Hindawi
Security and Communication Networks
Volume 2020, Article ID 5695943, 10 pages
https://doi.org/10.1155/2020/5695943

Research Article

Research and Implementation on Power Analysis Attacks for Unbalanced Data

Xiaoyi Duan, Dong Chen, Xiaohong Fan, Xiuying Li, Ding Ding, and You Li

Beijing Electronic Science and Technology Institute, Beijing 100070, China

Correspondence should be addressed to Xiaoyi Duan; duanxiaoyi@besti.edu.cn

Received 19 November 2019; Accepted 1 May 2020; Published 22 May 2020

Academic Editor: Gregorio Martinez Perez

In the power analysis attack, when the Hamming weight model is used to describe the power consumption of the chip operation data, the result of the random forest (RF) algorithm is not ideal, so a random forest classification method based on synthetic minority oversampling technique (SMOTE) is proposed. It compensates for the problem that the random forest algorithm is affected by the data imbalance and the classification accuracy of the minority classification is low, which improves the overall classification accuracy rate. The experimental results show that when the training set data is 800, the random forest algorithm predicts the correct rate of 84%, but the classification accuracy of the minority data is 0%, and the SMOTE-based random forest algorithm improves the prediction accuracy of the same set of test data by 91%. The classification accuracy rate of a few categories has increased from 0% to 100%.

1. Introduction

In the process of machine learning, the relevant learning algorithm is generated from the accumulated data and the data obtained from experience is provided to the learning algorithm to generate the corresponding model after processing. When solving new problems, the model can give corresponding judgments. The side channel attack is a method to attack crypto devices by collecting side channel information data (power consumption, electromagnetic radiation, running time, etc.) leaked by cryptographic devices during operation. Machine learning is a method of "learning" from data and being able to analyze unknown data. It conforms to the needs of side channel attack to crack unknown encryption information. In recent years, many researchers have applied machine learning to side channel attacks. The fusion of machine learning and side channel attacks not only improves the reliability and automation of attacks, but also improves the efficiency compared with traditional side channel attacks. In 2011, Gabriel et al. first applied machine learning techniques to power analysis attacks in side channel attack [1]. Using the data set with obvious hamming weight leakage, they successfully attacked some software implementations of Advanced Encryption Standard (AES) by using the least squares support vector machine (LS-SVM). They pointed out that the traditional template attack is a multiclassification question. Meanwhile, the experiment shows that the parameter setting of support vector machine (SVM) has a great impact on the classification performance, but the number of power curves and sampling points has a small impact on the results. In 2012, Hera He et al. used SVM classification algorithm to attack Data Encryption Standard (DES) algorithm running on 8-bit smart cards [2]. This was the first paper that used SVM to crack the complete key. In COSADE 2012, Heuser et al. based on the hamming weight model, used the multiclassification SVM with probability to divide the keys into nine categories [3]. Their experiments show that in order to achieve the same goal under strong noise, SVM attacks require less training power curves than template attacks, so SVM attacks are more general. At the 2015 CHES conference, Whitnall et al. proposed a classifier that can tolerate some differences between analysis and attack marks by using unsupervised machine learning methods combined with average trace and PCA algorithm. The experimental results show that the method is effective [4]. In 2019, Kim et al.

proposed a new convolutional neural network to side channel analysis [5] and Mathieu et al. used deep learning to evaluate secure RSA implementations [6].

When considering imbalanced data, Geetha et al. used the SMOTE to analyze imbalance of medical data in 2019 [7]. Luo et al. proposed a novel Divergence-Encouraging Autoencoder (DEA) to solve imbalanced data by encouraging maximization of divergence loss between different classes in the bottleneck layer to solve [8].

The power analysis attack based on machine learning is a classification problem and it is similar to the traditional template attack. The data imbalance in the classification is also worth paying attention to. Due to unbalanced data, machine learning algorithms will pay too much attention to the majority of classes in all classifications, resulting in the degradation of classification performance of minority classes [9]. In order to reduce the number of templates, most of the existing studies choose the intermediate hamming weight model to create templates. This leads to data imbalances; there may be only three hamming weights in the 1000 curves, which leads to a high overall accuracy rate for data classification, but low accuracy rate for individual categories, even 0%.

When adopting hamming weight model, researchers generally adopt the method of controlling the number of acquisition curves; that is, the curves with the same number are collected by each type of hamming weight, or generate a few types of data or eliminate most types of data through algorithms to achieve data balance. With the development of side-channel attack, the technology against side-channel attack is also developing. In real life, there will be many restrictions that make it difficult to the control curve quantity method. For example, the attack chip encryption times are limited, encryption time is limited, etc., which leads to the possibility of collecting the power consumption curve on the chip model only a few hundred or less. In this paper, the SMOTE technique is introduced in the attack of power consumption, and the authors suggest a method of random forest classification based on SMOTE, using the SMOTE algorithm to synthesize the minority samples to achieve the balance and improve the accuracy of the minority and the whole.

This paper is structured as follows. In Section 2, the disadvantages of unbalanced data distribution on random forest algorithm are pointed out. In Section 3, random forest algorithm is introduced, analyses the unbalanced power data attack based on RF-SMOTE algorithm and discuss unbalanced power consumption data. Section 4 analyses the characteristics of power data and the feature point is selected. In Section 5, experimental results are discussed; analysis and comparison are shown. In Section 6, conclusion and future work are presented.

