Original Research Article:
full paper

(2024), «EUREKA: Physics and Engineering»
Number 5

ANALYSIS OF AN ENHANCED RANDOM FOREST ALGORITHM FOR IDENTIFYING ENCRYPTED NETWORK TRAFFIC

Xiaoqing Yang
Research Unit for Electrical and Computer Engineering Technology (RECENT)
Faculty of Engineering[1]

Niwat Angkawisittpan✉
Research Unit for Electrical and Computer Engineering Technology (RECENT)
Faculty of Engineering[1]
niwat.a@msu.ac.th

Xinyue Feng
School of Electronic Information[2]

[1]*Mahasarakham University*
Kantarawichai, Maha Sarakham, Thailand, 44150

[2]*Foshan Polytechnic*
3 Zhijiao, Foshan, China, 528137

✉**Corresponding author**

Abstract

The focus of this paper is to apply an improved machine learning algorithm to realize the efficient and reliable identification and classification of network communication encrypted traffic, and to solve the challenges faced by traditional algorithms in analyzing encrypted traffic after adding encryption protocols. In this study, an enhanced random forest (ERF) algorithm is introduced to optimize the accuracy and efficiency of the identification and classification of encrypted network traffic. Compared with traditional methods, it aims to improve the identification ability of encrypted traffic and fill the knowledge gap in this field. Using the publicly available datasets and preprocessing the original PCAP format packets, the optimal combination of the relevant parameters of the tree was determined by grid search cross-validation, and the experimental results were evaluated in terms of performance using accuracy, precision, recall and F1 score, which showed that the average precision was more than 98 %, and that compared with the traditional algorithm, the error rate of the traffic test set was reduced, and the data of each performance evaluation index were better, which It shows that the advantages of the improved algorithm are obvious. In the experiment, the enhanced random forest and traditional random forest models were trained and tested on a series of data sets and the corresponding test errors were listed as the basis for judging the model quality. The experimental results show that the enhanced algorithm has good competitiveness. These findings have implications for cybersecurity professionals, researchers, and organizations, providing a practical solution to enhance threat detection and data privacy in the face of evolving encryption technologies. This study provides valuable insights for practitioners and decision-makers in the cybersecurity field.

Keywords: enhanced random forest, encrypted network traffic, traffic classification, identifying granularity.

DOI: 10.21303/2461-4262.2024.003372

1. Introduction

With the rapid development of computer technology and the comprehensive coverage of internet infrastructure, the scale of internet users has steadily increased. The rapid popularization of the Internet has driven the digital transformation of various industries and the vigorous development of internet applications, injecting new impetus into the country's economic development. However, with the rapid growth of network users and data traffic, servers face greater burdens and network security risks [1]. Effective management of massive network traffic is a key measure to ensure the efficient and stable operation of network infrastructure. Identifying encrypted traffic is crucial for network security as it enables prompt detection of potential attacks and protects sensitive data from leaks or malicious acquisition.

Encryption traffic recognition is mainly used in the field of network security. Classifying network traffic is essential for network management and security. It helps in identifying and prioritizing different types of traffic to optimize network performance, allocate resources, and detect potential security threats. Traditional traffic recognition and classification techniques have ideal classification results for non-encrypted traffic, but it is difficult to classify encrypted traffic [2]. This is because traditional traffic recognition and classification techniques mainly identify and classify network traffic based on plaintext data, such as port numbers, payloads, packet sizes, and protocols in network traffic during the transmission process, encrypted traffic uses encryption protocols to encrypt data, making traditional traffic recognition techniques unable to analyze the content of data packets and unable to accurately identify and classify encrypted traffic. With the increasing importance of personal privacy and data security, encryption protocols are also widely used, such as secure socket layer, transport layer security protocol, secure shell protocol, virtual private network, and other encryption protocols, the proportion of encrypted traffic in network traffic is constantly increasing, which not only protects personal privacy and data security, but also brings security risks. More and more network attacks use various encryption and obfuscation technologies to mask their true network traffic characteristics, making it difficult for traditional traffic recognition and classification techniques to correctly identify and classify them, in order to achieve the goals of stealing information and disrupting network security [3]. Therefore, identifying and classifying encrypted traffic has become an important challenge in the field of network management and security.

