Journal of Computers Vol. 35 No. 5, October 2024, pp. 61-72
doi: 10.53106/199115992024103505005

Machine Learning-Based Algorithms Applied to Identifying Drug Smuggling via Postal and Express Delivery Channels

Mingyue Qiu[1*], Xueying Zhang[2], Linsheng Jiang[1], and Xinmeng Wang[1]

[1] School of information Technology, Nanjing Forest Police College, Nanjing, 210023, China
[2] Key Laboratory of Virtual Geographic Environment, Nanjing Normal University, Nanjing, 210023, China
qiumy@nfpc.edu.cn

Received 22 September 2023; Revised 26 September 2023; Accepted 28 September 2023

Abstract. Globalization and automation of logistics and delivery services is accompanied with increasing risks of illegal goods and drugs being smuggled in postal and express packages, falsifying declaration information, and evading customs supervision and crackdown. Focusing on the schemes of drug smuggling through delivery channels, this study adopts the method of semi-supervised learning, using the features extracted from the case data of drug smuggling crimes through postal channels as labels, utilizing a small amount of labeled data and a large amount of unlabeled data for learning and prediction, and carrying out optimization and adjustment to improve accuracy and stability. In particular, the valid fields of 2299 cases data and validation data were obtained through text mining. The study applies four machine learning algorithms (decision tree, multilayer perceptron, support vector machine, and naive Bayes) to accurately predict presence or absence of hidden drugs in parcels sent through postal channels. Furthermore, this article compared and analyzed the specific code implementation from the perspectives of algorithm accuracy, cross-validation accuracy, and code response time.

Keywords: drug smuggling, delivery channels, big data, machine learning

1 Introduction

In the context of today's globalization, the flow of various goods, capital, personnel, and information between countries has become increasingly frequent. However, at the same time, smuggling crimes are also increasing. Postal channels, as a common smuggling channel, are used by criminals to transport dangerous goods, infringe upon intellectual property goods, and counterfeit products. Among them, drugs have become this chain's main target of smuggling crimes. Drug smuggling destroys individuals' health and families and affects social stability and security. Therefore, it is particularly important to crack down on drug smuggling crimes through postal channels.

Postal channels refer to the information and physical delivery channels formed based on postal companies, express delivery enterprises, and other delivery service activities. Smuggling drugs through postal channels refers to the intentional violation of customs regulations, evasion of customs supervision, and the delivery of mail, packages, and other items disguised or hidden with drugs to postal companies, express delivery enterprises, and other delivery industries. It is a criminal method of drug transfer through the postal channels at customs ports using cross-border mail and parcel delivery services provided by delivery enterprises.

Criminal suspects use false sender and recipient information to disguise themselves as ordinary express or mail items for cross-border transportation of drugs, including but not limited to marijuana, heroin, cocaine, and methamphetamine, to obtain illegal profits. This form of crime takes advantage of modern delivery channels' convenience and wide coverage. It has the characteristics of concealment and efficiency, making it more difficult to detect and combat drug trafficking.

This article uses the features extracted from the case data of drug smuggling crimes through postal channels as labels, uses a small amount of labeled data and a large amount of unlabeled data for learning and prediction, and performs optimization and adjustment to improve accuracy and stability. Four algorithms, namely decision tree (DT), multilayer perceptron (MLP), support vector machine (SVM), and naive Bayes (NB), are used to predict presence or absence of drugs hidden in particular parcels sent through postal channels.

The main contributions of this work are as follows:

* Corresponding Author

- The data in this article are classified into two categories, which are case data and inspection data. The valid fields are extracted after text mining.
- Different models (namely, decision tree (DT), multilayer perceptron (MLP), support vector machine (SVM), and naive Bayes (NB)) based on various algorithms were established to judge whether presence of drugs hidden in particular parcels sent through postal channels.
- The probability related to the presence of hidden drugs is obtained, which is convenient for the police to assist in the intelligent discovery of drug smuggling through postal channels and help customs departments better identify and combat drug smuggling in this field.
- This article compared and analyzed the specific code implementation from the perspectives of algorithm accuracy, cross-validation accuracy, and code response time. SVM performs best in terms of accuracy, cross-validation accuracy, and relatively short response times. Therefore, it can be considered feasible choice for solving classification problems.

The remaining material is structured as follows. Section 2 introduces the relevant research on machine learning algorithms in recognition. Section 3 mainly preprocesses the data based on case data and inspection data of drug smuggling crimes through the postal channel. Section 4 studies the identification method of whether drugs are hidden in packages through four machine learning algorithms: support vector machine, naive Bayes, multilayer perceptron, and decision tree, and conducts cross-validation and comparative research. Section 5 mainly summarizes this article's research results and the proposed model's practical application.