2. The Influence of Unbalanced Data Distribution on Random Forest Algorithm

In this paper, S-box output is used as the intermediate value. AES S-box encryption is operated every 8 bits, so only the 8 bits of hamming weight are used as the label each time. Therefore, labels can be divided into 9 categories (hamming weight is 0–8), which are represented as H0~H8, respectively. However, the key and plaintext are random in the process of data acquisition, so the hamming weight ratio of every 8 bits will be different if the probability of each 0 and 1 is guaranteed to be the same. The weight and power consumption ratio of each hamming is

$$\begin{aligned} &\text{H0: H1: H2: H3: H4: H5: H6: H7: H8} \\ &= 1\text{: } 8\text{: } 28\text{: } 56\text{: } 70\text{: } 56\text{: } 28\text{: } 8\text{: } 1. \end{aligned} \tag{1}$$

It can be seen that at least 256 training set curves are required to contain all categories, among which the proportion of the least minority category is 1/256, and the proportion of the most majority category is 70/256. With hamming weights of 0 and 8, if only a few curves are used as references, the generalization ability of the trained model is not sufficient.

Suppose that there is an unbalanced training data set $X\_\text{Train} = \{(x_i, y_i \mid i = 1, 2, \ldots, n)\}$, where $x_i \in R^m$ is the sample value, n is the total number of samples, m is the sample dimension, and $y_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$ is the corresponding label, and suppose that the number of samples of the class with the highest proportion is h, and the number of samples of the class with the lowest proportion is l. The random forest algorithm uses the random sampling technique with putting back to form each decision tree; that is, the probability of each sample being sampled is $1/n$; then the probability of the highest proportion class samples being sampled is h/n, and the probability of the lowest proportion class samples being sampled is l/n. Therefore, the probability difference is obvious. That is to say, when there is too little data in a certain category, the sampling will only take the majority of classes, and the accuracy of classification of minority classes is very low, or even the extreme phenomenon that the classification results of minority classes are completely incorrect, thus affecting the prediction results.

In view of the above deficiencies, this paper introduces the data enhancement technology into the random forest algorithm and makes the probability of random sampling of each type of data equal by balancing data. The result of this method is to make up for the shortcomings of the original algorithm, improve the accuracy of a few classes and the accuracy of the overall.

3. Introduction to Algorithms

3.1. Random Forest Algorithm. The random forest algorithm is a common and effective supervised machine learning algorithm which is proposed by Leo Breiman and Adele Cutler [10]. This algorithm is an integrated algorithm based on decision trees, and it improves the disadvantages of decision trees and makes classification results better. Random forest algorithm is one of the commonly used algorithms by researchers because of its simple parameter setting.

Random forest is composed of many decision trees without pruning. It is an integrated learning algorithm

combined with voting method [11], and its performance is better than the integrated algorithm of a single decision tree. Its structure diagram is shown in Figure 1. First of all, m subsets $D = \{D_1, D_2, \ldots, D_m\}$ are obtained from random samples put back from data set $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$. Then, m subsets are used to construct m decision trees, and each subdecision tree outputs a result, where Ti represents a single decision tree algorithm and ti represents the output of the subdecision tree. At last, the majority voting method is adopted for the judgment results of subdecision trees; that is, the output results of random forest are obtained by finding the majority in matrix $t = \{t_1, t_2, \ldots, t_m\}$.

3.2. Random Forest Algorithm Based on SMOTE

3.2.1. SMOTE Algorithm. In order to solve the problem of unbalanced data set in the classification problem of machine learning, data sampling technology is adopted in most researches to make the training set become a balanced data set, which will improve the accuracy of classification results in most cases. Data sampling consists of oversampling and undersampling [12]. Oversampling is increasing the number of minority classification. Undersampling removes some samples from majority classification or selects only some samples from majority classification [13]. The training data used in this paper contains only a few minority classification. When the randomly selected training data is 800, only 3 hamming weights are 0. Due to the less number of minority classification, undersampling will cause a large amount of data to be lost in the final training set, so oversampling is selected as the data enhancement technology.

Data synthesis is a method to synthesize new samples according to existing sample rules. Oversampling data enhancement technique includes adaptive sampling arithmetic and the synthetic minority oversampling technique [14]. Adaptive synthetic sampling (ADASYN) oversampled different quantity categories by calculating the imbalance degree and obtained different sample numbers for each category. The basic idea of the SMOTE method is to analyze the minority classification samples and add new samples to the data set artificially based on the minority samples [15].

In contrast to simple oversampling, the SMOTE increases the variety of the data and gets the same amount of each type of classification. In this way, for a random forest, the probability of each type of curve obtained by a random sampling is equal. Compared with ADASYN, SMOTE data is more balanced after enhancement.

The schematic diagram of the SMOTE algorithm is shown in Figure 2, where the coordinate axes represent the two different attributes of data v1 and v2, and the circle represents the majority of samples and the triangle represents the minority of samples. Its algorithm is described as follows:

Let us say that there are T minority classification samples; one of them is t, and its eigenvector is X_t, where $t \in \{1, \ldots, T\}$:

(1) K-nearest neighbors of sample t are found by using Euclidean distance from T samples, denoted as $x_{t(\text{near})}$, where near $\in \{1, \ldots, K\}$.

(2) For each randomly selected sample $x_{t(\text{near})}$ in K-nearest neighbors, a random number a is generated and a new sample $x_{t(\text{new})}$ is synthesized according to the following formula, where $t \in \{1, \ldots, T\}$, near $\in \{1, \ldots, K\}$:

$$x_{t(new)} = x_t + a\left(x_{t(near)} - x_t\right). \tag{2}$$

(3) Repeat step 2 for N times to synthesize N new samples $x_{t(new)}$, new $\in \{1, \ldots, N\}$. All T minority classification samples are performed above, and NT new samples can be synthesized to achieve data balance.

