Block Ciphers Classification Based on Randomness Test Statistic Value via LightGBM

Sijia Liu[1], Min Luo[1(✉)], Cong Peng[1], and Debiao He[1,2]

[1] Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan, China
{liusijia,mluo,cpeng}@whu.edu.cn

[2] Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center, Qilu University of Technology (Shandong Academy of Sciences), Jinan, China

Abstract. Cryptographic algorithms classification, which can detect the underlying encryption algorithm on sufficient large ciphertexts, is essential to encrypted traffic analysis and protocol compliance detection. Previous studies have typically employed various feature quantities and models for feature learning in analyzing encryption algorithms. Unlike these, this work performs a broader feature selection and extracts features from the P-values of the randomness test and their data distributions for different block cipher algorithms. This work utilizes the LightGBM framework to focus on block cipher algorithms classification in ECB mode. It takes six algorithms to test the classification scheme, including AES-128, AES-192, AES-256, DES, 3DES and SM4, with an average accuracy of 82%. To compare the accuracy, this work analyzes the influence weights of random features and experiments with the classification accuracy of different schemes on the same ciphertext blocks. The experiment results show that our scheme is effective in classifying block ciphers.

Keywords: Block cipher · Cryptographic algorithm classification · LightGBM · Randomness test statistic value

1 Introduction

With the high-frequency occurrence of security incidents, network security technologies, especially cryptographic technologies, have become an essential component of information systems. Specifically, encryption algorithms are extremely and widely used not only for transmission security [18], such as TLS and SSH protocols, but also for storage security [2], like XTS, etc. However, malicious adversaries may also use encryption to carry out attacks or hide their attacks. Although there will be relevant fields to mark the encrypted information, links

D. Wang et al. (Eds.): ICICS 2023, LNCS 14252, pp. 35–50, 2023.
https://doi.org/10.1007/978-981-99-7356-9_3

will be overlayed on the dark web data, making the cipher algorithms classification more difficult [15]. The indistinguishability of the ciphertext is a large obstacle to cryptanalysis.

In traffic transmission, data is encrypted by some security protocols. Classifying the encryption algorithms allows access to some transmission information and enables effective supervision of the network environment [17]. Moreover, there is no single authority in the blockchain system to manage all the fully open and autonomously managed data for decentralized applications (DApps). The classification of encryption algorithms plays an important role in identifying DApps and helps blockchain platforms manage user behavior [21]. Classifying cipher algorithm is significant for traffic analysis and network supervision.

Cryptosystems can be divided into two categories: symmetric-key algorithm and public-key algorithm. Public-key algorithms utilize different keys for encryption and decryption in which distinguishing curve parameters exist. Due to their high computational consumption, it is usually used with symmetric-key algorithms in practical applications. DES, AES, SM4 and other commonly used block cipher algorithms are widely applied in data encryption, message authentication and other information secure scenarios owing to their high encryption efficiency and convenient key management [25].

The indistinguishability of ciphertexts is a key obstacle to cryptanalysis in the security requirements of real scenarios. Effective cryptanalysis requires accurate encryption algorithms classification, enabling cryptanalysis to understand its structure, weaknesses, characteristics and encryption techniques. Analysts employ different methods for cryptanalysis, depending on the cryptographic regime [3]. These methods include cipher breaking, differential attacks, linear attacks and side-channel attacks [7]. Therefore, identifying the cryptographic regime provides a foundation for subsequent analysis.

Most previous studies on encryption algorithms classification performed randomness tests on ciphertext and learned the test pass rate at different randomness metrics as features. Random forest [13,29] and support vector machine classifiers [8] are mostly used to classify the selected features. However, since their accuracy can be raised and the analysis of the accuracy variation in the multi-key case is lacking, this work performs improved experiments. In this work, we take ten randomness test statistic values for the ciphertext and incorporate the statistical characteristics and data distributions into the feature set of the ciphertext. These methods effectively extract randomness features of ciphertext and provide important feature vectors for subsequent machine learning tasks. Later, we utilize the LightGBM framework for feature training and select six block cipher algorithms on three datasets to test the scheme.

Our contributions can be briefly summarized as follows:

- The proposed scheme extracts feature from ciphertext encrypted by multiple algorithms and forms a block cipher algorithms classifier consisting of feature extraction and classification.

- In this experiment, we select ten randomness test statistic values and data distributions of ciphertext as features of different encryption algorithms and analyze the importance of feature terms for classification.
- We experiment with this classification scheme on three datasets using six block cipher algorithms and compare it with previous studies, where the experimental result outperforms existing classification methods.

