Detection of Improved Collusive Interest Flooding Attacks Using BO-GBM Fusion Algorithm in NDN

Zhijun Wu, Silin Peng, *Student Member, IEEE*, Liang Liu, and Meng Yue

***Abstract*—In Named Data Networking (NDN), Collusive Interest Flooding Attacks (CIFA) is a new type of Distributed Denial of Service (DDoS) attacks, which can effectively affect the performance of NDN by sending malicious Interests intermittently. Since the concealment of CIFA is strong, the existing detection methods for Interest Flooding Attacks (IFA) are difficult to find the malicious Interests in the NDN network. However, the subsequent attack strength of CIFA is weaker than that of IFA, resulting in the attack range of CIFA is much smaller than that of IFA in large network topologies. In order to launch the most serious attack with the least cost, the attack model of CIFA has been improved by our previous work, namely Improved Collusive Interest Flooding Attacks (I-CIFA). To better take the countermeasures against I-CIFA, this paper studies the adverse effects of I-CIFA in NDN and proposes a detection mechanism for I-CIFA. Foremost, we extract the corresponding network traffic and analyze the impact of I-CIFA on malicious routing nodes in different locations of the network. Furthermore, the detection mechanism based on BO-GBM fusion algorithm is proposed to detect I-CIFA through classifying the network traffic. Finally, several specific performance metrics are adopted to evaluate the practicability of BO-GBM fusion algorithm in detecting I-CIFA. The results show that BO-GBM fusion algorithm has better detection performance than other existing detection schemes, with the detection rate of 98.69%, false alarm rate of 1.36% and missing alarm rate of 1.43%.**

***Index Terms*—Bo-gbm fusion algorithm, improved collusive interest flooding attacks, named data networking.**

I. Introduction

THE promising architecture in the future networks, Named Data Networking (NDN) [1], which can significantly improve data delivery by changing the content dissemination mechanism from the traditional host-centric to content-centric [2]. After ten years of research and development, it is mature enough to implement NDN today. Not only the relevant NDN Forwarding Daemon (NFD), but also stakeholders (such as telecommunications companies) will use NDN architecture for actual deployment [2]. With the advent of Big Data era and the explosive expansion of global networks, the current TCP/IP networks have exposed insecurity and poor mobility, making it more difficult to meet human needs for larger-scale networks in the future. In order to solve these problems, NDN is proposed as a feasible and efficient network architecture [1], [3].

Each router consists of a structure of the following three elements in NDN: CS (Content Store), PIT (Pending Interest Table), and FIB (Forwarding Information Base). On the one hand, the CS in the NDN router can cache the content in data packets, and the cache is directly established at the transport network layer, so it can save bandwidth. Compared with the IP router, the NDN router can reuse the forwarded data because the naming of data in the NDN network remains unchanged, thereby improving the content sharing rate. In other words, the consumer does not need to make a request to the producer again, because the cached copy is forwarded to any consumer that requests it. On the other hand, the PIT records the Faces on which Interests arrive at routing nodes before forwarding Interests, and Interests that have been forwarded but not satisfied. The FIB is a table that routes the incoming Interests based on the name prefixes of Interests. And there are two types of packets, Interests and data packets.

The working process of NDN is shown in Fig. 1. On upstream, the consumer sends an Interest to request corresponding content. When the Interest arrives at the routing node R1, the routing node R1 first checks the CS. If corresponding content is found in CS, it will be forwarded to the consumer in response. Otherwise, the routing node R1 continues to check the PIT, if there is a matching Interest in the PIT, the Face that the Interest arrives at will be added to the PIT but not forwarded. If not, the routing node R1 continues to check the FIB, the Interest will be forwarded if there is a matching entry in FIB. If none of the FIB entries match the Interest, the Interest will be forwarded through all Faces of the routing node R1. On downstream, when the data packet arrives at the routing node R2, the data packet will be forwarded by the routing node R2 in response to the consumer if there is a matching entry in the PIT. And the routing node R2 decides whether to cache the data packet in its CS according to its cache replacement policy. Besides, all Interests in other cases are discarded.

Manuscript received 1 March 2022; revised 31 May 2022; accepted 11 September 2022. Date of publication 14 September 2022; date of current version 6 January 2023. This work was supported in part by the Civil Aviation Joint Fund of National Natural Science Foundation under Grant U1933108, in part by the National Natural Science Foundation of China under Grant 62172418, in part by the Natural Science Foundation of Tianjin China under Grant 21JCZDJZ00830, and in part by the Fundamental Research Funds for the Central Universities of China under Grants ZXH2012P004 and 3122021026. Recommended for acceptance by Dr. Christian Poellabauer. *(Corresponding author: Zhijun Wu.)*

Zhijun Wu, Liang Liu, and Meng Yue are with the College of Safety Science and Engineering, Civil Aviation University of China, Tianjin 300300, China (e-mail: caucwu@263.net; liul@cauc.edu.cn; myue@cauc.edu.cn).

Silin Peng is with the College of Electronic Information and Automation, Civil Aviation University of China, Tianjin 300300, China (e-mail: stepeng30@163.com).

Digital Object Identifier 10.1109/TNSE.2022.3206581

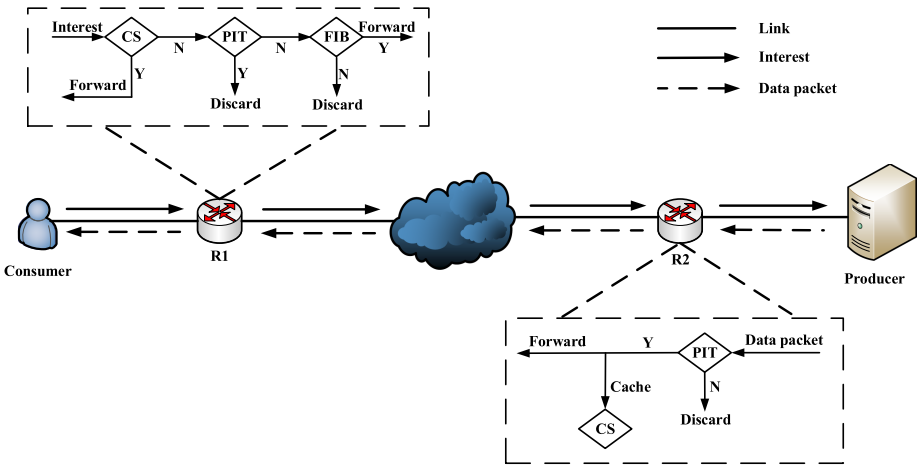


Fig. 1. The working process of NDN.

It is worth mentioning that the security issues of NDN are considered by the designers at the beginning. Meanwhile, for the purpose of effectively preventing the data packets from being forged or tampered with, the data packets requested by consumers are signed by the producers in NDN [4]. However, Distributed Denial of Service (DDoS) attacks in NDN still exist [5], such as IFA, CIFA, and recently proposed I-CIFA [4]. Most importantly, detecting and defending against interest flooding attacks such as I-CIFA are still the main difficulties in NDN.

The novel attack proposed by combining IFA and CIFA, I-CIFA, has been proven to have better attack effect on large-scale network topologies [4]. Before launching I-CIFA, the I-CIFA attackers can quickly detect the PIT capacity of downstream bottleneck routing nodes through the probing mode, which can provide prerequisites for subsequent attacks. Meanwhile, the attacker's cost can be reduced more effectively with the help of the probing mode. As a result, the Interest satisfaction of attackers keeps a high value while the Interest satisfaction of consumers keeps a low value.

At present, the existing detection schemes are difficult to detect the new type of I-CIFA. Through analysis and research, detection of I-CIFA can be regarded as judging whether the network state is normal at each moment, so we can predict and classify the network states based on multi-dimensional network traffic characteristics. Specially, using Machine Learning algorithms such as Gradient Boosting Machines (GBMs) which includes GBDT (Gradient Boosting Decision Tree), XGBoost (eXtreme Gradient Boosting) and LightGBM (Light Gradient Boosting Machine), outperforms other detection methods in attack detection [6], [7], [8]. In order to reduce the time of attack detection, GBMs will be Bayesian optimized for the network traffic data to obtain BO-GBMs, including BO-GBDT, BO-XGBoost and BO-LightGBM. Finally, BO-GBM fusion algorithm is proposed to detect I-CIFA, which will be compared with other Machine Learning algorithms, including SVM (Support Vector Machines) [9], KNN (K-Nearest Neighbor) [10], Decision Tree [11], BO-GBDT, BO-XGBoost and BO-LightGBM. Furthermore, BO-GBM fusion algorithm is compared with three representative detection method for IFA, including Cumulative Entropy [12], Gini Impurity [13] and IForest [20], as well as two representative detection methods for CIFA, including Wavelet Analysis [14] and Prediction Error [15], to verify the performance of BO-GBM fusion algorithm in detecting I-CIFA.

The four contributions of this study can be concluded as follows:

- The multi-dimensional network traffic characteristics of I-CIFA are extracted, such as the number of PIT entries, the number of CacheHits in CS, the number of OutInterests and the number of SatisfiedInterests, which can significantly reflect the changes in network traffic.
- Since the number of network traffic samples are imbalanced generally, in order to ensure that the percentages of the two types of network traffic data are consistent, including normal samples and attack samples, the stratified 10-fold cross-validation method is proposed to solve the problem of data imbalance and ensure the reliability of attack detection results.
- Propose a detection method for I-CIFA, BO-GBM fusion algorithm, which is constructed a two-layer structure with BO-GBMs and Logistic Regression by using Stacking method.
- BO-GBM fusion algorithm is compared with other eleven schemes: SVM, Decision Tree, KNN, BO-GBDT, BO-XGBoost, BO-LightGBM, Cumulative Entropy, Gini Impurity, IForest, Wavelet Analysis and Prediction Error. The substantial simulations demonstrate our scheme outperforms the other schemes for detecting I-CIFA by evaluating performance metrics.

