frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 22 April 2024
DOI 10.3389/fphy.2024.1350117

OPEN ACCESS

EDITED BY
Jianping Gou,
Southwest University, China

REVIEWED BY
Peilin He,
University of Pittsburgh, United States
Xiaolei Liu,
China Academy of Engineering Physics, China

*CORRESPONDENCE
Weisha Zhang,
✉ zhangweisha@uestc.edu.cn

RECEIVED 05 December 2023
ACCEPTED 20 March 2024
PUBLISHED 22 April 2024

CITATION
Zhang W, Liu J, Peng J, Liu Q and Yu K (2024), Malware traffic detection based on type II fuzzy recognition.
Front. Phys. 12:1350117.
doi: 10.3389/fphy.2024.1350117

Malware traffic detection based on type II fuzzy recognition

Weisha Zhang[1]*, Jiajia Liu[2], Jimin Peng[2], Qiang Liu[2] and Kun Yu[2]

[1]School of Foreign Languages, University of Electronic Science and Technology of China, Chengdu, China, [2]Advanced Cryptography and System Security Key Laboratory of Sichuan Province, Chengdu, China

In recent years, a surge in malicious network incidents and instances of network information theft has taken place, with malware identified as the primary culprit. The primary objective of malware is to disrupt the normal functioning of computers and networks, all the while surreptitiously gathering users' private and sensitive information. The formidable concealment and latency capabilities of malware pose significant challenges to its detection. In light of the operational characteristics of malware, this paper conducts an initial analysis of prevailing malware detection schemes. Subsequently, it extracts fuzzy features based on the distinct characteristics of malware traffic. The approach then integrates traffic detection techniques with Type II fuzzy recognition theory to effectively monitor malware-related traffic. Finally, the paper classifies the identified malware instances according to fuzzy association rules. Experimental results showcase that the proposed method achieves a detection accuracy exceeding 90%, with a remarkably low false alarm rate of approximately 5%. This method adeptly addresses the challenges associated with malware detection, thereby making a meaningful contribution to enhancing our country's cybersecurity.

KEYWORDS

traffic detection, Malware, fuzzy recognition, feature extraction, association rules

1 Introduction

The Internet hosts various forms of malware, including botnets, network worms, and malicious phishing websites. This category of malware exhibits distinct characteristics of malicious network behaviors, such as spam dissemination, the presence of malicious crawlers, dos attack, and port scanning. These activities have a detrimental impact on the network data security of both users and enterprises, posing a significant threat to the network information security of society and the country. Malware has the capability to establish a persistent, malicious controlling network topology that is highly contagious. Port-scanning malware conducts polling attacks on the ports of target computers, particularly targeting commonly used 80. Once the port is attacked and occupied by malware, it significantly disrupts the normal operation of web pages and hampers users' regular Internet activities.

Computer users frequently navigate through a substantial number of web pages during their internet browsing activities. Consequently, numerous malware instances are deployed across a plethora of websites. This situation exposes users to considerable risks as they traverse the web, increasing the likelihood of falling into traps that can result in the compromise of their information, privacy, and property. This phenomenon poses a significant threat to both social and national internet security, eroding the trust of the majority of internet users in the national internet security system.

Simultaneously, the structure of internet cybersecurity is intricate, making it challenging to precisely characterize the evolving features of data traffic in certain research fields due to the inherent uncertainty in research outcomes. Model mathematics emerges as a solution to address this challenge. The primary role of fuzzy mathematics is to effectively blur the boundaries of dichotomous problems. Consequently, this paper employs fuzzy recognition methodology for the detection of malware traffic in network communication.

The article has two innovative points:

Firstly, it involves conducting a statistical analysis of malware traffic through the application of fuzzy recognition theory. Utilizing fuzzy membership functions, a substantial volume of traffic is assessed, and the ambiguity characteristics are employed to extract the value range denoting the maliciousness of each traffic.

Secondly, the malware detection technique, supported by the innovative fuzzy mathematics theory, is aptly tailored for the demands of the current Internet era. Furthermore, tools for malware identification based on fuzzy recognition are anticipated to gain widespread adoption among a diverse range of cybersecurity enterprises and professionals, fueled by continuous innovation and improvement.

The remainder of this article is structured as follows. Section 2 is the related works of this paper and introduces the theoretical foundation and methodology of this paper. Section 3 describes Malware Feature Extraction. Section 4 describes Malware identification and classification. Section 5 discusses the experiments and experimental results. Section 6 concludes the paper with some final remarks and future research directions.

2 Related works

With the rapid proliferation of malware, traditional static analysis techniques are no longer sufficient to meet the demands of detection. Consequently, the adoption of fuzzy recognition theory for classifying and detecting malware has become increasingly prominent. An illustrative example of this is the application of fuzzy recognition to analyze network patterns and behavioral styles. Fuzzy mathematics [1] represents a contemporary branch of mathematics that emerged in the 20th century. It relies on fuzzy concepts to enable the estimation and computation of subjects that are not readily addressed by classical mathematics.

In some foreign countries, current practices involve employing two-way traffic analysis [2] and sensory inspection of network data packets [3] to detect specific states of malware, such as inward scanning, exploits, egg downloading, outward parallel sessions, *etc.* When these particular states align with predefined rules, they are identified as malicious traffic.