Rest of the Paper. The rest of the paper is organized as follows. Section 2 reviews research in recent years concerning block ciphers classification; Sect. 3 introduces the primary operating mode of cryptography algorithm and the technical foundation of ciphers classification; Sect. 4 presents the block ciphers classification scheme based on test statistic values of randomness proposed in this work; Sect. 5 shows the experimental result of the research and analyses the accuracy; Finally, Sect. 6 summarizes this research and looks ahead to our future work.

2 Related Work

As information technology and cryptanalysis develop, more researchers are using artificial intelligence techniques for cryptographic algorithms classification. Cryptanalysis involves recovering plaintext by extracting valid information with only the ciphertext known, and identifying cryptographic algorithms is an integral part of the cryptanalysis process. The resistance of an encryption algorithm to cryptanalysis is also a crucial indicator of its security. For the analysis of block ciphers, we focus on their characteristics, such as diffusion property and randomness, then extract and analyze their features using machine learning techniques.

To our knowledge, Dileep et al. [8] first attempted a support vector machine-based identification method for block cipher encryption algorithms to classify five encryption methods in ECB and CBC modes. This experiment significantly affected the recognition system's performance when the encrypted files were generated using different keys. Sharif et al. [20] used pattern recognition techniques and eight kinds of artificial intelligence classification ciphers to identify four block encryption algorithms in ECB mode. The experiment showed no significant classification accuracy difference between the multiple AES versions.

Based on previous studies, Huang et al. [13] proposed a random forest-based hierarchical identification scheme for cryptographic regimes. They first classified the three groups of cryptographic regimes to which ciphers belong in a coarse-grained way, then carried out a single classification of thirteen cryptographic algorithms in a fine-grained way. This experiment showed that the hierarchical recognition scheme performs better than the single-layer recognition scheme. However, it was still challenging to distinguish between different encryption algorithms that share the same structure.

As in the previous study, the random forest algorithm is often applied to cryptographic classification. Hu et al. [12] selected the random forest algorithm

to categorize eight encryption algorithms in both ECB and CBC modes. This experiment showed a significantly lower recognition accuracy in CBC mode compared to ECB mode. Zhao et al. [31] also used a random forest algorithm based on the NIST standard to extract ciphertext features and performed the two-differentiation experiment for six encryption algorithms. Additionally, the scheme could differentiate between the encrypted files' different block cipher modes.

Some studies used the random forest algorithm for comparison experiments. Zhang et al. [30] presented a feature extraction scheme by reducing the data preprocessing dimensionality based on ciphertext ASCII statistics. The researchers converted images into arrays, analyzed the encrypted data differences in ASCII code distributions, and classified eight encryption algorithms using two machine learning classifiers: random forest and support vector machine. The two classification algorithms used in this experiment showed significant differences in precision and recall, leading to unstable recognition results.

Some works utilized the NIST statistical test suite of random as a feature extraction method. Ke et al. [29] combined integrated learning and proposed an encryption algorithms identification scheme based on hybrid random forest and logistic regression models. This work selected the NIST random test features, encapsulating each feature in a sub-module so that they would be free from interference. Yu et al. [28] selected five of the NIST statistical test suite of random, extracted ciphertext features for four encryption algorithms in ECB and CBC modes. They used the MLP classifier for classification experiments, and it was found that the classification accuracy could reach 42% in ECB mode. In contrast, it was lower in CBC mode due to its higher randomness.

Several recent studies have focused on classifying encryption algorithms in CBC mode. Swapna et al. [23] proposed two block cipher recognition methods based on support vector machine for classifying five algorithms in CBC mode. They used the ciphertext and partially decrypted text extracted from the ciphertext with a heterogeneous association model as the input to the classifier. In addition, there was a study for single algorithm classification. Ji et al. [14] proposed the classification of SM4 cipher based on randomness features. They identified the state-secret SM4 algorithm with six other block cipher algorithms, analyzed the four randomness features of the algorithms, and then used the SVM and decision tree single algorithms for classification.

Wu et al. [26] applied statistical analysis based on the metrics of codeword frequency, intra-block frequency detection and runs frequency to classify the randomness of four block encryption algorithms. It was clustered and divided by the K-means algorithm. Tan et al. [24] conducted a study on multi-class recognition using genetic algorithms, graph theory, neural networks and clustering for five algorithms in CBC mode. They found high recognition was possible only when the ciphertext keys were identical.

3 Preliminaries

This section briefly reviews the ECB mode of block cipher, the randomness test statistic value and the LightGBM framework used in this work.

