CIT. Journal of Computing and Information Technology, Vol. 30, No. 1, March 2022, 35–50
doi: 10.20532/cit.2022.1005459

Anomaly Identification Model for Telecom Users Based on Machine Learning Model Fusion

Jianhong Lin[1, 2], Peng Wang[1] and Chunming Wu[1]

[1]College of Computer Science and Technology, Zhejiang University, Hangzhou, China
[2]Zhejiang Ponshine Information Technology Co., Ltd., Hangzhou, China

With the development of economic globalization and modern information and communication technology, the situation of communication fraud is becoming more and more serious. How to identify fraudulent calls accurately and effectively has become an urgent task in current telecommunications operations. Affected by the sample set and the current state of the art, the current machine learning methods used to identify the imbalanced distribution dataset of positive and negative samples have low recognition accuracy. Therefore, in this paper, we propose a new hybrid model solution that uses feature construction, feature selection and imbalanced classes handling. A stacking model fusion algorithm composed of a two-layer stacking framework with several state-of-the-art machine learning classifiers is adopted. The results show that the risk user identification model based on mobile network communication behavior established by our stacking model fusion algorithm can accurately predict the category labels of telecom users and improve the risk of telecom users. The generalization performance of the identification is high, which provides a certain reference for the telecommunications industry to identify risk users based on mobile network communication behaviors.

ACM CCS (2012) Classification: Security and privacy → Intrusion/anomaly detection and malware mitigation → Social engineering attacks → Spoofing attacks

Keywords: telecom fraud detection, model fusion, user behavior analysis, abnormal behavior prediction

1. Introduction

Two problems are often encountered in analyzing high-dimensional data, one of which is the Euclidean distance problem. Euclidean distance is the most widely used distance in machine learning, which can be used to measure the closeness and similarity between sample points, and is applicable to the low-dimensional space of 2~10 dimensions. In high-dimensional spaces, data sparsity can lead to the failure of a large number of traditional statistical methods, which is due to the distance between sample points tending to be approximately equal as the number of spatial dimensions increases, and thus the comparability of the distance measure loses its usefulness. The other is the dimensional inflation problem, commonly known as the "dimensional disaster", which is the biggest problem in the process of high-dimensional data analysis. As the number of dimensions increases, the amount of data computation increases rapidly, and the number of samples required grows exponentially, rapidly increasing complexity and cost of analyzing and processing multidimensional data. Therefore, dimensionality reduction of high-dimensional data is a key step in the process of high-dimensional data analysis, which can not only solve the dilemma that many traditional statistical methods cannot be applied to high-dimensional data, but also avoid the analysis difficulties brought by the characteristics of high-dimensional data itself.

There are two general methods for data dimensionality reduction: one is feature transformation, also known as feature extraction, which is the projection of high-dimensional data into a low-dimensional data space, *i.e.*, the original features of the data are transformed in a certain way to obtain new unrelated integrated features. The other is feature selection, also known as attribute selection [1], which is the use of certain evaluation criteria to screen the data features without changing the nature of the feature space and finally determine an effective feature subset with strong judgment. Although data dimensionality reduction can reduce the time complexity of data processing and facilitate the discovery of structural information of data, it also implies the loss of information. Considering that the actual data itself is often redundant, it is possible to extract features of interest or provide effective information for the research problem from a higher dimensional space through the process of dimensionality reduction, while retaining the maximum effective information of the data (minimizing information loss).

Commonly used algorithms for risk identification include logistic regression, random forest, and support vector machine. Although logistic regression has the advantages of a simple model and high accuracy, it may underfit when dealing with a dataset with a large number of features, like telecommunication subscribers, and is sensitive to multicollinearity of features [2]. For a typical unbalanced dataset like telecom subscriber data, random forest can balance the error, but the algorithm is not well interpreted. Although the decision performance of each decision tree in random forest is different, the decision weights given to each tree are the same, which can weaken the accuracy of the model to some extent [2]. The use of a single algorithm is prone to poor model identification performance due to the randomness of the samples; however, model fusion works well for problems such as user credit identification, risk prediction, and power equipment fault diagnosis [3-6]. Therefore, in this paper, we applied the model fusion approach to risky user identification based on mobile network communication behavior and propose a Stacking model fusion mobile network risky user identification model based on feature selection and hybrid sampling to fully combine the advantages of various algorithms and complement each other's strengths to achieve an overall model performance improvement using model fusion.

In the research of communication behavior records, we first extract communication behavior features, and then perform feature normalization, feature selection and hybrid sampling to address the problems of high-dimensional and complicated user behavior features and skewed samples: a mutual information-based automatic selection of relevant features is used to remove irrelevant features, and then the Support Vector Machine-Recursive Feature Elimination (SVM-RFE) feature selection algorithm is combined to remove redundant features and reasonably select the optimal feature subset that is more suitable for classification detection. Next, the data set is mixed-sampled to achieve a balanced distribution of data samples, and finally, a Stacking strategy is used to fuse multiple base classifiers to identify risky users based on mobile network communication behavior, which achieves accurate data classification.

2. Data Introduction and Feature Construction

This section introduces the source and content of the data and explains the original features. Since the dataset is the original detailed list of telecom users' communication behaviors such as calls, Short Message Service (SMS), website and app access, which cannot be directly input into the classification model for analysis, feature construction is needed first to realize the extraction of telecom users' behavioral features, and a regular high-dimensional imbalanced dataset is constructed, followed by feature normalization. Feature normalization is performed after constructing a regular high-dimensional imbalanced dataset.

2.1. Data Source and Introduction

The dataset used in this article comes from the DATA algorithm competition in the author's company. The content of the competition is the data of "Identification of Suspected Telecom Fraud Users Based on Mobile Network Communication Behavior", which is provided by China Unicom Big Data Co., Ltd. The data set includes

45 desensitization data of 5,609 users' daily voice call records, SMS sending and receiving records, website and app access records and other mobile network usage behaviors during 2 consecutive natural days. The purpose of this paper is to identify risk users based on the mobile communication network usage behaviors of these users. The original data is user behavior, which contains a lot of useful information, and feature construction is needed to effectively mine the unique attributes hidden in the analysis data. The content of the data set is shown in Table 1. There are a total of 5609 users, of which 4500 are normal users, but there are only 1109 risk users. From Figure 1, we can see the approximate ratio of the two, and the risk users only account for a small part. Figure 1 visually demonstrates the highly imbalanced problem of risky user identification datasets. Although the sample size of this data set is not very large, the behavior data of these users based on mobile communication network calls, SMS, website and APP access are far larger than the sample size, and some of them are included in the user website/APP access record data table. The existence of null values or NULLs in a column fully reflects the complexity and irregularity of data information in real life. A null value or NULL means that the user did not have corresponding calls, text messages or Internet access during this period, which is of no practical significance to the research problem, so these 801 missing records were deleted.