In certain security domains in China, the analysis of malware traffic predominantly relies on the WinPcap [4] function library, supplemented by external dependent software and applications. Enterprises leverage their internal functions. An application designed for monitoring network traffic or a user-friendly desktop application is developed in alignment with the specified software system functionality and the assessment of malicious traffic [5]. Subsequently, the proposed scheme outlines the specific system structure and optimization process diagram for each module.

In general, numerous cybersecurity projects, both domestically and internationally, have proposed effective solutions to mitigate excessive reliance on source IP, target IP, and the number of host ports during traffic monitoring. This particular scheme involves analyzing whether the uplink effective load and total downlink load of Internet traffic contain distinctive signatures or markers associated with known malicious programs or software for traffic classification [6]. It subsequently calculates fuzzy feature values, thereby achieving a high level of accuracy to some extent. Despite its accuracy, this solution entails a high analytical cost and demands significant effort. To alleviate resource consumption in terms of cost, time, and space, it can be synergistically employed with already analyzed and low-cost monitoring methods. This way, it can efficiently filter out straightforward and easily analyzable traffic in the initial stages.

In addressing the aforementioned challenges, this article employs malware detection tools grounded in fuzzy mathematics as the theoretical foundation, with fuzzy recognition theory serving as the detection method. Through extensive experimentation involving data statistics and analysis of data traffic packet captures, the study aims to offer practical assistance in enhancing the security of personal computers or enterprise extranets. The objective is to furnish efficient tools for inspecting malware traffic and analyzing data packets. By scrutinizing traffic characteristics, along with the expansive range of malware traffic, and employing fuzzy membership function calculations, the proposed approach aims to effectively and efficiently identify malware within IoT or personal computer network cards.

3 Malware feature extraction

3.1 Feature classification

Cluster analysis serves as a pivotal method in fuzzy feature classification. Cluster analysis serves as a valuable tool in identifying fuzzy patterns and similarities within data, providing a means to navigate the inherent ambiguity and uncertainty associated with the classification of such fuzzy features. Additionally, cluster analysis aids in the exploration of the intrinsic structure and patterns embedded in the data. This facet is particularly crucial for the classification of fuzzy features, as they may be inherent in the data's structure and can be better comprehended and recognized through the clustering process.

Moreover, the method's applicability to large-scale datasets further enhances its significance in the realm of fuzzy feature classification. These attributes collectively render cluster analysis advantageous and highly applicable in addressing the challenges posed by fuzzy features in classification tasks.

In this experiment, the classification of traffic features is categorized into five intervals based on the similarity of features and their close relationships: malicious traffic, approximate

TABLE 1 Fuzzy feature range of traffic.

	1 (%)	2	3	4 (%)	5 (%)
all_pkts	94.3–100	66.8%–84.4%	44.1%–47.1%	18.5–32.7	0–4.2
up_pkts	218–230	122.8%–160%	75.9%–97.2%	25.1–58.2	0–22.2
dw_pkts	97.2–99.9	92.9%	35.7%–42.4%	27–25.2	0–8.5
up_pl_pkts	84.7–100	66.7%–71%	38.5%–52.8%	22.3–36.3	0–9.7
dw_pl_pkts	94.3–100	66.8%–84.4%	44.1%–47.1%	18.5–32.7	0–4.2
up_pl_byte	218–230	122.8%–160%	75.9%–97.2%	25.1–58.2	0–22.2
dw_pl_byte	97.2–99.9	92.9%	35.7%–42.4%	27–25.2	0–8.5
duration	84.7–100	66.7%–71%	38.5%–52.8%	22.3–36.3	0–9.7
up_avg_plsize	94.3–100	66.8%–84.4%	44.1%–47.1%	18.5–32.7	0–4.2
dw_avg_plsize	218–230	122.8%–160%	75.9%–97.2%	25.1–58.2	0–22.2
up_min_plsize	97.2–99.9	92.9%	35.7%–42.4%	27–25.2	0–8.5
dw_min_plsize	84.7–100	66.7%–71%	38.5%–52.8%	22.3–36.3	0–9.7
up_max_plsize	218–230	122.8%–160%	75.9%–97.2%	25.1–58.2	0–22.2
dw_max_plsize	97.2–99.9	92.9%	35.7%–42.4%	27–25.2	0–8.5
up_stdev_size	84.7–100	66.7%–71%	38.5%–52.8%	22.3–36.3	0–9.7
dw_stdev_size	94.3–100	66.8%–84.4%	44.1%–47.1%	18.5–32.7	0–4.2
up_avg_ipt	218–230	122.8%–160%	75.9%–97.2%	25.1–58.2	0–22.2
dw_avg_ipt	97.2–99.9	92.9%	35.7%–42.4%	27–25.2	0–8.5
up_min_ipt	84.7–100	66.7%–71%	38.5%–52.8%	22.3–36.3	0–9.7
dw_min_ipt	84.7–100	66.7%–71%	38.5%–52.8%	22.3–36.3	0–9.7
up_max_ipt	94.3–100	66.8%–84.4%	44.1%–47.1%	18.5–32.7	0–4.2
dw_max_ipt	218–230	122.8%–160%	75.9%–97.2%	25.1–58.2	0–22.2
up_stdev_ipt	97.2–99.9	92.9%	35.7%–42.4%	27–25.2	0–8.5
dw_stdev_ipt	84.7–100	66.7%–71%	38.5%–52.8%	22.3–36.3	0–9.7