3.1 Electronic Codebook Mode

Electronic Cipherbook (ECB) is a typical operation mode of the block cipher. The plaintext requires an integer multiple of the block and the encryption can be computed in parallel. It is diffuse and computational errors in one block will not affect subsequent computations. Since the encryption of each block does not depend on each other, the adversary can replace any block with a previously intercepted block without being detected and can decrypt the message without knowing the key [10]. The encryption and decryption are shown in Fig. 1, the specific process is as follows:

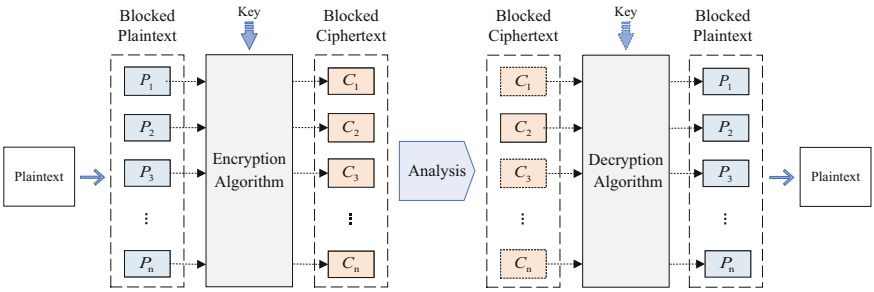

Fig. 1. Encryption and decryption in ECB mode.

- Encrypt the first plaintext block P_1 with the key to generate the first ciphertext block C_1.
- Use the key for the second plaintext block, generate the second ciphertext block, perform the same process, etc.
- Decrypt the first ciphertext block C_1 with the same key and restore the first plaintext block P_1.
- Use the key for the second ciphertext block, generate the second plaintext block, perform the same process, etc.

If the plaintext block P_m is encrypted twice using the same key in ECB mode, the output ciphertext block will be the same. We could create a corresponding cipher book for all plaintext blocks in a given key. Encryption will only require finding the ciphertext corresponding to the plaintext. The encryption and decryption parts of the ECB mode have no dependencies, so the input block of each iteration does not require feedback data from the previous iteration.

3.2 Randomness Test Statistic Value

The main evaluation methods of the randomness test statistic values selected in this paper are as follows:

- **Frequency Test.** The frequency test evaluates the ratio of 0 bits and 1 bits in data. The frequency test is the foundation of data randomness.
- **Runs Test.** The runs test assesses if the expected number of 1 bits sequences and 0 bits sequences of different lengths are identical. It uses the bit oscillation amplitude to reflect the randomness of the data.
- **Longest Run Test.** The longest run test evaluates the randomness of the test sequence by comparing the longest run length with the expected longest run length in a truly random sequence.
- **Cumulative Sums Test.** The cumulative sums test evaluates the randomness of 0 bits and 1 bits. In this test, 0 bits are converted to negative numbers. The test is based on the maximum random wander distance of 0, with larger distances indicating greater non-randomness of the test sequence.
- **Approximate Entropy Test.** The approximate entropy test compares the expected results of two overlapping blocks of adjacent lengths (i and $i + 1$) and the normal sequence.
- **Block Frequency Test.** The block frequency test evaluates whether the number of 0 bits and 1 bits in the m non-overlapping blocks created by the tested sequence is consistent with randomness.
- **Linear Complexity Test.** The linear complexity test is to divide the data into m non-overlapping modules. It checks the randomness of each module by examining the shortest length distribution of the linear feedback registers.
- **Serial Test.** The serial test evaluates whether the m-bit patterns of overlap are close to the expected number of occurrences in the random sequence.
- **Non-overlapping Template Matching Test.** The non-overlapping template matching test evaluates the occurrences of a given string in a sequence.
- **Fast Fourier Transform(FFT) Test.** The FFT is to convert a random data sequence into a frequency domain representation in order to examine its spectral characteristics and periodicity. For the input data sequence, the FFT converts it into a complex sequence of the same length and then calculates its frequency domain representation. The NIST statistical test suite for random using FFT for spectral testing, period testing and many other tasks [27].

The randomness test statistic values check whether the data has any recognizable patterns. Each statistic test is designed to check a hypothesis, and the statistical results of the experiment are evaluated to determine whether the hypothesis is valid [19]. The randomness of the ciphertext is an important factor in assessing the security of an encryption algorithm. In this work, we employ ten randomness test statistic values to analyze the ciphertext generated by various block ciphers.

In 2001, NIST published a standard for applying cryptographic random number tests, which can be applied to measure the deviation of binary sequences from

randomness. It comprises several tests to assess the system's randomness, including cryptographic algorithms, simulators and models [5]. Under the assumption of randomness, the statistic will satisfy a reference distribution, and the experiment will set a critical value (e.g., 99%). The test results are compared with the critical value. The hypothesis is considered valid if the test results are less than the critical value or invalid if the opposite is true.