2 State-of-the Art Solutions to This Problem

With the advancement of the national customs integration reform, traditional customs areas have also weakened, and the convenience of customs clearance has brought about more convenient illegal activities and crimes in import and export processes. Crime deduction and statistical calculation-based smuggling intelligence analysis can no longer fully meet the needs of practical operations, and machine learning is needed for intelligent intelligence analysis [1]. Currently, customs use supervised learning to study the discrimination and risk identification of smuggling cases, including false declaration identification based on k-means and logistic regression [2], customs risk classification based on fuzzy logic and backpropagation neural network [3, 4] and smuggling fishing vessel identification [5], density calculation based on k-nearest neighbors [6] and false declaration identification based on Bayesian and Markov chains [7], smuggling ship route prediction based on k-nearest neighbors [8], identification of smuggling goods and smuggling vessels based on classification trees and Bayesian network algorithms [9, 10], drug detection in milk powder cans based on support vector machines and backpropagation neural networks [11], encoding risk identification based on text classification and support vector machines, decision-making system based on Internet of Things and natural language processing [12], and analysis of smuggling crime organization structure based on natural language processing technology and clustering algorithm [13].

Unsupervised learning is used to analyze and discriminate the relationship between smuggling cases, such as using hash-based association algorithms to identify the traversal path of offshore money laundering [14], the association between smuggling goods and other commercial factors based on filtering *apriori* and *posteriori* [15], and false declaration based on *apriori* [16]. Research shows that supervised learning tools are more frequently used than unsupervised learning in customs smuggling research, and machine learning-based smuggling intelligence analysis is mainly focused on risk identification and discrimination of smuggling cases [17]. Certainly, there has been more recent research on unsupervised learning published in information science and/or computer science outlets [18-20].

Although, the aforesaid research proposed different methods and models for the discrimination and risk identification of smuggling cases but have following limitations:

- More researchers have worked only on policy and management of the smuggling cases but limited work is reported on the management and analysis of the smuggling discrimination based on the text analysis.
- Barring few researches on smuggled goods identification based on text data, the rest studies propose only the theoretical considerations with machine learning in the identification of things, and fail to carry out

the text extraction of the files, so as to truly sort out the research on the identification of smuggled drugs in the express delivery channels.

This study adopts the method of semi-supervised learning, using the features extracted from the case data of drug smuggling crimes through postal channels as labels, utilizing a small amount of labeled data and a large amount of unlabeled data for learning and prediction, and carrying out optimization and adjustment to improve accuracy and stability. Four algorithms (decision tree, random forest, support vector machine, and naive Bayes) are used to achieve accurate identification of whether drugs are hidden in parcels sent through postal channels.

3 Preprocessing of Case Data on Drug Smuggling through Postal Channels

3.1 Data Source

The data in this article are classified into two categories. The first one covers the case data from handling smuggling drug crime cases through postal channels by a certain customs anti-smuggling bureau. It includes 44 fields, such as the name of the drug, illegal methods, seizure methods, illegal dates, administrative divisions of seizure locations, specifications, quantities, and brief case details. The second category covers the data generated by certain customs in routine inspections, such as mail supervision and customs clearance operations, namely, inspection data. It comprises 15 fields, including declared item names, mail (shipping) numbers, item codes, source locations, and packaging specifications. Both types of data are semi-structured data. Although they have fixed fields and attributes that can be organized and managed to some extent, the structure is incomplete, and some fields have missing and duplicate information, which needs further processing. For protective reasons, all data used in this article have been anonymized.

3.2 Data Preprocessing

Case Data. By conducting preliminary data validation, it was decided to use the "serial number" field as the unique identifier for the data and delete the three fields of "Case number", "Case name", and "Property owner". This decision was made based on data security and confidentiality considerations, as well as the fact that the data in these three fields do not affect subsequent research. Through observation, it was found that the "brief case description" field has the most analytical value, as it contains information such as seizure time, waybill number, declared product name, packaging characteristics, drug name, drug specifications, and recipient's name. This information is of great significance for subsequent research. This article used the Python language to extract key information from the "brief case description" field and generate new fields.

Table 1. Presentation of fields in "case summary" (desensitized)

On November 14, 2016, G Customs Office in Post Office inspected an express delivery box (numbered E0000 and declared as foodstuff) at the express mail inspection site. Upon opening the box, eight packages of square-shaped items bundled with adhesive tapes to the exterior of the inner box, which were declared to contain foodstuff, were discovered. The square-shaped items were unpacked to reveal the branches and leaves of a dry plant. The preliminary assay suggested that the plant was marijuana, with a gross weight of 4600 g. The recipient of the above-mentioned express delivery box was XXX (XXX). On November 15, 2016, the G Customs Office in Post Office handed the case over to our bureau.