malicious traffic, no obvious features, approximate normal traffic, and normal traffic. Striking a balance is crucial; excessive intervals (>5) can diminish recognition accuracy, leading to frequent results spanning two intervals simultaneously, thereby introducing ambiguity. Conversely, few classification intervals (0–5) can also elevate recognition ambiguity, making it challenging to clearly discern the malicious nature of the traffic. Achieving an optimal number of intervals is key to ensuring accurate and unambiguous traffic feature classification results.

KM fuzzy clustering [7] classifies the feature classification of malicious traffic into five intervals. This method groups the values recorded under the same quantitative feature into the target dataset. For each interval, it extracts the maximum and minimum values, using the minimum value as the closed left endpoint and the maximum value as the closed right endpoint of the interval. A comprehensive analysis and summary of numerous sets of malicious data have been conducted, as illustrated in Table 1. The data characteristics represented are: total number of data packets, number of uplink data packets, number of downlink data packets, number of uplink loads, number of downlink loads, total uplink load, total downlink load, flow duration, uplink data Avg, downlink data Avg, Uplink minimum load, downlink minimum load, uplink maximum load, downlink maximum load, uplink load variance, downlink load variance, uplink load Avg, downlink load Avg, uplink minimum data, downlink minimum data, uplink maximum data, downlink maximum data, uplink data Variance, downward data variance.

3.2 Feature extraction

3.2.1 Feature calculation

Feature extraction represents a pivotal step in fuzzy recognition. To extract fuzzy features, the initial task involves determining the weight of each traffic feature. Subsequently, an extensive examination and analysis of the range of each feature across all malicious traffic instances are conducted using big data. The testing data for this experiment is sourced from a malware simulator, which generates malicious traffic. This traffic is then combined with normal traffic. The membership function is pre-established based on the characteristics of traffic emitted by known malware, resulting in a unique function. Following this, the dataset is utilized for testing, aiming to identify the number of malicious traffic instances, analyze the type of malware, and ultimately calculate the proportion and false alarm rate.

In the experimental section, we scrutinize the features of captured malicious network samples and public datasets, extracting distinct characteristics of malware traffic. These characteristics encompass the five-tuple [8], packet size, port number, DNS response time, and data packet load. Each type of malware exhibits its unique traits. During the statistical analysis, we filter out all malicious traffic instances, focusing on extracting abnormal characteristics from malicious traffic and HTTP network traffic.

This experiment primarily employs the method of fuzzy cluster analysis for malware identification. In traffic clustering, we utilize common clustering algorithms such as database scanning, memory sharing, K-Means, and design pattern [9]. Particularly, when handling substantial data with high concurrency, the KM algorithm proves effective in revealing the actual distribution and transmission of traffic.

As a result, the membership function will be translated into program code, and the likelihood of malicious traffic will be calculated by executing the program.

TABLE 2 Traffic feature range and weight.

	Weights (%)	Upper range (%)	Lower bound of range (%)	Optimal number of classifications
all_pkts	84.4	44.1–47.1	18.5–32.7	4
up_pkts	31	75.9–97.2	25.1–58.2	3
dw_pkts	92.9	35.7–42.4	27–25.2	5
up_pl_pkts	71.3	38.5–52.8	22.3–36.3	4
dw_pl_pkts	84.4	44.1–47.1	18.5–32.7	5
up_pl_byte	45	75.9–97.2	25.1–58.2	3
dw_pl_byte	92.9	35.7–42.4	27–25.2	3
duration	71	38.5–52.8	22.3–36.3	4
up_avg_plsize	84.4	44.1–47.1	18.5–32.7	5
dw_avg_plsize	21.1	75.9–97.2	25.1–58.2	5
up_min_plsize	92.9	35.7–42.4	27–25.2	3
dw_min_plsize	71	38.5–52.8	22.3–36.3	3
up_max_plsize	36.2	75.9–97.2	25.1–58.2	4
dw_max_plsize	92.9	35.7–42.4	27–25.2	5
up_stdev_plsize	1	38.5–52.8	22.3–36.3	4
dw_stdev_plsize	3.3	44.1–47.1	18.5–32.7	5
up_avg_ipt	9.1	75.9–97.2	25.1–58.2	3
dw_avg_ipt	92.9	35.7–42.4	27–25.2	4
up_min_ipt	71	38.5–52.8	22.3–36.3	5
dw_min_ipt	71	38.5–52.8	22.3–36.3	3
up_max_ipt	84.4	44.1–47.1	18.5–32.7	3
dw_max_ipt	7.4	75.9–97.2	25.1–58.2	5
up_stdev_ipt	92.9	35.7–42.4	27–25.2	3
dw_stdev_ipt	1	38.5–52.8	22.3–36.3	4

We can summarize that there are three types of fuzzy membership functions with a normal distribution:

$$A(x)\begin{cases} 1 & x \le a \\ e^{\frac{-(x-a)}{\delta}} & x > b \end{cases} \tag{1}$$

$$A(x)\begin{cases} 1 & x \le a \\ e^{\frac{-(x-a)}{\delta}} & x > a \end{cases} \tag{2}$$

$$A(x)\begin{cases} 1 & x \le a \\ e^{\frac{-(x-a)}{\delta}} & x > b \end{cases} \tag{3}$$

In this experiment, the network is modeled, analyzed, and detected based on the ecological characteristics of malicious traffic. These ecological characteristics encompass the utilization of commands to control communication traffic, which is generated during the propagation of the network and when the malware reaches a certain scale.