In the procedure, we compute a P-value for each test, which represents the strength of the data randomness. If the P-value is calculated close to 1, the detected bit sequence is ideally random. A P-value of 0 means that it is entirely non-random. If the P-value is set to 0.01, indicating that the data is a random sequence with 99% confidence. A P-value less than 0.01 means that the bit sequence fails the check, and a confidence level of 99% indicates that the data is non-random.

3.3 LightGBM

LightGBM is a decision tree-based algorithm developed by Microsoft that is commonly used for classification and ranking tasks. LightGBM has many advantages of the XGBoost framework, including sparse optimization, parallel training, regularization, bagging, multiple loss functions and early stopping. In this work, we use this algorithm to learn the features of multiple block cipher algorithms in order to categorize them.

LightGBM differs from other algorithms in its tree construction by not using an algorithm that grows the tree line-by-line; instead, it selects the leaf that will generate the maximum loss reduction. Furthermore, LightGBM does not search for the optimal split point on the sorted eigenvalues like XGBoost or other algorithms; instead, it implements a histogram-based decision tree learning algorithm that offers significant efficiency and memory consumption advantages. Compared to SVM and RF, LightGBM has superior robustness in managing large data instances [22]. Additionally, LightGBM is faster and more accurate than CatBoost and XGBoost in certain classifications, particularly for ranking and feature selection with different numbers of characteristics [1].

The LightGBM framework has the following advantages:

- Employ splitting and tree-based techniques significantly reducing training time.
- Enable processing of large amounts of data in limited memory space.
- Have a built-in feature selection function to reduce the noise in the training data and improve the scheme's accuracy.
- Support parallel computation, resulting in faster training tasks.

The LightGBM framework utilizes two new techniques, gradient-based one-sided sampling (GOSS) and exclusive feature bundling (EFB), to accelerate model training by reducing the number of samples and to further reduce the features number to make the data size smaller, enabling the algorithm to perform both better in terms of accuracy and runtime [4].

Algorithm 1: Gradient-based One-Side Sampling

Input: D: training data; *iter_num*: iteration number
lgd: sampling ratio of large gradient data
sgd: sampling ratio of small gradient data
loss: loss function, L: weak learner
Output: newModel: optimized model
models ← {}, fact ← $\frac{1-lgd}{sgd}$
topN ← lgd× len(D), randN ← sgd× len(D)
for $i = 1 \rightarrow iter\_num$ **do**
  preds ← models.predict(D)
  loss_array ← loss(D, preds), w ← {1,1,...}
  sorted_array ← GetSortedIndexes(abs(loss_array))
  lagreSet ← sorted_array[1:topN]
  randSet ← RandomPick(sorted_array[topN:len(D)], randN)
  newSet ← lagreSet + randSet
  w[randSet] × = fact
  newModel ← L(D[newSet],loss_array[newSet],w[newSet])
  models.append(newModel)

The basic idea of the GOSS algorithm is to rank the training data according to the gradient first, set a ratio, and retain the samples with a gradient higher than this ratio. Instead of directly throwing away the samples with gradients below this percentage, a certain percentage of them are taken for training. The GOSS algorithm computes the information gained by scaling up the dataset with smaller gradients, which can counteract the effect on the sample distributions. The exact algorithm is shown in Algorithm 1 [16].

GOSS greatly reduces the computational effort by estimating gains on smaller sample datasets without excessively reducing training accuracy. In addition, the LightGBM framework utilizes the EFB technique to reduce the number of variables. It improves the operational efficiency of the algorithm by linking features that are not mutually exclusive, and ranking features according to the degree of fixed points reflecting feature conflicts, and setting a maximum conflict threshold to merge features.

4 Block Ciphers Classification

The classification of block ciphers can be divided into two main parts. Use machine learning techniques to learn these features and then classify the block ciphers.

4.1 Feature Selection and Extraction

As we learned from the introduction in Sect. 3.2, the randomness test statistic is typically used to verify the random number generators of cryptographic applications. This study utilizes the test statistic values to analyze the randomness

differences among various block cipher algorithms by applying the suite to the ciphertexts.

Block cipher algorithms provide data confidentiality, integrity and authenticity, making them key tools for secure communication and data protection, so it is widely used in various fields, such as network work security, data storage and financial transactions. The ciphertext generated by block cipher algorithms exhibits high levels of randomness, making it challenging to select and extract features for classification manually. Due to the high randomness of ciphertexts generated by block cipher algorithms, it is challenging to select and extract features for classification manually. Therefore, we choose distinguishable test statistic-valued features to extract features from various block cipher algorithms in this study.

We utilize two features to represent this metric as the cumulative sums test is evaluated in forward and backward modes. Similarly, the serial test results generate two sets of features to express their property. In addition, we performed statistical analysis on the results of ten tests, describing the distribution of each data in terms of its concentration trend, degree of dispersion and shape. We select effective and distinguishable features from them to expand the feature set of the algorithm.