In the above context, this article applies the enhanced random forest (ERF) algorithm to the field of encrypted traffic recognition and classification, and identifies and classifies encrypted traffic without damaging the privacy of encrypted traffic data. Compared with traditional traffic recognition and classification technologies, encrypted traffic recognition and classification technology based on deep learning can automatically learn features from large-scale datasets, reducing the need for manual intervention while improving the efficiency and effectiveness of recognition and classification. It can also be quickly updated and optimized by retraining deep learning classification models, better adapting to complex network environments. This technology can also be applied to other fields, such as the Internet of Things, blockchain, etc. Therefore, applying deep learning to the field of encrypted traffic recognition and classification is of great significance.

2. Materials and methods

2. 1. Random forest concept

As an excellent classifier model, Random Forest was proposed by Leo Breiman mathematicians in the early 20th century, and its innovation lies in the effective fusion of ensemble classifiers and random subspaces. The basic model of a random forest is a decision tree, which is a strong classifier formed by concatenating decision trees obtained through the Bagging method [4]. The samples to be classified are judged using the voting method of each tree, as shown in **Fig. 1**.

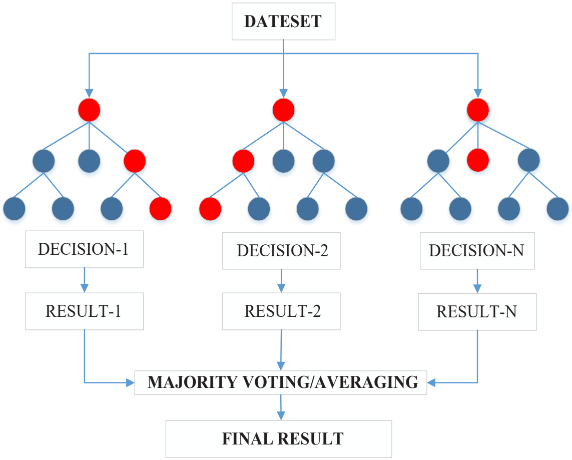

Fig. 1. The structure of random forest model

The biggest difference between random forests and traditional classifiers is that by introducing data in a random manner, the prediction accuracy of data classification is greatly improved without increasing computational complexity. Moreover, random forests can handle high latitude data, which is not prone to overfitting and has strong resistance to abnormal noise. In addition, random forests do not need to record prior probability information of classification samples, but rather learn from given samples according to training rules [5]. Due to its high stability and accuracy, the algorithm has been widely applied in fields such as medical processing, business modeling, financial economics, and data mining, and has achieved significant results. Compared with random forests in solving practical problems, there is less theoretical analysis and research, and there are still shortcomings. In the field of high-dimensional data processing, random forest algorithms are becoming a new research direction, and the rise of machine learning is also of great significance for its development and in-depth expansion research.

2. 2. Random forest expression

Definition 1: random forest is a model established using a large number of decision trees. Among them, *W* represents a random vector that follows independent and identically distributed distribution, *K* is the number of decision trees, and each tree is the known variable *x* for optimal voting. *N* represents the total number of samples, the objects in *X* have *M*-dimensional feature vectors, and *Y* includes *F* different categories of information [6].

In order to reduce the correlation between trees, the hyperparameters can be optimized to increase tree diversity and reduce overfitting. Commonly used methods include: increasing the number of trees, which makes the model more stable and reduces the effect of randomness in a single tree; limiting the depth of the tree to prevent the tree from being too complex, thus reducing the correlation between the trees; increasing the minimum number of sample splits and the minimum number of sample leaf nodes to control the complexity of the tree and to make each tree simpler and more independent; decreasing the maximum number of features, which increases the diversity among trees and reduces the total number of features by limiting the use of features per tree.

For example, the number of trees can be set to 50, 150, 250, 350, the maximum depth of the tree can be set to 5, 15, 25, and None, the minimum number of sample splits can be set to 4, 6, and 8, the minimum number of sample leaf nodes can be set to 1, 3, and 5, and the maximum number of features can be set to auto, sqrt, and log2. These parameter combinations will be evaluated by grid search cross-validation to get the optimal hyperparameters configuration, optimal cross-validation score and test set accuracy.

The classifier excessively emphasizes the classification of training samples, resulting in poor prediction of test samples, which is called overfitting. In the process of model construction, various error analyses will be introduced, and for random forests, generalization error is a key point in expressing overfitting problems. The generalization error is described by the edge function, as follows:

$$\mathrm{mg}(X,Y)=\alpha\mathrm{v}_k I\left(h_k(X)=Y\right)-\max_{j\neq Y}\alpha\mathrm{v}_k I\left(h_k(X)=j\right). \tag{1}$$

Among them, *I* represents the indicator function, and $\alpha\mathrm{v}_k$ represents the average value. This function expresses the difference between the average number of correct votes obtained by correctly classifying the random vector *X* into *Y* and the average number of votes obtained by other categories. The larger the function value, the higher the accuracy of the classification. The definition of generalization error is as follows:

$$PE=P_{X,Y}\left(mg(X,Y)<0\right). \tag{2}$$

Among them, the subscripts *X* and *Y* represent the generalization error obtained in the random variables *X* and *Y*.