During the generation and dissemination of malicious traffic by malicious application software, fuzzy mathematics establishes its own models and conducts extensive calculations and statistical analysis on the fuzzy characteristics of network data structures. By considering the fuzzy characteristics specific to malicious network traffic, we ultimately perform clustering on such traffic. Leveraging big data investigation methods, we monitor malicious network traffic to detect common malicious software and applications. The fuzzy recognition scheme, based on cluster analysis, utilizes fuzzy clustering to analyze malware and identify the technical core of the operating system. Employing high-speed mirroring [10] for saving malicious network traffic, it acts as a snapshot, subsequently stored on the computer's hard disk. This traffic is then input into a malicious network identification system based on cluster analysis. The system filters, monitors, and analyzes the traffic, extracts features, and finally conducts cluster analysis to determine the accuracy or success rate of all enterprise or personal traffic received.

3.2.2 Flow characteristics

The optimal number of classifications is a crucial concept in data feature segmentation within fuzzy recognition theory. Given the intricate nature of traffic features, different characteristics exhibit variations in their thresholds after the application of the fuzzy clustering analysis algorithm. The optimal number of classifications is defined as the number at which the threshold is maximized. At this point, the membership function is most accurate in determining the feature, resulting in the highest fuzzy recognition rate. Subsequently, we calculate the similarity of each feature after determining the optimal number of classifications for individual features. Ultimately, the optimal number of classifications for the entire set of traffic features is determined using the maximum number algorithm, resulting in five intervals.

To align with the feature function described in fuzzy mathematics, this experiment aims to automatically identify fuzzy features, learning their range and weight, as illustrated in Table 2.

3.3 Feature membership function

Since malicious traffic generated by different malware exhibits significant variations in the range of fuzzy features, the fuzzy features' membership function is fine-tuned with the aid of artificial learning. Leveraging an extensive analysis of massive malicious traffic techniques, we scrutinize and compare identical features to ascertain the average value and range of each fuzzy data level. Subsequently, we incorporate this information into the standard fuzzy membership function. Through program recognition and the application of a series of mathematical algorithms, we amalgamate normal distribution characteristics with the membership function of fuzzy features. Each traffic encompasses dozens of fuzzy features, and distinct types of traffic are associated with unique fuzzy features. To illustrate, consider the following fuzzy feature along with its corresponding membership function:

up_pkts:

$$W(x)\begin{cases} 0 & 0 \le x \le 23.7 \\ 1-e\left(-\dfrac{x-5}{10}\right) & 23.7 \le x \le 41.2 \\ 1-e^{\left(-\frac{x}{8}\right)} & 41.2 \le x \le 60 \\ e^{\left(-\frac{x-5}{6}\right)} & 60 \le x \le 81 \\ 1 & 81 \le x \le 100 \end{cases} \tag{4}$$

dw_pkts:

$$W(x)\begin{cases} 0 & 0 \le x \le 23.7 \\ 1-e\left(-\dfrac{x-5}{10}\right) & 23.7 \le x \le 41.2 \\ 1-e^{\left(-\frac{x}{8}\right)} & 41.2 \le x \le 60 \\ e^{\left(-\frac{x-5}{6}\right)} & 60 < x \le 81 \\ 1 & 81 \le x \le 100 \end{cases} \tag{5}$$

up_pl_pkts:

$$W(x)\begin{cases} 0 & 0 \le x \le 23.7 \\ 1-e\left(-\dfrac{x-5}{10}\right) & 23.7 \le x \le 41.2 \\ 1-e^{\left(-\frac{x}{8}\right)} & 41.2 \le x \le 60 \\ e^{\left(-\frac{x-5}{6}\right)} & 60 < x \le 81 \\ 1 & 81 \le x \le 100 \end{cases} \tag{6}$$

dw_pl_pkts:

$$W(x)\begin{cases} 0 & 0 \le x \le 23.7 \\ 1-e\left(-\dfrac{x-5}{10}\right) & 23.7 \le x \le 41.2 \\ 1-e^{\left(-\frac{x}{8}\right)} & 41.2 \le x \le 60 \\ e^{\left(-\frac{x-5}{6}\right)} & 60 < x \le 81 \\ 1 & 81 \le x \le 100 \end{cases} \tag{7}$$

Upstream Payload Variance:

$$W(x)\begin{cases} 0 & 0 \le x \le 23.7 \\ 1-e\left(-\dfrac{x-5}{10}\right) & 23.7 \le x \le 41.2 \\ 1-e^{\left(-\frac{x}{8}\right)} & 41.2 \le x \le 60 \\ e^{\left(-\frac{x-5}{6}\right)} & 60 < x \le 81 \\ 1 & 81 \le x \le 100 \end{cases} \tag{8}$$