Table 1. Raw data information.

Data Sheet	Number of records	Number of invalid records
User Risk Label	5609	0
User call log data	1150778	0
User SMS log data	302976	0
User website/APP access log data	4808343	801

Next, we introduce the original features of the risky user identification dataset, as shown in Table 2. Since the data of the risky user identification dataset comes from the original detailed list based on mobile network communication behavior, it is not suitable for the classification task of identifying risky users directly. Specifically, it requires feature construction of the detailed behavior data, which is combined with expertise in the communication industry to form the structured matrix required for general machine learning algorithm input before being fed into the model for classification and prediction.

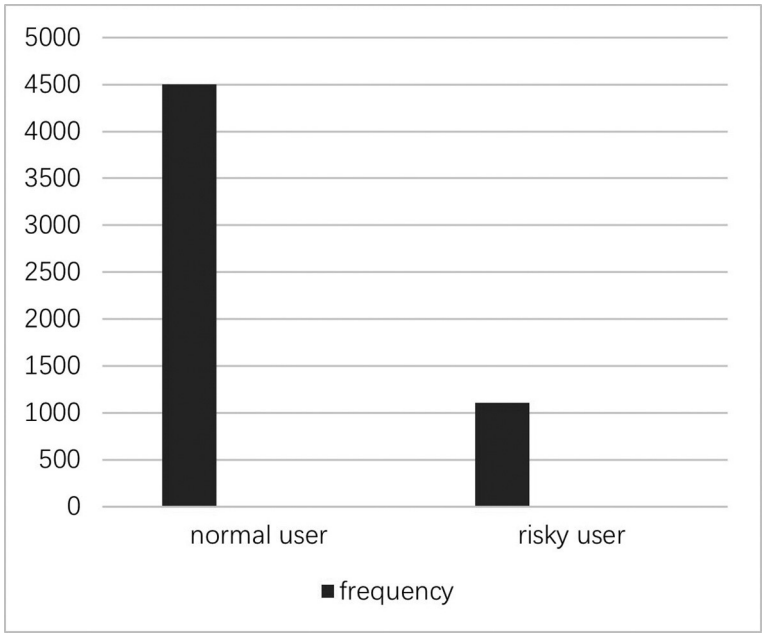

Figure 1. Bar chart of user label distribution.

Table 2. Original features and description of the risky user identification dataset.

Field Name Description	
Uid	**On behalf of a cell phone user**
Opp_num	In the user call record table, it represents the call number of the other side, and in the user SMS record table, it represents the SMS number of the other side: encryption was performed, and each number is kept unique and consistent after encryption.
Opp_head	The first n digits of the opposite number: the first three digits are taken if the number length is >5; the first 1 digit is taken if the number length is <=5
Opp_len	Length digits of the opposite end number
Start_time	In the user call record table represents the call initiation time, in the user SMS record table represents the SMS sending time, the format is DD-HH-MM-SS, such as 01081045 represents the 01 day and time 08:10:45
End_time	Call end time, formatted as above
Call_type	Call type: 1-Local, 2-Provincial long distance, 3-Provincial long distance, 4-Hong Kong, Macao and Taiwan long distance, 5-International long distance
In_out	In the user's call record table, it represents the caller/receiver type of the call: 0 is the uid initiated caller, 1 is the uid called; in the user's SMS record table, it represents the send/receive type of the SMS: 0 is the uid sender, 1 is the uid receiver
Wa_name	Name of the website or APP visited (each record corresponds to one website or APP)
Visit_cnt	Number of visits to the website or app on the same day
Visit_dura	Total time spent visiting the site or app on that day, in seconds
Up_flow	Total uplink traffic visiting the site or app on that day, in B
Down_flow	Total downlink traffic visiting the site or app on that day, in B
Wa_type	Website or APP distinguishing labels: 0 - Website, 1 - APP
Date	Date in DD format, *e.g.*, 01 for the 01st day
Label	Risky user label: 0-normal user, 1-risky user

2.2. Feature Construction

Features, also known as indicators, independent variables, explanatory variables, *etc.*, are used to describe the characteristics of a sample, which are fed back through features, while the category of a sample is usually represented by a label, also known as the dependent variable or response variable in regression [7].

In order to extract more useful information, some original new data based features with practical significance need to be constructed manually according to the needs of the research problem. Meanwhile, a standardized sample data set can be formed which is convenient for the training of machine learning algorithms. The constructed features not only reflect the original data information effectively but also have higher relevance to the research problem, which is more helpful for the prediction results, thus further improving the performance of the algorithm. In this paper, mobile network communication behavior is analyzed in detail. Specifically, a large number of communication user features are constructed by feature derivation, feature combination and feature discretization processing on the detailed mobile network communication behavior data combined with expertise in the communication industry.

2.2.1. Feature Derivation

Simple numerical operations, such as addition, subtraction, multiplication and division, are performed using the original data features to form new features, which are generally derived in conjunction with actual business needs or experience. In this paper, we subtract the two fields end_time and start_time from the call record table to obtain the talk_time of each call record, and we can also use the last_end_time of the last call and the start_time of the next call to calculate the time interval between two calls last_gap. The date of the call can be extracted from the end_time of each call record voice_date; similarly, the date of receiving or sending SMS sms_date can be extracted from the start_time field in the SMS record table, and the interval between two SMS messages can be calculated using the last_start_time of the last SMS message and the start_time of the next SMS message. start_time is used to calculate the last_startgap_time between two SMS messages; website/APP access record table by dividing the total access time visit_dura by the number of visits visit_cnt to get the average access time visit_per_dura, add up_flow and down_flow to get the total traffic flow. The total traffic (flow_amount) is obtained by adding upstream traffic (up_flow) and downstream traffic (down_flow), where up_flow is divided by the total access time (visit_dura) to calculate the upstream speed (upload_speed), downstream speed download_speed and total speed amount_speed can be obtained in the same way.