By solving the expected value of the edge function and inferring the Chebyshev inequality, an upper bound for the generalization error can be obtained:

$$PE\leq\frac{\bar{\rho}\left(1-s^2\right)}{s^2}. \tag{3}$$

Among them, $\bar{\rho}$ represents the mean correlation factor between trees, and s represents the performance strength of the classifier. In order to better analyze the final performance of the classifier, the correlation factors and performance strength are described by the c/s^2 ratio, and the smaller the ratio, the better the performance of the classifier. It is defined as:

$$c/s^2 = \frac{\bar{\rho}}{s^2}. \tag{4}$$

The random forest integrates the bagging framework and CART (Classification and Regression Trees) process. Specifically, each tree randomly selects some variables as candidates for the split variables during each split to produce child nodes, in order to reduce the correlation between individuals.

2. 3. Enhanced random forest algorithm

Build more trees than traditional models, and then select some more accurate trees for aggregation. There is a variable interval between the correlation coefficient and the strength, within which the correlation coefficient increases and the strength will increase as compensation, and vice versa. Therefore, from the perspective of balancing the degree of correlation and strength, the above modifications attempt to find individuals with higher strength, while ensuring that the correlation coefficient between these individuals does not increase significantly. In addition, building more trees will inevitably increase the training time of the model, but as discussed earlier, due to the good parallelism of bagging, the time cost can easily be solved by increasing computational resources.

Due to the fact that the enhanced random forest aggregates the most accurate subset of individuals in the early stages, it can achieve more accurate prediction results than traditional algorithms as the number of aggregated trees increased, both models eventually aggregated all the constructed trees. The algorithm steps for enhancing random forests are as follows:

Step 1. Generate a bootstrap sample Z with a sample size of N from the training set.

Step 2. Generate a tree in a random forest based on each bootstrap sample during the process of tree growth, recursively repeat the following steps for each node until the node reaches the predetermined size limit.

Step 3. Randomly select r variables from all p variables, and select the best splitting variable and splitting point from r variables as candidates.

Step 4. Split the current node into two sub nodes based on the selected split points.

Step 5. Calculate the individual accuracy of each tree using Out-of-Bag (OOB) samples, and then sort the K trees in descending order of their individual accuracy.

Step6. Output the set T ($N<K$) of the first N trees, give a new point x, and obtain its predicted value.

If T is a classification tree, then:

$$\frac{1}{OOB^m} \sum_{x \in OOB^m} I\left(T^m\left(x_i\right) = y_i\right). \tag{5}$$

Among them, $T^m(x_i)$ represents the predicted value of tree m for the i-th observation, y_i represents the true value of the i-th observation, and i represents an indicative function. This indicator is called individual accuracy, and the larger its value, the higher the accuracy of tree m.

In order to get the least correlation between the trees, in the optimization process of Random Forest, in addition to the OOB method to select the number of predictors r and the depth of the tree d, another commonly used and effective method is Grid Search Cross-Validation. The GridSearchCV module of the sklearn library is used to perform the grid search, which automatically tries all combinations of parameters and uses cross-validation to evaluate the performance of each combination. Through the detailed output after each training, the accuracy rate is used as the evaluation criterion, and the model training process uses 10-fold cross-validation for the best results, and the optimal parameter combination is obtained after the grid search is completed.

2. 4. Encrypted traffic

Encrypted traffic refers to network communication data processed by encryption algorithms and protocols to ensure the security of data during transmission [7]. In computer networks, in order to protect the confidentiality and integrity of data, various encryption algorithms and protocols are usually used to encrypt data. This encryption can effectively prevent data from being stolen, tampered with, or forged, improving the security and reliability of data transmission.

The identification of encrypted traffic has multiple importance in network security. By identifying encrypted traffic, it is promptly possible to detect potential network attack behaviors and take corresponding preventive measures. Encrypted traffic identification helps to protect the security of sensitive data, prevent data leakage or malicious acquisition. By identifying different types of encrypted traffic, it is possible to improve our network security defense capabilities and better respond to various network threats.