Downstream Payload Variance:

$$W(x)\begin{cases} 0 & 0 \le x \le 23.7 \\ 1-e\left(-\dfrac{x-5}{10}\right) & 23.7 \le x \le 41.2 \\ 1-e^{\left(-\frac{x}{8}\right)} & 41.2 \le x \le 60 \\ e^{\left(-\frac{x-5}{6}\right)} & 60 < x \le 81 \\ 1 & 81 \le x \le 100 \end{cases} \tag{9}$$

In the traffic analysis of certain software, the time interval proves to be a crucial feature. Therefore, it is essential to examine the fuzzy feature of time intervals. This involves calculating the minimum, maximum, average, and variance of the uplink and downlink time intervals. Subsequently, the weight of the time interval in the overall fuzzy recognition feature is determined through computation:

Upstream Mean Time Interval:

$$W(x)\begin{cases} 0 & 0 \le x \le 23.7 \\ 1-e\left(-\dfrac{x-5}{10}\right) & 23.7 \le x \le 41.2 \\ 1-e^{\left(-\frac{x}{8}\right)} & 41.2 \le x \le 60 \\ e^{\left(-\frac{x-5}{6}\right)} & 60 < x \le 81 \\ 1 & 81 \le x \le 100 \end{cases} \tag{10}$$

Downstream Mean Time Interval:

$$W(x)\begin{cases} 0 & 0 \le x \le 23.7 \\ 1-e\left(-\dfrac{x-5}{10}\right) & 23.7 \le x \le 41.2 \\ 1-e^{\left(-\frac{x}{8}\right)} & 41.2 \le x \le 60 \\ e^{\left(-\frac{x-5}{6}\right)} & 60 < x \le 81 \\ e^{\left(-\frac{x-5}{6}\right)} & 81 < x \le 100 \end{cases} \tag{11}$$

4 Malware identification and classification

4.1 Fuzzy recognition process

Firstly, in fuzzy recognition theory, the identification process begins with defining the object to be recognized. Subsequently, the fuzzy characteristics of the target object are analyzed. Finally, ambiguity is calculated through functions, and the result interval is determined to achieve the recognition and classification of malware. For the experiment, the identification process is divided into the following steps:

1. Identifying the target object involves the initial step of capturing and filtering data packets that satisfy specific identification criteria. Subsequently, the traffic is regarded as the object of identification in accordance with the five-tuple principle.
2. Conducting fuzzy feature calculation and classification involves the determination of fuzzy features within the data flow. Subsequently, cluster analysis is applied to ascertain the optimal number of interval classifications for each identified feature. Ultimately, this process culminates in the establishment of classification intervals for the entirety of the fuzzy recognition procedure.
3. Following the establishment of classification intervals, the evaluation of fuzzy features involves assessing the degree of variation using a normal distribution. Employing the fuzzy membership functions derived from the normal distribution, determine the membership functions for each feature within the data flow.
4. Upon completing all prerequisites, execute fuzzy recognition computations within the program. Integrate the interval values of data flow features and the pre-determined membership functions of features into the program to achieve comprehensive fuzzy software recognition.
5. Ultimately, leveraging the fluctuation range of recognized data flow features, proceed with the identification and classification of malicious software.

4.2 Type I fuzzy recognition

Fuzzy recognition theory posits that the attributes of the object undergoing recognition exhibit fuzziness throughout the recognition process. In other words, the standard fuzzy model is inherently fuzzy. Type I fuzzy recognition theory involves the manual determination of the variable range of data characteristic ambiguity. This is achieved through human learning and experiential judgment, leading to the subdivision of intervals for data characteristics. By iteratively tuning and calculating, we identify the maximum threshold through cluster analysis, facilitating subsequent stages of identification.

Type I fuzzy recognition theory focuses on the proximity of data features. The proximity of fuzzy sets is inversely proportional to the size of the outer product: the closer the fuzzy set, the smaller the outer product. Conversely, the larger the inner product, the closer the fuzzy set. Hence, closeness serves as a metric to depict the similarity between two fuzzy sets.

The algorithmic principles guiding the design of the first type of fuzzy recognition theory include:

1. Maximum Membership Principle;
2. Threshold Principle; This fuzzy algorithm employs a fuzzy decision-making method to prescribe a specific design plan, addressing issues that may arise in the current or future selection of the optimal plan. The objective of fuzzy decision-making [11] is to rank objects in the domain by considering their superiority and inferiority, or to choose a satisfactory plan from the domain using a predefined method. Ultimately, the application of fuzzy decision-making is specifically manifested in the realms of scientific technology and economic management.

4.3 Type II fuzzy recognition

In practical datasets, instances often emerge where data display ambiguity and uncertainty. Traditional binary classification methods may fall short in effectively addressing such complexities. The Type II fuzzy recognition theory excels in handling issues related to fuzziness and uncertainty, providing a robust framework for the classification and recognition of data characterized by fuzziness.