3.2.2. Unbalanced Power Data Attack Based on RF-SMOTE Algorithm. Under unbalanced data, the implementation process of random forest classification method based on SOMTE algorithm in power analysis attack is described as follows:

(1) N power consumption curves are selected as the training data set, each curve has m sampling points, the original data matrix is $V_{nm} = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1m} \\ v_{21} & v_{22} & \cdots & v_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ v_{n1} & v_{n1} & \cdots & v_{nm} \end{bmatrix}$, and the label matrix corresponding to each curve is $H\_\text{Train}_{n1} = \begin{bmatrix} h_1 \\ h_2 \\ \cdots \\ h_n \end{bmatrix}$.

(2) Calculate the Pearson correlation coefficient between the sampling point of $V_$Train and $H_$Train at each moment to get the coefficient matrix $P = [\rho_1, \rho_2, \ldots, \rho_m]$, $\rho_1, \rho_2, \ldots, \rho_m$ is arranged from small to large in absolute value, and the sampling points corresponding to the first k points are taken to form the training set matrix $V\_\text{Train}_{nk} = \begin{bmatrix} v'_{11} & v'_{12} & \cdots & v'_{1k} \\ v'_{21} & v'_{22} & \cdots & v'_{2k} \\ \cdots & \cdots & \cdots & \cdots \\ v'_{n1} & v'_{n2} & \cdots & v'_{nk} \end{bmatrix}$.

(3) Use SMOTE to expand the data set of V_Train_{nk} to get the new training set matrix $V\_\text{Train}'_{Nk} = \begin{bmatrix} v''_{11} & v''_{12} & \cdots & v''_{1k} \\ v''_{21} & v''_{22} & \cdots & v''_{2k} \\ \cdots & \cdots & \cdots & \cdots \\ v''_{N1} & v''_{N2} & \cdots & v''_{Nk} \end{bmatrix}$ and its corresponding label matrix $H\_\text{Train}'_{N1} = \begin{bmatrix} h_1 \\ h_2 \\ \cdots \\ h_N \end{bmatrix}$ is obtained.

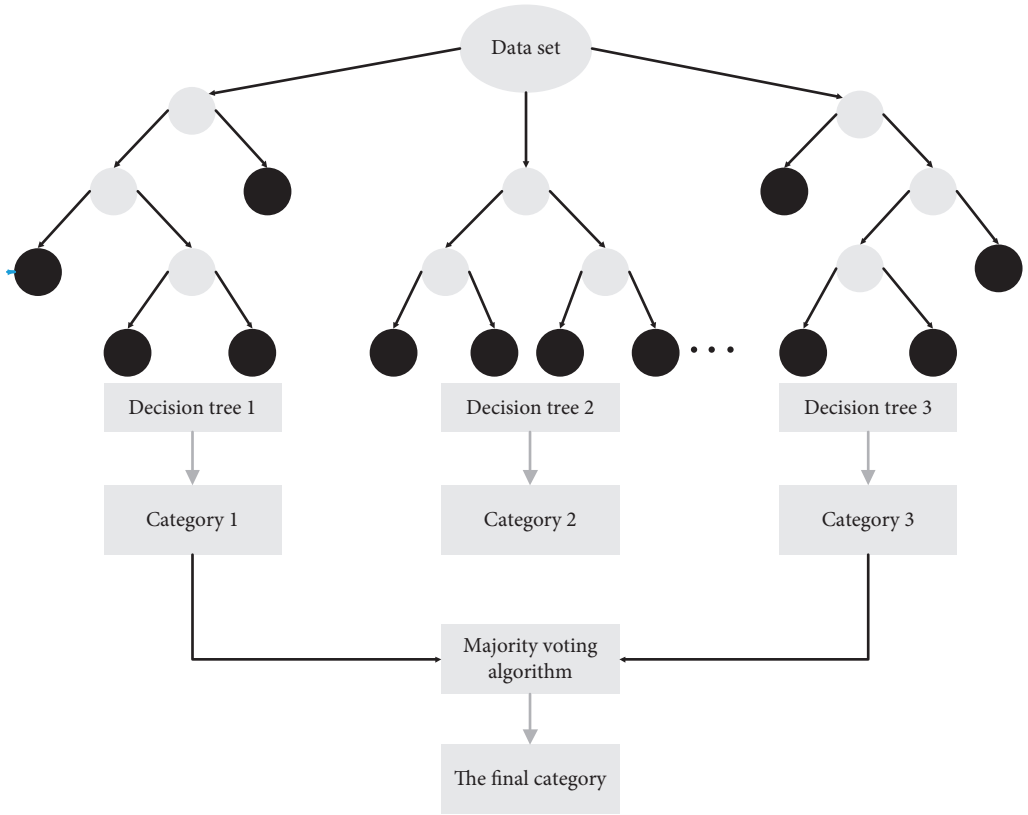

Figure 1: Schematic diagram of random forest structure.