4.2 Classification Scheme Based on LightGBM

For a set Enc of cryptographic algorithms with n block cipher algorithms, let $Enc = \{e_1, e_2, ..., e_n\}$. When a block cipher algorithm e_i is given, a cryptographic classification scheme is used to determine which of the Enc algorithms it belongs to in the absence of other information. The workflow of the cryptographic recognition scheme is shown in Fig. 2.

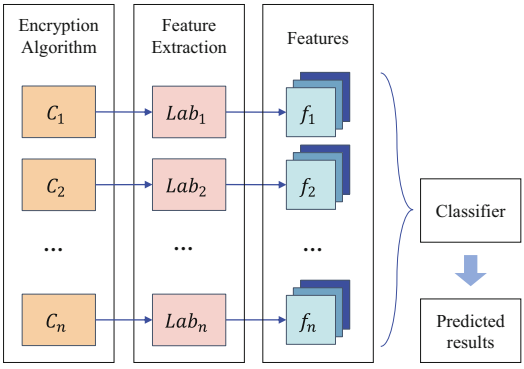

Fig. 2. Encryption algorithms classification flow chart

We normally consider that a cryptographic recognition scheme consists of a combination of algorithms to be classified, a collection of feature and a recognition algorithm. we express as $C = \{Enc, Fea, Alg\}$, where C denotes the work's

Algorithm 2: Construction of block ciphers classification

Input: Encryption algorithm set $Enc = \{e_1, e_2, ..., e_n\}$
Output: Block ciphers classification scheme $C = \{Enc, Fea, Alg\}$
Initialize the test statistic values Val and features Fea as empty sets
Val, $Fea \leftarrow \{\}$
Generate the ciphertext set
$Cip = \{c_1, c_2, ..., c_n\} \leftarrow$ plaintext.$Enc()$
for $c_i \in Cip$ **do**
  Group c_i in equal lengths
  $j \leftarrow 1$
  while ! *end_test* **do**
    $Val_i \leftarrow$ get_test_sta_val($c_{i,j}$)
  $f_i \leftarrow$ get_fea(Val_i)
  Add the f_i in Fea
Disrupt the data and split the training and test set
tra_data $\leftarrow$ 4/5 of data, *test_data* $\leftarrow$ 1/5 of data
Transform the training and test datasets sequentially
Optimize parameters for Alg with Fea
return Block ciphers classification scheme

proposed block cipher algorithm recognition scheme. Enc denotes the set of cryptographic algorithms to be classified. Fea is the set of features corresponding to the encryption algorithm in Enc,denoted as $Fea = \{f_1, f_2, ..., f_n\}$. Alg is the selected machine learning-based classification method. The block ciphers classification scheme's construction is described in Algorithm 2.

We adopt a 5-fold cross-validation strategy in the training process, which means dividing the dataset into five subsets, training and testing the model on different combinations of these subsets. The number of training examples is increased to optimize the model, while the test set is used to evaluate the model's performance and deviation values. Furthermore, cross-validation ensures that all data are involved in the model's training and prediction processes, effectively mitigating overfitting and enhancing the model's accuracy.

The 5-fold cross-validation method is implemented as follows:

- Divide the data into five equal-sized groups.
- Take 4/5 of the data to train and the remaining 1/5 to test.
- Repeat the whole process five times.

The accuracy of the experiments is calculated as the average of the training results of each partition. All classifiers are trained using the same training set, and the related metrics are measured using the same test set.

5 Experimental Results

5.1 Evaluation Metrics

We choose the following metrics to evaluate the classification scheme.

- TP (True Positive): the number of positive cases is predicted to be positive.
- FP (False Positive): the number of actual negative cases that are predicted to be positive.
- TN (True Negative): the number of negative cases predicted to be negative.
- FN (False Negative): the number of positive cases predicted to be negative.

Precision is a metric used to evaluate the performance of a classifier, which represents the ratio of the correctly classified samples to the total number of samples. The precision score ranges between 0 and 1, with a value closer to 1 indicating that the model's prediction results are more accurate. Precision is calculated using the following formula:

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

The recall is the percentage of correctly classified positive samples to total positive samples. It measures the proportion of true positive cases correctly identified by the model. Recall has a value range between 0 and 1, where a higher value indicates that the model can identify positive cases correctly. The recall is calculated using the following formula:

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

The F1-Score is a metric that combines precision and recall. It is more informative when there is a large imbalance between the two. It ranges from 0 to 1, with a higher value indicating better model performance. In cases of an uneven distribution of positive and negative samples, the F1-Score can consider both precision and recall. The formula for calculating the F1-Score is:

$$F1 - Score = \frac{2 * (Precision * Recall)}{Precision + Recall} \tag{3}$$

Our work employed a 5-fold cross-validation strategy to calculate the classification accuracy record, average precision, recall and F1-Score values.