Using the Python language and necessary libraries such as jieba, basic processing methods such as removing stop words are used to clean the data shown in Table 1. After running the Python program, taking the fields shown in Table 1 as an example, we processed the data as shown in Table 2.

Table 2. Presentation of the processing results

Example	Extraction result	Label
On November 14, 2016	November 14, 2016	Date
G Customs Office in Post Office inspected an express delivery box (numbered E0000 and declared as foodstuff) at the express mail inspection site.	E0000	Mail number
	"FOODSTUFF"	Declared product name
Upon opening the box, eight packages of square-shaped items bundled with adhesive tapes to the exterior of the inner box, which was declared to contain foodstuff, were discovered.	Bundled with adhesive tapes, square-shaped	Packaging features
	Branches and leaves of a dry plant	Specification features
The square-shaped items were unpacked to reveal the branches and leaves of a dry plant.	Marijuana	Drug name
The preliminary assay suggested that the plant was marijuana, with a gross weight of 4600 g.	4600 g	Weight
The recipient of the above-mentioned express delivery box was XXX (XXX).	XXX (XXX)	Recipient

Check the Data. By verifying the inspection data in the same way, this part is standard structured data with a high level of standardization. However, there are still some duplicate values and missing values. After deleting the columns containing duplicate and missing values, the remaining data can still meet the testing requirements for later machine learning analysis.

4 Discovery and Identification of Drug Smuggling Crimes through Postal Channels

4.1 Data Preparation and Feature Extraction

This article combines case data with inspection data of the same category, retains the determined feature fields, adds a column "whether to conceal drugs" field, marks the case data as "yes", and marks the inspection data as "no". The "Seizure location" field in the case data and the "Inspection location" field in the inspection data are merged into "Location", and the "Seizure time" field in the case data and the "Inspection time" field in the inspection data are merged into "Time". They are randomly sorted, with a total of 2299 data entries. At the same time, the "Declared product name", "Drug name", "Time", "Location", "Source location", "Specification feature", and "Packaging feature" are determined as the feature labels for machine learning for further research.

4.2 Algorithm Selection

In general, different forms of learning in machine learning have different algorithms, and different algorithms have different characteristics. It is necessary to compare and select the most suitable algorithm for this study and the data in this article to ensure the accuracy of the final prediction results. This article combines supervised and semi supervised learning to improve the accuracy of prediction by labeling data, improve the generalization ability of the model, and reduce the workload of manually annotating data. This article has high data quality and low noise in unlabeled data. Choosing semi supervised learning can fully leverage its advantages and avoid overfitting problems caused by data quality and data noise. After annotating the dossier data (in Section 3), this article intends to use four supervised learning algorithms: support vector machine, naive Bayes, neural network, and decision tree for research.

Support Vector Machine. Support Vector Machine (SVM) is a supervised learning algorithm mainly used for classification and regression problems. SVM maps the data to a high-dimensional space and constructs a hyperplane in this space to maximize the classification boundary. SVM performs particularly well in small sample situations. "Performs well" here refers to the ability of the algorithm to maintain high classification accuracy even with a small number of samples. This is because SVM seeks the maximum margin hyperplane, meaning that among all possible classification hyperplanes, SVM selects the largest margin. This approach allows SVM to

classify a small number of samples effectively, thus performing well in small sample situations.

First, we import the necessary libraries. Next, we read the test set and training set saved in CSV UTF-8 (comma-separated) format, and extract the features of the fields "Declared product name", "Drug name", "Time", "Location", "Source Location", "Specification Features", and "Packaging Features". Concatenate multiple text fields in the training data together, calculate the TF-IDF value for each text field, and store the results in "train_features". After extraction, scale the range of feature values to between 0 and 1, and extract "Whether Carrying Drugs" as the label.

1. Import the data
2. Feature extraction
   Create a TfidfVectorizer object tfidf_vec, stop the word in English, and use L2 norm normalization. Concatenate each feature of the training data into a string, and carry out TF-IDF feature extraction on the string
3. Feature scaling
   Create a MinMaxScaler object, Scale the features of the training data and save the results in train_features.
4. Label extraction
   Extract the labels of training data and save them in train labels.

Then, we started adjusting the hyperparameters. In this article, we used GridSearchCV for hyperparameter tuning. We tried different combinations of C and gamma parameter values, setting C parameters as 1, 10, 100, and 1000 and gamma parameters as 0.01, 0.001, and 0.0001. We output the best parameters and their corresponding best scores. After setting the hyperparameters, you can make predictions on the test data and write the corresponding prediction results into a new csv file.