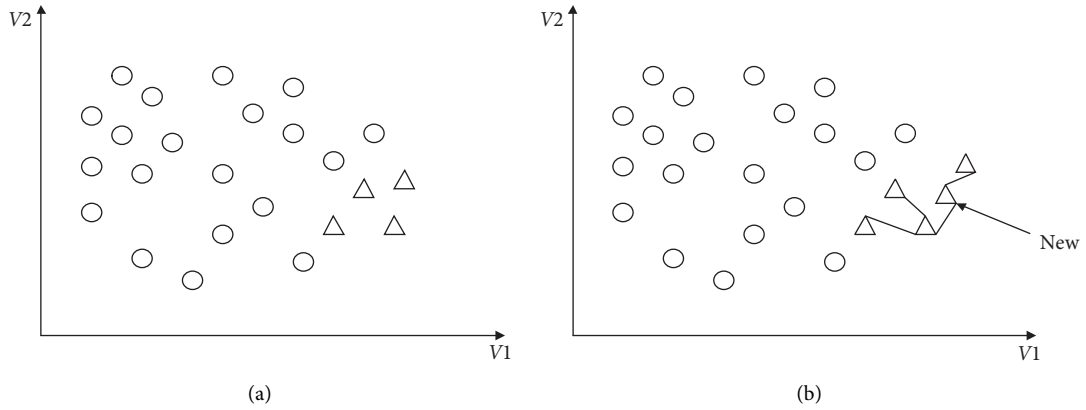

Figure 2: SMOTE process diagram. (a) Original data. (b) SMOTE synthetic data.

(4) Execute random forest algorithm, learn a model from the training set, and establish a mapping from $V\_\text{Train}'_{Nk}$ to $H\_\text{Train}'_{N1}$: $f : V\_\text{Train}'_{Nk} \longrightarrow H\_\text{Train}'_{N1}$.

(5) The model is used to predict the unknown sample x, labeled M.

3.3. The Research on Unbalanced Power Consumption Data. The single decision tree algorithm can directly reflect the characteristics of data according to attribute classification. In a relatively short time, the judgment result of big data is very good and the quality of attributes is the key factor to determine the performance of decision trees. If the attribute differentiation between classes is small, the test accuracy will be lower. But errors will increase with the increase of categories and overfitting is easy to occur. The random forest algorithm uses several different decision trees to classify the data, which improves the generalization ability of the algorithm and alleviates the occurrence of overfitting to a large extent.

The data set used in this paper has the characteristics of large data volume and high dimension. Take 1000 random power curves as an example, and their distribution is shown in Figure 5. Its characteristics are composed of the power

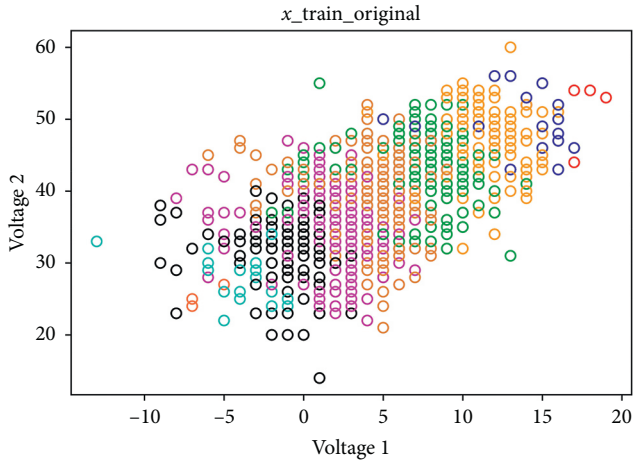

Figure 3: Original data.

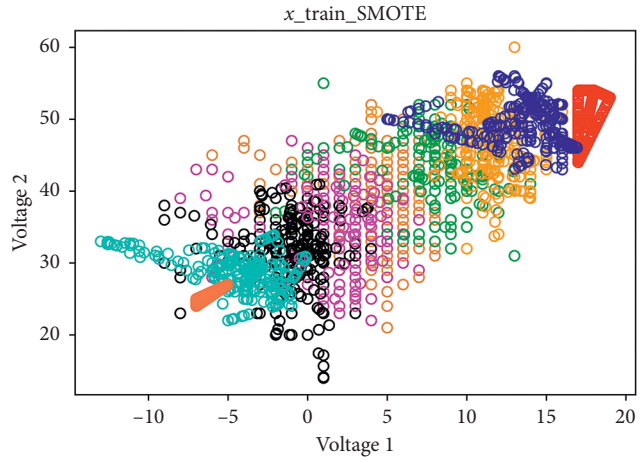

Figure 4: SMOTE synthetic data.

consumption voltage collected in the encryption process. In Figure 3, the horizontal and vertical coordinates are different voltage properties of the curve, and the circles with different colors represent different hamming weights. It can be seen from the figure that different power consumption curves of the same hamming weight have different voltage values at the same time. That is to say, one attribute contains a variety of situations, and the amount of data contained by different hamming weights is unbalanced.

As shown in Figure 3, when the random forest algorithm constructs the decision tree by randomly sampling samples due to the unbalance of data, some minority classifications may not be included in the samples sampled, thus reducing the prediction effect. The training set after data set enhancement is shown in Figure 4. The increase of data brings benefits to the random forest. When random data is extracted, the probability of each type of data being extracted into samples is greatly increased so that good prediction results can be made.

4. Characteristics of Data and Feature Point Selection

In this paper, we use Differential Power Analysis (DPA) contest v4 as the experimental target. The DPA international academic competition began in August 2008 and was jointly sponsored by the National Academy of Sciences of France and the Institute of Advanced Telecommunications of High-Tech Paris. Its official website is http://www.dpacontest.org. The DPA Contest v4 analyzed the mask-class AES-256 cryptographic algorithm on ATMega163 chips. A total of 100,000 power curves were collected, and each power curve contained 435,002 sampling points. Because encryption operations on smart cards last a very long time (AES is coded in C, not assembly language), these power consumption curves only include the first and second rounds of AES encryption. All the key, plaintext, offset, and mask used for the power consumption curves are known. Since this paper focuses on the study of unbalanced data, the mask and offset are taken as known values, and the hamming weight of s-box