2. 5. Encrypted traffic identification

Encrypted traffic identification, also known as encrypted traffic analysis, is a technique that analyzes the content of network packets to determine whether they have been encrypted and the type of encryption. In the field of network security, encrypted traffic identification is of great significance because it can help better understand network attacks, track malicious behavior, and identify potential security threats [8].

The principle of encrypted traffic recognition is mainly based on the header information of network packets. The header information contains key information such as protocol type, source address, destination address, etc. By analyzing this information, it is possible to determine whether the data packet has been encrypted and the type of encryption (such as SSL, TLS, UDP encryption, etc.). In addition, some advanced encryption traffic recognition technologies can also be combined with the content of data packets to further improve the accuracy of recognition.

There are many methods for identifying encrypted traffic, mainly including but not limited to the following:

1) protocol analysis: by analyzing the packet structure of common encryption protocols such as SSL and TLS, it is possible to determine whether the data packet has been encrypted [9];

2) fingerprint recognition: using known encryption algorithm features, by comparing the information in the data packet, determine its encryption type;

3) deep learning: using machine learning algorithms to train a large number of encrypted data packets and establish classification models to improve recognition accuracy.

2. 6. Encryption algorithms

Encryption algorithms are an important means of protecting information security, which can convert plaintext into ciphertext to protect the security of data. According to the usage of keys, encryption algorithms are mainly divided into two types: symmetric encryption algorithm and asymmetric encryption algorithm. Symmetric encryption algorithms use the same key for encryption and decryption, and common symmetric encryption algorithms include DES, 3DES, AES, etc. Asymmetric encryption algorithms require the use of public and private keys for encryption and decryption [10]. The public key can be made public and can be obtained by anyone for encrypting data, while the private key is only owned by the private holder for decrypting data. Common asymmetric encryption algorithms include RSA, DSA, ECC, etc.

The use of these encryption algorithms can effectively ensure the security of data during transmission and storage, avoid data theft, tampering, or forgery, and protect user privacy and data security.

3. Results and discussion

3. 1. Acquisition of datasets

3. 1. 1. Self-collected data

There are many ways to collect traffic in the Internet. The basic principle is to mirror the traffic in the network. For example, using port mirroring, traffic redirection based on Web Cache

Communication Protocol (WCCP), and using splitters for traffic collection [11]. In specific practical applications, users can adopt corresponding collection methods based on the actual network structure, network traffic, and characteristics of the network equipment used.

3. 1. 2. Public datasets

In addition to building your own datasets to train the model, it is also possible to use publicly available datasets. Usually, building a data set by oneself requires a lot of humans, financial and material resources as well as time. It may also require a lot of engineering design, which is not conducive to the rapid implementation and verification of scientific research ideas. Fortunately, due to the improvement of information infrastructure, researchers around the world can share their own data sets to others through the Internet, so that they can use these open data sets to carry out their own research.

In the field of machine learning, the importance of datasets is self-evident, and datasets are the foundation of research. In addition to self-made data sets, today, with the development of the Internet, many classic data sets have been distributed online for scholars around the world to study. When using public data sets on the vast Internet, it is possible to choose a data set suitable for our research direction according to our research topic. In the field of traffic recognition, there are many specialized research institutions that collect traffic data, such as DARPA, DEFCON, CAIDA, CAPTURES, ADFA, and so on [12].

3. 2. Data preprocessing

3. 2. 1. Raw traffic data segmentation

Whether it is the captured datasets or the publicly available datasets downloaded from the internet, the most primitive data format is the PCAP package format. It is obvious that the data format of the PCAP package cannot be used directly. Before use, it must be segmented and necessary cleaning work must be carried out. It is necessary for us to understand the PCAP package format before splitting [13]. Each PCAP file consists of three parts: GlobalHeader, PacketHeader, and PacketData. The initial position of the file is GlobalHeader, followed by paired PacketHeader and PacketData.

When dividing traffic, the first step is to determine whether to divide it according to bidirectional flow or unidirectional flow. Traffic data has a direction, which refers to whether a specific message is sent from the source IP to the destination or from the destination IP to the source IP. In the actual conversation process, these two directions of flow exist simultaneously. In the process of capturing traffic, real-time data flowing through the host network card is read out, and these two different directions of flow are not distinguished. However, in practical use, sometimes it may be necessary to distinguish between traffic data from two different directions. Therefore, dividing the direction of flow here can provide more targeted data analysis. In other scenarios such as application recognition, whether it is traffic data from the client to the server or from the server to the client, they all contain specific application characteristics. In this scenario, there is no need to distinguish the direction of diversion.