2.2.2. Combination of Features

For the categorical variable opp_number, opp_head, there are 550 categories in the call behavior record and 72 categories in the SMS behavior record, which we extend to 622 feature variables, indicating whether the user has used the number for communication behavior, and we can also get the number of communications using the number; the length of the opp_number opp_len takes only 18 values, and we can also get 36 features about call and SMS user behavior. The same can get 36 user behavior features about calls and SMS; and there are hundreds of thousands of different opp_num, in order to dig the information in this paper, the number of calls, SMS top 1000 opp_num as variables for communication behavior frequency statistics, the number of visits, the length of visits top 1000 website or APP name (Wa_name),the same operation is performed.

2.3. Feature Normalization

For example, the time-related features in the risky user dataset in this paper are measured in seconds, and the traffic unit used is bit. At the same time, their value ranges are also very different, which will amplify the effect of features with large value ranges and have higher "weights" when brought into the model directly. This will amplify the effect of features with a large value range and a high "weight", thus ignoring the effect of features with a small value range, and ultimately reducing the prediction effect of the model. In order to eliminate the influence of dimensionality between features, it is necessary to scale the data of constructed

continuous-type features, so that each feature is unified into roughly the same interval range, making the data comparable and ensuring that each feature has a consistent weight on the influence of the objective function. At present, there are two common methods of data scaling: standardization and normalization. In this paper, we adopt normalization, *i.e.*, Min-Max normalization, to make the data fall in the [0,1] interval by linear transformation, to achieve the isometric scaling of the original data and ensure the consistency of the feature value domain, as shown in Equation (1).

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

x is the original feature fetch, x' is the normalized feature fetch, x_{max}, x_{min} are the maximum and minimum values of the feature fetch, respectively.

2.4. Feature Selection

Feature selection and data balancing are common pre-processing tools to deal with high-dimensional unbalanced data. In this section, the extracted user communication behavior features are pre-processed with the high-dimensional problem before the unbalanced problem to obtain a more reliable dataset, which is convenient for subsequent training to obtain a more stable and reliable classification model and produce better classification performance.

After feature construction, 5609 × 5153 ultra-high-dimensional sample data is obtained, which is accompanied by a large number of irrelevant and redundant features. Related research [8] showed the number of samples required by most data mining classification algorithms increases exponentially with the increase of irrelevant features. Moreover, the classification ability decreases with the increase in redundant features. High data dimensionality affects the computational speed of the model and increases the training time overhead, which is not conducive to identifying important features that cannot be correctly expressed in the model and even cause the model to fail to converge [7]. Therefore, it is necessary to pre-process the high-dimensional data by data dimensionality reduction to improve the training efficiency and classification performance of the model. Feature selection, as the main technique of data dimensionality reduction, reduces model complexity and improves the performance of data mining models by selecting important features from the original feature set space based on certain criteria, removing irrelevant and redundant features, and reducing the risk of overfitting.

2.4.1. Automatic Selection of Relevant Features

In this paper, we first use mutual information as an evaluation criterion to measure the correlation between features and class labels, the stronger the correlation, the more information the feature contains and the higher the feature evaluation given. The mutual information is defined as follows.

Let X and Y be two discrete (continuous) random variables, $p(x)$ and $p(y)$ be the respective marginal probability mass (density) functions, and $p(x, y)$ be the joint probability mass (density) function of the two, then the mutual information (MI) of the random variables with can be defined as

$$MI(x,y) = \sum_x \sum_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{2}$$

or

$$MI(x,y) = \int\int p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dxdy. \tag{3}$$

Mutual information is a measure of the interdependence between variables and can measure any relationship between variables, and mutual information is invariant under feature space transformation. Mutual information has the following four main properties.

- Symmetry:
$$MI(X, Y) = MI(Y, X). \tag{4}$$
- Non-negativity:
$$MI(X, Y) \geq 0. \tag{5}$$
- Extremality:
$$\begin{aligned} MI(X, Y) &\leq MI(X, X), \\ MI(X, Y) &\leq MI(X, Y). \end{aligned} \tag{6}$$

Algorithm 1. Automatic feature selection algorithm.

Input: $n*d$ dimensional data matrix X, sample label vector Y
Output: Characteristics associated with tag Y

Algorithm 1: Calculate the mutual information $MI(X_i, Y)$ between d features X_i and labels Y respectively, $i = 1, ..., d$;

2. Sort the mutual information $MI(X_i, Y)$ in descending order and sort the d features X_i accordingly to obtain the sorted feature set $X^* = (X_1^*, X_2^*, ..., X_d^*)^T$;
3. Select the first k features with the largest mutual information MI to form the initial subset of relevant features $S_0 = (X_1^*, X_2^*, ..., X_k^*)^T$;
   Let the remaining subset consisting of d-k uncorrelated features be $S_1 = (X_{k+1}^*, X_{k+2}^*, ..., X_d^*)^T$;
4. Perform t-test
5. Update $S_0 = (X_1^*, X_2^*, ..., X_k^*, X_{k+1}^*)^T$ and $S_1 = (X_{k+2}^*, ..., X_d^*)^T$
6. Repeat steps 4 and 5 until $S_0 = (X_1^*, X_2^*, ..., X_k^*, X_{k+1}^*, ..., X_{d-2}^*)^T$, $S_1 = (X_{d-1}^*, X_d^*)^T$
7. Select the largest tmax from the $d-k-1$ t-tests that have been performed.
8. If the hypothesis test corresponding to t_{max} is significant, the features in the set of features S_0 for computing t_{max} are considered to be the features associated with label Y.

- When the variables X and Y are independent of each other, the mutual information $MI(X, Y) = 0$.