5. Adjust the hyperparameters
   Create a dictionary param_grid that contains the hyperparameter grid to search for. Initialize an SVM model and use grid search and cross validation to find the best combination of hyperparameters. Output optimal parameters and best score.
6. Create a new CSV file and open it, and set the write mode
7. Write each prediction result in a row of the CSV file

Naive Bayes. The naive Bayes (NB) algorithm is a classification algorithm based on Bayes' theorem and feature independence assumption. Given the target value, it assumes that the attributes are conditionally independent of each other. Its basic idea is to use the features and class information in the training set to learn a classification model and then use this model to classify new unknown samples, thus predicting a certain outcome. Specifically, the NB algorithm simplifies the Bayes algorithm:

$$P(C/X) = P(X/C)P(C)/P(X) \tag{1}$$

where C is the category, X is the feature, $P(C|X)$ is the probability that a sample belongs to category C given feature X, $P(X|C)$ is the probability that feature X appears given category C, $P(C)$ is the probability of category C appearing, and $P(X)$ is the probability of feature X appearing.

Since $P(X)$ is the same for all categories, this term can be omitted. The naive Bayes algorithm assumes that all features are independent of each other given the category; that is, $P(X|C)$ can be represented as the product of individual feature conditional probabilities:

$$P(X|C) = P(X_1 \mid C)P(X_2 \mid C)...P(X_n \mid C) \tag{2}$$

Among them, X_1, X_2, …, and X_n represent the n features of the sample.

According to the above formula, the NB algorithm can determine the category of a sample by calculating

$P(C|X)$; that is, for an unknown sample, the naive Bayes algorithm calculates the conditional probability of it belonging to each category and then selects the category with the highest probability as its classification result. The specific steps are as follows:

Assuming there is no correlation between each feature, given a training dataset where each sample x includes n-dimensional features, i.e., $x = (x_1, x_2, \ldots, x_n)$, the set of class labels contains k categories, i.e., $y = (y_1, y_2, \ldots, y_k)$.

For a given new sample x, to determine which category it belongs to, according to Bayes' theorem, we can obtain the probability $P(y_k|x)$ that x belongs to category y_k.

$$P(y_k \mid x) = \frac{P(x \mid y) \times P(y_k)}{\sum_k P(x \mid y_k) \times P(y_k)} \tag{3}$$

The class with the highest posterior probability is referred to as the predicted class, that is: $argmax_{y_k} P(y_k|x)$

The NB algorithm assumes independence on the conditional probability distribution. In simple terms, it assumes that each dimension's features $x_1, x_2, \ldots, x_n$ are independent. Based on this assumption, the conditional probability can be transformed into:

$$P(x|y_k) = P(x_1, x_2, \ldots, x_n|y_k) = \prod_{i=1}^{n} P(x_i \mid y_k) \tag{4}$$

Substituting into the above Bayesian formula, we obtain:

$$P(y_k \mid x) = \frac{P(y_k) \times \prod_{i=1}^{n} P(x_i \mid y_k)}{\sum_k P(y_k) \times \prod_{i=1}^{n} P(x_i \mid y_k)} \tag{5}$$

Therefore, the NB classifier can be represented as:

$$f(x) = argmax_{y_k} P(y_k) = argmax_{y_k} \frac{P(y_k) \times \prod_{i=1}^{n} P(x_i \mid y_k)}{\sum_k P(y_k) \times \prod_{i=1}^{n} P(x_i \mid y_k)} \tag{6}$$

Because, for all yk, the value of the denominator in the equation is the same, the denominator part can be ignored. The NB classifier is finally represented as follows:

$$f(x) = argmax_{y_k} P(y_k) \times \prod_{i=1}^{n} P(x_i \mid y_k) \tag{7}$$

Based on the above principle, we use the Python language to implement the NB algorithm, and the code is shown below:

First, we import the necessary libraries. Next, we read the dataset and extracted seven fields, including "Declared product name", "Drug name", "Time", "Location", "Source location", "Specification characteristics", and "Packaging characteristics", as features. Convert categorical variables into numerical features, and the target variable is "Whether it contains drugs".

1. Read the dataset
   Use pandas' read_csv function to read the csv file where the data is stored.
2. Feature extraction and preprocessing
   Select the corresponding column from the data dataset and store it in variable X. The get_dummies function of pandas is used for unique

thermal encoding of X. Select the column 'Drugs or not' from the data dataset and store it in a variable named Y.

Afterward, the dataset is divided into training and test sets. The training set includes the features "X_train" and the target variable "Y_train", while the test set includes the features "X_test" and the target variable "Y_test". The data are split with a ratio of 80% for training and 20% for testing, and the "random_state" parameter is set to a random seed for reproducibility. Once this is done, a GaussianNB object is created as the NB classifier model, and the model is trained using the training set data "X_train" and "Y_train". Finally, the model is tested using the test set data "X_test" to predict the trained model, and the accuracy between the predicted and actual results is calculated. The accuracy is printed and outputted.