In intricate scenarios, data features can become highly complex, posing a challenge for traditional classification methods to adapt effectively. The Type II fuzzy recognition theory demonstrates notable prowess in managing large volumes of complex data, enabling the classification and recognition of extensive datasets. This capability significantly enhances the accuracy and efficiency of data processing in such complex situations.

Type II fuzzy recognition theory differs from Type I in that it mandates that the feature set to be recognized possesses attributes that either completely belong or do not belong at all. In other words, there is a stringent [0,1] closed interval constraint between feature elements and fuzzy sets within the fuzzy model lib. Unlike Type I recognition, Type II recognition dispenses with the need for fuzzy cluster analysis, as it eschews artificial feature interval classification. Instead, it directly determines the membership function based on the established fuzzy standard model lib, thereby facilitating the identification process.

The design of Type II fuzzy recognition theory adheres to the following principles:

1. The Proximity Principle.
2. Multi-characteristic Proximity Principle.

We leverage the fuzzy association rules within Type II fuzzy recognition theory for the classification of fuzzy features. The primary objective involves parsing and calculating the entire dataset of malware traffic using big data techniques. Subsequently, we determine the degree of membership for each fuzzy feature and assess fuzzy association rules based on support and trust criteria.

Compared to Type I fuzzy recognition, Type II fuzzy recognition can adjust recognition methods according to specific situations, better adapting to different scenarios and requirements, thereby improving the flexibility and applicability of recognition. Moreover, Type II fuzzy recognition methods have relatively lower requirements on hardware devices and software running memory, making them better suited to meet the needs in resource-limited environments.

Type II fuzzy recognition introduces a novel concept known as closeness [12]. In contrast to Type I's fuzzy recognition theory, Type II involves the comparison of two fuzzy sets: the model fuzzy set and the standard fuzzy set. The process entails identifying the affiliation between these sets, establishing fuzzy subsets, and determining the closeness between subsets and supersets.

4.4 Malware classification

Following the completion of fuzzy identification on the traffic data collection, a subsequent analysis is essential to extract malicious traffic and ascertain the type of malware. In this experiment, the calculation of the fuzzy degree of trust is conducted based on the fuzzy association rules outlined in Section 3. Ultimately, the degree of trust, represented by FConf, is employed for the classification of malware:

Dos attack: The predominant fuzzy feature, given its significant weight, is the time interval of the traffic. Additionally, the port number serves as a robust criterion for determining its nature. This type of malware is classified as C1.

Web crawlers: These malware entities engage in the unauthorized retrieval of users' or enterprises' data through the transmission of malicious crawler data. Consequently, this type of malware falls under the classification C2.

Mail interception: This category involves the interception or camouflage of Internet mail on the designated network card. Hence, this type of malware is classified as C3.

Phishing websites: This category involves the deployment of phishing advertisements or the malicious download of phishing software on specific websites. The primary goal is to illicitly obtain users' information, primarily targeting individual users. This type of malware is classified as C4.

Port scanning: This type of malware engages in extensive port scanning processes on corporate extranets to identify available ports. It subsequently floods idle ports with numerous malicious and invalid data packets, creating confusion in corporate network data and disrupting the analysis of network traffic. Consequently, this type of malware is classified as C5.

In addition to the aforementioned malware, there are numerous comprehensive threats, including Pajio and Ofred, among others. These comprehensive malware variants use a diverse range of malicious attack methods. Consequently, a thorough traffic analysis of this type of malware necessitates multiple iterations for comprehensive understanding.

5 Experiments

5.1 Data sets

In this experiment, the dataset comprises network data packets, with the traffic data collection predominantly categorized into two segments: malware traffic data and normal traffic data, as outlined in Table 3 and Table 4:

In this experiment, the malware traffic was generated by the malicious traffic simulator, directing phishing website traffic to the network. Simultaneously, the Doser packet sender executed a simulated Dos attack on a designated port. Furthermore, the MBlocker mail [13] blocking simulation tool intercepted mail on the experimental machines. The normal traffic, on the other hand, involved regular Internet access by the experimental computer network card, encompassing both the sending and receiving processes of network data packets. Various protocols, such as HTTP, FTP, SMTP, RIP, DNS, ARP, were employed in the traffic packets. Once the normal traffic reached a specified volume, it was deliberately mixed with malware traffic packets at an appropriate ratio. Subsequently, the network simulated the reception of malware attacks based on this proportion, thus forming the dataset utilized in this experiment.

5.2 Environment and operation

For environmental configuration, Wireshark was employed in this experiment to capture data packets for testing purposes. The data to be tested originated from traffic transmitted through the SMTP or HTTP protocol. Malicious traffic was generated through the testing of phishing websites and malware, and subsequently captured on the experimental machine, as illustrated in Figure 1.

The equipment used in this experiment comprises an experimental computer equipped with 8.00 GB of memory and a 64-bit operating system, featuring an x64 processor. In addition to Wireshark, the primary software tools include Qt Creator and Visual Studio 2019 as programming tools. The characteristics of the traffics are saved in Excel tables, and the results are documented in Word text files.