The rest of this study is organized as follows: Section II presents and discusses the existing countermeasures against DDoS attacks in NDN. Section III introduces a more detailed description of the DDoS attacks in NDN, and makes an in-depth analysis of I-CIFA. Section IV analyzes the impact of I-CIFA in NDN. Section V proposes the detection mechanism for I-CIFA based on BO-GBM fusion algorithm. Section VI analyzes and verifies the detection performance of our proposed solution. Finally, this paper draws conclusions and gives suggestions to extend this work in Section VII.

II. Related Work

As we all know, DDoS attacks are seriously affect the network security of NDN. The attackers can use various methods to launch DDoS attacks [5], so many detection and mitigation methods have been proposed for DDoS attacks in NDN [35].

A. The Research Status of IFA

IFA can seriously affect network services was proven in the paper [16] and a few victimized routing nodes significantly reduced the performance of legitimate consumers. However, Satisfaction-Based Push Back (SBPB) was proposed by Afanasyev et al. [3] to defense against IFA successfully, and it determines the network state according to the Interests satisfaction of routing nodes. But this work was limited to a static and simple attack model and an assumption of the Interests do not satisfy the way of caching and forwarding. Moreover, a novel mechanism named Poseidon was proposed by Compagno et al. [17] to mitigate IFA, which detected IFA in a timely manner based on local metrics and collaborative techniques. But this method does not work if the adjacent routing node is hijacked. Since the existing IFA detection and mitigation methods based on PIT abnormal state statistics can lead to misjudge the network state and damage consumers, therefore, through monitoring the abnormal distribution of content request, a novel IFA detection method based on Cumulative Entropy was proposed by Xin et al. [12]. After detection, in order to restrain IFA, a countermeasure based on Interest traceback was proposed by them. However, this work was unable to identify the complex spoofed prefixes. From the perspective of the network, Cheng et al. [18] proposed the nonparametric CUSUM algorithm, which used a central controller to detect and mitigate more complex IFA. After determining the IFA, the source of attack can be directly located. Then it can be used without restraining legitimate Interests to prevent malicious Interests from entering the network. However, this work did not calculate the central controller's overhead. With the help of a central controller, Salah et al. [19] scheduled a group of monitoring routers to detect and defense against IFA with low overhead, but this mechanism can only work efficiently under the static network. An approach based on Gini Impurity and rate-limit was proposed by Zhi et al. [13] to detect and mitigate IFA, which use the statistical properties of name field in Interests. But this research did not verify the method in a more realistic topology.

Based on congestion-aware, Benmoussa et al. [32] proposed a novel detection and mitigation solution for IFA. To avoid false alarms about router behaviors, the network congestion was regarded as an important parameter. Simulations show that this approach can efficiently detect and mitigate IFA. However, the effect of this method on CIFA and I-CIFA has not been effectively implemented and verified. Through grouping Interests and using the distribution of names of different groups, Hou et al. [33] proposed the Theil-based Countermeasures (TC) to detect and mitigate IFAs. It turns out that TC has better performance than other typical IFA countermeasures. Nevertheless, this study did not validate the mechanism in various complex attack scenarios. The defense mechanism MSIDN was proposed by Benmoussa et al. [34] to mitigate sophisticated interest flooding-based (D)DoS attacks. Without affecting legitimate traffic and consumers, MSIDN reduces the impact of malicious traffic by mitigating attacks at the source of NDN network. Experiments demonstrate the efficiency of this method in different attack scenarios. Although this approach can reduce network overhead, it relies on the feedback of producers. A detection mechanism for IFA based on isolation forest (IForest) algorithm was designed by Chen et al. [20], which was constructed through the prefix of the Interests to isolate the normal prefix and the abnormal prefix. According to the occupancy rate of PIT, malicious prefixes were detected from abnormal prefixes. However, this work did not contain a defense mechanism for IFA. In this regard, Xing et al. [21] also introduced IForest algorithm to identify malicious prefixes in abnormal prefixes, thereby reduced the impact of IFA by restricting the forwarding of malicious Interests. But this mechanism has not been proven in a network environment with more sophisticated attacks. In order to realize the defense against IFA, Zhou et al. [22] proposed a new defense method based on deep reinforcement learning through designing a reward function to give timely feedback to the agent. Finally, the average request delay of consumers, the number of retransmissions of Interests and the number of received packets were compared to verify that the proposed solution can better resist IFA. But this research does not compare with other existing methods and is complicated to implement.

B. The Research Status of CIFA

Xin et al. [14] first proposed the concept of CIFA and Wavelet Analysis was used to detect CIFA. For this attack, the entropy-based detection method takes a long time to collect the prefix and other relevant information of malicious Interests. Even though the attacks were detected ultimately, the CIFA had already existed in NDN for a long time. A generic defense mechanism against CIFA, CoMon, was proposed by Salah et al. [23], which can detect and mitigate successfully with only a few routers at an early stage. Nevertheless, this research will cost more than other existing mechanisms. Liu et al. [15] proposed a detection method for CIFA based on Prediction Error. Although, this detection method helped to design safe and efficient routing forwarding strategies in NDN and detected CIFA through modular deployment without changing the front-end mechanism, this work was limited to detect CIFA modularly without corresponding defense measures. Therefore, a lightweight defense scheme based on PIT space management was proposed by Wu et al. [24], which can reduce the adverse effects of CIFA in NDN. And they proposed Time Scrolling Window Algorithm (TSWA) to detect CIFA previously. However, this algorithm is not so efficient in detecting I-CIFA. Shigeyasu et al. [25] proposed a novel distributed algorithm for detecting CIFA through three checks to improve the accuracy of CIFA attack detection, which can detect malicious routing nodes that do real harm. In addition, the malicious routing nodes can be well suppressed by introducing the penalty mechanism, however,

TABLE I
COMPARISON OF EXISTING STUDIES

Paper	Type of attack	Motivation	Method	Applicable to I-CIFA or not
Afanasyev 2013 [3]	IFA	Detection and mitigation	SBPB	No
Compagno 2013 [17]	IFA	Detection and mitigation	Poseidon	No
Xin 2016 [12]	IFA	Detection and mitigation	Cumulative Entropy	Yes but not efficient
Cheng 2019 [18]	IFA	Detection and mitigation	Non-parametric CUSUM algorithm	Yes but high cost
Salah 2015 [19]	IFA	Detection and mitigation	Monitoring routers	Yes but not efficient
Zhi 2018 [13]	IFA	Detection and mitigation	Gini Impurity and rate-limit	Yes but not efficient
Benmoussa 2020 [32]	IFA	Detection and mitigation	Congestion-aware	No
Hou 2019 [33]	IFA	Detection and mitigation	Theil-based	Yes but not efficient
Benmoussa 2019 [34]	IFA	Mitigation	MSIDN	No
Chen 2019 [20]	IFA	Detection	IForest	Yes but not efficient
Xing 2021 [21]	IFA	Detection and mitigation	IForest	Yes but not efficient
Zhou 2020 [22]	IFA	Mitigation	Deep reinforcement learning	Yes but complex
Xin 2017 [14]	CIFA	Detection	Wavelet Analysis	Yes but not efficient
Salah 2016 [23]	CIFA	Detection and mitigation	CoMon	Yes but high cost
Liu 2020 [15]	CIFA	Detection	Prediction Error	Yes but not efficient
Wu 2020 [24]	CIFA	Detection and mitigation	TWSA and PIT space management	Yes but not efficient
Shigeyasu 2020 [25]	CIFA	Detection	Distributed algorithm	Yes but complex
This paper	I-CIFA	Detection	BO-GBM fusion algorithm	Yes and efficient

this penalty mechanism is only effective when the network's runtime becomes longer, and the adaptability of the detection mechanism is insufficient. For improving the attack mode of CIFA, then Wu et al. [4] designed a new type of attacks, I-CIFA, which has the strength of IFA and evades the existing detection mechanisms such as CIFA. Finally, they proved the effectiveness of I-CIFA.

Comprehensively consider the above analysis, the deficiencies of previous studies in detecting and mitigating IFA and CIFA can be summarized as follows:

1) High cost and complex;
2) The legitimate traffic and consumers would be affected;
3) Not efficient.

Aiming at overcoming the above disadvantages, foremost, we use the trace helpers that comes with ndnSIM to collect legitimate and malicious network traffic data without affecting the legitimate traffic and consumers. Due to the concealment of I-CIFA explained in Section III, it is difficult for us to detect malicious traffic in the network. The previous methods are not efficient enough to detect this highly stealthy attack [15], so it is necessary to analyze the multi-dimensional network traffic characteristics to judge the network states. Therefore, the attack detection of I-CIFA is equivalent to classify the extracted multi-dimensional network traffic into legitimate or malicious traffic, then predict the network state and judge whether it is normal or not. Furthermore, an efficient scheme, the BO-GBM fusion algorithm, is proposed to judge the network states to realize the attack detection of I-CIFA. Finally, the experiment results show that BO-GBM fusion algorithm has better performance than the existing detection mechanisms.