3. 2. 2. Feature selection

In order to extract as many features as possible from the original data, a huge feature library is established, which contains a large number of original features, of course, its dimensionality is also very high. However, in this step, let's reduce the dimensionality of samples in high-dimensional space by removing redundant and irrelevant features through feature selection. The availability of data tends to focus on its features which have capability for its considerable representation. The process of features selection is dependent to computational tasks which are the challenging phase for any selection criterion [14]. The reason for doing this is that using a large number of features to design a classifier with a limited number of samples is too computationally expensive and may not necessarily guarantee good classifier performance [15]. Therefore, in the step of feature selection, the main task is to filter out features that have good discriminative properties for the datasets from the feature library. And there are some basic principles that should be followed when performing feature selection work here, such as obtaining the smallest possible subset of features, not significantly reducing classification accuracy, not affecting class distribution, and the feature subset should have stable and strong adaptability.

3. 3. Simulation experiment

3. 3. 1. The dataset used

The dataset used in this chapter is the publicly available CTU dataset, which was publicly released by the MCFP (Malware CaptureFacility Project) project team at Stratosphere Laboratory, Czech Polytechnic University. The main purpose of the MCFP project is to build a machine learning based intrusion detection system (IDS) based on the latest research progress in the field of malicious attack detection to defend against specific network attacks [16]. Based on this project, three main types of datasets have been released: malicious traffic data, normal traffic data, and mixed traffic data. In this paper, combined with the research direction of this paper, encrypted malicious traffic data is mainly used. Whether encrypted traffic is collected is determined by determining whether the PCAP packet in the collected traffic uses an encryption protocol.

A detailed introduction to using datasets is shown in **Table 1**.

Table 1
List of dataset details

Label	0	1	2	3	4	5	6	7
Name	Artemis	Coin Miner XMRig	Trick Bot	None	Trickster	Web Companion	Dridex	CoinMiner
Number	10473	17896	8211	20376	45462	59303	27998	50225

3. 3. 2. Simulation analysis

The experimental hardware environment is based on Windows 10 64-bit operating system with Intel Core i7 CPU and 8GB of RAM, message queue kafka, and the Python machine learning library auto-sklearn is installed. All models including the comparison methods are trained using the enhanced random forest algorithm proposed in this paper.

In order to ensure the rapid convergence of the Enhanced Random Forest (ERF) algorithm and eliminate redundant information as much as possible, it is necessary to unify the length of the input encrypted traffic. When unifying the length of the encrypted traffic, it is necessary to determine the number of intercepted data packets and the byte length in the data packet. Different data packet numbers and byte lengths will have an impact on the classification performance of the model. Therefore, the evaluation index value is used as a performance evaluation indicator for data packet numbers, respectively, Extract bytes of different lengths for comparative experiments, and display the average values of experiments. The experimental results are shown in **Fig. 2**.

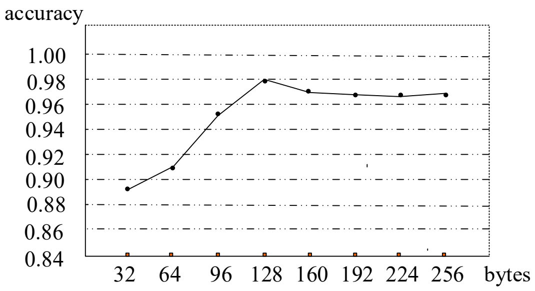

Fig. 2. The effect of different byte lengths by the ERF algorithm

The horizontal axis represents the number of bytes intercepted by the data packet, and the vertical axis use accuracy as the evaluation index. From **Fig. 2**, it can be seen that the evaluation index value of the ERF algorithm performs best when intercepting the first 128 bytes of data packets.

After determining the length of the intercepted bytes to be 128, the evaluation index value is used as a performance evaluation index to extract different numbers of data packets from session traffic for comparative experiments. The experimental results are shown in **Fig. 3**.

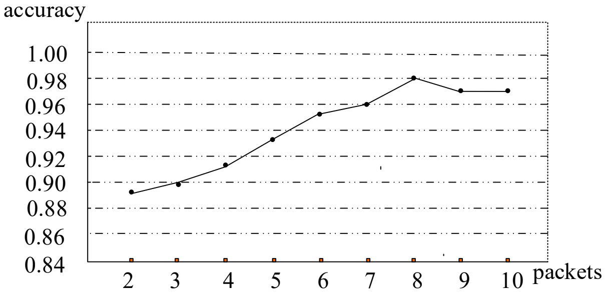

Fig. 3. The effect of different packet numbers by the ERF algorithm

The horizontal axis represents the number of data packets, and the vertical axis use accuracy as the evaluation index. From the comparative experiment, it can be seen that the model performs best in classification when intercepting the first 8 data packets of the session.