3. Data partitioning
   The train_test_split function was used to divide the data sets X and Y into training sets and test sets, and the partitioned data sets were stored in variables X_train, X_test, Y_train, Y_test respectively.
4. Model training
   Create a Gaussian naive Bayes classifier model and store it in the variable model.Use the training set data to train the model, that is, call the fit function of the model, and pass the training set feature X_train and the training set label Y_train as parameters.
5. Model testing
   Use the trained model to predict the feature X_test of the test set, and store the prediction results in variable Y_pred. Accuracy_score function is used to calculate the accuracy of the model on the test set, and the calculated results are stored in the variable accuracy. Output the accuracy.

Multilayer Perceptron. The multilayer perceptron (MLP) is a classic artificial neural network model that imitates the structure and function of biological neural networks. It achieves complex mapping of input data through multiple layers of nonlinear transformations. The MLP consists of an input layer, hidden layers, and an output layer, with the option of having multiple hidden layers. Each layer comprises multiple neurons, each receiving input signals from the neurons in the previous layer. After undergoing weighted and nonlinear transformations, the output signals are passed on to the neurons in the next layer. Ultimately, the output layer neurons' output results can be considered the classification or regression results of the entire neural network for the input data, as shown in Fig. 1.

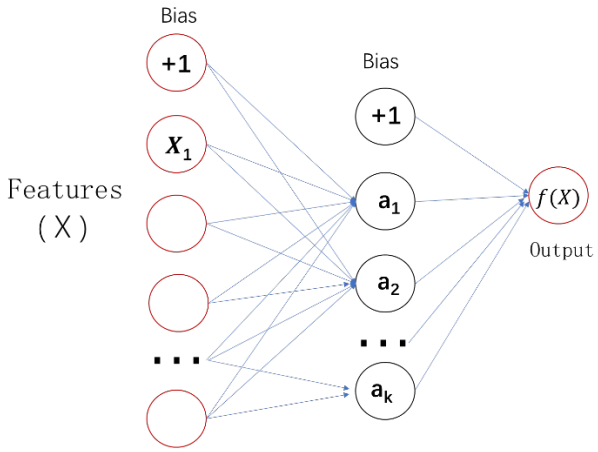

Fig. 1. Multilayer perceptron schematic diagram

Specifically, the algorithm first initializes the weights and biases of the neural network randomly. Then, it calculates the error gradient of each neuron based on the difference between the training data's true output and the neural network's predicted output. It then adjusts the weights and biases of the neurons accordingly. This process is repeated iteratively until the predicted results of the network have a sufficiently small error compared to the true output of the training data or until a certain number of iterations is reached. The MLP algorithm achieves complex mappings of the input data by using multiple layers of nonlinear transformations. It then trains the network parameters using the backpropagation algorithm, allowing the network to classify or regress the training data.

Based on the above principle, we use the Python language to implement the multilayer perceptron (MLP) algorithm, and the code is shown below:

First, we import the necessary libraries. After reading the dataset, features and target variables are extracted from the test data. Define *X* as the features and *Y* as the target variable. Similarly, we still use the seven fields "Declared product name", "Drug name", "Time", "Location", "Source location", "Specification features", and "Packaging features" as features and "Whether carrying drugs" as the target variable. Use "pd.get_dummies" for one-hot encoding.

```
1. Read the dataset
2. Feature extraction and preprocessing
   Select the corresponding column from the data dataset and store it in
   a variable named X. The get_dummies function of pandas is used for
   unique thermal encoding of X. Select the column 'Drugs or not' from
   the data dataset and store it in a variable named Y.
```

Afterward, the dataset is split into training and test sets. Similarly, the training set includes the features "X_train" and the target variable "Y_train", while the test set includes the features "X_test" and the target variable "Y_test". The data are divided into a ratio of 80% for training and 20% for testing. Here, it is necessary to create a multilayer perceptron neural network classifier object "self" and define the neural network's hidden layer structure as two layers with 100 and 50 neurons, respectively. Then, the model is trained using the training set data "X_train" and "Y_train". Finally, the model is tested using the test set data "X_test" to predict the trained model, and the accuracy between the predicted and actual results is calculated. The accuracy is printed and outputted.

```
3. Data partitioning
   The train_test_split function was used to divide the data sets X and Y
   into training and test sets, and the partitioned data sets were stored
   in variables X_train, X_test, Y_train, Y_test, respectively.
4. Construct the neural network classifier
   Create a multi-layer perceptron classifier model and store it in the
   variable clf.
5. Model training
   The model is trained using the training set data, and the training set
   feature X_train and the training set label Y_train are passed in as
   parameters.
6. Model testing
   Use the trained model to predict the feature X_test of the test set,
   and store the prediction results in variable Y_pred. Accuracy_score
   function is used to calculate the accuracy of the model on the test
   set, and the calculated results are stored in the variable accuracy.
   Output the accuracy.
```