The initial phase of this experimental program involves parsing Pcap data packets as the primary input. Specifically, it parses all data packets within files designated with the. pcap suffix and subsequently classifies the traffic. The analysis process is as shown below:

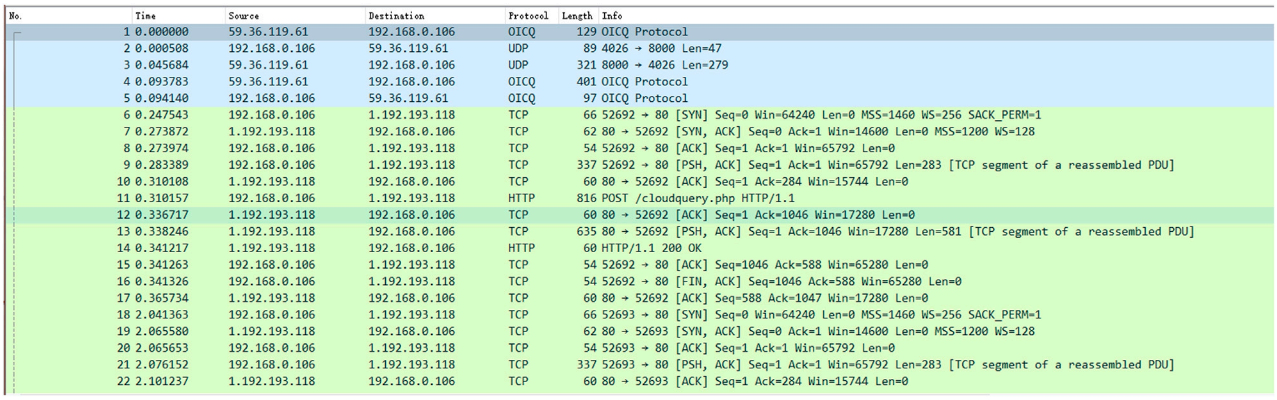
```
Serial Number:90.
89:725906(Len:60) (capLen:60).
5254 00 12 35 02 08 00 27 e6 9f 5f 08 00 45 00.
0028 00 82 40 00 80 06 2b a2 0a 00 02 0f 1f aa
a2 f3 04 1a 00 50 84 31 f7 72 00 03 f9 c8 50 10
fa f0 6c 5d 00 00 00 00 00 00 00 00.
Serial Number:91.
89:725981(Len:60) (cpLen60).
52 54 00 12 35 02 08 00 27 e6 9f 5f 08 00 45 00.
0028 00 83 40 00 80 06 2b a1 0a 00 02 0f 1f aa
a2 f3 04 1a 00 50 84 31 f7 72 00 04 04 e0 50 10
fa f0 61 45 00 00 00 00 00 00 00 00.
Serial Number:92.
89:726024(Len:60) (capLen:60).
52 54 00 12 35 02 08 00 27 e6 9f 5f 08 00 45 00.
00 28 00 82 40 00 80 06 2b a2 0a 00 02 0f 1f aa
a2 f3 04 1a 00 50 84 31 f7 72 00 04 0a 6c 50 10
f5 64 61 45 00 00 00 00 00 00 00 00.
```

TABLE 3 Malicious traffic dataset.

	Total malicious flows (Entries)	Packet size (MB)	Traffic percentage (%)
Dos Attack	1,387	765	1.15
Mail Interception	2,456	874	2.04
Malicious Crawlers	3,399	3,240	2.81
Phishing Websites	2,365	2,310	1.96
Port Scanning	1777	965	1.48
Comprehensive Malware	8,365	1,028	6.97

TABLE 4 Normal traffic dataset.

	Total malicious flows (Entries)	Packet size (MB)	Traffic percentage (%)
HTTP Requests	45630	54420	38
SMTP Mail Requests	14721	12351	12.2
FTP File Requests	11023	9,897	9.19
DNS Domain Requests	21100	7,684	17.5
Telnet	350	5568	0.29
SNMP	765	6,327	0.63



```
No.          Time       Source           Destination      Protocol  Length Info
           1 0.000000   59.36.119.61     192.168.0.106    OICQ        129 OICQ Protocol
           2 0.000508   192.168.0.106    59.36.119.61     UDP          89 4026 → 8000 Len=47
           3 0.045684   59.36.119.61     192.168.0.106    UDP         321 8000 → 4026 Len=279
           4 0.093783   59.36.119.61     192.168.0.106    OICQ        401 OICQ Protocol
           5 0.094140   192.168.0.106    59.36.119.61     OICQ         97 OICQ Protocol
           6 0.247543   192.168.0.106    1.192.193.118    TCP          66 52692 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
           7 0.273872   1.192.193.118    192.168.0.106    TCP          62 80 → 52692 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1200 WS=128
           8 0.273974   192.168.0.106    1.192.193.118    TCP          54 52692 → 80 [ACK] Seq=1 Ack=1 Win=65792 Len=0
           9 0.283389   192.168.0.106    1.192.193.118    TCP         337 52692 → 80 [PSH, ACK] Seq=1 Ack=1 Win=65792 Len=283 [TCP segment of a reassembled PDU]
          10 0.310108   1.192.193.118    192.168.0.106    TCP          60 80 → 52692 [ACK] Seq=1 Ack=284 Win=15744 Len=0
          11 0.310157   192.168.0.106    1.192.193.118    HTTP        816 POST /cloudquery.php HTTP/1.1
          12 0.336717   1.192.193.118    192.168.0.106    TCP          60 80 → 52692 [ACK] Seq=1 Ack=1046 Win=17280 Len=0
          13 0.338246   1.192.193.118    192.168.0.106    TCP         635 80 → 52692 [PSH, ACK] Seq=1 Ack=1046 Win=17280 Len=581 [TCP segment of a reassembled PDU]
          14 0.341217   1.192.193.118    192.168.0.106    HTTP         60 HTTP/1.1 200 OK
          15 0.341263   192.168.0.106    1.192.193.118    TCP          54 52692 → 80 [ACK] Seq=1046 Ack=588 Win=65280 Len=0
          16 0.341326   192.168.0.106    1.192.193.118    TCP          54 52692 → 80 [FIN, ACK] Seq=1046 Ack=588 Win=65280 Len=0
          17 0.365734   1.192.193.118    192.168.0.106    TCP          60 80 → 52692 [ACK] Seq=588 Ack=1047 Win=17280 Len=0
          18 2.041363   192.168.0.106    1.192.193.118    TCP          66 52693 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
          19 2.065580   1.192.193.118    192.168.0.106    TCP          62 80 → 52693 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1200 WS=128
          20 2.065653   192.168.0.106    1.192.193.118    TCP          54 52693 → 80 [ACK] Seq=1 Ack=1 Win=65792 Len=0
          21 2.076152   192.168.0.106    1.192.193.118    TCP         337 52693 → 80 [PSH, ACK] Seq=1 Ack=1 Win=65792 Len=283 [TCP segment of a reassembled PDU]
          22 2.101237   1.192.193.118    192.168.0.106    TCP          60 80 → 52693 [ACK] Seq=1 Ack=284 Win=15744 Len=0
```