We summarize and compare the above solutions as shown in Table I.

III. OVERVIEW OF DDOS ATTACKS IN NDN

In this section, we mainly introduce the DDoS attacks in NDN, namely IFA, CIFA and I-CIFA.

A. Overview of IFA and CIFA

With the emergence of IFA in NDN, various attacks have continued to evolve, and the NDN networks are subject to great security threats. According to the different types of content requested, they can be divided into three different content:

1) Content that static or existing;
2) Content that dynamically generated;
3) Content that does not exist.

Among them, IFA attackers request 3) to launch attacks, while CIFA request 1) and 2) to launch attacks. During the attack phase of IFA, the PIT entries are the status information generated by malicious Interests, which will fill up the PIT space until they are expired. After the PIT entries are deleted when they expired, PIT space will be quickly filled again by these entries due to the IFA attackers continue to send malicious Interests, causing the PIT space always being overloaded. Consequently, numerous legitimate Interests will not be received by the malicious routing nodes or even the producers, so there are no legitimate data packets generated by the producers in the network, resulting in the consumers will be denied service by normal producers.

However, in order to enhance the concealment of IFA, CIFA attackers send a range of different malicious Interests to request real and different content with the help of collusive producers in NDN, and a unique PIT entry will be generated by each malicious Interest in passing routing nodes. Then collusive producers generate corresponding collusive data packets with different content in response to the different malicious Interests when the PIT entries are about to expire. In the next round of CIFA, the PIT space where just released the malicious Interests, will be filled with new malicious interests again, resulting in the PIT space being intermittently overloaded, causing a large amount of normal Interests will be discarded by the malicious routing nodes. Finally, the consumers will be denied service by normal producers.

From the above discussion, it can be seen that CIFA has four points that are different from IFA:

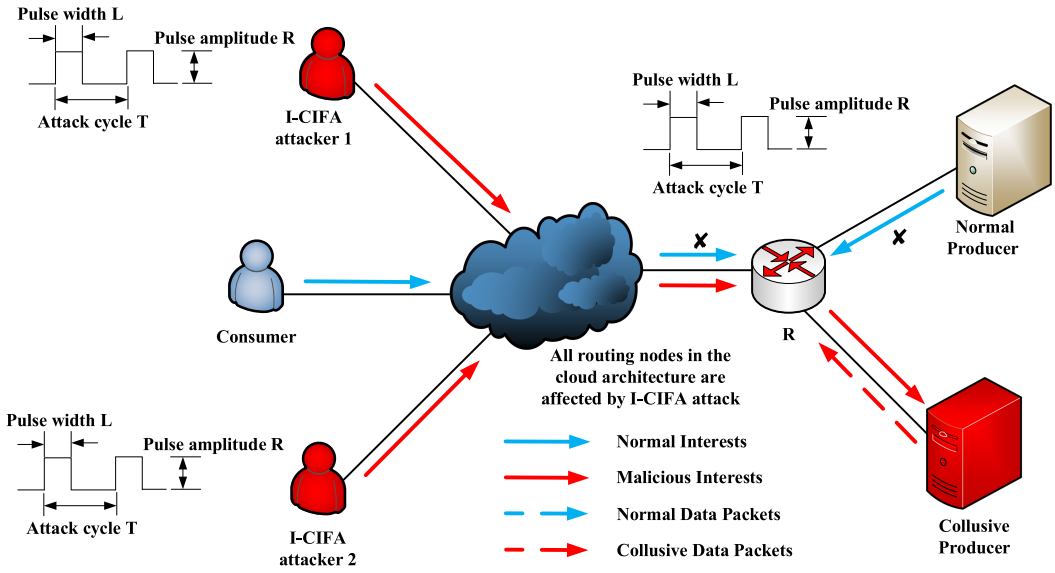


Fig. 2. The attack scenario of I-CIFA.

1) CIFA greatly affects the bottleneck routing nodes but the impact on upstream routing nodes is much smaller than IFA;
2) CIFA attackers send malicious Interests intermittently so that the CIFA is periodic, while IFA is persistent;
3) In order to improve the problem of insufficient concealment of IFA, CIFA attackers launch attacks in the form of periodic pulse data streams with low-rate by learning from the model of Low-rate Denial of Service (LDoS) attacks so that each attack period of CIFA can be represented by a triple (T, L, R).
4) Unlike IFA attackers requesting non-existent content, CIFA attackers request static or already existing and dynamically generated content.

The advantage of CIFA is that the PIT capacity of malicious routing nodes can be overloaded without the need for the aggregation of multiple attack traffic, because the number of malicious Interests sent by each CIFA attacker is the same as the setting PIT capacity during each attack cycle T. Hence, it has remarkable effect on bottleneck routing nodes at the beginning of CIFA. However, the CIFA attackers will compete with each other for the limited resources of PIT because the content requested is different so that many entries will be generated by the malicious Interests in the PIT of routing nodes. Finally, due to the overload of the PIT space, a large number of malicious Interests are discarded by routing nodes in each round of CIFA, so the strength of CIFA gradually weakened. It can be seen from the multi-dimensional network traffic characteristics of CIFA that the three disadvantages of CIFA are as follows [4]:

1) Ignore the state of the network infrastructure;
2) Ignore the unique forwarding network traffic characteristics of routing nodes in NDN;
3) Ignore the complexity and diversity in real network topology.

Even though CIFA can evade the existing attack detection mechanisms applied for IFA, they had little impact on large networks. Based on the above shortcomings of CIFA, a new type of attacks, I-CIFA, which is designed to solve the problems of the gradually weakening strength and the small attack range of CIFA.

B. Overview of I-CIFA

Before launching I-CIFA, there is a probing mode that the real PIT capacity can be faster detected on downstream routing nodes, because I-CIFA attackers request the same content from collusive producers simultaneously so that multiple identical malicious Interests will only generate one PIT entry. In the aspect of attack mode, I-CIFA combine IFA and CIFA, that is, the I-CIFA attackers periodically send malicious Interests to request the same content that does not exist in NDN, so the attack effect will not be weakened by the competition among the attackers for limited PIT resources. Thus, I-CIFA has stronger concealment than IFA, and has stronger attack strength and larger attack range than CIFA. From the above discussion, it can be seen that I-CIFA has three points that are different from IFA and CIFA:

1) The content requested by CIFA attackers is different, while I-CIFA attackers request the same content;
2) I-CIFA attackers send malicious Interests intermittently so that the CIFA is periodic, while IFA is persistent, so the concealment of I-CIFA is stronger than that of IFA;
3) I-CIFA has stronger attack strength and larger attack range than CIFA.

Fig. 2 shows the real attack scenario of I-CIFA [4]. When malicious Interests sent by I-CIFA attacker 1 first arrive at the routing node R, the PIT entries generated by malicious Interests will exhaust the limited PIT resources on the passing routing nodes. After the malicious Interests sent by I-CIFA attacker 2 arrive at the routing node R, the I-CIFA attacker 2 wait for the response from the collusive producer with the I-CIFA attacker 1 together because they request the same content, and the interface

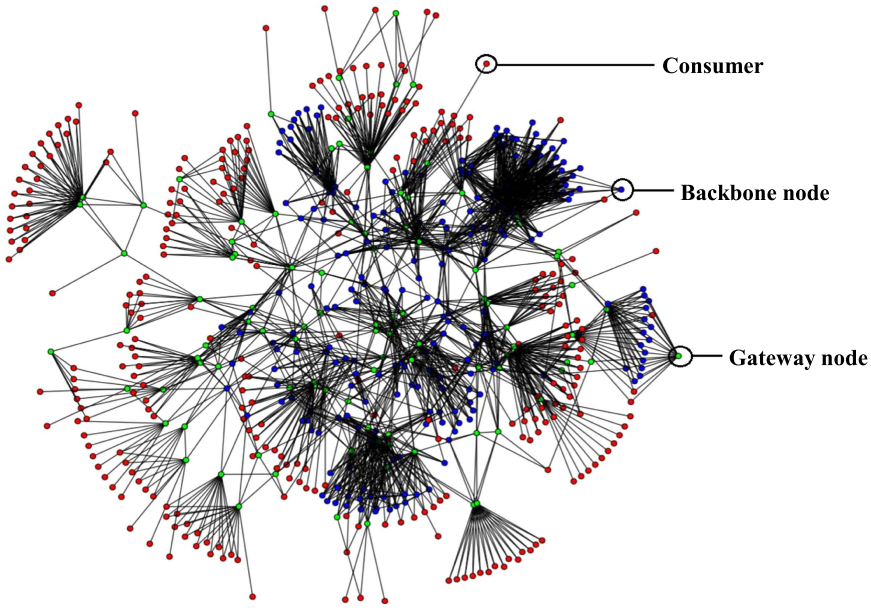


Fig. 3. Network topology 7018.r0.

on which the malicious interest reaches the routing node R will be added to the interface set of the PIT. In this way, the PITs of all routing nodes that the malicious Interests pass through will be overloaded during I-CIFA.

When the current round of attacks is over, the I-CIFA attackers immediately launch the next round of I-CIFA and send malicious Interests again to fill up the just released PIT space. As a result, the PIT space will always be overloaded so that a plenty of legitimate Interests will be discarded by the malicious routing nodes, resulting in consumers being denied service by normal producers. In this regard, I-CIFA overcomes the shortcoming that the strength of CIFA will gradually weaken afterwards, thus, the attack range of I-CIFA will be further increased. In general, I-CIFA combines the strong attack strength of IFA to increase the range of attacks, and combines the periodicity of CIFA to increase the concealment of attacks in NDN, thereby increasing the detection difficulty for I-CIFA.