Once the relationship metric between each feature and the class label is available, a threshold approach is usually used to select features that are relevant to the class label [9, 10]. For example, a feature X_i is considered relevant to a class label Y if the mutual information $\widehat{MI}(X_i, Y) \geq \eta$, where η is a threshold value set in advance. Another approach is to draw a graph for the descending mutual information estimate $\widehat{MI}(X_i^*, Y)$, $i = 1, 2, ..., d$. Similar to the Principal Component Analysis (PCA) when selecting the number of principal components using the gravel plot, the features with very small mutual information afterwards are eliminated by using the threshold at the cliff drop and the value close to 0 as the threshold. However, the selection of the threshold is difficult to determine and requires many trials for tuning, so in this paper, an algorithm is used to automatically select relevant features based on the correlation between the mutual information obtained features and class labels [11], which avoids the tediousness of tuning and does not need to consider the redundancy between features. The basic idea is to divide the set of features arranged in descending order according to the mutual information into two subsets. One is the set of relevant features with high mutual information, and the remaining features form the set of irrelevant features. To seek the best division, we select the t test statistic as the separation measure for the set division, as shown in Algorithm 1.

In this paper, the data set excluding user number id and class label has 5151 features. According to Algorithm 1, we first calculated the mutual information between 5151 features and class labels. Then after descending the order, we obtained the top four features with high relevance to the class labels, ie., the summation and average value of the total traffic to visit the website or APP, the average upload speed and the average total speed to visit the website or APP. The mutual information of their 4 features and class labels is 0.46913, so we select these 4 features to form the initial relevant feature subset, and the remaining 5147 features form the initial irrelevant feature subset, and then carry out two positive tests on the mutual information. The t-test of the overall mean difference of the state, where the t-test statistic considers the situation that the overall variance is unknown and unequal. t-test the feature with the largest mutual information in the irrelevant feature subset. After the t-test, the feature with the largest mutual information in the subset of uncorrelated features is put into the subset of correlated features for updating, and then the t-test is done again and the procedure is repeated until only two features remain in the subset of uncorrelated features, at which point a

total of 5146 *t*-tests are done. Looking for the maximum *t*-test statistic of 397.951235 in 5146 *t*-tests, which is obviously much larger than the *t*-quantile fraction $t_1 = \frac{\alpha}{5146} = 0.84131836$ with a Bonferroni modified degree of freedom of 5038.873878, where $\alpha = 0.05$ is the original hypothesis is rejected and the set division is considered significant, and finally 63 features are retained to form a subset of relevant features. The feature set was optimized by eliminating features with low correlation with class labels through the algorithm of automatic selection of relevant features, but redundant information still existed among the relevant features left behind.

2.4.2. RENN Under-Sampling Algorithm

After interpolating the samples with equal probability for each centroid, the minority class samples may be more than the majority class samples, and the sample removal operation is performed at this time. Together with the use of Repeated Edited Nearest Neighbours (RENN) under sampling method to remove the excessive minority class samples generated by the K-means Synthetic Minority Over-sampling Technique (SMOTE) algorithm oversampling until the balance in the dataset is reached.

The Edited Nearest Neighbours (ENN) algorithm uses the Nearest Neighbor K-Nearest Neighbors (KNN) algorithm to edit the dataset, and for each majority class sample, if more than half of its K-nearest neighbor samples or all of them do not agree with it, it will be deleted, thus reducing the number of majority class samples. The final majority class samples that are kept belong to the same class, and most or all of their nearest neighbors belong to the same class. The RENN algorithm is an extension of the ENN algorithm, which is formed by repeating the ENN algorithm several times. Although data cleaning using under sampling of data can only remove a very limited number of sample points and cannot control the number of samples in advance, this type of method can be used in combination with oversampling methods to eliminate the redundant new sample points generated in oversampling.

Since the imbalance of the data will have a great impact on the performance of the classification model, hybrid sampling of the down sampled dataset is used to equalize the distribution of the dataset, which is conducive to the subsequent training to obtain a more stable and reliable classification model and thus improve the classification effect of the model. In this paper, we adopt the hybrid sampling algorithm based on K-means SMOTE and RENN: firstly, we oversample the minority class samples in the dataset with optimal features by K-means SMOTE, and then combine the original samples with the newly generated minority class samples to expand the minority class sample space; then we merge the new oversampled samples with the majority class samples in the dataset to obtain the new complete data. The new complete dataset contains 8202 samples, including 4103 minority samples (risky users) and 4099 majority samples (normal users); finally, the complete dataset is under sampled using the RENN under sampling algorithm to clean up the fuzzy decision boundary samples and make the positive and negative class boundaries clearer, so as to obtain the dataset with balanced distribution. The data set of the equalized distribution contains 3666 normal users and 3372 risky users. The hybrid sampling algorithm cleverly avoids the drawbacks of using either the oversampling or under sampling algorithms alone and uses the advantages of both to reconstruct a sample dataset with a new equilibrium of class distribution in the dataset [12-14].

3. Mobile Network Risk User Identification

3.1. Model Evaluation Indicators

Conventional classification algorithms usually use classification accuracy or error rate as a metric to evaluate the performance of classification models, which can accurately and reliably reflect the performance of classifiers in low-dimensional balanced datasets. However, when it is used to evaluate the classification effect of the model on high-dimensional unbalanced data, the classification accuracy can mislead us to think that the classifier has good performance, but in fact, the classifier may misclassify a few classes of samples into most classes of samples,

which is often very costly and can cause incalculable losses in practice. For unbalanced data sets we tend to focus more on the classification results of minority samples, and modeling is to improve the recognition rate of minority samples, so we need to choose appropriate classification performance evaluation metrics to help us effectively identify minority samples.

Risky user identification based on mobile network communication behavior is a typical binary classification problem, and the confusion matrix of its classification results is shown in Table 3. NTP is the number of identified risky users *i.e.* true cases; NFN is the number of risky users predicted as normal users *i.e.* false negative cases; NFP is the number of normal users predicted as risky users *i.e.* false positive cases; NTN is the number of identified normal users *i.e.* true negative cases.

Table 3. Confusion matrix.

