Hindawi
Complexity
Volume 2021, Article ID 7190446, 14 pages
https://doi.org/10.1155/2021/7190446

Research Article

A Prediction Method of Electromagnetic Environment Effects for UAV LiDAR Detection System

Min Huang,[1,2] Dandan Liu,[2] Liyun Ma,[1] Jingyang Wang ,[2] Yuming Wang ,[1] and Yazhou Chen [1]

[1] *National Key Laboratory on Electromagnetic Environment Effects, Army Engineering University, Shijiazhuang Campus, Shijiazhuang 050003, China*
[2] *School of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang 050018, China*

Correspondence should be addressed to Yuming Wang; ymking2006@163.com and Yazhou Chen; chen_yazhou@sina.com

Received 25 September 2021; Accepted 17 November 2021; Published 6 December 2021

Academic Editor: Min Xia

With the rapid development of science and technology, UAVs (Unmanned Aerial Vehicles) have become a new type of weapon in the informatization battlefield by their advantages of low loss and zero casualty rate. In recent years, UAV navigation electromagnetic decoy and electromagnetic interference crashes have activated widespread international attention. The UAV LiDAR detection system is susceptible to electromagnetic interference in a complex electromagnetic environment, which results in inaccurate detection and causes the mission to fail. Therefore, it is very necessary to predict the effects of the electromagnetic environment. Traditional electromagnetic environment effect prediction methods mostly use a single model of mathematical model and machine learning, but the traditional prediction method has poor processing nonlinear ability and weak generalization ability. Therefore, this paper uses the Stacking fusion model algorithm in machine learning to study the electromagnetic environment effect prediction. This paper proposes a Stacking fusion model based on machine learning to predict electromagnetic environment effects. The method consists of Extreme Gradient Boosting algorithm (XGB), Gradient Boosting Decision Tree algorithm (GBDT), K Nearest Neighbor algorithm (KNN), and Decision Tree algorithm (DT). Experimental results show that, comprising with the other seven machine learning algorithms, the Stacking fusion model has a better classification prediction accuracy of 0.9762, a lower Hamming code distance of 0.0336, and a higher Kappa coefficient of 0.955. The fusion model proposed in this paper has a better predictive effect on electromagnetic environment effects and is of great significance for improving the accuracy and safety of UAV LiDAR detection systems under the complex electromagnetic environment on the battlefield.

1. Introduction

Modern warfares are information and electronic warfare. Many enemies and our radars are deployed on the battlefield, coupled with natural electromagnetic radiation and man-made electromagnetic radiation interference, making the electromagnetic environment of the battlefield more complicated [1]. The UAV LiDAR detection system plays an important role in informatization electronic warfare operations. The complex electromagnetic environment has caused serious interference to the UAV LiDAR detection system, threatening the safety and combat effectiveness of the UAV [2].

The LiDAR detection system plays an important role in the flight safety of UAV. It is easily affected by the electromagnetic environment of the complex battlefield, which makes the UAV LiDAR detection system have detection errors, affects the construction of point cloud maps, and causes inaccurate target detection. When the UAV LiDAR detection system is subjected to electromagnetic interference during the flight, to ensure safety, measures such as leaving the interference zone and returning home are generally taken, but it will have a great impact on the completion of the mission. Research is done on the prediction method of the complex electromagnetic environment effect of the battlefield, so that the UAV LiDAR detection system can realize the intelligent prediction of the electromagnetic risk area, to make intelligent decisions to avoid the electromagnetic risk area, thereby improving the detection accuracy of the UAV LiDAR detection system and safety.

At present, the more popular learning methods are machine learning and deep learning. The research content of deep learning mainly involves methods such as convolutional neural networks, recurrent neural networks, and self-encoding neural networks, which usually mimic the mechanisms of the human brain to interpret data such as images, time series, and text. Electromagnetic environment effect prediction is an artificial intelligence process to complete machine decision-making with the help of a large number of experimental data. So we chose machine learning algorithms rather than deep learning to solve this problem. The traditional prediction methods of electromagnetic environment effects are mainly mathematical models and machine learning single models, such as the method of moments and Support Vector Machine algorithm (SVM) [3]. Traditional forecasting methods have relatively simple models and weaker ability to deal with nonlinear problems, and errors will occur in the process of forecasting. Because this model has many prerequisites and conditional restrictions, this model is not universal.

Machine learning is a multifield subject, involving statistics, probability, etc. Machine learning algorithms can handle nonlinear problems better and have the advantages of fast calculation and automatic learning. In this paper, machine learning algorithms are used to predict the effects of electromagnetic environment. In the process of research, the theory of fusion algorithm is introduced, and the electromagnetic environment effect prediction model of Stacking model fusion algorithm is constructed.

In this paper, experiments will be used to demonstrate the effectiveness of the electromagnetic environment effect analysis and prediction model based on machine learning. In this experiment, from adaptive boosting algorithm (ADB), SVM, Random Forest algorithm (RF), DT, XGB, and GBDT, KNN selects the model with better prediction effect from seven algorithms to form the Stacking fusion model to predict the electromagnetic environment effect. In the rest of this paper, we will focus on the details of the method.

The main contributions of this paper are as follows:

(1) By analyzing the experimental data, a Stacking fusion model based on machine learning (composed of XGB, GBDT, KNN, and DT algorithms) is proposed to predict the electromagnetic environment effects of the UAV LiDAR detection system.

(2) This method has proved its effectiveness by comparing it with seven other classification prediction methods of electromagnetic environment effects. Experimental results show that this method is more suitable for predicting the electromagnetic environment effects of UAV LiDAR detection systems.

2. Related Work

Due to the wide application of high-tech electronic technology in the military field, any military activity is under a certain electromagnetic environment. The electromagnetic radiation power of current navigation, radar, and communication equipment is increasing, and the frequency spectrum is constantly widening, making the electromagnetic environment of the battlefield increasingly complex. The emergence of electronic pulse weapons, the application of electronic warfare systems, and electromagnetic sources such as lightning and natural electromagnetic fields have made the electromagnetic environment of the battlefield worse [4]. UAV may encounter interference from radiation systems such as communications equipment, electronic interference, electronic deception, lightning, antiradiation weapons, radar, high-power microwave pulses, and nuclear battery pulses during their missions on the battlefield. The electromagnetic environment facing the drone is shown in Figure 1.

The traditional electromagnetic environment effect prediction method uses artificial mathematical modeling and single algorithm model. In 1999, Antonini et al. used numerical calculation methods to predict the electromagnetic interference of the electric drive system [5]. In 2009, Coco et al. used GRID-based methods to predict the electromagnetic field of the urban environment [6]. In 2010, Chen et al. used the entropy principle to predict complex electromagnetic signals in the battlefield [7]. In 2013, Ying et al. used statistical model methods to predict the electromagnetic environment [8]. In 2015, Alligier et al. used ridge regression and multiple linear regression methods to predict the climb of aircraft on the ground [9]. In 2016, Zhang et al. used the SVM algorithm to predict the UAV data link interference in the complex electromagnetic environment. Experiments showed that the SVM algorithm had advantages in nonlinear data prediction, but the accuracy of the prediction results needed to be improved [10]. In 2017, Yuan et al. used Bayesian networks to predict and evaluate the complex electromagnetic environment [11]. In 2019, Shu et al. used Artificial Neural Network (ANN) to predict electromagnetic interference [12]. In 2021, Zhang et al. used the GPR algorithm to predict the electromagnetic interference of the UAV dynamic data link [13]. In 2021, Kogut and Slowik used the Multilayer Perceptron (MLP) method to classify airborne laser sounding data. Compared with algorithms such as SVM and K-means, the classification accuracy had been improved to a certain extent [14].

From this, it can be seen that most of the current predictions of electromagnetic environment effects use traditional artificial mathematical models and single algorithm models, but the complex electromagnetic environment effects have nonlinearity, ambiguity, uncertainty, etc., so traditional predictions are used. The method is not effective in predicting the effects of a complex electromagnetic environment.