5.2 Experimental Results and Analysis

We utilize three types of datasets, including the Caltech-256 image dataset [11], the Caltech Resident-Intruder Mouse dataset (CRIM13) [6] and VPN-nonVPN dataset (ISCXVPN2016) [9]. Caltech-256 includes 256 categories of images with over 80 items per category, the common image dataset. It downloads from Google Images and manually filters those images that do not match the categories. CRIM13 contains 474 videos from pairs of mice performing social behaviors, with 88 h and 8 million video frames, a commonly used video dataset. ISCXVPN2016 contains 14 types of traffic data from VPN and regular scenarios, commonly used as a traffic dataset.

After collecting the dataset, we divide it into a number of files of about 1GB. Then, we use six block ciphers, AES-128, AES-192, AES-256, DES, 3DES

Table 1. Configurations of the algorithms

Settings	Algorithm	Key length	Key
Same key	AES-128	128	1234567812345678
	AES-192	192	123456781234567812345678
	AES-256	256	12345678123456781234567812345678
	DES	64	12345678
	3DES	128	1234567812345678
	SM4	128	1234567812345678
Different keys	AES-128	128	hd3UHq5UJpECzkNR
	AES-192	192	HJpEC1k9m5UyHdGcf23gvf4I
	AES-256	256	YQfGZJ16P8PfurygtPbKUUZRYepVAnTZ
	DES	64	wfrL7Ipm
	3DES	128	3m0pE9bMq7bAv2zU
	SM4	128	BKEhx9f6bzvFhZcH

and SM4 to encrypt these files. Table 1 shows the relevant configurations of the six block ciphers we used. We investigate the key's influence on the classification of encryption algorithms by setting the same and different keys for the six algorithms in our experiment. When setting different keys, we generate random strings as keys for encryption algorithms. When setting the same key, we set its repeat unit to the same value since the encryption algorithms have different key lengths.

In this work, we conduct 10 experiments on the above dataset separately and set the feature data extracted from the ciphertext to 1024×6 items each time. In the cross-validation, 4915 items were selected as the training data and 1229 items as the validation data. The accuracy of the block ciphers classification scheme based on Sect. 4 is verified experimentally.

When the classification scheme is tested on different datasets, the classification accuracy is shown in Fig. 3. The identifiers 'dk' and 'sk' of the dataset mean that the encryption algorithm works with different or similar keys. It can be seen from the figure that the classification accuracy of encryption algorithms is a little higher when using the same key than using different keys. We consider that the key's influence on the classification of encryption algorithms is diminished when using the same key, making it more accurate. However, in practical scenarios, different encryption algorithms generally use different keys, so we are more interested in classifying encryption algorithms under different keys.

Cross-validation is a method helping to reduce unstable training results that may arise from using a portion of the data as a validation set to evaluate the model. It is more reliable than a single evaluation as it considers the distribution of multiple data points. To improve the performance of our classification scheme, we employ cross-validation to ensure that each data point is used for training and testing to evaluate its stability and generalization performance.

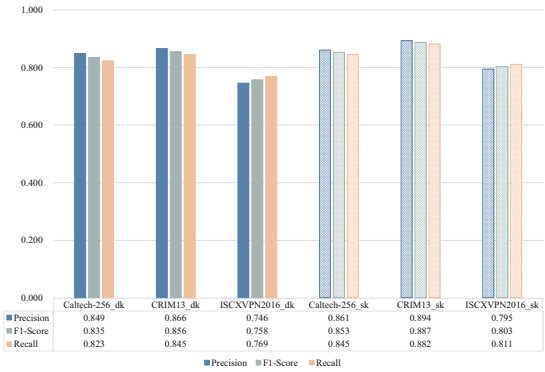

Fig. 3. Classification accuracies on different datasets

Besides, the above figure shows that the classification accuracy on the ISCXVPN2016 dataset is slightly lower than others. We analyze that because there are encrypted processes inside the traffic data, inner encryption features might cause some interference with the analysis of the outer encryption algorithm. The classification scheme is validated using six encryption algorithms in ECB mode with different keys, which achieve an average classification accuracy of 82%.

Table 2 compares the average accuracy achieved in this work with existing studies on the classification of block cipher algorithms. An experiment includes the symmetric-key and public-key algorithms [13], so we do not classify the work mode of the experiment's encryption algorithms. Notably, our work achieves the highest accuracy among the comparative studies. We conduct comparative experiments for the LightGBM-based classification scheme in this paper with the K-Nearest Neighbors model. Figure 4 shows the classification accuracy of the two algorithms in one independent experiment on the Caltech-256 dataset.