Real Category	**Predicted results**	
	Predicted to be at risk users	**Predicted to be a normal user**
At-Risk Users	N_{TP}	N_{FN}
Normal users	N_{FP}	N_{TN}

The accuracy can be obtained from the confusion matrix:

$$\text{precision} = \frac{N_{TP}}{N_{TP} + N_{FP}} \tag{7}$$

Checking completeness rate:

$$\text{recall} = \frac{N_{TP}}{N_{TP} + N_{FN}} \tag{8}$$

The accuracy rate and the completeness rate are a pair of incompatible evaluation metrics. In classification of unbalanced data, one of them alone cannot accurately reflect the performance of the classifier, and it is necessary to consider them together.

$$F\text{-}measure = \frac{1 + \beta^2 \cdot \text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}} \tag{9}$$

Among them, $\beta > 0$ measures the relative importance of the recall to the precision. $\beta = 1$ is the standard $F1$ measure, and the recall to the precision are of equal importance; $\beta > 1$, the recall is more important than the precision; $\beta < 1$, the precision is more important than the recall. In the face of unbalanced datasets, the focus of information is often concentrated on the minority class samples, so it is necessary to improve the recognition rate for the minority class samples, and F-measure focuses more on the evaluation of the classification performance of the minority class samples, and it is more reasonable and objective to use it as the classification performance evaluation index.

The Receiver Operating Characteristic (ROC) is a powerful tool to study the generalization ability of the model, which can still maintain good stability when the sample distribution of the test set of the model changes. The ROC curve takes the true rate ie., True Positive Rate (TPR) as the vertical axis, and the false positive rate False Positive Rate (FPR) as the horizontal axis. The formulas of TPR and FPR are as follows:

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

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

In this paper, the true rate indicates the proportion of risky users who are predicted to be risky users, and FPR indicates the proportion of normal users who are predicted to be risky users. The point (0, 1) indicates that the model classifies all samples correctly, which is the best case of the risky user identification model; the point (1, 0) indicates that the model misclassifies all samples, which is the worst case of the model; the point (0, 0) indicates that the model predicts all samples as normal users; the point (1, 1) indicates that the model predicts all samples as risky users. The ROC curve can visually see the overall generalization ability of each learning classifier. If the ROC curves of multiple learning classifiers are compared, the ROC curves of each model may cross on a single graph, and at this time it is impossible to judge the performance of learning classifiers directly by observation. The area under ROC curve solves

this problem by calculating the area, which distinguishes the overall generalization ability of each model [18]. The Area Under Curve (AUC) is more robust and less affected by the category distribution, so the AUC is a better measure of the model's prediction ability for category imbalance data.

3.2. Stacking Model Fusion

The algorithms for model fusion include Stacking, Blending, Bagging, Boosting, *etc.* In this paper, the Stacking [15] model fusion algorithm with a two-layer stacking framework is used to train a dataset with equalized distribution. The details are as follows.

1. First, divide the dataset of equalized distribution, 70% as the training set and 30% as the test set, and the sample situation after division is shown in Table 4.
2. The training set is divided into N equal-sized sub-training sets and input to the K base learners in the first layer, as shown in Figure 2. One of the sub-training sets is used as the validation set each time, and the remaining N–1 sub-training sets are used for the traning model. The operation traverses each sub-training set, and each base learner completes N training sessions. After that, the prediction is performed on the validation set and the test set, respectively. Finally, the results are output.
3. By stitching the N times validation set prediction results ($x_1 - x_N$) output by each base learner into a column, K base learners have K columns of validation set prediction results as the training input of the second layer meta-learner (S^1_{train}, S^2_{train}, ..., S^K_{train}). Similarly, by averaging the N times test set prediction results ($c_1 - c_N$) output by each base learner, K base learners have the average of K test set prediction results as the test input to the second layer of meta-learner (S^1_{train}, S^2_{train}, ..., S^K_{train}), as shown in Figure 3.
4. The training input (S^1_{train}, S^2_{train}, ..., S^K_{train}), is trained in the meta-learner of the second layer, and then the trained Stacking model fusion algorithm is used to predict the test input (S^1_{train}, S^2_{train}, ..., S^K_{train}), to obtain the prediction results of Stacking model fusion.

Table 4. Classification of datasets with homogenized distribution.

	Sample size	**Most categories of samples (normal users)**	**Minority sample (at-risk users)**
Training set	4926	2566	2360
Test set	2112	1100	1012

The Stacking model fusion algorithm improves the overall prediction accuracy of the model by generalizing the prediction results generated by all base learners using a meta-learner. The multi-classifier combination approach based on Stacking integrated learning strategy, in order to achieve the best prediction effect of the integrated learning model, it is necessary to ensure not only the individual prediction ability of each base learner, but also to consider the combined effect of each base learner. Since the base model with strong learning ability can improve the overall prediction effect of the model, the machine learning model with excellent prediction performance should be selected for the first layer of base learners, and the diversity of

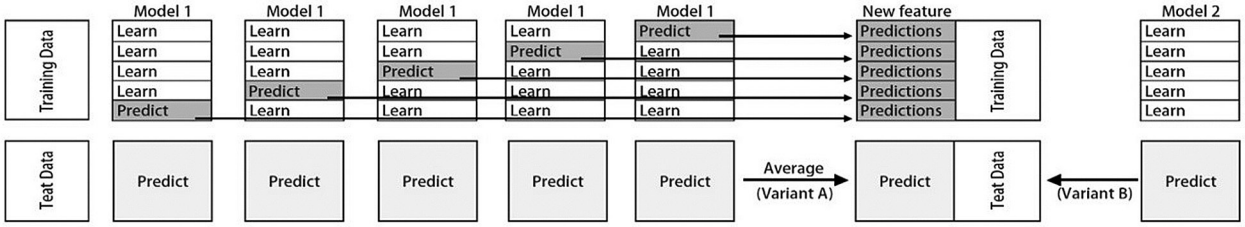

Figure 2. Stacking first layer schematic.