Evaluation metrics for the effectiveness of encrypted traffic identification involves measuring *TP* (true positives, correctly identified as normal), *FP* (false positives, incorrectly identified as normal), *FN* (false negatives, incorrectly identified as malicious), and *TN* (true negatives, correctly identified as malicious). These metrics gauge the accuracy in distinguishing between normal and malicious traffic types.

Recall is the ratio between the number of positive samples correctly classified and the number of actual positive samples, ranging between 0 and 1. *FPR* is the ratio between the number of negative samples incorrectly classified and the number of actual negative samples, ranging between 0 and 1 [17]. *P* (Precision) is the ratio between the number of positive samples correctly classified and the number of samples classified as positive, ranging between 0 and 1. *F*1 considers both *P* and Recall, which is represented as the harmonic mean of *P* and Recall. The percentage of correctness is expressed as Accuracy and is calculated as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{6}$$

$$P = \frac{TP}{TP + FP}, \tag{7}$$

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{8}$$

$$F1 = \frac{2 * P * \text{Recall}}{P + \text{Recall}}. \tag{9}$$

From **Table 2**, it can be seen that the average checking Accuracy reaches 98.95 % in the experimental results for different datasets, and the Precision, Recall rate, and the *F*1 data, which comprehensively reflect the overall performance, all show better results.

Comparison of the data results of the different datasets for the four performance evaluation metrics are shown in **Fig. 4**.

From **Table 3**, it can be seen that the ERF algorithm has a good recognition effect on encrypted traffic. Whether in conventional encrypted traffic or VPN encrypted traffic, the algorithm has a recognition accuracy of over 90 %, with an average recognition accuracy of 92 % in several conventional encrypted traffic and 94 % in several VPN encrypted traffic. This indicates that the model extracts more features from VPN encrypted traffic. Therefore, the classification effect is better.

Table 2

Data set recognition performance

Data set	*P* (%)	Recall (%)	*F*1 (%)	Accuracy (%)
N1	98.5	97.8	98.1	98.2
N2	99.4	98.2	98.4	99.7
N3	98.2	97.6	97.7	97.5
N4	97.3	99.5	99.8	98.4
N5	99.6	98.6	97.9	97.9
N6	98.5	99.3	99.1	98.9
N7	98.5	98.8	98.1	98.2
N8	99.4	98.2	98.4	99.7
N9	98.2	97.6	97.7	98.5