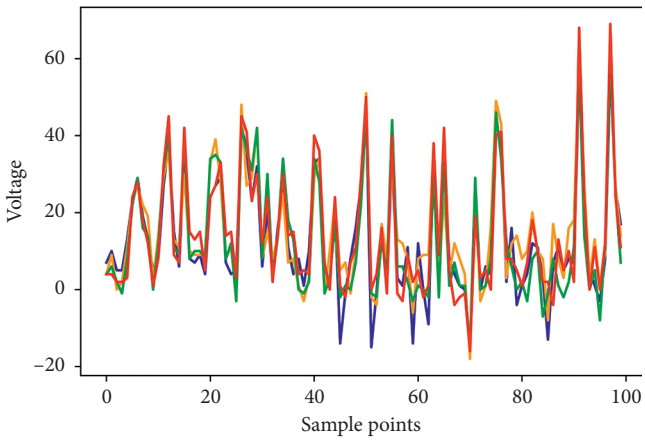

Figure 5: Power consumption data voltage diagram.

output value is used as the label to implement the first-order power consumption analysis attack based on random forest on these data.

4.1. Feature Point Selection. The mask and offset are taken as known values and the hamming weight of s-box output value is used as the label to implement the first-order power consumption analysis attack based on random forest on these data. Since a power consumption curve contains 435,002 sampling points, only a few points can represent the weight characteristics of s-box hamming and too many irrelevant redundant points will affect the performance of the machine learning algorithm. The common feature points are feature extraction and feature selection. The feature extraction refers to the generation of new feature sets based on the original features, such as Principal Component Analysis (PCA) [16], while the feature selection refers to the selection of subsets from the original data set, such as the selection of sampling points with the highest scores through correlation calculation.

The data used in this paper is the collected power consumption voltage, which amplifies the parts of any few bars, as shown in Figure 5. It can be seen from the figure that

the power consumption curve has obtained a relatively ideal alignment, which can be directly used for characteristic engineering processing.

The label selected in the experiment was the hamming weight of the s-box output value. Reference [17] points out that the power consumption voltage of the crypto chip is proportional to hamming weight, so Pearson's correlation coefficient is selected to extract feature points in this paper. Due to the large number of sampling points in the experiment, the extracted voltage forms a good linear relationship, so a good effect can be obtained by using a subset of the original voltage set.

Suppose there are N power curves, each of which has M points. The voltage at a sampling point on the power curve is denoted as $v_{(i,j)}$ $(i \in [1, N], j \in [1, M])$ and the label of each curve is H_j $(j \in [1, N])$. The correlation coefficient between voltage value and label is calculated according to Pearson's correlation coefficient formula. The absolute value of correlation coefficient is sorted from large to small and the voltage value corresponding to m points with the largest correlation coefficient is selected to form a new data matrix. The values of correlation coefficients between power consumption and hamming weight are calculated by using 800 power curves, as shown in Figure 6. The maximum value of correlation coefficients is at the sampling point of 101,589 and the maximum value is at 0.868736.

In the same way, the correlation coefficient between the second byte of the median value and the power consumption curve is calculated, as shown in Figure 7. As can be seen from the two figures, the feature points generate different bytes so that feature point selection is required before predicting each byte.

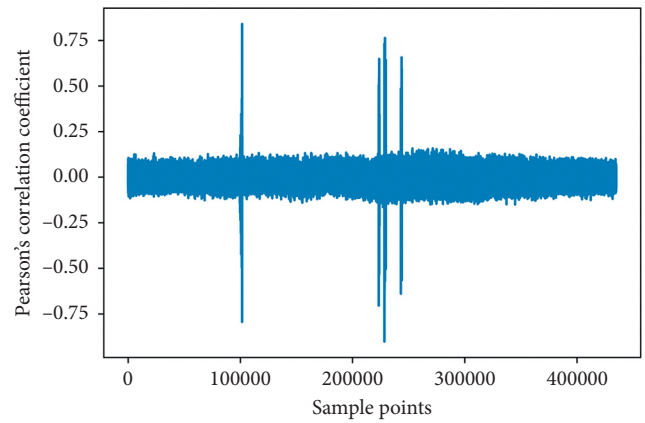

Figure 6: Pearson's correlation coefficient diagram.

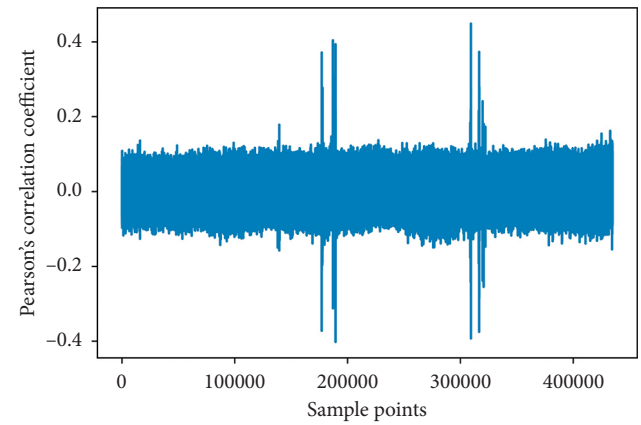

Figure 7: The second byte Pearson's correlation coefficient diagram.

5. Experimental Results and Analysis

In the classification task, error rate and accuracy are the evaluation criteria of model generalization ability [18].