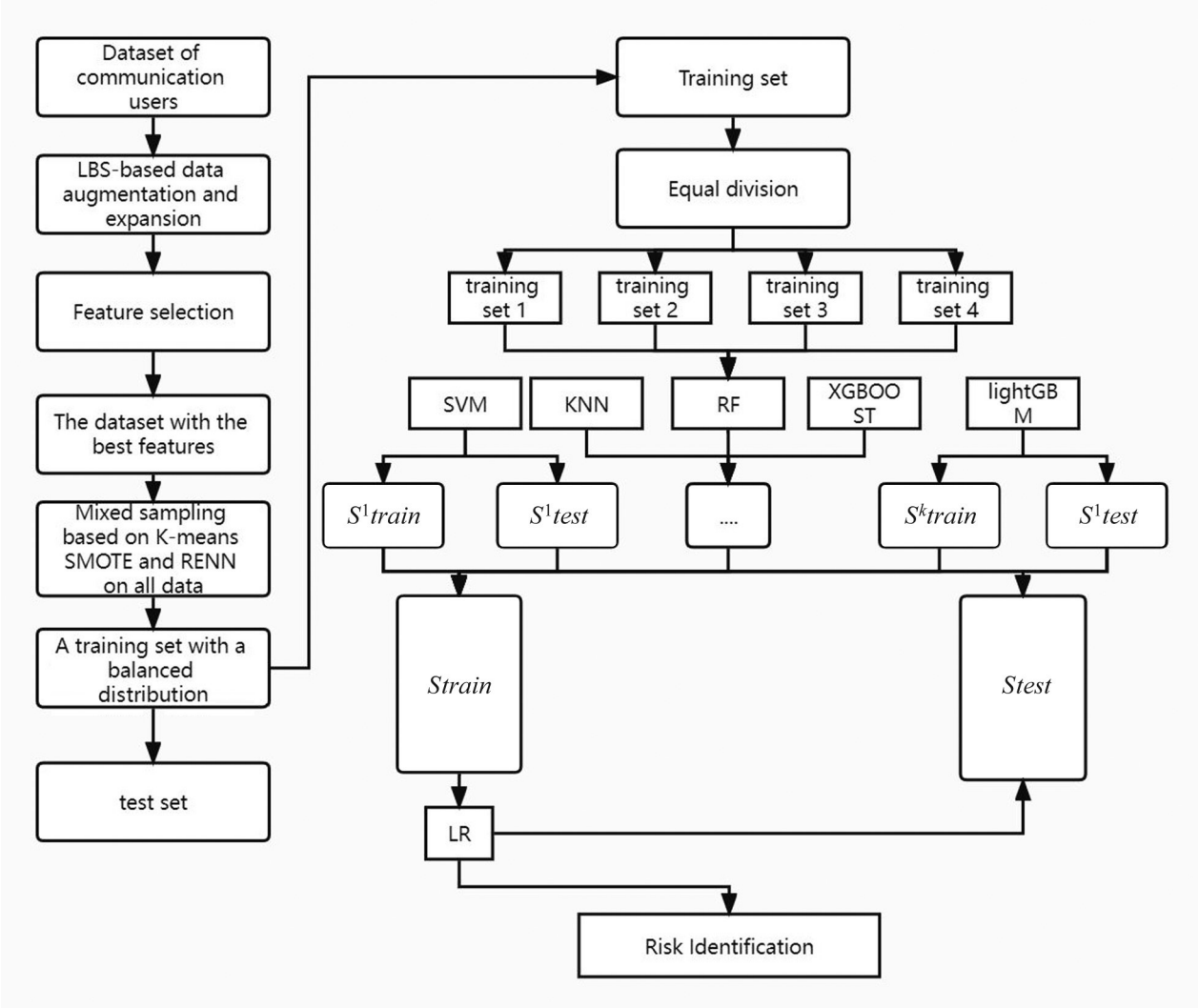

Figure 3. G-stacking model fusion algorithm.

models should be ensured, because considering the different perspectives of different algorithmic models to observe the data and their own algorithmic principles, different models can be built for the same data and then fused together to explore the data information in multiple directions. Therefore, it is important to select diverse base learners. Therefore, the selection of diverse base learners can concentrate the advantages of different algorithms and make each differentiated model complement each other. In this paper, classical machine algorithms including Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Random Forest (RF), eXtreme Gradient Boosting tree (XGBoost), and Light Gradient Boosting Machine (LightGBM) with excellent prediction performance and low algorithm time complexity are chosen as the base classifier in the first layer.

Among them, Random Forest (RF) is a representative algorithm of Bagging integrated learning, with small operation, fast classification speed and robust results; XGBoost and LightGBM are advanced algorithms based on Boosting integrated learning framework, with rigorous theory and excellent prediction ability; SVM can solve the problem of nonlinear features and large feature space, and is not easily affected by noise interference; KNN is simple in principle and has low training time complexity. Since the principles of SVM, KNN, RF, XGBoost and LightGBM are different, the correlation of prediction results is low, and the composition of model fusion is beneficial to improve the accuracy of final classification. The meta-learn-

er in the second layer is generally selected as a simple model with better stability to improve the model performance overall, so the Logistic Regression (LR) with simple model and good generalization performance is used as the meta-learner in the second layer of this paper.

According to the principle of Stacking algorithm, we know that the input of the second layer meta-learner is generated from the output of each base learner in the first layer, and if the training set of the base learner is used directly as the training set of the second layer meta-learner without the cross-validation method, the data is repeatedly learned by the two-layer learner, which is very likely to cause the overfitting problem of the Stacking model [16]. In this paper, in the process of Stacking model fusion, each base learner performs a 4-fold cross-validation process on the training set, using one data block as the validation set and the corresponding remaining three data blocks as the training set, and after each fold of cross-validation is completed, the validation set is predicted using the base learner, *i.e.*, each base learner produces the same number of training sets as the original training set at the end of the 4-fold cross-validation of the new datasets. The cross-validation process converts all the data in the dataset from input to output to achieve one data transformation, avoiding the risk of model overfitting due to data being reused, and the training set of the second layer meta-learner comes entirely from the predicted output of each base learner in the first layer, so the meta-learner fully integrates the model advantages of each base learner in the first layer for model construction, which has the overall effect of improving the model from effect.

The framework diagram of the mobile network risk user identification method based on Stacking model fusion is shown in Figure 3, and the training process is roughly as follows.