Decision Tree (DT). In the process of constructing a tree, we need to select the optimal attribute for splitting. Commonly used attribute selection criteria include information gain, gain ratio, and Gini index. Among them, information gain is the most commonly used criterion, based on the concept of information entropy and used to measure attributes' contribution to data classification.

$$Information\ gain = Entropy\ of\ the\ parent\ node - Conditional\ entropy$$

The information entropy of the parent node refers to the entropy of the probability distribution of different categories in the parent node. Its calculation formula is:

$$H(P) = -\sum(P_i \times \log_2 P_i) \tag{8}$$

Conditional entropy refers to the weighted average of the information entropy of the sub-nodes corresponding to different values under a certain feature. Its calculation formula is given below:

$$H(Y|X) = \sum(|X = i| / |D|) \times H(Y \mid X = i) \tag{9}$$

where $|X=i|$ is the number of samples with feature X taking the value i, $|D|$ is the total number of samples, and $H(Y|X=i)$ is the information entropy of Y when feature X takes the value i.

In addition to information gain, the gain ratio can also be used to select the optimal feature. The gain ratio is a modification of information gain, which avoids preference for attributes with more attribute values. Its calculation formula is:

$$Gain/ratio = Information\ gain \quad Feature\ entropy \tag{10}$$

Among them, feature entropy refers to the entropy of the probability distribution corresponding to different values of a certain feature. Its calculation formula is:

$$H(X) = -\sum(|X = i| / |D||) \times \log_2(|X = i| / |D|) \tag{11}$$

where $|X=i|$ is the number of samples, with feature X taking the value i, and $|D|$ is the total number of samples.

The Gini index is used to measure the uncertainty of a dataset, indicating the impurity of the dataset. The smaller the Gini index is, the more stable the data. Assuming there are K classes, the probability of a sample point belonging to the Kth class is P_k, and its calculation formula is as follows:

$$Gini(p) = \sum_{k=1}^{K} P_k(1 - P_k) = 1 - \sum_{k=1}^{K} P_k^2 \tag{12}$$

In the process of pruning, we prune the well-constructed decision tree to avoid overfitting problems. Common pruning methods include pre-pruning and post-pruning. Pre-pruning involves setting thresholds to limit the depth of the tree or the number of nodes during the tree construction process to avoid overfitting. Post-pruning involves determining whether pruning a subtree can improve the tree's generalization ability after the tree has been constructed.

Based on the above principles, we use the Python language to implement the decision tree (DT) algorithm, and the code is shown below:

First, we import the necessary libraries. After reading the dataset, because decision tree algorithms can usually only handle numerical data, the "LabelEncoder" is used to encode the string type features in DataFrame. The seven-string type features "Declared product name", "Drug name", "Time", "Location", "Source location", "Specification feature", and "Packaging feature" are converted into integer form.

```
1. Read the dataset
2. Use LabelEncoder for encoding features of string type into integers
   Create a LabelEncoder object le, adjust the data encoding in the df to
   a proper type.
```

Select features converted into integer form, and determine the target variable as "Whether carrying drugs". Divide 80% of the dataset as the training set and 20% as the test set. Assign the features to X and the target variable to Y. Build a decision tree model, train the model using the training set data "X_train" and "Y_train", use "X_test" for prediction, and output the accuracy.

3. Feature selection
   Select the feature columns in the data set and store them in the variable feature_cols.
4. Separate feature data from target data
   Select feature columns from the data set and store them in variable X, select the target column 'Drugs or not' from the data set and store it in variable Y.
5. Split the dataset into a training set and a test set
   The data sets X and Y are split into training and test sets using the train_test_split function, and the split data sets are stored in variables X_train, X_test, Y_train, Y_test, respectively.
6. Build the decision tree model
   Create a DecisionTreeClassifier object and store it in the variable dtree.
7. Train the model
   The training set data X_train and Y_train are used to train the decision tree model, and the training set feature X_train and the training set label Y_train are passed in as parameters.
8. Predict the result
   The trained decision tree model is used to predict the test set feature X_test, and the prediction results are stored in the variable Y_pred.
9. Assess the accuracy of the model
   Print out the accuracy of the model on the test set.

Model Results. The above is the specific code implementation process for using SVM, NB, MLP, and DT algorithms to predict the presence or absence of hidden drugs in packages in the inbound and outbound delivery channels. After adjusting the model parameters, only the best options are presented in this article. According to the running results of the code, the preliminary accuracy of the four algorithms reached over 75%, as shown in Table 3.