Table 2. Classification accuracies of algorithms

Sources	Classification objects	Mode	Accuracy
[8]	DES/3DES/Blowfish/AES/RC5	ECB/CBC	0.41/0.35
[20]	DES/IDEA/AES/RC2	ECB	0.53
[13]	Substitution/Permutation/Trivium/Sosemanuk/ grain/RC4/AES/Camellia/DES/SM4/RSA/ECC	-	0.21
[29]	AES/3DES/Blowfish/CAST/RC2	ECB	0.73
[28]	DES/3DES/AES/Blowfish	ECB/CBC	0.42/0.30
This Work	AES-128/AES-192/AES-256/DES/3DES/SM4	ECB	**0.82**

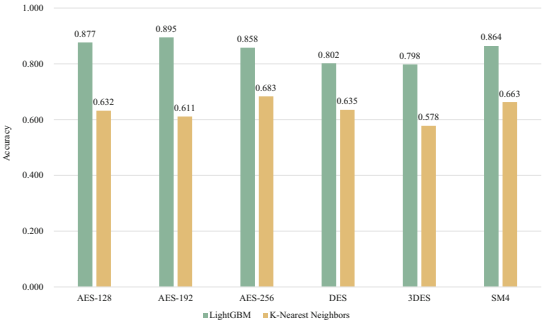

Fig. 4. Classification accuracies comparison of LightGBM and KNN

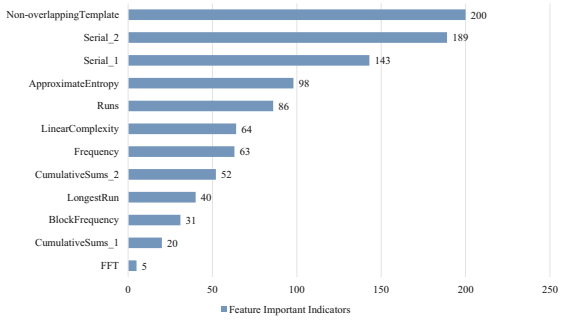

Fig. 5. The importance of features for classification

Figure 5 shows the features' importance for classifying encryption algorithms in this experiment. It can be seen from the figure that the non-overlapping and serial statistic values of different algorithms have more significant distinctions in their test of randomness characteristics. The LightGBM framework provides a built-in feature importance calculation function. It evaluates the importance of a feature by considering the number of times it is split in the decision tree and the gain achieved by these splits. The higher the score, the more significant the feature is in the classification process.

6 Conclusion

This work proposes a block ciphers classification scheme based on randomness test statistic value. We take the randomness test statistic values and their distributions of ciphertext as features of encryption algorithms and classify them via LightGBM. We experiment with this classification scheme on three datasets using six block cipher algorithms, AES-128, AES-192, AES-256, DES, 3DES and SM4. The classification accuracy reaches 82% when the algorithms are encrypted with different keys. The experiment results show that differences in keys and

datasets somewhat impact the classification accuracy. The classification accuracy in this work is significantly higher than random classification and above other classification schemes.

In future research, we would analyze the characteristics of block ciphers in CBC mode for encryption algorithm classification under multiple working modes. In addition, we would like to design an effective classifier to extract and classify multi-system encryption algorithms.

Acknowledgments. This research was supported by the Key Research and Development Program Project of Shandong Province under grants No. 2020CXGC010115.