IV. Impact Under I-CIFA

This section introduces the relevant environment and parameter settings of the experiment, and analyzes the impact of I-CIFA on NDN networks based on the network traffic characteristics extracted through the PCA dimensionality reduction method.

A. Experiment Environment and Parameter Settings

The large network topology 7018.r0 used in this experiment is shown in Fig. 3. The outermost nodes are consumers (296 red nodes), the nodes directly connected to the consumers are gateway nodes (108 green nodes), such as central routing nodes discussed above. And the remaining nodes are backbone nodes (221 blue nodes), such as downstream routing nodes discussed above.

TABLE II
Parameters of Experiment

Parameter	Value
Zipf-Mandelbrot distribution	(0,1)
The capacity of PIT	1000
The capacity of CS	1000
The size of packets	1024 Bytes
The lifetime of PIT entry	7s
The sending rate of packets	20/s
The attack duration L	6s
The attack cycle T	7s
The number of normal producer	1
The number of collusive producer	1
The time period of experiment	0-200s
The time period of I-CIFA	50-150s
Sampling interval	1s
The number of normal samples	806
The number of attack samples	1668

Foremost, we simulated the I-CIFA by using actual network topology 7018.r0 in ndnSIM, then extracted corresponding network traffic characteristics to analyze the impact of I-CIFA in NDN. Furthermore, we only did a 200-second simulation to better analyze the significant impact under I-CIFA on NDN networks, where the same simulation time in attack and normal states. However, the amount of network traffic data under attack and normal states is generally different in real network scenarios, so the network traffic data used for I-CIFA attack detection contains a 2474-second experiment. In order to clearly see the changes in the number of CS CacheHits and PIT entries, both the size of PIT capacity and the size of the CS capacity were set to 1000. The specific experiment parameters are shown in Table II.

B. Analysis of I-CIFA

I-CIFA prevent the forwarding of normal Interests by occupying the limited PIT resources that seriously affect network

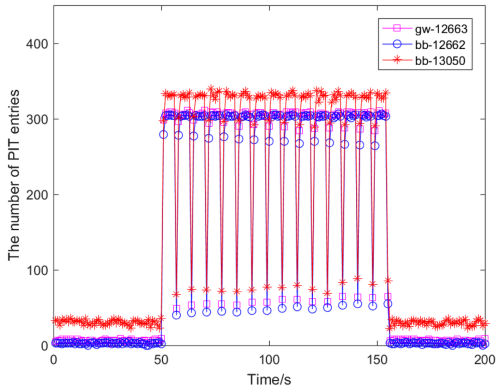


Fig. 4. The number of PIT entries.

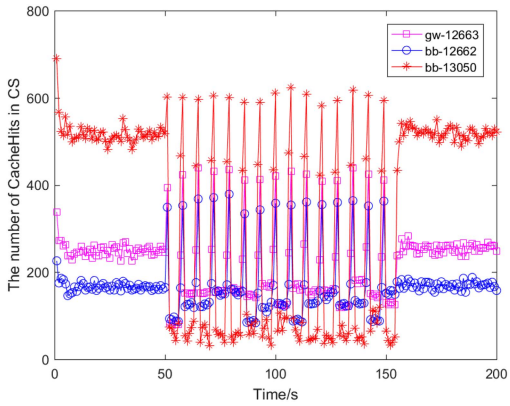


Fig. 5. The number of CacheHits in CS.

performance in NDN. Therefore, network traffic is a good indicator for evaluating the performance of NDN networks, and the multi-dimensional network traffic characteristics such as the number of PIT entries, the number of CS CacheHits, the number of OutInterests and the number of SatisfiedInterests will be extracted. The statistics on PIT entries can be acquired by using the module of ndn::Pit while the statistics on CacheHits, OutInterests and SatisfiedInterests are generated by the tracers in ndnSIM. By using ndn::CsTracer, it is feasible to get statistics on CacheHits on emulated routing nodes. Similarly, the statistics of OutInterests and SatisfiedInterests can be obtained by using ndn::L3RateTracer to output the traffic information of each emulated routing node. Importantly, network traffic data is collected by using trace helpers, so this method ensures the reliability of the detection method without affecting legitimate traffic and consumers on NDN networks.

Since the backbone nodes bb-12662 and bb-13050 are closer to the producers, they are most sensitive to changes in network traffic characteristics, so their corresponding network traffic characteristics were extracted. In order to reflect the wide-ranging impact of I-CIFA in NDN, the corresponding network traffic characteristics of gateway routing node gw-12663 were extracted. Moreover, the multi-dimensional network traffic characteristics extracted in large network topology 7018.r0 will be analyzed. As can be seen from the above, the simulation lasted for 200 seconds, in which I-CIFA was launched at 50 seconds and ended at 150 seconds. The significant changes in

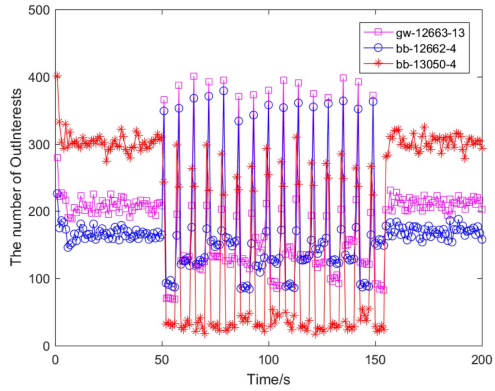


Fig. 6. The number of OutInterests.

the number of PIT entries of different routing nodes are shown in Fig. 4.

Fig. 4 shows that in the ideal scenario with PIT capacity set to 1000, the number of PIT entries of the routing nodes bb-12662, bb-13050 and gw-12663 increases intermittently in the form of periodic pulses during the attack phase. However, it quickly returns to its normal value after each attack pulse. Because the size of PIT capacity in the real network scenario is different, and if the capacity of PIT is set too large, it will cause a waste of resources. Consequently, when the size of the PIT capacity is limited, the PIT resources will be exhausted by the PIT entries generated by the malicious Interests to prevent the forwarding of legitimate Interests. As a result, consumers will not receive the response of legitimate data packets sent by the normal producer, resulting in denial of service for the consumers, which will seriously affect the performance of NDN networks. The notable changes in the number of CacheHits of different routing nodes are shown in Fig. 5.

Fig. 5 shows that in the ideal scenario with CS capacity set to 1000, the number of CS CacheHits of the routing nodes bb-12662, bb-13050 and gw-12663 decreases intermittently in the form of periodic pulses during the attack phase. However, it quickly returns to its normal value after each attack pulse. Obviously, the number of CS CacheHits is significantly reduced during the attack phase. In other words, I-CIFA will greatly reduce the content cache hit rate for normal data packets in CS, resulting in large content retrieval delay, which greatly affects the performance of NDN networks. The network traffic changes in the number of OutInterest on the different interfaces of different routing nodes are shown in Fig. 6.

Fig. 6 shows that the number of OutInterests on the different interfaces of the routing nodes bb-12662, bb-13050 and gw-12663 decreases intermittently in the form of periodic pulses during the attack phase and it quickly returns to its normal value after each attack pulse. These interfaces are Face 13 of routing node gw-12663, Face 4 of routing node bb-12662, Face 4 of routing node gw-13050, respectively. Since the number of OutInterests is obviously reduced during the attack phase, corresponding data packets will not satisfy a large amount of legitimate Interests sent by consumers, which significantly affects the performance of NDN networks. The network traffic

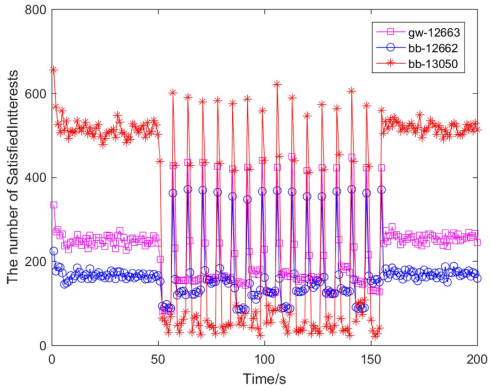


Fig. 7. The number of SatisfiedInterests.

changes in the number of SatisfiedInterest in different routing nodes are shown in Fig. 7.

Fig. 7 shows that the number of SatisfiedInterests of the routing nodes bb-12662, bb-13050 and gw-12663 decreases intermittently in the form of periodic pulses during the attack phase and it quickly returns to its normal value after each attack pulse. As the number of SatisfiedInterests is obviously reduced during the attack phase, consumers will have a low satisfaction with normal Interests, which apparently affects the quality of NDN network service.

V. Detection Mechanism

In this section, BO-GBDT, BO-XGBoost and BO-LightGBM will be proposed to design the BO-GBM fusion algorithm to detect I-CIFA in NDN.