Table 3. Comparison of the results of the code run

Algorithm	Accuracy, %
Support Vector Machine (SVM)	88.477
Naive Bayes (NB)	88.409
Multilayer Perceptron (MLP)	79.545
Decision Tree (DT)	84.848

Cross-validation. Only the accuracy data alone cannot fully explain the effectiveness of the algorithm's prediction. To further evaluate the generalization ability of the model established in this article and avoid the phenomenon of overfitting, where the model performs poorly on new data, the method of cross-validation is introduced in this section to further evaluate and validate the model.

For the SVM algorithm, we introduced the ten-fold cross-validation method as a cross-validation method. As a common cross-validation method, ten-fold cross-validation randomly divides the dataset into ten equally sized subsets. One of these ten subsets is randomly selected as the test set, while the remaining nine subsets are used as the training set to train and test the model. This validation method can fully utilize the dataset and provide more stable and reliable evaluation results for the model, reducing the accidental results introduced by specific dataset divisions and better evaluating the performance and generalization ability of the model.

The running result shows that the tenfold cross-validation of the SVM algorithm reached 89.623%. The support vector machine algorithm performs well in identifying drug smuggling in postal channels and has good generalization ability. It can also perform well in predicting new data in the future.

Similarly, we conducted cross-validation on NB, MLP, and DT algorithms using the same validation method. By performing tenfold cross-validation on the SVM, NB, MLP, and DT, the results obtained were 89.623, 86.335, 77.357, and 84.388%, respectively. In this study, the data were divided into ten equal subsets, and the

model was trained nine times, each training using nine subsets and one subset used for validation. The accuracy of the predictions in the validation phase already exceeded 75%.

Table 4. Accuracy of ten-fold cross-validation results obtained by the four algorithms

Algorithm	Ten-fold cross-validation accuracy, %
Support Vector Machine (SVM)	89.623
Naive Bayes (NB)	86.335
Multilayer Perceptron (MLP)	77.357
Decision Tree (DT)	84.388

The results shown in Table 4 indicate that the selection of the four algorithms in this article is more suitable for identifying drug-smuggling crimes in postal channels. On the one hand, it fits the data type and has a high degree of model generalization. On the other hand, it has high accuracy and can better handle the identification of drug-smuggling crimes in postal channels.

Comparative Study. This article compared and analyzed the specific code implementation from the perspectives of algorithm accuracy, cross-validation accuracy, and code response time. Accuracy is a commonly used indicator to evaluate the performance of classification algorithms, representing the proportion of correctly predicted samples by the classifier. A higher value indicates that the classifier predicts more samples correctly. Cross-validation is a technique for evaluating the generalization ability of machine learning models, which can better assess the model's performance on unknown data. Response time is the time required for algorithm execution, usually measured in seconds. A lower response time is generally considered a better choice because it indicates a faster algorithm execution speed.

Table 5. Comparison of operating parameters

Algorithm	Accuracy, %	Accuracy of cross-validation, %	Response time, s
Support Vector Machine (SVM)	0.884773	0.896232	0.281253
Naive Bayes (NB)	0.884090	0.863352	0.204682
Multilayer Perceptron (MLP)	0.795454	0.773572	3.234375
Decision Tree (DT)	0.848485	0.843879	0.359735

According to Table 5, SVM and NB algorithms perform well in terms of accuracy and cross-validation accuracy, and they also have relatively short response times. Therefore, they can be considered feasible choices for solving classification problems. The accuracy and cross-validation accuracy of the MLP algorithm are relatively low, and the response time is longer, so further optimization or consideration of other algorithms may be needed. The performance of the DT algorithm is between the previous two, and the specific application scenario can be chosen accordingly. However, the algorithm's performance not only depends on the algorithm itself but is also influenced by various aspects, such as data quality, data quantity, feature selection and extraction, parameter settings, and computer processing power. For the application scenario proposed in this paper, further evaluation is needed in the future when it is put into actual use.

5 Summary

This study used support vector machine (SVM), naive Bayes (NB), multilayer perceptron (MLP), and decision tree (DT) machine learning algorithms to identify drugs hidden in postal channel packages to assist in the intelligent discovery of drug smuggling through postal channels and help customs departments better identify and combat drug smuggling in this field. Through cross-validation and comparative analysis, the performance and accuracy of each algorithm were evaluated, providing a reference for identifying drug smuggling through postal channels. SVM and NB algorithms had the best accuracy and cross-validation accuracy, with relatively short response times. Therefore, they were found the most lucrative for solving the above tasks. The accuracy and

cross-validation accuracy of the MLP algorithm were the worst, and the response time was the longest. The DT algorithm performed better than MLP but worse than SVM and NB. However, the algorithm's performance not only depends on the algorithm itself but is also influenced by various aspects, such as data quality, data quantity, feature selection and extraction, parameter settings, and computer processing power. For the application scenario proposed in this paper, further evaluation is needed.