1. After feature construction and pre-processing to obtain the 5609 × 5153 ultra-high-dimensional communication user dataset, first use Algorithm 1 automatic feature selection algorithm for the communication user dataset to remove features irrelevant to class labels and obtain a subset of relevant features.
2. The SVM-RFE algorithm is used to further remove the redundant features on the basis of the relevant feature subsets to obtain the optimal feature subsets with strong discriminatory ability.
3. Hybrid sampling with a combination of K-means SMOTE and RENN for the dataset with optimal features to equalize the distribution of the dataset and obtain a more reliable dataset.
4. Dividing the data set of the equalization distribution, 70% as the training set and 30% as the test set.
5. The optimal hyperparameters of each model are selected by using grid search plus cross-validation on the training set for each of the five classifiers in the first layer of Stacking.
6. The five classifiers of the first layer of Stacking are trained on the training set using cross-validation to produce the training set of the second layer of meta-learner, and the five base classifiers are predicted on the test set and the test set of the second layer of meta-learner is obtained by averaging the multiple prediction results of each test sample.
7. The newly generated training set of each base learner in the first layer of Stacking is used to train the meta-learner in the second layer of Stacking integrated framework to obtain a risky user identification model based on mobile network communication behavior, and then the prediction results are obtained by inputting the newly generated test set of the first layer of Stacking into the model to achieve effective classification of high-dimensional imbalanced data.

4. Hyperparameter Selection and Performance Evaluation

In order to optimize the performance of Stacking model fusion, the learning ability of five base learner models, namely SVM, KNN, RF, XGBoost, and lightGBM, is analyzed on the basis of the equalized distributed dataset. The hyperparameter selection is performed by grid search with cross validation for each base

learner. Firstly, the training set of the equalization distribution is divided into training set and validation set, and the prediction effect of the model in the validation set after training with different hyperparameter sets is observed by cross validation, so as to select the optimal hyperparameter set for each model. In this way, we can avoid the chance of a situation arising from the division of data sets and ensure the reliability of classification detection results. The tuned base classifier models are then applied on the test set, and the model evaluation metrics are used to see the prediction and classification effects of the models.

In addition, six single models, SVM, KNN, RF, Logistic Regression (LR), XGBoost, and LightGBM, are used to model and analyze the equilibrium data, and then compared with the Stacking model fusion mobile network risk user identification model. The set of hyperparameters of each model and the prediction performance are shown in Table 5. Because the process of finding the optimal parameter model and evaluating the performance of the five base learners in the first layer of the Stacking fusion model is the process of modeling the equilibrium data by five single models, *i.e.*, the five single models of SVM, KNN, RF, XGBoost, and LightGBM have already completed the modeling. The analysis only needs to be done again for LR using the grid search method for algorithmic model parameter tuning, while the comprehensive performance evaluation of the established LR model is performed using 5-fold cross-validation.

Table 5 shows that the Stacking model has the truest cases and the least false negative cases among the models after hyperparameter optimization, which means that the risky users are the least likely to be predicted as normal users and the recognition rate of risky users is the highest; the AUC of each model is between 0.968 and 0.977, and the Stacking model has the largest AUC of 0.9767. The AUC of Stacking model fusion is 0.9767, with an advantage of 0.0001 over random forest; we prefer to have a small number of false negative cases, *i.e.*, we pay more attention to the recognition effect of a small number of samples, and the impact of the check-all rate is greater than the check-accuracy rate, so the β in the F-measure is taken as 2.

Table 5. Hyperparameter set and prediction performance of each model.

Model Name	Hyperparameters	Predictive Performance			
		AUC	F-measure (β = 2)	Real example	False negative example
KNN	n neighbors = 30, p = 5, weights = 'distance'	0.9760	0.9670	974	38
SVM	C = 10, gamma = 0.001, kernel = 'rbf'	0.9685	0.9644	974	38
RF	max depth = 7, min samples leaf = 10, min samples split = 20, n estimators = 110	0.9766	0.9672	973	39
XGBoost	n estimators = 145, max depth = 7, learning rate = 0.1, booster = 'gbtree', gamma = 2.3, reg lambda = 1, min child weight = 1, subsample = 0.7	0.9758	0.9674	974	38
LightGBM	n estimators = 42, max depth = 12, learning rate = 0.1, num leaves = 21, max bin = 15, min data in leaf = 101, subsample = 0.8, bagging fraction = 0.9, feature fraction = 1, bagging freq = 10	0.9757	0.9668	973	39
LR	C = 10, penalty = 'l2', tol = 0.001	0.9712	0.9655	974	38
Stacking		0.9767	0.9684	975	37

The F-measure of each model is concentrated between 0.964 and 0.969, with that of Stacking model fusion being greater than the other models. When all the evaluation metrics are combined, the Stacking model improves the AUC by 0.01% and the F-measure by 0.1% compared with the best algorithm models Random Forest (maximum AUC) and XGBoost (maximum F-measure) among the single models. Compared with SVM, the least effective algorithm among single-model classification algorithms, AUC and F-measure improved by about 0.85% and 0.35%, respectively. Therefore, stacking model has the best prediction effect and the greatest advantage. The classification performance of the five different algorithmic models as the first layer base learner of the Stacking model is good, and the evaluation index of each model is above 0.96, which meets the requirements of the Stacking framework for excellent prediction performance of the first layer base learner and inter-model diversity. In summary, using Stacking model fusion algorithm for risky user identification based on mobile network communication behavior can better identify risky user.

5. Conclusion

This paper considers mobile network calls, SMS, access traffic, and other communication behavior records which contain rich and valuable information. User communication behavior features are extracted from these historical records and analyzed using a Stacking model fusion method that combines feature selection and hybrid sampling. The method identifies risky users in telecommunication, which is crucial for preventing telecommunication fraud, crimes that threaten social security, and false network entries by users. The Stacking model fusion approach overcomes the limitations of single models in risk identification by combining the strengths of multiple machine learning algorithms for improved prediction. In this paper, a two-layer Stacking model fusion algorithm is applied to degraded and balanced user communication data using SVM, KNN, RF (Bagging algorithm), XGBoost, and LightGBM (Boosting algorithm) as the first layer base models. Hyperparameter tuning is performed to optimize the performance of each base model, and the LR model is chosen as the meta-classifier in the second layer for model fusion. Results indicate that the Stacking model fusion algorithm in this paper can accurately predict telecommunication user categories, enhance the generalization performance of telecommunication risky user identification, and provide a useful reference for the telecommunication industry.