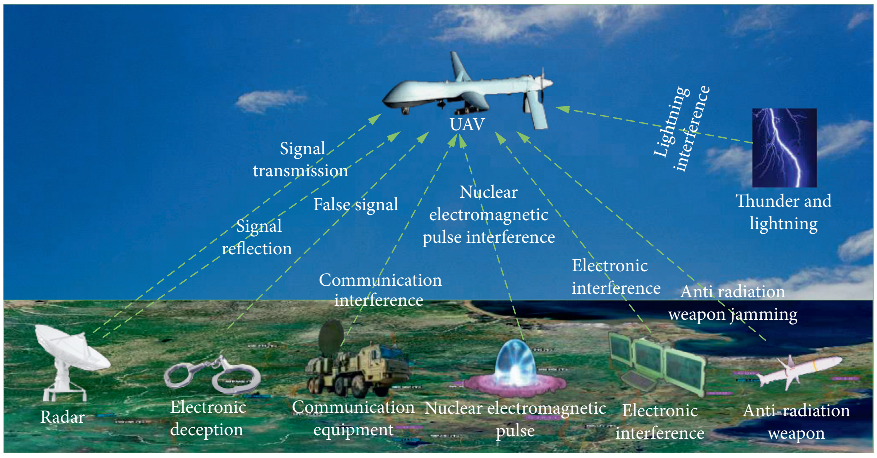

Figure 1: The electromagnetic environment that drones face.

3. Machine Learning Algorithm

3.1. Stacking Integrated Learning Algorithm. Ensemble learning is to combine different algorithm models for learning and use certain rules to merge different models to obtain better results. Integrated learning algorithms can solve problems such as classification and regression. In this experiment, the Stacking ensemble learning algorithm is used for classification prediction.

Stacking integrated learning algorithm is a hierarchical heterogeneous fusion model. The individual learner is called the primary learner, and the learner that combines the results is called the secondary learner. The training data used by the secondary learner is called the secondary training set, and the second training set data comes from the primary learner. Choose the XGB, GBDT, and KNN algorithm with better prediction effects from the seven algorithms of ADB, SVC, RF, DT, XGB, GBDT, and KNN as the primary learner, and choose the DT algorithm as the secondary learner. In this experiment, the data set is divided into the training set and test set. Use the training set to train the XGB, GBDT, and KNN models to obtain three primary learners, then predict the test set, and use the output value as the input value of the next stage, and the final label as the training output value. The DT secondary learner is trained, and the trained secondary learner is used for prediction. Since the data sets used in the two times are different, overfitting can be prevented to a certain extent.

3.2. Decision Tree Algorithm. A decision tree classification algorithm is a supervised machine learning algorithm, which trains a tree-type classification model from a given out-of-order training sample. In the process of classification training, a classification decision tree is established according to the principle of minimizing the loss function. In classification prediction, the test set data is used to predict the decision tree model. The CART algorithm is used in this experiment. The CRTA algorithm consists of feature selection, tree generation, and pruning, with CART decision tree feature selection. The Gini coefficient is used as the basis for splitting nodes in the CART algorithm [15]. The Gini coefficient is a judgment of the impurity of the model. The larger the coefficient, the higher the impurity and the bad characteristics. On the contrary, the impurity is low and the characteristics are better, as shown in

$$\mathrm{Gini}(p) = \sum_{k=1}^{K} P_k \left(1 - P_k\right) = 1 - \sum_{k=1}^{K} p_k^2. \tag{1}$$

K is the number of categories; p_k is the probability that the sample point belongs to the Kth category.

CART decision tree generation. Input test data set D and stop calculation conditions, and output CART decision tree.

(1) Suppose the training set is D, and calculate the Gini coefficients of all features on D. Suppose the possible value of feature A is a, test the correctness of $A = a$, divide the training data into D_1 and D_2, and calculate the Gini coefficient of D_1 and D_2 when $A = a$.

(2) From all A and all possible cutting points a, choose the cut point and feature with the smallest Gini coefficient as the best cut point and feature. Then produce two child nodes based on the best features and cut points, split the data set D, and assign it to two child nodes.

(3) Recursively call formula (1) and formula (2) on the two generated child nodes until the number of samples in the node is less than the threshold or the Gini coefficient is less than the threshold.

(4) Generate CART decision tree. CART decision tree pruning [16] algorithm is to subtract some subtrees from the bottom of the decision tree to make the model simple, which can improve the accuracy of predicting unknown data. Decision tree pruning is a dynamic process. Starting from the leaf node, the prediction error within the node and the prediction error after pruning are calculated from the bottom up. If the prediction error after pruning becomes smaller, then pruning is performed; otherwise, no pruning is performed. After pruning, the original nonleaf nodes inside will become leaf nodes. The

category of the new leaf node is determined by the Decision Tree algorithm, and the above steps are repeated until the minimum prediction error is found. The loss function is shown in

$$C_{\alpha}(T_t) = C(T_t) + \alpha|T_t|. \tag{2}$$

Note: α is the regularization parameter, $C(T_t)$ is the prediction error of the training set, and $|T_t|$ is the number of leaf nodes of the subtree.

3.3. KNN Algorithm. KNN algorithm, also known as the K nearest neighbor algorithm, is a machine learning algorithm that can solve classification and regression problems, and it is also a relatively mature algorithm in theory [17]. This experiment uses the KNN classification algorithm to classify according to the distance between different feature values [18]. The main idea of the algorithm is that when predicting a new value x, it is judged which category x belongs to according to the category of the nearest K points. In KNN, the dissimilarity between sample objects is determined by calculating the distance between objects. Generally, Manhattan distance or Euclidean distance is used to calculate the distance between sample objects [19], as shown in formulas (3) and (4):

$$d(x,y) = \sqrt{\sum_{k=1}^{n}|x_k - y_k|}, \tag{3}$$

$$d(x,y) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2}. \tag{4}$$

3.4. GBDT Algorithm. The full name of GBDT is a gradient descent tree. The main idea of the algorithm is to use an additive model to classify or regress data by continuously reducing the residuals generated during the training process [20]. This experiment uses the GBDT classification algorithm and uses the difference between the predicted probability value and the true predicted probability value to fit the loss. The flow of the GBDT classification algorithm is as follows.

(1) Suppose the number of classifications is k, and the log-likelihood loss function is shown in

$$L(y, f(x)) = -\sum_{k=1}^{K} y_k \log\, p_k(x). \tag{5}$$

(2) If the sample output category is k, then $y_k = 1$; the expression of probability p_k is shown in

$$p_k(x) = \frac{\exp(f_k(x))}{\sum_{k=1}^{K}\exp(f_l(x))}. \tag{6}$$

(3) According to formulas (5) and (6), the negative gradient error of category l corresponding to the i-th sample in the t-the round can be calculated; the negative gradient error formula is shown in

$$r_{til} = -\left[\frac{\partial L(y_{i,} f(x_i))}{\partial f(x_i)}\right]_{f_k(x)=f_{l,t-1}(x)} = y_{il} - p_{l,t-1}(x_i). \tag{7}$$

(4) Generate a decision tree; the best negative gradient fitting value of each leaf node is shown in

$$C_{tjl} = \frac{K-1}{K}\,\frac{\sum_{x_{i\in R_{tjl}}} r_{til}}{\sum_{x_{i\in R_{tjl}}}|r_{til}|(1-|r_{til}|)}. \tag{8}$$

3.5. XGBoost Algorithm. The full name of the XGB algorithm is Extreme Gradient Boosting. It is a gradient boosting tree algorithm based on decision trees. After multiple iterations, each iteration produces a weak classifier. Each classifier is performed based on the previous round of classifier residuals, training. Weak classifiers need to meet the basic requirements of high deviation and low variance, because the process of algorithm training is to continuously reduce the deviation, thereby improving the accuracy of the final classifier. In general, the weak classifier uses the CART decision tree. Due to the simplicity and high deviation requirements, the depth of each classification tree will not be very deep. The final classifier is obtained by the weighted summation of the weak classifiers obtained in each round of training. The objective function is shown in

$$\mathrm{Obj}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + \text{constant}. \tag{9}$$

Note: l is the loss function; $\Omega(f_t)$ is shown in

$$\Omega(f_t) = rT + \frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2. \tag{10}$$

4. Experiments

4.1. Data Sources. The experimental data in this paper comes from the radiation interference experiment of the UAV LiDAR detection system. The UAV LiDAR detection system experiment consists of two parts, the electromagnetic interference radiation emission system and the UAV working system. In an electromagnetic radiation interference emission system, a signal generator generates electromagnetic signals, a power amplifier is used to amplify the power, and then the directional coupler feeds the radiating antenna. The power meter measures the power of the power amplifier through the directional coupler, which can accurately measure the forward output power and the backward reflected power and monitor the working status of the experimental system. Adjust the gain multiple of the power amplifier and the output level of the signal generator; the intensity of the radiated electric field can be adjusted. The radiation interference experiment of the UAV LiDAR detection system is shown in Figure 2.