A. Bo-Gbdt, Bo-Xgboost and Bo-Lightgbm

Through analyzing the extracted multi-dimensional network traffic characteristics, attack detection can be regarded as classifying the network traffic characteristics to judge the network states at each moment. In Machine Learning algorithms, Gradient Boosting Machines (GBMs), including GBDT (Gradient Boosting Decision Tree) [6], XGBoost (eXtreme Gradient Boosting) [7] and LightGBM (Light Gradient Boosting Machine) [8], is a gradient boosting method used to solve the problems of classifying and predicting. In addition, multiple tree-based weak classifiers will be established by optimizing their own loss functions in the direction of gradient descent, and a series of weak classifiers will be combined through a specific strategy to obtain a strong classifier. It is worth mentioning that Decision Trees are adopted as the weak classifiers, so GBMs can be seen as a further optimization and improvement of decision trees for attack detection. All in all, GBMs output the probability of predicting whether the current network state is normal or not based on multi-dimensional network traffic characteristics. Among them, the normal network state is marked as 0, and the network state under I-CIFA is marked as 1. We set the threshold as 0.5, and the probability less than this value is judged as 0, that is, the network state is judged to be normal at this time, and the probability greater than this value is judged as 1, that is, the network state is judged to be subject to I-CIFA at this time. In addition, GBMs have their own advantages in attack detection. For example, GBDT can handle data flexibly, XGBoost can prevent overfitting and the training efficiency of LightGBM is great. Specially, they outperform other detection methods in attack detection [6], [7], [8] so we design a stronger classifier to combine them below.

Algorithm 1: Bayesian Optimization.

INPUT: $f, \mathcal{X}, \mathcal{AF}, \mathcal{M}$
$\mathcal{D} \leftarrow$ INITSAMPLES$(f, \mathcal{X})$
for $i \leftarrow |\mathcal{D}|$ **to** N **do**
$p(y|x, \mathcal{D}) \leftarrow$ FIT$(\mathcal{M}, \mathcal{D})$
$x_i \leftarrow \arg\max_{x \in \mathcal{X}} \mathcal{AF}(x, p(y|x, \mathcal{D}))$
$y_i \leftarrow f(x_i)$
$\mathcal{D} \leftarrow \mathcal{D} \cup (x_i, y_i)$
end for

However, the problem of optimizing the parameters of Machine Learning algorithms is difficult to address. Thereby, Bayesian optimization was proposed in the paper [26] to solve the problem. Compared with the parameter optimization methods of grid search and random search, the iterations and granularities of Bayesian optimization can be very small so that Bayesian optimization can efficiently optimize the parameters of GBMs, saving the detection time for I-CIFA.

Before Bayesian optimization, we must first determine the optimized objective function. Herein, GBMs adopt the AUC value with 10-fold cross-validation as their objective functions. AUC stands for Area Under Curve, which means the area under the ROC curve [29]. And the larger the AUC value, the better the model we used. Suppose that the actual samples are a_1, a_2, a_n, the predicted samples are $\hat{a_1}, \hat{a_2}, \hat{a_n}$, and the mean of a_i is $\bar{a_i}$. Thus, AUC can be calculated as

$$AUC = \frac{\sum_i P(M_i, N_i)}{M \times N}, \quad P(M_i, N_i) = \begin{cases} 1, & M_i < N_i \\ 0.5, & M_i = N_i \\ 0, & M_i > N_i \end{cases} \tag{1}$$

where M and N represent normal samples and attack samples, and $P(M_i, N_i)$ is the probability of detecting I-CIFA with a pair of samples consists of M_i and N_i. The details of Bayesian optimization are listed as follows.

Suppose $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ is a set of hyperparameters, where x_n represents the value of a certain hyperparameter. In addition, $f(x) : x \rightarrow R$ represents an unknown objective function with a set of hyperparameters which will be optimized. And the goal of Bayesian optimization is to find the maximum value of the unknown objective function f. The maximum value of the objective function can be expressed as

$$x^* = \arg\max_{x \in \mathcal{X}} f(x) \tag{2}$$

where $x \subseteq \mathcal{X}$, $\mathcal{X}$ represents the search space of f, and x^* represents the value after maximizing the objective function

TABLE III
OPTIMAL PARAMETER COMBINATIONS OF BO-GBDT, BO-XGBOOST AND BO-LIGHTGBM

GBM	Adjusted parameters	Interval	Value
GBDT	max_depth	[5, 15]	9
	max_features	[0.1, 0.999]	0.4545
	n_estimators	[10, 250]	119
	min_samples_split	[2, 25]	13
XGBoost	learning_rate	[0.01, 0.3]	0.01
	max_depth	[1, 10]	10
	reg_alpha	[0.001, 20]	0.001
	n_estimators	[1, 1000]	990
	colsample_bytree	[0.01, 1]	1
	subsample	[0.001, 1]	1
	gamma	[0.001, 10]	0.001
	min_child_weight	[0, 20]	15
LightGBM	colsample_bytree	[0.8, 1]	0.8
	max_depth	[2, 5]	3.2477
	num_leaves	[24, 45]	38
	subsample	[0.8, 1]	0.8
	learning_rate	[0.01, 0.3]	0.02

f. Specifically, the procedure of Bayesian optimization for GBMs is shown in algorithm 1.

Where $\mathcal{D}$ represents a dataset composed of several data pairs (x, y), obtained by initializing the samples $(f, \mathcal{X})$. And $\mathcal{AF}$ represents the Acquisition Function [27]. Through fitting the dataset $\mathcal{D}$, the model $\mathcal{M}$ which obeys the Gaussian distribution is obtained. Firstly, we assume that $y = f(x)$ with a prior probability, then calculate the posterior probability $p(y|x, \mathcal{D})$ as the posterior probability of the next data x based on $\mathcal{M}$. Moreover, optimizing $\mathcal{AF}$, which will be calculated according to the current posterior probability to obtain the maximum value x_i, then y_i is obtained by calculating $f(x_i)$. Finally, the optimal parameters are obtained by $\mathcal{AF}$ that satisfy the formula 2, and $\mathcal{D}$ will be updated afterwards.

There are two key steps during the process of Bayesian optimization [26], one is using Gaussian Process (GP) to fit data and update the posterior distribution of functions, and the other is determining the next evaluation point by $\mathcal{AF}$. There are several methods to meet with $\mathcal{AF}$, such as Probability of Improvement (PI), Expected Improvement (EI), Upper Confidence Bound (UCB) and GP Upper Confidence Bound (GP-UCB) [28], to realize high performance in hyperparameter tuning.

After 40 iterations of the Bayesian optimization for parameter tuning, the AUC values of GBDT, XGBoost and LightGBM reach 0.9731, 0.9815 and 0.9841, respectively. Finally, the optimal parameter combinations of GBMs obtained through Bayesian optimization are shown in Table III. After Bayesian optimization, BO-GBDT, BO-XGBoost and BO-LightGBM are obtained to reduce the detection time for I-CIFA.

B. Detection of I-CIFA

Since the number of the attack samples and the normal samples are inconsistent in general, so stratified K-fold cross-validation is adopted before constructing the BO-GBM fusion algorithm. Taking K=10 because 10-fold cross-validation can realize the best balance between deviation and variance [30]. If K is less than 10, more training data will be used for each iteration, which will cause smaller deviations for the data and make the model to be too rough. If K is greater than 10, due to the number of training copies are more similar to each other, it will cause high variance and lead the model to be overly idealized.

Firstly, the network traffic characteristics are divided into a training set and a test set at a ratio of 7:3 after extracting and preprocessing the network traffic. However, the number of the attack samples and the normal samples are imbalanced in real situations, so the number of attack samples extracted is 1668 and the number of normal samples extracted is 806 in the experiments. Thus, the ratio of attack samples to normal samples is about 2:1. Moreover, stratified 10-fold cross-validation is used to ensure that the proportion of each category in training set and test set is the same as the original network traffic data, which makes the detection results of BO-GBM fusion algorithm more reliable for I-CIFA. Finally, the training set is further divided into 10 folds after using stratified 10-fold cross-validation. Consequently, the 9-fold training set is used as the new training set for training in the current iteration. In order to obtain the model with the best generalization ability and the best predictions, the remaining 1-fold training set is used as the validation set for evaluating the performance of the model and predicting the network state in the current iteration.

From the explanation above, Stacking method is adopted to stack the detection results of validation set in the current model A_i after 10 iterations, where A_i represents BO-GBDT, BO-XGBoost and BO-LightGBM successively. The flow chart of stratified 10-fold cross-validation and Stacking method are shown in Fig. 8, the training set consists of 9-fold new training set in light grey and 1-fold validation set in orange. Through classifying and predicting the network traffic samples, the predictions $a_{i,j}$ are obtained by the current model A_i, which are the probabilities of judging whether the current network state is normal. Furthermore, these 10 predictions are averaged to get the final prediction $P(A_i)$, which will be stacked as the training set of the next-layer model through Stacking method. Finally, the performance of the next-layer model will be evaluated by the test set to get the predictions of the next-layer model.

However, the role of the test set is the same as the validation set, but the difference between them is validation set can be used for training hyperparameters. From the above description, it can be seen that Stacking method is the process of modeling the predictions of different models at the first layer, which can improve the performance of the models [31]. Specifically, the process of BO-GBM fusion algorithm for detecting I-CIFA is shown in algorithm 2.

Firstly, we input the first-layer model A_i, the second-layer model $\mathcal{F}$ and the network traffic dataset $\mathcal{N} = \{(x_1, y_1), (x_2, y_2), \ldots (x_m, y_m)\}$, then $\mathcal{R}_i$ is obtained by modeling A_i from the network traffic dataset $\mathcal{N}$. Obviously, $\mathcal{R}_i$ indicates BO-GBDT, BO-XGBoost and BO-LightGBM successively. Secondly, all the predictions of the first-layer models $\{z_{n1}, z_{n2}, z_{n3}\}$ and the validation set y_n will be added into $\mathcal{N}'$, where $\mathcal{N}'$ represents a preset empty set $\varnothing$. Furthermore, $\mathcal{N}'$ will be the training set of the next-layer model $\mathcal{F}$ which consists of all the

Fig. 8. Stratified 10-fold cross-validation and Stacking method.