In future research, the following issues need to be continued to be explored and improved. Because of the complex design of Stacking integrated learning framework, the selection of each layer of models and the combination of models will have a significant impact on the final model effect, so we need to carefully select the base learners and meta-learners and try other classification algorithms for model fusion. Moreover, the Stacking integrated learning framework requires high prediction performance of the base model, which requires hyperparametric merit selection of multiple base learners by using grid search plus cross-validation, a time-consuming and computationally intensive process, so the support of distributed computing technology is needed in future work to disassemble the task and model the base model at different terminals, which will greatly reduce the algorithm time complexity and improve the efficiency.

References

[1] R. Zebari *et al.*, "A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction", *Journal of Applied Science and Technology Trends*, vol. 1, no. 2, pp. 56–70, 2020.

[2] J. Liu *et al.*, "Improved Stacking Model Fusion Based on Weak Classifier and Word2vec", in *Proc. of the 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*, 2018, pp. 820–824. http://dx.doi.org/10.1109/ICIS.2018.8466463

[3] Y. Zhang *et al.*, "A Feature Selection and Multi-model Fusion-based Approach of Predicting air Quality", *ISA transactions*, vol. 100, pp. 210–220, 2020. http://dx.doi.org/10.1016/j.isatra.2019.11.023

[4] B. Liu *et al.*, "Research on Fault Diagnosis of IPMSM for Electric Vehicles Based on Multi-Level Feature Fusion SPP Network", *Symmetry*, vol. 13, no. 10, p. 1844, 2021. http://dx.doi.org/10.3390/sym13101844

[5] X. Zhang *et al.*, "Research on Transformer Fault Diagnosis: Based on Improved Firefly Algorithm Optimized LPboost-classification and Regression Tree", *IET Generation, Transmission & Distribution*, vol. 15, no. 20, pp. 2926–2942, 2021. http://dx.doi.org/10.1049/gtd2.12229

[6] Y. Liu *et al.*, "Transformer Fault Diagnosis Technique Based on AdaBoost-RBF Algorithm and DSmT", *Power Automation Equipment*, vol. 39, no. 6, pp. 166–172, 2019.

[7] C. Zhang *et al.*, "Research on Classification Method of High-dimensional Class-imbalanced Datasets Based on SVM", *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 7, pp. 1765–1778, 2019. http://dx.doi.org/10.1007/s13042-018-0853-2

[8] A. Destrero *et al.*, "Feature Selection for High-dimensional Data", *Computational Management Science*, vol. 6, no. 1, pp. 25–40, 2009. http://dx.doi.org/10.1007/s10287-008-0070-7

[9] W. Gómez *et al.*, "Mutual Information and Intrinsic Dimensionality for Feature Selection", in *Proc. of the 2010 7th International Conference on Electrical Engineering Computing Science and Automatic Control*, 2010, pp. 339–344. http://dx.doi.org/10.1109/ICEEE.2010.5608600

[10] C. Pascoal *et al.*, "Robust Feature Selection and Robust PCA for Internet Traffic Anomaly Detection", in *Proceedings of the IEEE INFOCOM*, 2012, pp. 1755–1763. http://dx.doi.org/10.1109/INFCOM.2012.6195548

[11] D. Zhang, "Research on Classification Problems Based on Unbalanced Datasets", *Yunnan University of Finance and Economics*, 2020. (in Chinese)

[12] D. H. Wolpert, "Stacked Generalization", *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992. http://dx.doi.org/10.1016/S0893-6080(05)80023-1

[13] J. Jiang *et al.*, "Electrical Load Forecasting Based on Multi-model Combination by Stacking Ensemble Learning Algorithm", in *Proc. of the 2021 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, 2021, pp. 739–743. http://dx.doi.org/10.1109/ICAICA52286.2021.9498248

[14] X. Guo *et al.*, "Study on Short-term Photovoltaic Power Prediction Model Based on the Stacking Ensemble Learning", *Energy Reports*, vol. 6, pp. 1424–1431, 2020. http://dx.doi.org/10.1016/j.egyr.2020.11.006

[15] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System", in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), New York, NY, USA*, 2016, pp. 785–794. http://dx.doi.org/10.1145/2939672.2939785

[16] R. K. Mazumder *et al.*, "Failure Risk Analysis of Pipelines Using Data-driven Machine Learning Algorithms", *Structural Safety*, vol. 89, no. 102047, 2021. http://dx.doi.org/10.1016/j.strusafe.2021.02.011

Received: September 2022
Revised: Februrary 2023
Accepted: February 2023

Contact addresses:
Jianhong Lin
College of Computer Science and Technology
Zhejiang University
Hangzhou
China
Zhejiang Ponshine Information Technology Co., Ltd.
Hangzhou
China
e-mail: linjianhong2022@163.com

Peng Wang
College of Computer Science and Technology
Zhejiang University
Hangzhou
China
e-mail: wangpengla@163.com

Chunming Wu*
College of Computer Science and Technology
Zhejiang University
Hangzhou
China
e-mail: wuchunming@zju.edu.cn
*Corresponding author

Jianhong Lin received the BSc degree in electrochemical engineering from Zhejiang University of Technology, Hangzhou, China, in 1998, and the MSc degree in business administration from Zhejiang University, Hangzhou, China, in 2014. He is currently pursuing the PhD degree with Zhejiang University. He is also the Chief Technology Officer of Zhejiang Ponshine Information Technology Company Ltd., Hangzhou, China. His current research interests include network security and security management.

Peng Wang received the BSc degree in Communication Engineering from Shandong Normal University, Jinan, China, in 2013, and the MSc degree in Computer technology from University of Chinese Academy of Sciences, Beijing, China, in 2016. She is currently pursuing the PhD degree at Zhejiang University. She is also a senior engineer of H3C Technology Company Ltd., Hangzhou, China. Her main research interests include network communication and network security.

Chunming Wu received the PhD degree in computer science from Zhejiang University in 1995. He is now a Professor with the College of Computer Science and Technology, Zhejiang University. He is also the Associate Director of the Research Institute of Computer System Architecture and Network Security, Zhejiang University, and the Director of the NGNT Laboratory. His current research fields include software-defined networks, reconfigurable networks, network security, proactive network defense, and the architecture of next-generation Internet.