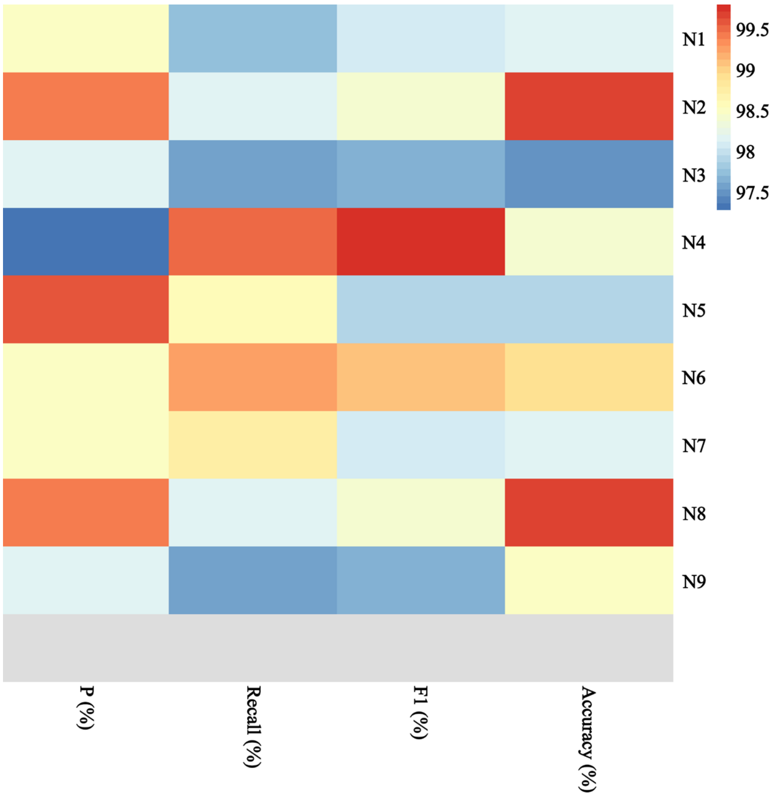

Fig. 4. Data comparison of datasets on performance evaluation metrics

Table 3

Error rate of traffic test set

Data set	SVM	GBM	RF	ERF1	ERF2	ERF3	ERF4
N1	3.36	3.96	3.16	3.47	3.37	3.57	3.37
N2	23.52	25.51	25.25	25.25	24.91	25.15	25.11
N3	23.89	23.32	23.54	23.78	23.28	23.53	23.27
N4	37.68	3.26	2.11	2.12	2.03	2.01	2.02
N5	4.60	3.72	3.98	4.00	4.02	4.07	4.02
N6	29.82	25.80	26.29	27.74	27.2	28.06	27.43
N7	6.85	5.82	4.9	4.86	4.82	4.87	4.79
N8	5.72	20.53	3.86	4.03	3.79	4.03	3.76
N9	10.38	11.74	8.69	8.83	8.64	8.83	8.65

3. 4. Limitations of the study and future directions for its development

3. 4. 1. Limitations and applicability of research results

The applicability of the Enhanced Random Forest algorithm for identifying encrypted traffic is limited by factors such as the quality and diversity of the datasets, changes in encryption protocols and technologies, computational resources and efficiency, feature selection and engineering, and variations in network environments. In practice, the most important aspects are feature selection and extraction, the generalization ability of the model, and the diversity and representativeness of the data. If new encryption methods are not included in the model, or computational resources are insufficient, the algorithm may perform poorly. Additionally, different network environments and inappropriate feature selection can affect the algorithm's accuracy, necessitating tuning and validation in various environments.

The algorithm proposed in this paper can be replicated and implemented. First, network traffic data is collected and appropriate features are selected, such as packet size, transmission time, etc. Then, the Enhanced Random Forest algorithm is used for training to optimize the model parameters, such as the number and depth of trees, through grid search and cross-validation. Finally, the model performance is evaluated using a test set to ensure that it can effectively recognize encrypted traffic. Through this method, network traffic classification and security monitoring can be achieved in the field of network security.

3. 4. 2. Future research directions

Future research in identifying encrypted traffic will continue to search for algorithm choices and improvements with optimal efficiency, and explore refined classification and anomaly detection techniques for encrypted traffic.

Further research will be conducted on how to efficiently deploy algorithms in real-time environments to reduce latency and resource consumption, as well as to improve the robustness and generalization ability of the model under different network environments and traffic patterns.

4. Conclusions

This research uses the publicly available CTU datasets and performs data segmentation and data cleaning on the original PCAP package format. Features with good discriminative properties for the datasets are filtered from the feature library. On the basis of the traditional random forest model, more trees than the traditional model are constructed, and then some more accurate trees are selected for aggregation, and the values of the parameters such as the number and depth of the trees are constantly adjusted using grid search cross-validation, while the performance of each set of parameter combinations is evaluated, and the optimal parameter combinations are ultimately found to form the enhanced random forest algorithm, which can obtain more accurate prediction results than the traditional algorithm. The experimental results are evaluated using accuracy, precision, recall and $F1$ score, and the results show that the average precision reaches 98.95 %, and the data of each performance evaluation metric is good, indicating that the advantages of the improved algorithm are obvious. In summary, the enhanced Random Forest algorithm proposed in this paper helps to identify malicious traffic in the network communication process and improve the defense ability of network attacks, especially for various encryption protocols and new types of traffic characteristics appear can still be effectively identified.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

Financing

This research project was financially supported by Mahasarakham University (Thailand).

Data availability

Manuscript has no associated data.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

Acknowledgments

The authors would like thank Scientific and Technological Innovation Programs of Higher Education Institutions in Shanxi (STIP, 2023L422), Shanxi, China, for their support in scholarship. Special thanks are extended to all the experts who generously provided research tools and equipment, as well as valuable suggestions.