The classification error rate of sample set D refers to the proportion of classification error samples to the whole sample, while the accuracy refers to the proportion of correct classification samples to the whole sample. The expression of error rate is shown as follows:

$$E(f; D) = \frac{1}{m} \sum_{i=1}^{m} II(f(x_i) \neq y_i). \tag{3}$$

The expression of accuracy is shown as follows:

$$acc(f; D) = \frac{1}{m} \sum_{i=1}^{m} II(f(x_i) = y_i) = 1 - E(f; D), \tag{4}$$

where f represents the machine learning algorithm and D represents the whole sample set marked by f. Accuracy and error rate reflect the quality of the algorithm model, but for unbalanced data, classification accuracy also needs to be paid attention to. Because the classifier is more inclined to classify the predicted categories into the majority classifications, it leads to low classification accuracy of the minority classification. Therefore, recall and precision are used to evaluate the performance of the model more comprehensively when precision is used as a performance measure.

For binary classification, recall and precision are calculated by using the confusion matrix.

The confusion matrix is shown in Table 1, where TP, FN, FP, and FN, respectively, represent the sample numbers of various types and the sum of the four represents the whole sample numbers.

The precision rate P is

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}. \tag{5}$$

The recall rate R is

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{6}$$

For multiclassification problems, the classification results include not only positive and negative classes. The experiment in this paper consists of nine classifications, namely, hamming weight H0-H8. We hope that the accuracy rate of each classification can be high, but not that of some classifications. All the nine classifications are regarded as positive classifications. The number of correctly classified samples are denoted as T0, T1, ... T8 and the sample number of misclassification is denoted as F01(indicating that H0 of the real sample is predicted as H1), F02 (indicating that H0 of the real sample is predicted as H2), ... F08, F12, ... F78.

Table 1: Confusion matrix.

Real value	Predictive value	
	Positive	Negative
Positive	TP	FN
Negative	FP	TN

Table 2: Multiple classification confusion matrix.

Real value	Predictive value								
	H0	H1	H2	H3	H4	H5	H6	H7	H8
H0	**T0**	F01	F02	F03	F04	F05	F06	F07	F08
H1	F10	**T1**	F12	F13	F14	F15	F16	F17	F18
H2	F20	F21	**T2**	F23	F24	F25	F26	F27	F28
H3	F30	F31	F32	**T3**	F34	F35	F36	F37	F38
H4	F40	F41	F42	F43	**T4**	F45	F46	F47	F48
H5	F50	F51	F52	F53	F54	**T5**	F56	F57	F58
H6	F60	F61	F62	F63	F64	F65	**T6**	F67	F68
H7	F70	F71	F72	F73	F74	F75	F76	**T7**	F78
H8	F80	F81	F82	F83	F84	F85	F86	F87	**T8**

Table 3: The number distribution of each class in the training set.

Pieces	Label								
	0	1	2	3	4	5	6	7	8
800	4	21	86	166	226	188	87	19	3
1000	5	28	107	209	279	238	105	25	4
1500	8	45	163	297	441	349	150	43	5

Then, each classification is evaluated using precision and recall. According to the experimental conditions, the confusion matrix was adjusted to Table 2.

We only care about the proportion of the predicted results that are correctly classified, but the classification of true values is predicted and leads to errors. The precision and recall rates of each category can be expressed as follows where $i \in [0, 8]$:

$$P_i = \frac{T_i}{T_i + \sum_{j=0, j\neq i}^{8} F_{ji}},$$
$$R_i = \frac{T_i}{T_i + \sum_{j=0, j\neq i}^{8} F_{ij}}. \quad (7)$$

800, 1000, and 1500 pieces of data are used as training sets, respectively. The number distribution of each class in the training set is shown in Table 3. It is obvious that the data are unbalanced.

Fixed 1000 pieces of data are used as the test set and the distribution of various types is shown in Table 4. The accuracy of the test set is shown in Table 5 without data enhancement.

As we can be seen from Table 3, random forest is affected by unbalanced data resulting in the accuracy rate of a few classifications of 0% and most categories of 92%. At the same time, it can be seen that the increase of data numbers can appropriately improve the accuracy and the time, but the accuracy rate of random forest minority data set is always 0%. So, we enhanced the training set. After processing, the data numbers are the same for each class. After the expansion of 800, 1000, and 1500 data, the number distribution of each class is shown in Table 6 and the test set is still 1000 pieces in Table 4.

The training set after data set enhancement uses random forest algorithm to predict the test set and the accuracy rates of each classification are shown in Table 7.

The accuracy and running time of all kinds of data after enhancement are represented by broken-line graph as shown in Figures 8 and 9, respectively.

It can be seen from Figure 8 that the accuracy of random forest is greatly improved after data enhancement. Compared with Tables 5 and 7, the curve accuracy rate of random forest minority classification increased from 0% to 100% and other tag categories also improved. As it can be seen from Figures 8 and 9, the running time of the random forest increases with the increase of the pieces of power curves.

In the case of 800 pieces of data, the time before and after data enhancement increased by 0.627s and the accuracy increased by 7%; in the case of 1000 pieces of data, the time increased by 0.86s and the accuracy increased by 7%; in the case of 1500 pieces of data, the running time increased by 1.695s and the accuracy increased by only 5%. Therefore, it can be concluded that data enhancement technology is more suitable for data collection limited to 1000 pieces.

Table 4: The test set number distribution.

Pieces	Label								
	0	1	2	3	4	5	6	7	8
1000	3	27	108	216	169	222	118	33	4

Table 5: The test accuracy of unbalanced data set.

Pieces		Label									Accuracy	Time(s)
		0	1	2	3	4	5	6	7	8		
800	P_i	0	77	83%	91%	92	87	75	50	0	84	1.233
	R_i	0	37	77%	92%	96	91	80	9	0		
1000	P_i	0	80	85%	92%	91	86	76	67	0	86	1.284
	R_i	0	37	81%	91%	94	95	76	18	0		
1500	P_i	0	81	85%	93%	92	90	86	82	0	88	1.415
	R_i	0	48	84%	88%	97	91	83	52	0		

Table 6: Data set number distribution after expansion.