The radar technical indicators are shown in Table 1.

In the radiation interference experiment of the UAV LiDAR detection system, use the laser detection radar in the drone as the test equipment and use strong electromagnetic fields to conduct radiation interference experiments. Through experiments, verify the interference of the equipment under test in different electromagnetic field environments. 135,658 pieces of data are obtained through experiments. The data that needs to be collected in real-time during the experiment include error, angle, frequency, and field strength. The range of distance is 0–150 m, and the range of angle is 0°–360°. The frequency range is 1.2 Hz–2.5 Hz, and the field strength range is 25 V/m–200 V/m. The target value is obtained according to the error. The data is divided into four categories by *K*-means clustering classification. The divided intervals are [0, 0.03], [0.03, 0.06], [0.06, 0.09], greater than 0.09. The device under test is slightly sensitive to electromagnetic interference in the interval [0, 0.03], the device under test is slightly sensitive to electromagnetic interference in the interval [0.03, 0.06], and the device under test is slightly sensitive to electromagnetic interference in the interval [0.06, 0.09]. Degree of sensitivity: the device under test is highly sensitive to electromagnetic interference in the interval greater than 0.09. Part of the sample data is shown in Table 2.

4.2. Data Sources. The data preprocessing in this experiment mainly includes three aspects: abnormal point processing [21], sample equalization, and data standardization.

4.2.1. Handling of Abnormal Points. The LiDAR detection system on the UAV has problems such as gaps and nonsmooth surfaces during the detection and scanning process, so abnormal points will inevitably appear. We use the K-means algorithm to deal with the abnormal points. The main idea of the algorithm is to use the elbow method to determine the number of clusters. According to the results of the clustering, calculate the distance from each point to the cluster center, and compare the distance with the threshold. The abnormal point is the abnormal point that is greater than the threshold. Click to delete it. The SSE formula is shown in formula (11). The Euclidean distance formula is shown in formula (12):

$$\mathrm{SSE} = \sum_{i=1}^{k} \sum_{p \in C_i} \left| p - m_t \right|^2, \tag{11}$$

$$\mathrm{dist}(X, Y) = \sqrt{\sum_{i=1}^{n} \left(x_i - y_i\right)^2}. \tag{12}$$

4.2.2. Sample Equalization. Unbalanced sample categories will result in fewer features in the classification with small sample size, and it is difficult to find the regular pattern. After the model is trained, it is easy to rely on a small number of data samples to cause overfitting, which makes the model predict new data. The accuracy obtained is poor, so the data set needs to be equalized. In this experiment, the SMOTE algorithm is used to solve the problem of unbalanced data set samples [22]. The SMOTE algorithm analyzes and simulates a small number of category samples and then adds the simulated data to the data set to balance the unbalanced data set. The simulation process of a few categories of samples of the SMOTE algorithm draws on the KNN algorithm. Select a sample in a minority category, use Euclidean distance to calculate the distance from this sample to all samples in the minority category sample data set, and get its *K* nearest neighbors. The Euclidean distance formula is shown in

$$d = \sqrt{\left(x_1 - x_2\right)^2 + \left(y_1 - y_2\right)^2}. \tag{13}$$

The sampling ratio is set according to the sample imbalance ratio, and then the sampling magnification n is determined, and several samples are randomly selected from the *K* nearest neighbors of each minority category. Randomly select a number from [0, 1], multiply it by the randomly selected neighbor, and add x. The formula is shown in

$$x_{\text{new}} = x + \mathrm{rand}(0, 1) * (\tilde{x} - x). \tag{14}$$

The SMOTE algorithm does not use random oversampling, which effectively prevents the problem of overfitting and makes the model have better generalization [23]. The sample before sampling is shown in Figure 3. Figure 4 shows the sample after sampling. It can be seen from Figure 3 that the data set has a sample imbalance. It can be seen from Figure 4 that the sample data set has reached equilibrium after sample equalization using the SMOTE algorithm.

4.3. Evaluation Index. This paper uses accuracy, Kappa coefficient, and Hamming distance to evaluate the prediction effect of electromagnetic environment effects.

4.3.1. Accuracy. Accuracy is one of the evaluation indicators commonly used in classification problems. Accuracy refers to the percentage of the correct result of classification prediction to the total number of classified samples. The formula is shown in

$$\text{Accurancy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}. \tag{15}$$

TP means that the classifier identified the sample correctly, and the classifier considered the sample as positive. TN means that the classifier identified the sample correctly, and the classifier considered the sample as negative. FP means that the classifier identified the sample incorrectly, and the classifier considered the sample as positive; therefore, the sample is actually negative. FN means that the classifier identified the sample incorrectly, and the classifier considered the sample as negative; therefore, the sample is actually positive. The sample is actually a positive sample.

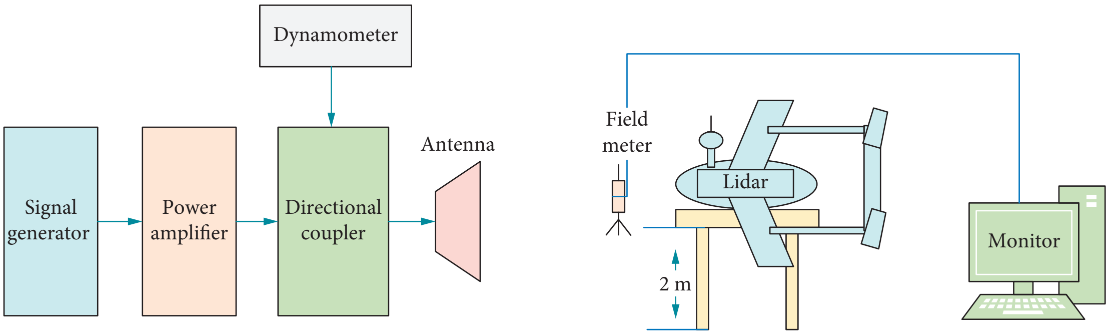

Figure 2: Radiation interference experiment of UAV LiDAR detection system.

Table 1: Detailed technical indicators.

Technical name	Parameter
Laser class	Class1
Laser	905 nm
Field of view (horizontal)	360°
Horizontal angle resolution	0.07°
Ranging accuracy	±3 cm
Frame rate	10 Hz
Number of point clouds	25 kpt/s
Ranging range	0.5~100 m
Light resistance	80 klux
Communication interface	Ethernet
Power supply	DC 12 V
Power consumption	<5 W
Operating temperature	−10°C~60°C
Storage temperature	−40°C~105°C
Relative humidity	0–95%
Weight	465 g
Equipment size (mm)	85 (D) ∗ 64.5 (H)
Protection level	IP65

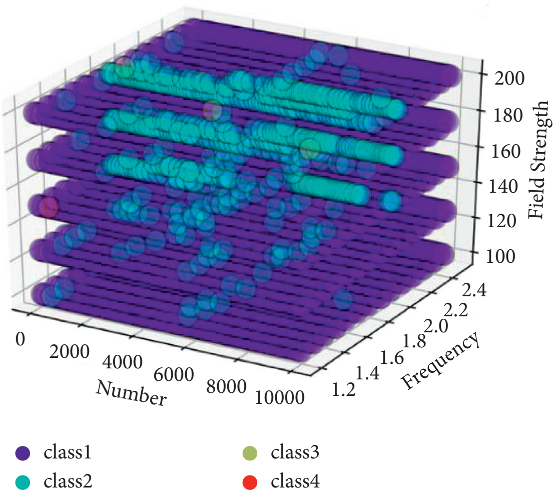

Figure 3: Before sample sampling.