References

1. Al Daoud, E.: Comparison between XGBoost, LightGBM and CatBoost using a home credit dataset. Int. J. Comput. Inf. Eng. **13**(1), 6–10 (2019)
2. Benadjila, R., Khati, L., Vergnaud, D.: Secure storage-confidentiality and authentication. Comput. Sci. Rev. **44**, 100465 (2022)
3. Benamira, A., Gerault, D., Peyrin, T., Tan, Q.Q.: A deeper look at machine learning-based cryptanalysis. In: Canteaut, A., Standaert, F.-X. (eds.) EUROCRYPT 2021. LNCS, vol. 12696, pp. 805–835. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77870-5_28
4. Bentéjac, C., Csörgő, A., Martínez-Muñoz, G.: A comparative analysis of gradient boosting algorithms. Artif. Intell. Rev. **54**, 1937–1967 (2021)
5. Bogos, C.E., Mocanu, R., Simion, E.: A remark on NIST SP 800–22 serial test. Cryptology ePrint Archive (2022)
6. Burgos-Artizzu, X.P., Dollár, P., Lin, D., Anderson, D.J., Perona, P.: Social behavior recognition in continuous video. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1322–1329. IEEE (2012)
7. Devi, M., Majumder, A.: Side-channel attack in internet of things: a survey. In: Mandal, J.K., Mukhopadhyay, S., Roy, A. (eds.) Applications of Internet of Things. LNNS, vol. 137, pp. 213–222. Springer, Singapore (2021). https://doi.org/10.1007/978-981-15-6198-6_20
8. Dileep, A.D., Sekhar, C.C.: Identification of block ciphers using support vector machines. In: The 2006 IEEE International Joint Conference on Neural Network Proceedings, pp. 2696–2701. IEEE (2006)
9. Draper-Gil, G., Lashkari, A.H., Mamun, M.S.I., Ghorbani, A.A.: Characterization of encrypted and VPN traffic using time-related. In: Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP), pp. 407–414 (2016)
10. Elashry, I.F., Allah, O., Abbas, A.M., El-Rabaie, S.: A new diffusion mechanism for data encryption in the ECB mode. In: IEEE (2010)
11. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset (2007)
12. Hu, X., Zhao, Y.: Block ciphers classification based on random forest. In: Journal of Physics: Conference Series, vol. 1168, p. 032015. IOP Publishing (2019)
13. Huang, L., Zhao, Z., Zhao, Y.: A two-stage cryptosystem recognition scheme based on random forest. Chin. J. Comput. **41**(2), 382–399 (2018)
14. Ji, W., Li, Y., Qin, B.: Identification of SM4 block cipher system based on random features. Appl. Res. Comput. (2021)

15. Kaur, S., Randhawa, S.: Dark web: a web of crimes. Wirel. Pers. Commun. **112**, 2131–2158 (2020)
16. Ke, G., et al.: LightGBM: a highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
17. Liu, C., He, L., Xiong, G., Cao, Z., Li, Z.: FS-Net: a flow sequence network for encrypted traffic classification. In: IEEE INFOCOM 2019-IEEE Conference On Computer Communications, pp. 1171–1179. IEEE (2019)
18. Manfredi, S., Ranise, S., Sciarretta, G.: Lost in TLS? no more! assisted deployment of secure TLS configurations. In: Foley, S.N. (ed.) DBSec 2019. LNCS, vol. 11559, pp. 201–220. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-22479-0_11
19. Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E.: A statistical test suite for random and pseudorandom number generators for cryptographic applications. Technical report, Booz-allen and hamilton inc mclean va (2001)
20. Sharif, S.O., Kuncheva, L., Mansoor, S.: Classifying encryption algorithms using pattern recognition techniques. In: 2010 IEEE International Conference on Information Theory and Information Security, pp. 1168–1172. IEEE (2010)
21. Shen, M., Zhang, J., Zhu, L., Xu, K., Du, X.: Accurate decentralized application identification via encrypted traffic analysis using graph neural networks. IEEE Trans. Inf. Forensics Secur. **16**, 2367–2380 (2021)
22. Sun, X., Liu, M., Sima, Z.: A novel cryptocurrency price trend forecasting model based on LightGBM. Financ. Res. Lett. **32**, 101084 (2020)
23. Swapna, S., Dileep, A., Sekhar, C.C., Kant, S.: Block cipher identification using support vector classification and regression. J. Discr. Math. Sci. Crypt. **13**(4), 305–318 (2010)
24. Tan, C., Deng, X., Zhang, L.: Identification of block ciphers under CBC mode. Procedia Comput. Sci. **131**, 65–71 (2018)
25. Usmonov, M.: Fundamentals of Symmetric Cryptosystem. Scienceweb Academic Papers Collection (2021)
26. Wu, Y., Wang, T., Xing, M., Li, J.: Block ciphers identification scheme based on the distribution character of randomness test values of ciphertext. J. Commun. **36**(4), 146–155 (2015)
27. Xing, M., Wu, Y., Wang, T., Li, J.: Identification of encrypted bit stream based on runs test and fast Fourier transform. Comput. Sci. **42**(1), 164–169 (2015)
28. Yu, X., Shi, K.: Block ciphers identification scheme based on randomness test. In: 6th International Workshop on Advanced Algorithms and Control Engineering (IWAACE 2022), vol. 12350, pp. 375–380. SPIE (2022)
29. Yuan, K., Huang, Y., Li, J., Jia, C., Yu, D.: A block cipher algorithm identification scheme based on hybrid random forest and logistic regression model. In: Neural Processing Letters, pp. 1–19 (2022)
30. Zhang, W., Zhao, Y., Fan, S.: Cryptosystem identification scheme based on ascii code statistics. Secur. Commun. Netw. **2020**, 1–10 (2020)
31. Zhao, Z., Zhao, Y., Liu, F.: Scheme of block ciphers recognition based on randomness test. J. Cryptol. Res. **6**(2), 177–190 (2018)