Original pieces	Each class pieces after expansion	The whole pieces after expansion
800	226	2034
1000	279	2511
1500	441	3969

Table 7: The test accuracy after data enhancement.

Pieces		Label									Accuracy (%)	Time(s)
		0 (%)	1 (%)	2 (%)	3 (%)	4 (%)	5 (%)	6 (%)	7 (%)	8 (%)		
800	P_i	100	76	90	93	93	92	90	92	80	91	1.856
	R_i	67	78	84	91	92	91	89	79	50		
1000	P_i	100	91	91	95	93	94	92	93	75	93	2.144
	R_i	67	85	89	91	90	94	83	82	50		
1500	P_i	100	91	89	97	92	93	92	85	100	93	3.112
	R_i	67	81	88	92	93	90	89	82	50		

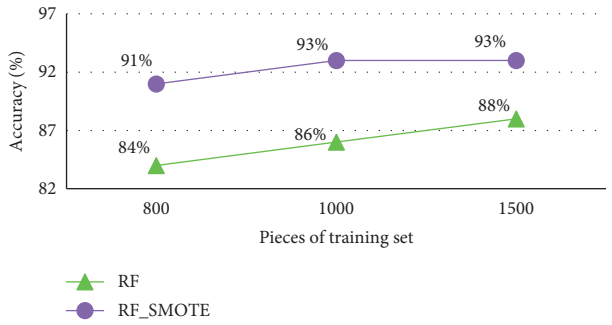

Figure 8: Comparison of accuracy before and after data enhancement.

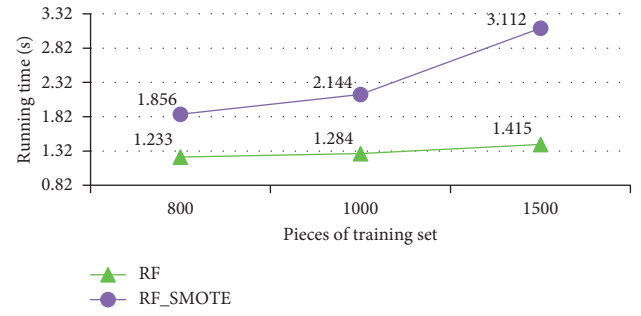

Figure 9: Running time comparison before and after data enhancement.

Like random forest, AdaBoost algorithm is also a kind of integrated algorithm, which is a typical iterative algorithm. The upper classifier modifies the weight of samples and changes the data distribution by predicting the accuracy of samples and judging the overall accuracy. The lower classifier uses the new sample set with adjusted weights as the training set and finally combines the training results of each time as the final result of the whole classification algorithm. To some extent, the AdaBoost algorithm can overcome the data imbalance. As it can be seen from Table 8 and Figure 10, this algorithm has high accuracy compared with RF, but its accuracy is lower than the modified random forest algorithm RF_SMOTE. Also, SVM and ANN have lower accuracy than RF_SMOTE.

The popular imbalance algorithm is ADASYN and SMOTENN. The SMOTENN is a combination of oversampling and undersampling (Edited Nearest Neighbor, ENN) algorithm. ADASYN oversampled different quantity

Table 8: Comparison of accuracy rates with other algorithms.

Pieces	SVM (%)	AdaBoost (%)	RF (%)	ANN (%)	RF_SMOTE (%)
800	88	87	84	85	91
1000	90	88	86	86	93
1500	91	89	88	87	93

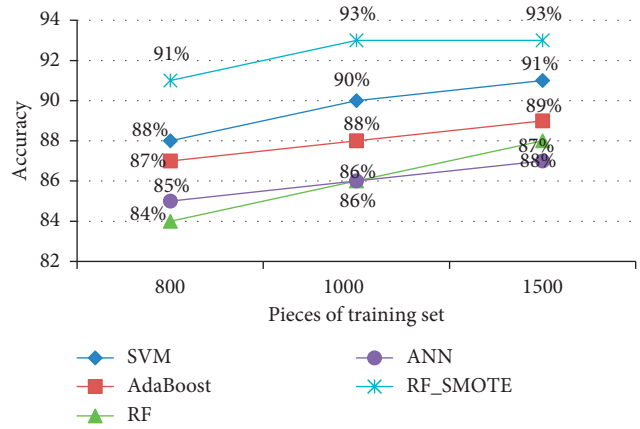

Figure 10: Comparison of accuracy with other algorithms.

Table 9: Comparison of accuracy with other data enhancement methods.

Pieces	Random Forest_ADASYN (%)	Random Forest_SMOTENN (%)	Random Forest_SMOTE (%)
800	79.8	74.6	91.2
1000	83.4	86.6	93.1
1500	84.3	87.7	93.6
2000	89.3	88.3	94.2

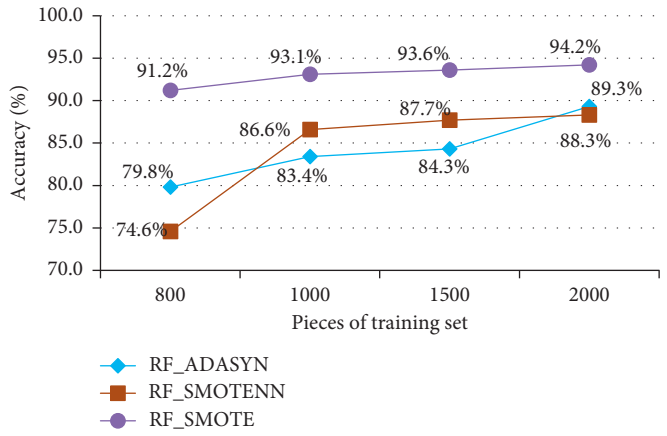

Figure 11: Comparison of accuracy rate with other data enhancement methods.

classifications by calculating the imbalance degree and obtained different sample numbers for each classification.