Table 2: Part of sample data.

Angle	Frequency (Hz)	Field strength (V/m)	Error (cm)	Classification
12	1.2	50	0.002	1
24	1.2	100	0.006	1
228	2.2	150	0.04	2
240	2.1	150	0.05	2
100	1.9	200	0.07	3
168	1.9	200	0.09	3
220	1.2	100	0.12	4
320	1.3	100	0.15	4

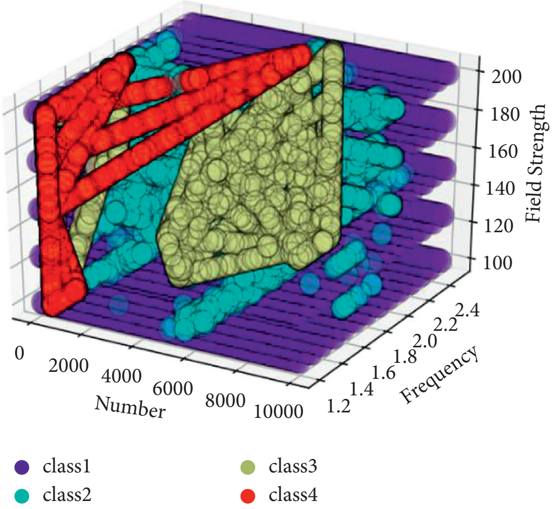

Figure 4: After sample sampling.

4.3.2. Kappa Coefficient. The value range is [−1, 1]. The larger the Kappa coefficient, the more accurate the model classification result. The formula is shown in

$$K = \frac{P_0 - P_e}{1 - P_e}. \tag{16}$$

P_0 represents the total classification accuracy; P_e represents (the number of real samples of the i-th type multiplied by the number of predicted samples)/the square of the total number of samples.

4.3.3. Hamming Distance. The Hamming distance is used to measure the distance between the predicted label and the real label, and the value range is [0, 1]. The distance is 0, indicating that the real result is the same as the predicted result. If the distance is 1, it means that the actual result is opposite to the predicted result. The smaller the Hamming distance, the better. The formula is shown in

$$\text{Hamming} = \frac{1}{N}\sum_{i=1}^{N}\frac{\text{XOR}\left(Y_{i,j}, P_{i,j}\right)}{L}. \tag{17}$$

Note: N represents the number of samples, L represents the number of tags, $Y_{i,j}$ represents the true value of the j-th component in the i-th prediction result, $P_{i,j}$ represents the predicted value of the i-th component in the j-th prediction result, and XOR represents exclusive OR.

4.4. Model Flow Chart. The main voting algorithms in machine learning are the bagging algorithm and the boosting algorithm. The bagging algorithm and the boosting algorithm are relatively simple to average or vote on the results of the basic model, and there may be large learning errors. Therefore, this article uses another learning method, Stacking model fusion algorithm. The Stacking model fusion algorithm does not perform simple logic processing on the results of the model but adds a layer outside the model. There are two layers of models in total. The first layer model is established through the prediction training set, and then the result of the training set prediction model is used as input, and then the second layer new model is trained to obtain the final result. Stacking model fusion algorithm can reduce the deviation of bagging algorithm or boosting algorithm.

From the model prediction results of ADB algorithm, SVC algorithm, RF algorithm, DT algorithm, XGB algorithm, GBDT algorithm, and KNN algorithm, we can see that DT algorithm, XGB algorithm, GBDT algorithm, and KNN algorithm have better prediction results, so using these four models as the base model of Stacking, the algorithm and input of the metamodel have an important impact on Stacking. The input features of the metamodel are the combination of all the prediction results of the base model, and the method splices all the features of the base model without missing and fully uses all the data. The metamodel usually selects the best prediction result from the base model, so the KNN algorithm is chosen as the metamodel in this experiment to ensure the accuracy of Stacking model prediction.

Flowchart of seven models is shown in Figure 5.

The main process is as follows:

(1) Data preprocessing. There are unbalanced sample category distribution and abnormal points in the data set, and the SMOTE algorithm is used for sample category equalization processing. Use the K-means clustering algorithm to find outliers and delete them. The Z-score algorithm is used to standardize the data set.

(2) Model selection, training, and prediction. Choose a machine learning model from the seven models, in turn, use the training set to train the model, and then use the test set to test and predict the model.

(3) Model evaluation. The accuracy, Kappa coefficient, and Hamming distance are used to evaluate the model classification prediction results.

Stacking (DXGK) fusion model flowchart is shown in Figure 6.

The main process is as follows:

(1) Data preprocessing. There are unbalanced sample category distribution and abnormal points in the data set, and the SMOTE algorithm is used for sample category equalization processing. Use K-means clustering algorithm to find outliers and delete them. The Z-score algorithm is used to standardize the data set.

(2) Model building and training. In this experiment, ADB algorithm, SVC algorithm, RF algorithm, DT algorithm, XGB algorithm, GBDT algorithm, and KNN algorithm are used to predict the electromagnetic environment effect. Through the comparison of the prediction results of each model, it can be seen that the DT algorithm, the XGBoost algorithm, the GBDT algorithm, and the KNN algorithm have good prediction results. In the experiment, a two-layer Stacking fusion algorithm model is constructed. The first layer is composed of multiple basic learners, and the second layer of metamodel is based on the output of the first layer of basic learners as features and added to the training set for retraining, thereby obtaining Complete Stacking model. The Stacking model fusion construction generally chooses the algorithm with better prediction effect as the base model, so choose the DT algorithm and XGBoost algorithm, GBDT algorithm, and KNN algorithm as the base model of Stacking model fusion algorithm. The metamodel usually chooses the best prediction effect in the base model, so the KNN algorithm is used as the metamodel to construct the Stacking model fusion algorithm.

(3) Model evaluation. The accuracy, Kappa coefficient, and Hamming distance are used to evaluate the model classification prediction results.

4.5. Experimental Comparison. In this section, the electromagnetic interference comparison diagram of the UAV LiDAR detection system is shown, and the electromagnetic environment effect prediction diagram of ADB, SVC, RF, DT, XGB GBDT, KNN, and Stacking (DXGK) is shown.

(1) Electromagnetic interference comparison chart of UAV LiDAR detection system. The complex electromagnetic environment will cause electromagnetic interference to the UAV LiDAR detection system. The electromagnetic interference comparison diagram of the UAV LiDAR detection system is shown in Figure 7.

The red line in Figure 7 represents the data before electromagnetic interference, and the blue line represents the data after electromagnetic interference. It can be seen that electromagnetic interference will cause strong interference to the UAV LiDAR detection system.

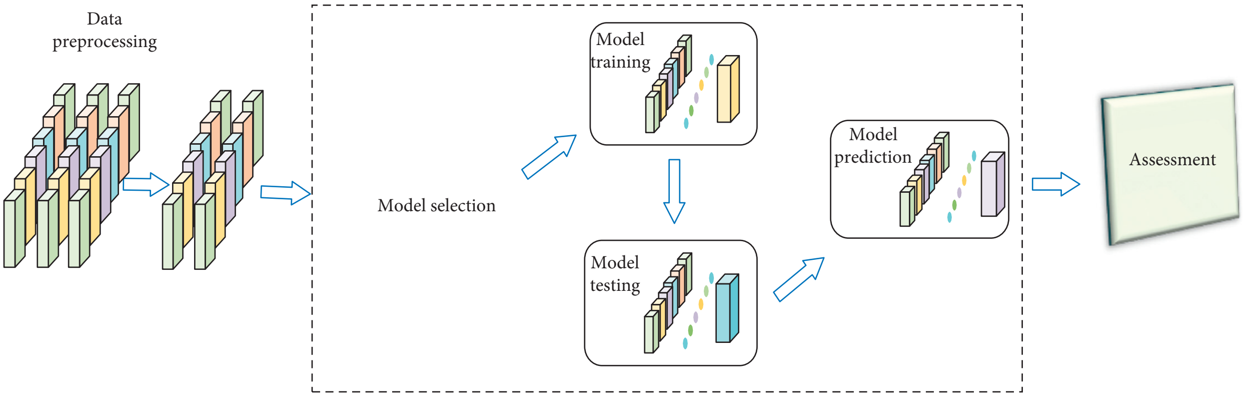

Figure 5: Classification and prediction flowchart of seven models.