Algorithm 2: BO-GBM Fusion Algorithm for Detecting I-CIFA.

INPUT: $A_i, \mathcal{F}, \mathcal{N} = \{(x_1, y_1), (x_2, y_2), \ldots (x_m, y_m)\}$
for $i = 1, 2, 3$ **do**
  $\mathcal{R}_i \leftarrow A_i(\mathcal{N})$
end for
$\mathcal{N}' = \varnothing$
for $n = 1, 2, \ldots m$ **do**
  for $i = 1, 2, 3$ **do**
    $z_{ni} = \mathcal{R}_i(x_n)$
  end for
  $\mathcal{N}' = \mathcal{N}' \cup ((z_{n1}, z_{n2}, z_{n3}), y_n)$
end for
$\mathcal{R}' \leftarrow \mathcal{F}(\mathcal{N}')$
P $\leftarrow \mathcal{R}'(\mathcal{R}_1(x), \mathcal{R}_2(x), \mathcal{R}_3(x))$
return P

predictions of first-layer models $\{z_{n1}, z_{n2}, z_{n3}\}$ and the test set $\{y_1, y_2, \ldots y_m\}$ will be the validation set of the second-layer model to construct the BO-GBM fusion model $\mathcal{R}'$ by evaluating the training set $\mathcal{N}'$. Finally, the final predictions **P** for the network states are obtained by using the BO-GBM fusion model $\mathcal{R}'(\mathcal{R}_1(x), \mathcal{R}_2(x), \mathcal{R}_3(x))$. It can be understood as the nesting process of the models.

However, it is enough to build a two-layer model. In order to avoid overfitting, the second-layer model $\mathcal{F}$ we use is Logistic Regression. For the detection of I-CIFA, an efficient detection mechanism based on BO-GBM fusion algorithm is finally constructed. The flow chart of the detection mechanism is shown in Fig. 9. Firstly, the relevant multi-dimensional features are extracted from network traffic in the experiment, which will be preprocessed and divided into training set and test set. Secondly, stratified 10-fold cross-validation is adopted to further split the training set into 9-fold new training set and 1-fold validation set. Moreover, BO-GBDT, BO-XGBoost and BO-LightGBM are modeling by the 9-fold training set and 1-fold validation set, which are the first-layer models of BO-GBM fusion algorithm. After that, we use Stacking method to construct a two-layer model, namely BO-GBM fusion model. Specially, in order to realize detection of I-CIFA, the predictions of the BO-GBM fusion algorithm **P** are obtained for predicting and judging the network states. Last but not least, the detection performance of BO-GBM fusion algorithm will be evaluated by some specific metrics described below.

VI. Experiment Results and Algorithm Complexity Analysis

In this section, we verify the detection performance of BO-GBM fusion algorithm compared with SVM, Decision Tree, KNN, BO-GBDT, BO-XGBoost, BO-LightGBM, Cumulative Entropy, Gini Impurity, IForest, Wavelet Analysis and Prediction Error in terms of missing alarm rate, false alarm rate, detection rate, complexity, processing time and memory usage.

A. Model Evaluation Metrics

The ROC curve and confusion matrix are excellent metrics to evaluate various models for attack detection. The ordinate label of the ROC curve, namely True Positive Rate (TPR), which represents the recall rate of normal samples. As well as the abscissa label of the ROC curve, namely False Positive Rate (FPR), which represents the recall rate of attack samples. When the closer the ROC curve is to the coordinate (0,1), the better the model, while the AUC value discussed above are obtained from the ROC curve.

The confusion matrix diagram is shown in Fig. 10, where *TP* represents that both the actual and predicted samples are normal samples, and *FP* means that the predicted samples are normal samples, but the actual samples are attack samples.

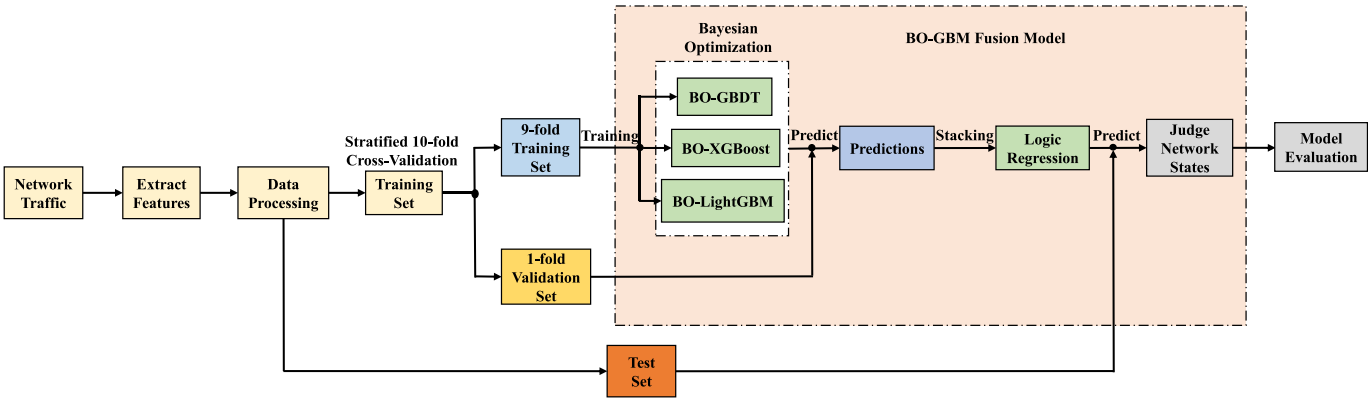


Fig. 9. The flow chart of detecting I-CIFA based on BO-GBM fusion algorithm.

Confusion matrix		Predicted label	
		Normal	Attack
Actual label	Normal	True positive (TP)	False negative (FN)
	Attack	False positive (FP)	Ture negative (TN)

Fig. 10. The confusion matrix diagram.

Similarly, FN means that the predicted samples are attack samples, but the actual samples are normal samples, and TN indicates that both the actual and predicted samples are attack samples. We use the six effective metrics, including missing alarm rate, false alarm rate, detection rate, complexity, processing time, and memory usage, to analyze and evaluate the BO-GBM fusion algorithm.

Herein, the detection rate of attack samples can be expressed as $TN/(TN+FN)$. However, the false alarm rate of attack samples, denoted as FAR, reflects how many attack samples are misclassified as normal, namely $FP/(TN+FP)$. While the missing alarm rate of attack samples, recorded as MAR, reflects how many attack samples are missed to be detected, which can be calculated as $FP/(TP+FP)$. Complexity is a measure of the efficiency of algorithm execution, actually, algorithm complexity includes time complexity and space complexity. On the one hand, time complexity means the increasing trend of code execution time which can be represented by $O(n)$, where n represents the amount of input data and $O(n)$ reflects the proportional relationship between n and the time consuming of programs. On the other hand, space complexity represents the amount of storage space temporarily occupied by an algorithm during its operation. Consequently, processing time is calculated to reflect time complexity and memory usage is calculated to reflect space complexity.

B. Experiment Results

Firstly, we have conducted many experiments and calculated the average values of evaluation metrics as detection results which are shown in Table IV. It can be concluded that BO-GBM fusion algorithm performs better than SVM, Decision Tree, KNN, BO-GBDT, BO-XGBoost and BO-LightGBM in detecting I-CIFA. The detection rate of BO-GBM fusion algorithm is 98.69%, which is higher than the other six Machine Learning algorithms. Additional, the FAR and MAR of BO-GBM fusion algorithm are 1.36% and 1.43%, respectively, which are lower than the other six schemes.

Moreover, Fig. 11 shows the comparison of ROC curves of different Machine Learning algorithms for detecting I-CIFA, the ROC curve of BO-GBM fusion algorithm is the closest to the coordinate (0, 1) compared with the other six algorithms. From the AUC values in the Fig. 11, it can be seen that the AUC value of BO-GBM fusion algorithm is higher than the other algorithms for detecting I-CIFA. These results verify that BO-GBM fusion algorithm has better performance for detecting I-CIFA.

For further verification, the BO-GBM fusion algorithm will be compared with other algorithms which were applied to IFA and CIFA, including Cumulative Entropy [12], Gini Impurity [13] and IForest [20], Wavelet Analysis [14] and Prediction Error [15]. By the way, Cumulative Entropy, Gini Impurity and IForest are applied to detect IFA, while Wavelet Analysis and Prediction Error are designed to detect CIFA. As shown in Table V, the detection rate of IFA detection method based on Cumulative Entropy and Gini Impurity are only 52.42%, 69.38% and 58.34% in detecting I-CIFA, respectively. In addition, the MARs and FARs of Cumulative Entropy, Gini Impurity and IForest are higher than Wavelet Analysis, Prediction Error and BO-GBM fusion algorithm, because the concealment of CIFA and I-CIFA is much higher than that of IFA. Although Wavelet Analysis and Prediction Error have a certain detection effect on I-CIFA, they are worse than the our proposed scheme. It further demonstrates that the BO-GBM fusion algorithm has better detection performance for detecting I-CIFA than the approaches applied to IFA and CIFA.