As can be seen from Table 9 and Figure 11, the combination of random forest with other techniques for oversaturation is not as accurate as the SMOTE method. Therefore, the combination of the two algorithms in this paper is better than the combination of other methods and random forest. The closer each data piece after enhancement, the better effect will be obtained after combining with random forest algorithm.

6. Summary

The random forest algorithm is affected by the unbalanced data that leads to low accuracy of minority classification in the energy analysis attack. To overcome this problem, this paper gives a method of random forest classification based on the Synthetic Minority Oversampling Technique (SMOTE). The method improves the classification accuracy by increasing the data diversity. Moreover, the confusion matrix of dichotomies is transformed into a

multiclassification confusion matrix, which can more directly reflect the classification accuracy of various classifications. Under the condition of unbalanced data, the RF algorithm is not correct in the classification of minority classification. But the experimental results show that the improved random forest algorithm can make up for the lack of RF and improve the accuracy of model prediction. The data enhancement techniques improve the accuracy of random forest tests but also take longer run time. Considering time and accuracy, this paper considers that when the training data are 800 pieces, the random forest algorithm combined with data enhancement technology will have the best performance. In the future work, we will continue to study methods to reduce the number of energy trace requirements and improve the attack efficiency.

Data Availability

The data used to support the findings of this study are from the DPA Contest v4 database, which is available at http://www.dpacontest.org/v4/42_traces.php.

Conflicts of Interest

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

Acknowledgments

This work has been funded by the Fundamental Research Funds for the Central Universities, China (Grant nos. 328201914 and 328201913).

References

[1] G. Hospodar, B. Gierlichs, E. De Mulder, I. Verbauwhede, and J. Vandewalle, "Machine learning in side-channel analysis: a first study," *Journal of Cryptographic Engineering*, vol. 1, no. 4, pp. 293–302, 2011.

[2] H. Hera, Josh, and Z. Long, "Side channel cryptanalysis using machine learning," 2012, http://cs229.stanford.edu/proj2012/HeJaffeZou-SideChannelCryptanalysisUsingMachineLearning.pdf.

[3] A. Heuser and M. Zohner, "Intelligent homicide," *Constructive Side-Channel Analysis and Secure Design*, pp. 249–264, Springer, Berlin, Germany, 2012.

[4] C. Whitnall and E. Oswald, "Robust profiling for DPA-Style Attacks," *Cryptographic Hardware and Embedded Systems-CHES 2015*, pp. 3–21, Springer, Berlin, Germany, 2015.

[5] J. Kim, S. Picek, A. Heuser, S. Bhasin, and A. Hanjalic, "Make some noise unleashing the power of convolutional neural networks for profiled side-channel analysis," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2019, no. 3, pp. 148–179, 2019.

[6] M. Carbone, C. Vincent, M.-A. Cornelie et al., "Deep learning to evaluate secure RSA implementations," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2019, no. 2, pp. 132–161, 2019.

[7] R. Geetha, S. Sivasubramanian, M. Kaliappan et al., "Cervical cancer identification with synthetic minority oversampling technique and PCA analysis using random forest classifier," *Journal of Medical Systems*, vol. 43, no. 9, 2019.

[8] R. Luo, Q. Feng, C. Wang et al., "Feature learning with a divergence-encouraging autoencoder for imbalanced data classification," *IEEE Access*, vol. 6, no. 6, pp. 70197–70211, 2018.

[9] Z. Cao, *Study on Optimization of Random Forest Algorithm*, Capital University of Economics and Business, Beijing, China, 2014.

[10] F. Pang, *Study on Application of Random Forest in Mass Appraisal of Second-Hand House*, Chongqing Jiaotong University, Chongqing, China, 2017.

[11] Y. Li, Z. Liu, and H. jun Zhang, "Review on ensemble algorithms for imbalanced data classification," *Application Research of Computers*, vol. 31, no. 5, pp. 1287–1291, 2014.

[12] K. D. Duncan and I. Lanekoff, "Oversampling to improve spatial resolution for liquid extraction mass spectrometry imaging," *Analytical Chemistry*, vol. 90, no. 4, pp. 2451–2455, 2018.

[13] M. Chen, *Study on Under-sampling and Unbalanced Ensemble Classification for Web Spam Detection*, Nanchang University, Nanchang, China, 2018.

[14] X. Tan and S. Tan, "Anomaly detection based on synthetic minority oversampling technique and deep belief network," *Journal of Computer Applications*, vol. 38, no. 7, pp. 1941–1945, 2018.

[15] X. Sun, J. Li, L. Gu et al., "Identifying the characteristics of the hypusination sites using SMOTE and SVM algorithm with feature selection," *Current Proteomics*, vol. 15, no. 2, pp. 111–118, 2018.

[16] Q. Zhao, "A review of principal component analysis," *Software Engineering*, vol. 19, no. 6, pp. 1–3, 2016.

[17] P. Luo, D. Feng, and Y. Zhou, "Power model in power analysis attack," *Journal on Communications*, vol. 33, no. 1, pp. 276–281, 2012.

[18] Z. Zhou, *Machine Learning*, pp. 29-30, Tsinghua press, Beijing, China, 2016.