245	0.0197	0.7804	0.0941

The mean squared error (MSE) and root mean squared error (RMSE) of the SVM, Random Forest, and GBDT are close, but the XGBoost and Stacking models perform better on these two metrics, with lower values, indicating that these two models have smaller errors in prediction. In terms of mean absolute error (MAE), the Stacking model performs best, followed by XGBoost, suggesting that these two models have relatively smaller prediction errors. The coefficient of determination (R^2) reflects the goodness of fit of the model, and the R^2 value of the Stacking model is the highest, reaching 0.7804, indicating that the model fits the observed data well. In terms of maximum absolute error (MaxAE), the Stacking model also performs the best, with a maximum error value of 0.0941, which is smaller than other models. In summary, the Stacking model performs best on these evaluation metrics and is a more optimal model for predicting the probability of flood occurrence.

5. Conclusion and Future Works

In conclusion, this study has successfully developed an ensemble machine learning model integrating the entropy weight method and TOPSIS for the prediction of flood occurrence probabilities. The model has demonstrated high accuracy and robustness through comprehensive evaluation metrics, providing a valuable tool for flood risk assessment and early warning systems. Looking ahead, further research is warranted to refine the model with more extensive datasets, explore its adaptability across different geographical and climatic conditions, and enhance its real-time predictive capabilities, thereby contributing to more effective flood management strategies and community resilience.

References

[1] Izinyon O C, Ajumuka H N. Probability distribution models for flood prediction in Upper Benue River Basin-Part II[J]. Civil and Environmental Research, 2013, 3(2): 62-74.

[2] Panahi M, Jaafari A, Shirzadi A, et al. Deep learning neural networks for spatially explicit prediction of flash flood probability[J]. Geoscience Frontiers, 2021, 12(3): 101076.

[3] Dong X, Yu Z, Cao W, et al. A survey on ensemble learning[J]. Frontiers of Computer Science, 2020, 14: 241-258.

[4] Zhu Y, Tian D, Yan F. Effectiveness of entropy weight method in decision-making[J]. Mathematical Problems in Engineering, 2020, 2020(1): 3564835.

[5] Behzadian M, Otaghsara S K, Yazdani M, et al. A state-of the-art survey of TOPSIS applications[J]. Expert Systems with applications, 2012, 39(17): 13051-13069.

[6] Na S, Xumin L, Yong G. Research on k-means clustering algorithm: An improved k-means clustering algorithm[C]//2010 Third International Symposium on intelligent information technology

and security informatics. Ieee, 2010: 63-67.

[7] Xu J. Construction of the Evaluation Index System of “Five Development Concepts” Based on R-Type Clustering and Variation Coefficient in Hunan Province[C]//IOP Conference Series: Materials Science and Engineering. IOP Publishing, 2019, 490(6): 062045.