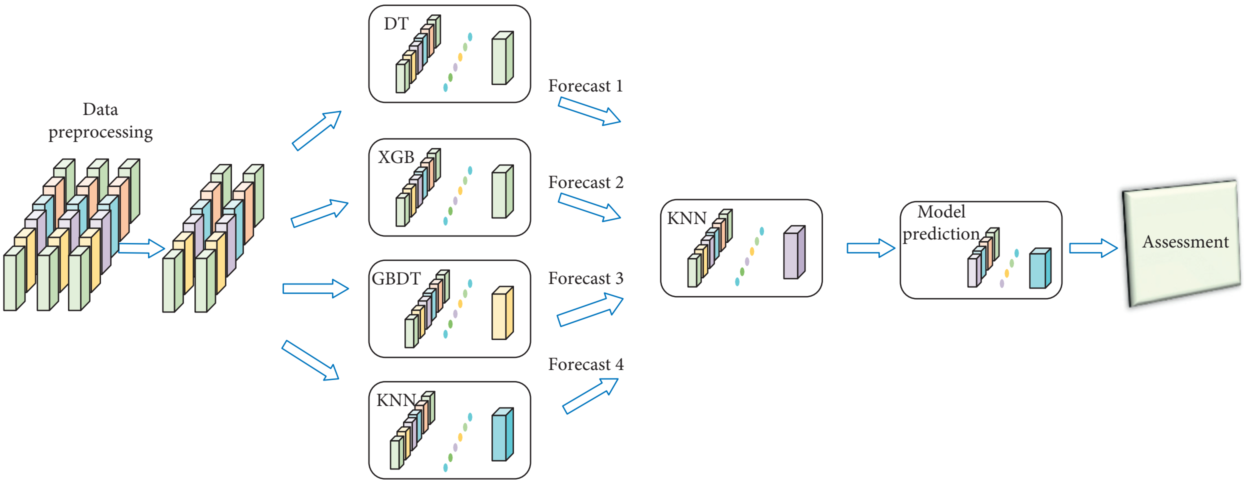

Figure 6: Classification and prediction flowchart of seven models.

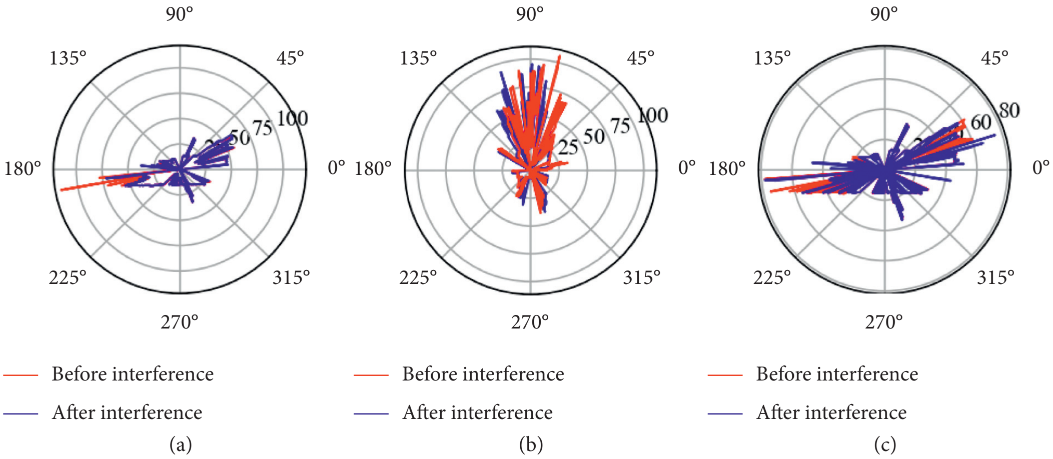

Figure 7: Comparison of electromagnetic interference of UAV LiDAR detection system ((a) is the frequency of 2.5 Hz, the field strength is 145 V/m, (b) is frequency of 86 Hz, field strength is 205 V/m, (c) is the frequency of 1.2 Hz, the field strength is 83 V/m).

(2) Use the grid search method to optimize each parameter, and the parameter optimization is shown in Table 3.

(3) Classification prediction results of each model. Use ADB, SVC, RF, DT, XGB, GBDT, KNN, Stacking (DXGK) fusion model, and other algorithms to classify and predict electromagnetic environment effects. The fitting graph of the predicted value and the true value is shown in Figures 8–15.

From Figures 8–15, it can be seen that, among the eight prediction methods, the true value and predicted value fit

Table 3: Parameter optimization table.

Model	Parameter
ADB	AdaBoostClassifier (n_estimators = 100, learning_rate = 0.8, algorithm = SAMME.R)
SVC	SVC (kernel = rbf, gamma = 0.2, decision_function_shape = ovo, C = 1)
RF	RandomForestClassifier (criterion = "gini," max_depth = 12, max_leaf_nodes = 20)
DT	DecisionTreeClassifier (min_samples_split = 10, max_depth = 20, splitter = random)
XGB	XGBClassifier (eta = 0.1, objective = multi:softmax, num_class = 4)
GDBT	GradientBoostingClassifier (n_estimators = 90, learning_rate = 0.3, loss = deviance)
KNN	KNeighborsClassifier (n_neighbors = 85, Algorithm = auto)

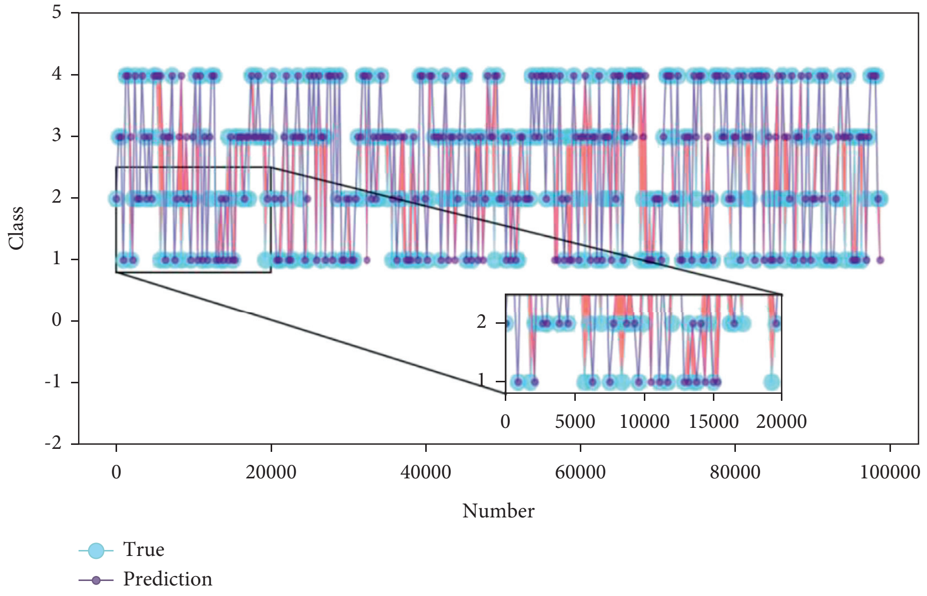

Figure 8: Fitting diagram of the true and predicted values of the ADB model (the accuracy value is 0.7360, the Hamming value is 0.2639, and the Kappa coefficient value is 0.6480).