References

[1] Cisco Annual Cybersecurity Report. Available at: https://www.cisco.com/

[2] Hu, G., Fukuda, K. (2023). Characterizing Privacy Leakage in Encrypted DNS Traffic. IEICE Transactions on Communications, E106.B (2), 156–165. https://doi.org/10.1587/transcom.2022ebp3014

[3] Tadini, M., Borruso, G. (2022). Sea-Rail Intermodal Transport in Italian Gateway Ports: A Sustainable Solution? The Examples of La Spezia and Trieste. Lecture Notes in Computer Science, 156–172. https://doi.org/10.1007/978-3-031-10548-7_12

[4] Moharamkhani, E., Yahyaei Feriz Hendi, M., Bandar, E., Izadkhasti, A., Sirwan Raza, R. (2022). Intrusion detection system based firefly algorithm-random forest for cloud computing. Concurrency and Computation: Practice and Experience, 34 (24). https://doi.org/10.1002/cpe.7220

[5] Park, S., Ye, J. C., Lee, E. S., Cho, G., Yoon, J. W., Choi, J. H. et al. (2023). Deep Learning-Enabled Detection of Pneumoperitoneum in Supine and Erect Abdominal Radiography: Modeling Using Transfer Learning and Semi-Supervised Learning. Korean Journal of Radiology, 24 (6), 541. https://doi.org/10.3348/kjr.2022.1032

[6] Zhu, L., Tian, N., Li, W., Yang, J. (2022). A Text Classification Algorithm for Power Equipment Defects Based on Random Forest. International Journal of Reliability, Quality and Safety Engineering, 29 (05). https://doi.org/10.1142/s0218539322400010

[7] Kurita, Y., Meguro, S., Tsuyama, N., Kosugi, I., Enomoto, Y., Kawasaki, H. et al. (2023). Accurate deep learning model using semi-supervised learning and Noisy Student for cervical cancer screening in low magnification images. PLOS ONE, 18 (5), e0285996. https://doi.org/10.1371/journal.pone.0285996

[8] Shen, M., Ye, K., Liu, X., Zhu, L., Kang, J., Yu, S. et al. (2023). Machine Learning-Powered Encrypted Network Traffic Analysis: A Comprehensive Survey. IEEE Communications Surveys & Tutorials, 25 (1), 791–824. https://doi.org/10.1109/comst.2022.3208196

[9] Hu, Y., Cheng, G., Chen, W., Jiang, B. (2022). Attribute-Based Zero-Shot Learning for Encrypted Traffic Classification. IEEE Transactions on Network and Service Management, 19 (4), 4583–4599. https://doi.org/10.1109/tnsm.2022.3183247

[10] Wassie Geremew, G., Ding, J. (2023). Elephant Flows Detection Using Deep Neural Network, Convolutional Neural Network, Long Short-Term Memory, and Autoencoder. Journal of Computer Networks and Communications, 2023, 1–18. https://doi.org/10.1155/2023/1495642

[11] Yao, H., Liu, C., Zhang, P., Wu, S., Jiang, C., Yu, S. (2022). Identification of Encrypted Traffic Through Attention Mechanism Based Long Short Term Memory. IEEE Transactions on Big Data, 8 (1), 241–252. https://doi.org/10.1109/tbdata.2019.2940675

[12] Tong, V. V., Souihi, S., Tran, H.-A., Mellouk, A. (2023). Novel Global Troubleshooting Framework fo Encrypted Traffic. Troubleshooting for Network Operators, 25–43. https://doi.org/10.1002/9781394236664.ch2

[13] Ren, Y., Zhu, X., Bai, K., Zhang, R. (2023). A New Random Forest Ensemble of Intuitionistic Fuzzy Decision Trees. IEEE Transactions on Fuzzy Systems, 31 (5), 1729–1741. https://doi.org/10.1109/tfuzz.2022.3215725

[14] Ali, A., Jillani, F., Zaheer, R., Karim, A., Alharbi, Y. O., Alsaffar, M., Alhamazani, K. (2022). Practically Implementation of Information Loss: Sensitivity, Risk by Different Feature Selection Techniques. IEEE Access, 10, 27643–27654. https://doi.org/10.1109/access.2022.3152963

[15] Gantzer, T. D. (2019). Security Bug Report Classification using Feature Selection, Clustering, and Deep Learning. Statler College of Engineering and Mineral Resources. https://doi.org/10.33915/etd.4022

[16] Obasi, T. Encrypted Network Traffic Classification using Ensemble Learning Techniques. https://doi.org/10.22215/etd/2020-14171

[17] Liu, J., Tian, Z., Zheng, R., Liu, L. (2019). A Distance-Based Method for Building an Encrypted Malware Traffic Identification Framework. IEEE Access, 7, 100014–100028. https://doi.org/10.1109/access.2019.2930717

Received date 29.04.2024
Accepted date 01.07.2024
Published date 10.09.2024

How to cite: *Yang, X., Angkawisittpan, N., Feng, X. (2024). Analysis of an enhanced random forest algorithm for identifying encrypted network traffic. EUREKA: Physics and Engineering, 5, 201–212. https://doi.org/10.21303/2461-4262.2024.003372*