Acknowledgement

This study is partially supported by Project on No. LGZD202404 the Fundamental Research Funds for the Central Universities.

References

[1] Z.-Y. Pan, Design and implementation of crime early warning system based on machine learning algorithm, [Master's Dissertation] Chengdu: University of Electronic Science and Technology of China, 2019.
[2] Z. Hua, S. Li, Z. Tao, A rule-based risk decision-making approach and its application in China's customs inspection decision, Journal of the Operational Research Society 57(11)(2006) 1313-1322.
[3] J.-H. Duan, Risk identification and evaluation of customs management based on fuzzy neural network algorithm, Applied Mechanics and Materials 291-294(2013) 2924-2927
[4] Y.-Q. Jia, The risk identification and its control strategies of custom in the context of non-traditional security, [Master's Dissertation] Shanghai: Fudan University, 2011.
[5] C.-H. Wen, P.-Y. Hsu, C.-Y. Wang, T.-L. Wu, Identifying smuggling vessels with artificial neural network and logistics regression in criminal intelligence using vessels smuggling case data, in: Proc. 2012 Asian Conference on Intelligent Information and Database Systems, 2012.
[6] H.A. Rad, S. Arash, F. Rahbar, R. Rahmani, Z. Heshmati, M.M. Fard, A novel unsupervised classification method for customs fraud detection, Indian Journal of Science and Technology 8(35)(2015) 1-7.
[7] R. Triepels, A. Feelders, H. Daniels, Uncovering document fraud in maritime freight transport based on probabilistic classification, in: Proc. 2015 International Conference on Computer Information Systems and Industrial Management, 2015.
[8] A.L. Duca, C. Bacciu, A. Marchetti, A K-nearest neighbor classifier for ship route prediction, in: Proc. 2017 Oceans, 2017.
[9] R. Triepels, H. Daniels, A. Feelders, Data-driven fraud detection in international shipping, Expert Systems with Applications 99(2018) 193-202.
[10] C.-H. Wen, P.-Y. Hsu, M.-S. Cheng, Applying intelligent methods in detecting maritime smuggling, Maritime Economics and Logistics 19(3)(2017) 573-599.
[11] Y.-P. Zhu, L. Wang, W. Zhang, Detection of contraband in milk powder cans by using stacked auto-encoders combination with support vector machine, in: Proc. 2018 IOP Conference Series: Earth and Environmental Science, 2018.
[12] N. Deng, X. Chen, C.-Q. Xiong, Design and construction of intelligent decision-making system for marine protection and law enforcement, in: Proc. 2019 International Conference on Broadband and Wireless Computing, Communication and Applications, 2019.
[13] F. Iqbal, B.C.M. Fung, M. Debbabi, R. Batool, A. Marrington, Wordnet-based criminal networks mining for cybercrime investig[ation, IEEE Access 7(2019) 22740-22755.
[14] C. Suresh, K.T. Reddy, N. Sweta, A hybrid approach for detecting suspicious accounts in money laundering using data mining techniques, International Journal of Information Technology and Computer Science 8(5)(2016) 37-43.
[15] M.M. Gebru, Association pattern discovery of import export items in Ethiopia, American Scientific Research Journal for Engineering, Technology, and Sciences 44(1)(2018) 240-256.
[16] B.B. Zehero, E. Soro, Y. Gondo, P. Brou, O. Asseu, Elicitation of association rules from information on customs offences on the basis of frequent motives, Engineering 10(9)(2018) 588-605.
[17] K. Tachtsidou, E. Kirkos, Risk analysis and decision support systems with data mining techniques in customs administrators, in: Proc. 2019 International Conference on Quantitative, Social, Biomedical and Economic Issues, 2019.
[18] O. Syrotkina, M. Aleksieiev, B. Moroz, S. Matsiuk, O. Shevtsova, A. Kozlovskyi, Mathematical Methods for optimizing Big Data Processing, in: Proc. 2020 10th International Conference on Advanced Computer Information Technologies (ACIT), 2020.
[19] Z.-H. Fang, S.-D. Du, X.-C. Lin, J.-B. Yang, S.-P. Wang, Y.-Q. Shi, DBO-Net: Differentiable bi-level optimization network for multi-view clustering, Information Sciences 626(2023) 572-585.
[20] K. Nirmaladevi, K. Prabha, A selfish node trust aware with Optimized Clustering for reliable routing protocol in Manet, Measurement: Sensors 26(2023) 1-10.