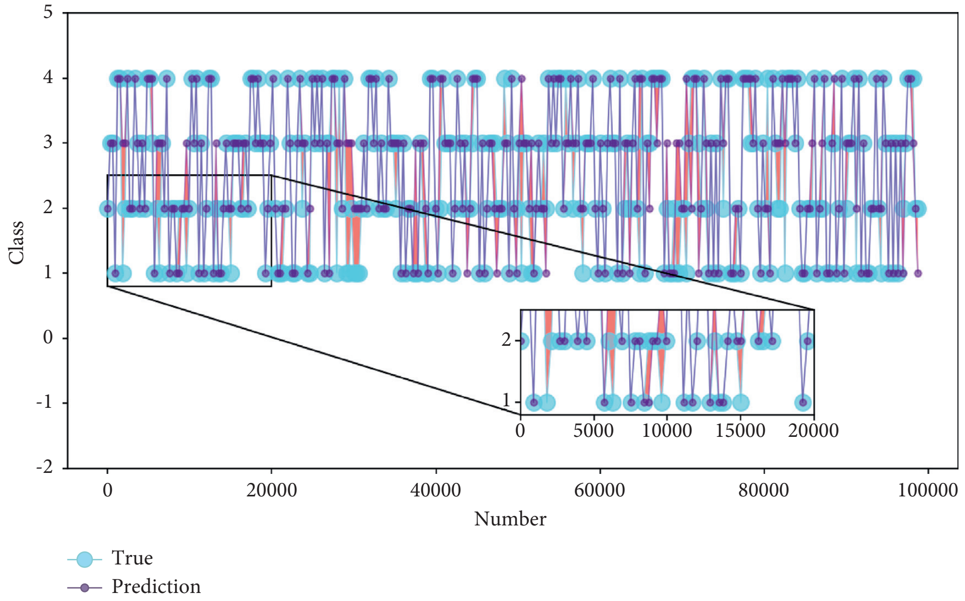

Figure 9: Fitting diagram of the true and predicted values of the SVC model (the accuracy value is 0.8150, the Hamming value is 0.1849, and the Kappa coefficient value is 0.7534).

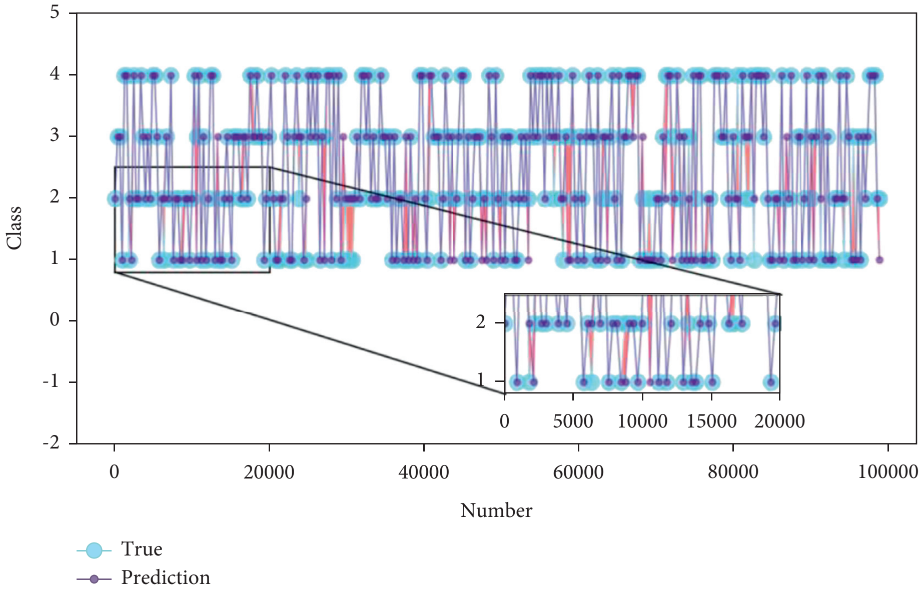

Figure 10: Fitting diagram of the true and predicted values of the RF model (the accuracy value is 0.8717, the Hamming value is 0.1282, and the Kappa coefficient value is 0.8290).

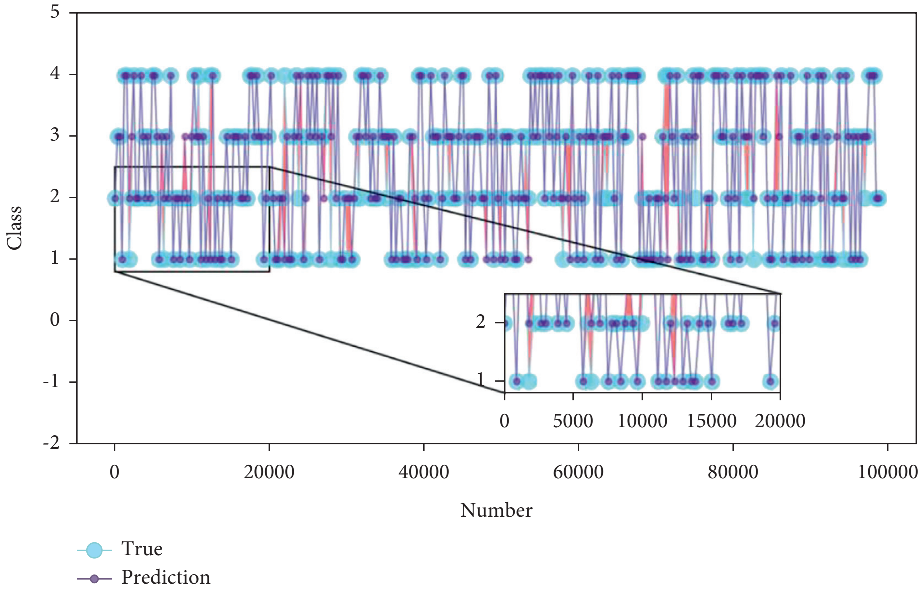

Figure 11: Fitting diagram of the true and predicted values of the DT model (the accuracy value is 0.8850, the Hamming value is 0.1149, and the Kappa coefficient value is 0.8467).

from low to high as ADB, SVC, RF, DT, XGB, GBDT, KNN, and Stacking (DXGK). It can be seen from the figure and evaluation indicators that the true value of the ADB algorithm fits poorly with the predicted value, and the true value and predicted value of the Stacking (DXGK) algorithm fit better.

Using accuracy, Hamming distance, and Kappa coefficient as the evaluation indicators of each model, the comparison results of the evaluation indicators of the eight algorithm models are shown in Table 4. The comparison results of the evaluation indicators of the eight algorithm models are shown in Figure 16.

It can be seen from Table 4 and Figure 16 that the performance of the eight algorithm models is ranked from low to high, ADB, SVC, RF, DT, XGB, GBDT, KNN, and Stacking (DXGK). From the evaluation results of the algorithm model, it can be seen that the accuracy of the ADB model is 0.7360, the Hamming distance is 0.2639, and the Kappa coefficient is 0.6480. The various model evaluation indicators of the ADB model are the lowest among the eight algorithm models. The Stacking (DXGK) fusion model accuracy rate is 0.9762, Hamming distance is 0.0336, and Kappa coefficient is 0.9552. The Stacking (DXGK) model is compared with the other seven models, and it can be concluded that the various evaluation

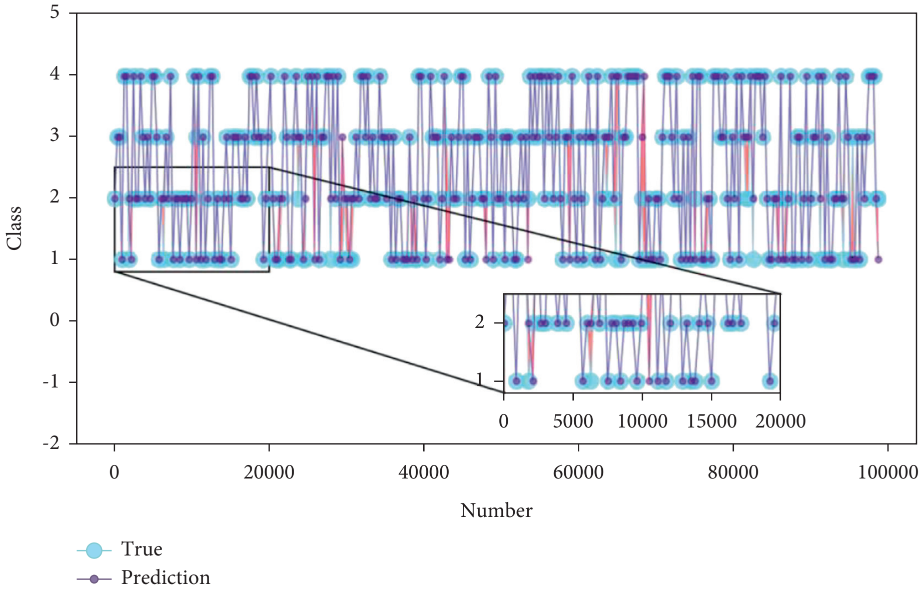

Figure 12: Fitting diagram of the true and predicted values of the XGB model (the accuracy value is 0.9085, the Hamming value is 0.0914, and the Kappa coefficient value is 0.8780).