C. Algorithm Complexity Analysis

Since $O(n)$ reflects the proportional relationship between the amount of input data and the time consuming of programs,

TABLE IV
COMPARISON OF EXPERIMENT RESULTS WITH THE OTHER MACHINE LEARNING SCHEMES FOR DETECTING I-CIFA

Algorithm	SVM	Decision Tree	KNN	BO-GBDT	BO-XGBoost	BO-LightGBM	BO-GBM fusion algorithm
Detection rate	93.05%	95.48%	95.14%	97.05%	97.48%	97.94%	98.69%
FAR	6.47%	4.17%	5.82%	2.94%	2.35%	2.14%	1.36%
MAR	5.94%	4.15%	5.14%	2.85%	2.16%	2.28%	1.43%
AUC	0.9317	0.9661	0.9649	0.9731	0.9815	0.9841	0.9887

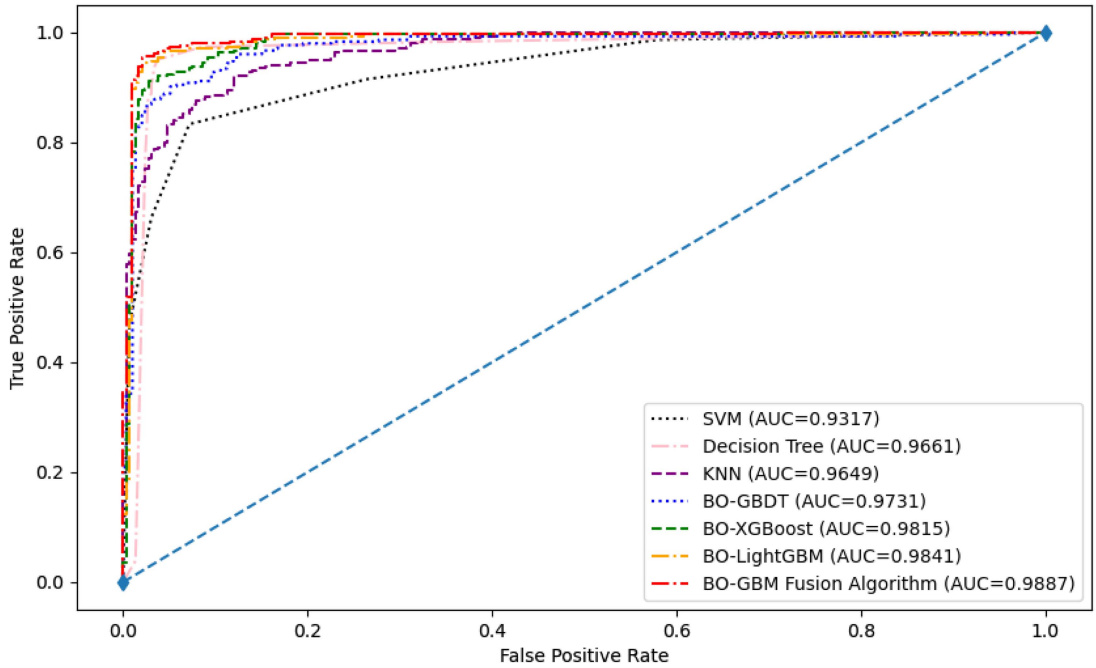


Fig. 11. Comparison of the ROC curves of Machine Learning algorithms for detecting I-CIFA.

TABLE V
COMPARISON OF EXPERIMENT RESULTS WITH CUMULATIVE ENTROPY, GINI IMPURITY, IFOREST, WAVELET ANALYSIS AND PREDICTION ERROR

Algorithm	Cumulative Entropy	Gini Impurity	IForest	Wavelet Analysis	Prediction Error	BO-GBM fusion algorithm
Detection rate	52.42%	69.38%	58.34%	80.18%	90.91%	98.69%
FAR	39.37%	28.65%	36.13%	11.85%	9.37%	1.36%
MAR	33.12%	35.13%	40.89%	18.56%	6.41%	1.43%

the time consumption of Bayesian optimization, Decision Tree, KNN, Cumulative Entropy Gini Impurity, IForest, Wavelet Analysis and Prediction Error in detecting I-CIFA has a linear relationship with the amount of input data, so their time complexity is $O(n)$. Nevertheless, the proportional relationship between the amount of input data and the time consuming of BO-XGBoost in detecting I-CIFA is $O(nlog_2(n))$, that is, BO-XGBoost detects I-CIFA with the complexity $O(nlog_2(n))$. Additional, the time complexity of BO-LightGBM is $O(n)$, because both LightGBM and Bayesian optimization have the time complexity $O(n)$.

Furthermore, stratified 10-fold cross-validation and Stacking method are adopted to fuse BO-GBDT, BO-XGBoost and BO-LightGBM with the complexity $O(n)$. However, the time complexity is sorted from largest to smallest as $O(n^2) > O(nlog_2(n)) > O(n)$, so the time complexity of the first-layer of BO-GBM fusion algorithm is $O(n^2) + O(nlog_2(n)) + O(n) + O(n) = O(n^2)$, whereas the time complexity of the second-layer of BO-GBM fusion algorithm is $O(n)$. Thus, the overall complexity of BO-GBM fusion algorithm is $O(n^2) + O(n) = O(n^2)$. Similarly, the overall complexity of GBM fusion algorithm is also $O(n^2)$. We conducted extensive experiments on the processing time and memory usage for detecting I-CIFA and averaged the results, as shown in Table VI.

As shown in Table VI, the processing time and memory usage required for the BO-GBM fusion algorithm is 0.3601 s and 243.12 MB, while the GBM fusion algorithm requires 2.1071 s and 445.93 MB, respectively. Thus, it verifies that Bayesian optimization can reduce the detection time and overhead for detecting I-CIFA because GBM fusion algorithm needs to find the optimal parameters constantly. The BO-GBM fusion algorithm greatly reduces more detection time and overhead than that of GBM fusion algorithm, SVM and BO-GBDT for detecting I-CIFA, which shows that the BO-GBM fusion

TABLE VI
ALGORITHM COMPLEXITY EVALUATION

Algorithm	Complexity	Processing time/s	Memory usage/MB
BO-GBM fusion algorithm	$O(n^2)$	0.3601	243.12
GBM fusion algorithm	$O(n^2)$	2.1071	445.93
BO-GBDT	$O(n^2)$	1.7081	371.32
BO-XGBoost	$O(nlog_2(n))$	0.2462	201.92
BO-LightGBM	$O(n)$	0.0889	93.73
Decision Tree	$O(n)$	0.0964	104.06
KNN	$O(n)$	0.0987	108.68
SVM	$O(n^2)$	1.9351	395.87
Cumulative Entropy	$O(n)$	0.0922	121.03
Gini Impurity	$O(n)$	0.1573	172.81
IForest	$O(n)$	0.1226	153.14
Wavelet Analysis	$O(n)$	0.1824	195.98
Prediction Error	$O(n)$	0.1356	156.79

algorithm has better detection performance. But the detection time of our proposed mechanism is greater than that of KNN, Decision Tree, Cumulative Entropy, Gini Impurity, IForest, Wavelet Analysis, Prediction Error, BO-XGBoost and BO-LightGBM. Consequently, BO-GBM fusion algorithm trades a little more detection time and overhead for higher detection rate. However, the memory usage and the detection time of BO-GBM fusion algorithm are close to those of the above night algorithms. It also demonstrates that our proposed scheme has real-time performance in detecting I-CIFA.

VII. CONCLUSION AND FUTURE DIRECTIONS

In this article, we experimentally extract and analyze the multi-dimensional network traffic characteristics under I-CIFA on NDN networks. For detecting I-CIFA in NDN, we adopt stratified 10-fold cross-validation and Stacking method to fuse BO-GBDT, BO-XGBoost and BO-LightGBM and then construct a two-layer structure to get the BO-GBM fusion algorithm to judge the network states. The experiment results show that BO-GBM fusion algorithm has better detection performance than SVM, Decision Tree, KNN, BO-GBDT, BO-XGBoost, BO-LightGBM, Cumulative Entropy, Gini Impurity, IForest, Wavelet Analysis and Prediction Error, and it's practicality has been validated through algorithm complexity analysis. However, the original network traffic data is large and complex, network traffic characteristics are difficult to extract and analyze. In the future, more effective methods will be proposed to process network traffic and evaluate the correlation between network traffic characteristics and network states. In order to improve our detection performance, it is feasible to extract more effective network traffic characteristics and use more efficient mechanisms. If this work is to be extended, some effective countermeasures against I-CIFA in NDN will be introduced in the future.

REFERENCES

[1] L. Zhang et al., "Named data networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, 2014.

[2] N. Tan et al., "Reliable detection of interest flooding attack in real deployment of named data networking," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 9, pp. 2470–2485, Sep. 2019.

[3] A. Afanasyev, P. Mahadevan, I. Moiseenko, E. Uzun, and L. Zhang, "Interest flooding attack and countermeasures in named data networking," in *Proc. IFIP Netw. Conf.*, 2013, pp. 1–9.

[4] Z. Wu et al., "I-CIFA: An improved collusive interest flooding attack in named data networking," *J. Inf. Secur. Appl.*, vol. 61, no. 2, 2021, Art. no. 102912.

[5] Q. Li, P. P. C. Lee, P. Zhang, P. Su, L. He, and K. Ren, "Capability-based security enforcement in named data networking," *IEEE/ACM Trans. Netw.*, vol. 25, no. 5, pp. 2719–2730, Oct. 2017, doi: 10.1109/TNET.2017.2715822.