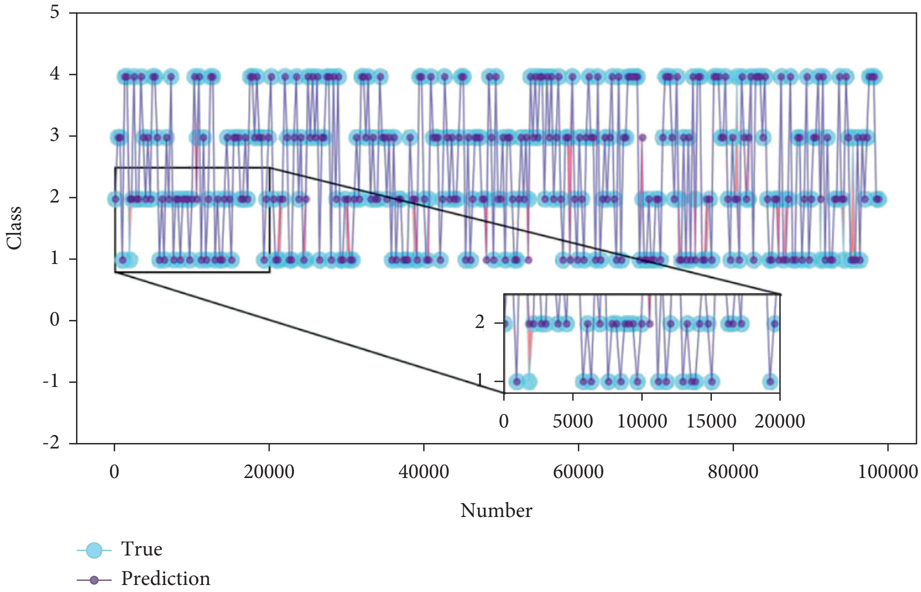

Figure 13: Fitting diagram of the true and predicted values of the GBDT model (the accuracy value is 0.9497, the Hamming value is 0.0502, and the Kappa coefficient value is 0.9329).

indicators are good. Therefore, if you choose a machine learning algorithm to predict the electromagnetic environment effects and improve the detection accuracy and safety of the UAV LiDAR detection system in the complex battlefield electromagnetic environment, the Stacking (DXGK) model is a more suitable one.

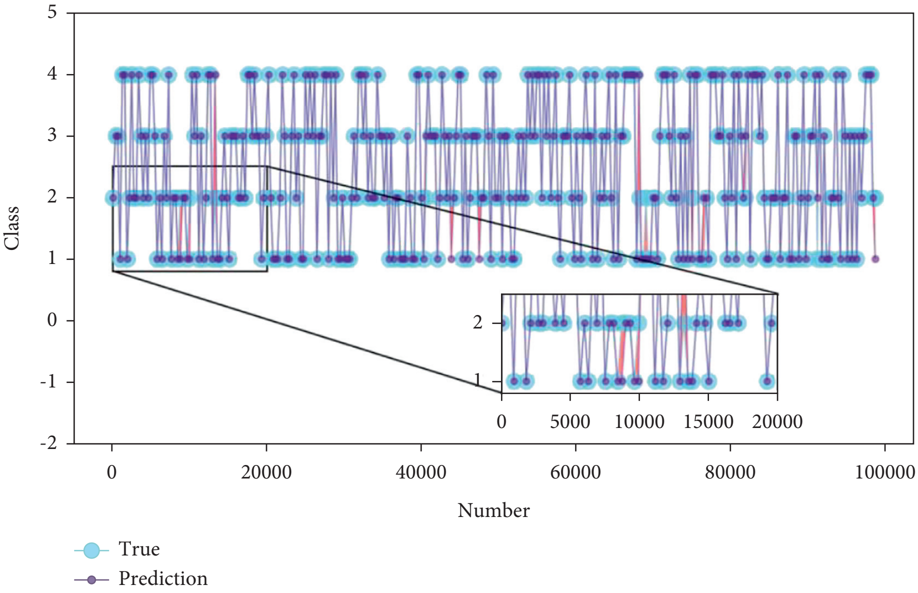

Figure 14: Fitting diagram of the true and predicted values of the KNN model (the accuracy value is 0.9636, the Hamming value is 0.0363, and the Kappa coefficient value is 0.9515).

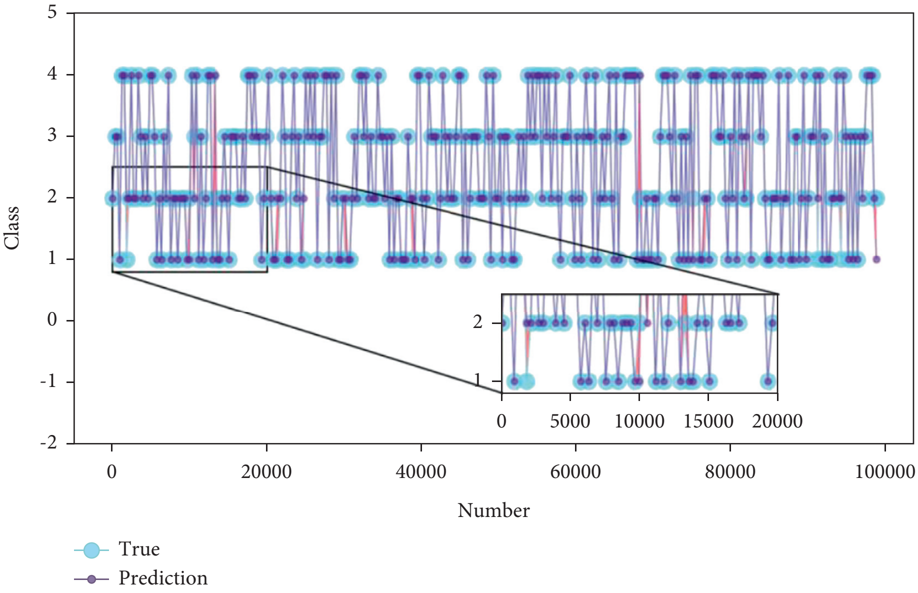

Figure 15: Stacking (DXGK) fusion model true value and predicted value fitting diagram (the accuracy value is 0.9762, the Hamming value is 0.0336, and the Kappa coefficient value is 0.9652).

Table 4: Comparison of eight model indicators.

Model	Accuracy	Hamming	Kappa
ADB	0.7360	0.2639	0.6480
SVC	0.8150	0.1849	0.7534
RF	0.8717	0.1282	0.8290
DT	0.8850	0.1149	0.8467
XGB	0.9085	0.0914	0.8780
GBDT	0.9497	0.0502	0.9329
KNN	0.9636	0.0363	0.9515
Staking (DXGK)	**0.9762**	**0.0336**	**0.9652**

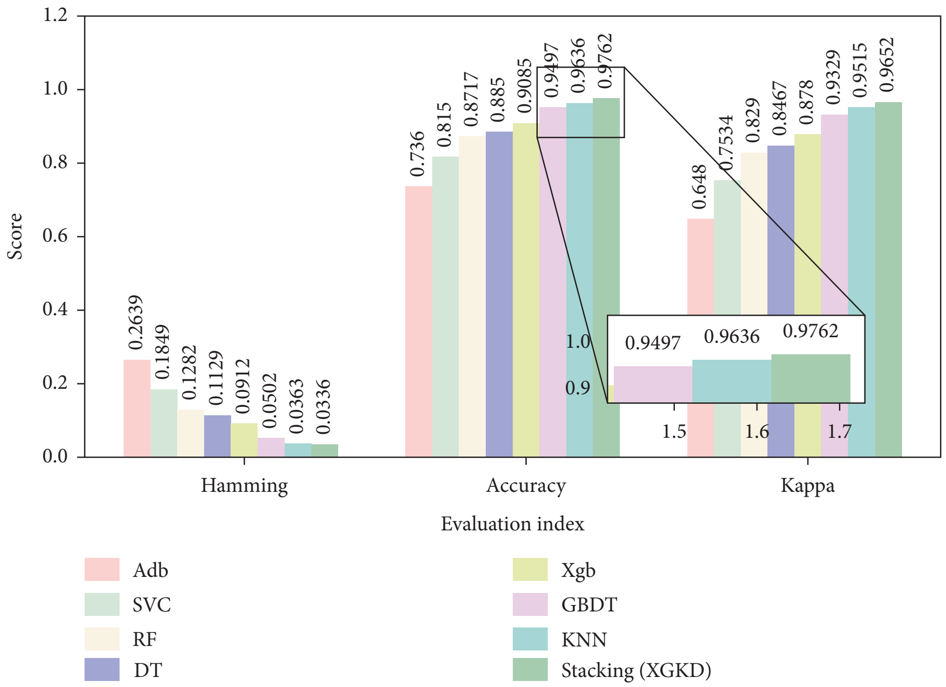

Figure 16: Comparison of results of eight algorithm models.

5. Conclusions

In this paper, a Stacking (DXGK) fusion model based on machine learning is proposed to predict electromagnetic environment effects. Compared with traditional mathematical modeling methods and single models, the fusion model has a higher accuracy rate, better robustness, and generalization ability. The experimental results show that the Stacking (DXGK) fusion model has a better prediction effect than the ADB, SVC, RF, DT, XGB, GBDT, and KNN. It can be seen that the classification prediction accuracy of the single model is low, and the use of a multiple fusion model can effectively improve the accuracy of the classification prediction. Therefore, the Stacking (DXGK) fusion model is more suitable for predicting electromagnetic environment effects and can provide a corresponding reference for improving the detection accuracy and safety of the UAV LiDAR detection system.

The future research work mainly focuses on how to adjust the model parameter values to further improve the prediction accuracy. It is necessary to further study the antielectromagnetic interference of UAV LiDAR detection system in the complex electromagnetic environment.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by Natural Science Foundation of Hebei Province under Grant ZD2018236 and Foundation of Hebei University of Science and Technology under Grant 2019-ZDB02.

References

[1] Q. Qu, S. Wei, S. Liu, J. Liang, and J. Shi, “JRNet: jamming recognition networks for radar compound suppression jamming signals,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15035–15045, 2020.

[2] K.-J. Li, Y.-Z. Xie, Y.-H. Chen, and Y.-C. Hui, “Multinomial regression model for the evaluation of multilevel effects caused by high-power electromagnetic environments,” *IEEE Transactions on Electromagnetic Compatibility*, vol. 61, no. 1, pp. 149–156, 2019.

[3] H. Jiang, B. Zou, C. Xu, J. Xu, and Y. Y. Tang, “SVM-Boosting based on Markov resampling: theory and algorithm,” *Neural Networks*, vol. 131, pp. 276–290, 2020.

[4] J. Zheng and G. Wei, “New development of electromagnetic compatibility in the future: cognitive electromagnetic environment adaptation,” in *Proceedings of the Global Symposium on Millimeter-Waves & Terahertz*, pp. 1–3, GSMM, Nanjing, China, May 2021.

[5] G. Antonini, S. Cristina, and A. Orlandi, “A prediction model for electromagnetic interferences radiated by an industrial power drive system,” *IEEE Transactions on Industry Applications*, vol. 35, no. 4, pp. 870–876, 1999.

[6] S. Coco, A. Laudani, and G. Pollicino, “GRID-based prediction of electromagnetic fields in urban environment,” *IEEE Transactions on Magnetics*, vol. 45, no. 3, pp. 1060–1063, 2009.

[7] H. Y. Chen, D. Z. Zhang, Y. Wang et al., “Maximum complexity principle prediction of battlefield electromagnetic signal

environment," *Chinese Journal of Radio Science*, vol. 25, pp. 1157–1162, 2010.

[8] G. Ying, W. Feng-Hua, H. U. Zhan-Tao et al., "Design and implementation of prediction system for complex electromagnetic environment," *Modern Electronics Technique*, vol. 36, pp. 146–150, 2013.

[9] R. Alligier, D. Gianazza, and N. Durand, "Machine learning and mass estimation methods for ground-based aircraft climb prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3138–3149, 2015.

[10] W. W. Zhang, W. R. Ding, and C. H. Liu, "Prediction of interference effect on UAV data link in complex environment," *Systems Engineering and Electronics*, vol. 38, 2016.

[11] H. Yuan, D. Wenrui, and L. Chunlei, "Assessment and prediction of complex electromagnetic environment based on Bayesian network," in *Proceedings of the IEEE International Conference on Unmanned Systems (ICUS)*, pp. 120–125, Beijing, China, 2017.

[12] Y.-F. Shu, X.-C. Wei, J. Fan, R. Yang, and Y.-B. Yang, "An equivalent dipole model hybrid with artificial neural network for electromagnetic interference prediction," *IEEE Transactions on Microwave Theory and Techniques*, vol. 67, no. 5, pp. 1790–1797, 2019.

[13] D. Zhang, M. Zhao, E. Cheng, and Y. Chen, "GPR-based EMI prediction for UAV's dynamic datalink," *IEEE Transactions on Electromagnetic Compatibility*, vol. 63, no. 1, pp. 19–29, 2021.

[14] T. Kogut and A. Slowik, "Classification of airborne laser bathymetry data using artificial neural networks," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 1959–1966, 2021.

[15] R. Rosati, L. Romeo, C. A. Goday, T. Menga, and E. Frontoni, "Machine learning in capital markets: decision support system for outcome analysis," *IEEE Access*, vol. 8, pp. 109080–109091, 2020.

[16] S. Buschjäger and K. Morik, "Decision tree and random forest implementations for fast filtering of sensor data," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 1, pp. 209–222, 2018.

[17] P. Kumar Sinha, "Modifying one of the machine learning algorithms kNN to make it independent of the parameter k by Re-defining neighbor," *International Journal of Mathematics and Soft Computing*, vol. 6, no. 4, pp. 12–25, 2020.

[18] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient kNN classification with different numbers of nearest neighbors," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1774–1785, 2018.

[19] L. Cui, Y. Zhang, R. Zhang, and Q. H. Liu, "A modified efficient KNN method for antenna optimization and design," *IEEE Transactions on Antennas and Propagation*, vol. 68, no. 10, pp. 6858–6866, 2020.

[20] Z. Wen, J. Shi, B. He, J. Chen, K. Ramamohanarao, and Q. Li, "Exploiting GPUs for efficient gradient boosting decision tree training," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 12, pp. 2706–2717, 2019.

[21] C. Ge, Z. Wang, and X. Zhang, "Robust long-term spectrum prediction with missing values and sparse anomalies," *IEEE Access*, vol. 7, pp. 16655–16664, 2019.

[22] Y. Yan, R. Liu, Z. Ding, X. Du, J. Chen, and Y. Zhang, "A parameter-free cleaning method for SMOTE in imbalanced classification," *IEEE Access*, vol. 7, pp. 23537–23548, 2019.

[23] K. Cheng, C. Zhang, H. Yu, X. Yang, H. Zou, and S. Gao, "Grouped SMOTE with noise filtering mechanism for classifying imbalanced data," *IEEE Access*, vol. 7, pp. 170668–170681, 2019.