[6] T. Yang, L. Chen, J. Wang, Z. Cui, and J. Qi, "Anomaly detection of dust removal system through gradient boosting decision tree algorithm," in *Proc. IEEE Int. Conf. Commun., Inf. Syst. Comput. Eng.*, 2021, pp. 685–688, doi: 10.1109/CISCE52179.2021.9445934.

[7] Y. Sun, C. Song, S. Yu, H. Pan, T. Li, and Y. Liu, "A novel genetic Algorithm-XGBoost based intrusion detection method," in *Proc. IEEE 4th Adv. Inf. Management, Communicates, Electron. Automat. Control Conf.*, 2021, pp. 1–5, doi: 10.1109/IMCEC51613.2021.9482357.

[8] W. Tong, B. Liu, Z. Li, J. Lin, and X. Jin, "Intrusion detection method of industrial control network based on lightgbm," in *Proc. IEEE Int. Conf. Commun., Inf. Syst. Comput. Eng.*, 2021, pp. 631–635, doi: 10.1109/CISCE52179.2021.9445956.

[9] G. Xiaoqing, G. Hebin, and C. Luyi, "Network intrusion detection method based on agent and SVM," in *Proc. IEEE 2nd Int. Conf. Inf. Manage. Eng.*, 2010, pp. 399–402, doi: 10.1109/ICIME.2010.5477694.

[10] Z. Ma and A. Kaban, "K-Nearest-Neighbours with a novel similarity measure for intrusion detection," in *Proc. IEEE 13th U.K. Workshop Comput. Intell.*, 2013, pp. 266–271, doi: 10.1109/UKCI.2013.6651315.

[11] J. Wang, Q. Yang, and D. Ren, "An intrusion detection algorithm based on decision tree technology," in *Proc. IEEE Asia-Pacific Conf. Inf. Process.*, 2009, pp. 333–35, doi: 10.1109/APCIP.2009.218.

[12] Y. Xin, Y. Li, W. Wang, W. Li, and X. Chen, "A novel interest flooding attacks detection and countermeasure scheme in NDN," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, 2016, pp. 1–7, doi: 10.1109/GLOCOM.2016.7841526.

[13] T. Zhi, H. Luo, and Y. Liu, "A gini impurity-based interest flooding attack defence mechanism in NDN," *IEEE Commun. Lett.*, vol. 22, no. 3, pp. 538–541, Mar. 2018.

[14] Y. Xin, Y. Li, W. Wang, W. Li, and X. Chen, "Detection of collusive interest flooding attacks in named data networking using wavelet analysis," in *Proc. IEEE Mil. Commun. Conf.*, 2017, pp. 557–562.

[15] L. Liu et al., "The detection method of collusive interest flooding attacks based on prediction error in NDN," *IEEE Access*, vol. 8, pp. 128005–128017, 2020.

[16] R. Jeet, P. Arun Raj Kumar, "A survey on interest packet flooding attacks and its countermeasures in named data networking," *Int. J. Inf. Secur.*, pp. 1–25, 2022. [Online]. Available: https://doi.org/10.1007/s10207-022-00591-w

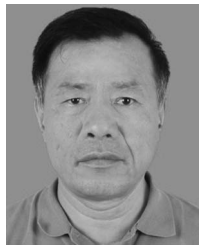
[17] A. Compagno, M. Conti, P. Gasti, and G. Tsudik, "Poseidon: Mitigating interest flooding DDoS attacks in named data networking," in *Proc. IEEE Conf. Local Comput. Netw.*, Sydney, 2013, pp. 630–638.

[18] G. Cheng et al., "Detecting and mitigating A sophisticated interest flooding attack in NDN from the network-wide view," in *Proc. IEEE 1st Int. Workshop Netw. Meets Intell. Computations*, 2019, pp. 7–12.

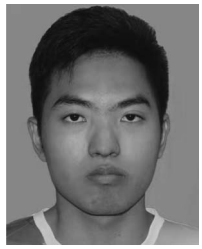
[19] H. Salah, J. Wulfheide, and T. Strufe, "Lightweight coordinated defence against interest flooding attacks in NDN," in *Proc. INFOCOM Workshops*, Hong Kong, China, 2015, pp. 103–104.

[20] J. Chen, G. Xing, M. Cui, H. Huo, and R. Hou, "Isolation forest based interest flooding attack detection mechanism in NDN," in *Proc. IEEE 2nd Int. Conf. Hot Inf.-Centric Netw.*, 2019, pp. 58–62, doi: 10.1109/HotICN48464.2019.9063205.

[21] G. Xing et al., "Isolation forest-based mechanism to defend against interest flooding attacks in named data networking," *IEEE Commun. Mag.*, vol. 59, no. 3, pp. 98–103, Mar. 2021, doi: 10.1109/MCOM.001.2000368.
[22] J. Zhou, J. Luo, L. Deng, and J. Wang, "Defense mechanism of interest flooding attack based on deep reinforcement learning," in *Proc. IEEE 3rd Int. Conf. Hot Inf.-Centric Netw.*, 2020, pp. 65–70, doi: 10.1109/HotICN50779.2020.9350852.
[23] H. Salah and T. Strufe, "Evaluating and mitigating a collusive version of the interest flooding attack in NDN," in *Proc. IEEE Symp. Comput. Commun.*, 2016, pp. 938–945, doi: 10.1109/ISCC.2016.7543857.
[24] Z. Wu et al., "Mitigation measures of collusive interest flooding attacks in named data networking - ScienceDirect," *Comput. Secur.*, vol. 97, 2020, Art. no. 101971.
[25] T. Shigeyasu and A. Sonoda, "Detection and mitigation of collusive interest flooding attack on content centric networking," *Int. J. Grid Utility Comput.*, vol. 11, no. 1, pp. 21–29, 2020.
[26] A. Klein et al., "Fast Bayesian Optimization of machine learning hyperparameters on large datasetsm," in *Artificial Intelligence and Statistics (AISTATS)*. Fort Lauderdale, Florida, USA: PMLR, 2016, pp. 528–536.
[27] E. Brochu, V. M. Cora, and N. D. Freitas, "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," *Comput. Sci.*, pp. 1–49, 2010.
[28] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proc. IEEE*, vol. 104, no. 1, pp. 148–175, Jan. 2016, doi: 10.1109/JPROC.2015.2494218
[29] A. M. Carrington et al., "Deep ROC analysis and AUC as balanced average accuracy, for improved classifier selection, audit and explanation," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jan. 25, 2022, doi: 10.1109/TPAMI.2022.3145392.
[30] R. MilidiúLuiz, and L. Müller, "SeismoGlow - Data augmentation for the class imbalance problem," 2020, *arXiv:2007.12229*.
[31] B. Pavlyshenko, "Using stacking approaches for machine learning models," in *Proc. IEEE 2nd Int. Conf. Data Stream Mining Process.*, 2018, pp. 255–258, doi: 10.1109/DSMP.2018.8478522.
[32] A. Benmoussa et al., "A novel congestion-aware interest flooding attacks detection mechanism in named data networking," in *Proc. IEEE 28th Int. Conf. Comput. Commun. Netw.*, 2019, pp. 1–6, doi: 10.1109/ICCCN.2019.8847146.
[33] R. Hou et al., "Theil-based countermeasure against interest flooding attacks for named data networks," *IEEE Netw.*, vol. 33, no. 3, pp. 116–121, May/Jun. 2019, doi: 10.1109/MNET.2019.1800350.
[34] A. Benmoussa et al., "MSIDN: Mitigation of sophisticated interest flooding-based DDoS attacks in named data networking," *Future Gener. Comput. Syst.*, vol. 107, pp. 293–306, Jun. 2020.
[35] A. Benmoussa, C. A. Kerrache, N. Lagraa, S. Mastorakis, A. Lakas, and A. el Karim Tahari, "Interest flooding attacks in named data networking: Survey of existing solutions, open issues, requirements and future directions," *ACM Comput. Surv.*, to be published, doi: 10.1145/3539730.

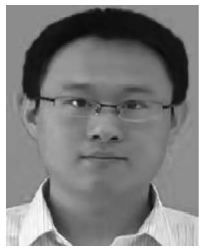


Zhijun Wu received the B.S. and M.S. degrees in information processing from Xidian University, Xi'an, China, and the Ph.D. degree in cryptography from the Beijing University of Posts and Telecommunications, Beijing, China. He was a Professor with the College of Safety Science and Engineering, Civil Aviation University of China, Tianjin, China. His research interests include denial-of-service attacks, and security in Big Data and cloud computing.



Silin Peng (Student Member, IEEE) received the B.S. degree from Guangdong Polytechnic Normal University, Guangzhou, China. He is currently working toward the master's degree in information security with the College of Electronic Information and Automation, Civil Aviation University of China, Tianjin, China. His research focuses on security of named data networking.

Liang Liu received the master's degree in communication and information system from the Civil Aviation University of China, Tianjin, China. He is currently an Assistant Experimenter with the College of Safety Science and Engineering, Civil Aviation University of China. His main research interests include network information security, including defense of future network security, and denial of service attacks.



Meng Yue received the Ph.D. degree in information and communication engineering from Tianjin University, Tianjin, China, in 2017. He is currently an Associate Professor with the College of Safety Science and Engineering, Civil Aviation University of China, Tianjin, China. His research interests include information security and cloud computing.