FIGURE 1
Packet capture diagram.

5.3 Experimental results

Due to constraints in the program operating environment and limited CPU memory, this experiment will selectively choose a subset of traffic from the dataset as samples for testing and analysis. The goal is to make comprehensive assessments through multiple analysis and comparison processes, as depicted in Table 5 and Table 6.

The analysis results can be accessed and reviewed from the Word document. Each analysis produces a distinct Word document, as illustrated in Table 7.

The malware traffic statistics are presented in Table 8.

The outcomes of the malware identification process are displayed in Table 9.

In addition to checking through the result Word log document, the final parsing results can also be directly obtained from the program's execution results, as shown in Figure 2. The final analysis result of the data in the figure is malware. The total number of data flows is 1,384, the total number of malicious data flows is 11, and the current percentage of malicious data flows is 0.008%, of which 0.0022% is email interception, 0.00122% is Dos attack, 0.00138% is malicious crawler, and 0.00122% is phishing website traffic. 0.00122% is a port scan, the identification success rate is about 96.00%, and the false positive rate is 1%.

In contrast to specific technologies employing approaches focused on monitoring malware traffic, particularly those dependent on the fuzzy characteristics of network data to differentiate between diverse network applications—whether benign or malicious—with the ultimate goal of identifying traffic using fuzzy mathematics. Although most of

TABLE 5 Malicious data flow samples.

	Total malicious flows (Entries)	Packet size (MB)	Traffic percentage (%)
Dos Attack	121	45.4	1.15
Malicious Crawlers	54	36.7	2.04
Phishing Websites	79	79.5	2.81
Port Scanning	83	31.4	1.96
Mail Interception	34	42.6	1.48

TABLE 6 Normal data flow samples.

	Total malicious flows (Entries)	Packet size (MB)	Traffic percentage (%)
HTTP Requests	1,235	32.4	38
SMTP Mail Requests	897	19.9	12.2
FTP File Requests	2,548	32.2	9.19
DNS Domain Requests	1,241	49.1	17.5
Telnet	1,090	14.1	0.29
SNMP	765	22.3	0.63

TABLE 7 Data flow recognition results table.

	Malicious traffic	Approximate malicious traffic	No clear characteristics	Approximate normal traffic	Normal traffic
Fuzziness Level Classification	92.193%	71.020%	51.531%	20.389%	0.896%
Malware Determination	Yes	Yes	Pending	No	No
Number of Data Flows	135	452	654	124	781

TABLE 8 Malware traffic statistics table.

Malware status	Yes
Total Number of Data Flows (Entries)	1,389
Total Number of Malicious Data Flows (Entries)	16
Current Packet Malicious Flow Percentage (%)	0.012%
Recognition Success Rate (%)	96.000%
False Positive Rate (%)	3.000%

TABLE 9 Malware identification results table.

Mail interception	2.67
Dos Attack	4.01
Malicious Crawlers	1.90
Phishing Website Traffic	2.10
Port Scanning	1.88
Trojan Virus	3.00

these methods showcase a low false alarm rate coupled with high accuracy, the primary challenge resides in establishing an appropriate classification basis for the categorization of network traffic.

Our approach demonstrates a feature balance, referring to a rational weighing and adjustment of different features in malicious traffic detection. This ensures that the model comprehensively considers the contribution of each feature, preventing any single feature from becoming overly prominent or dominant, thereby affecting the overall accuracy and stability of detection.

We use fuzzy recognition theory for malicious traffic monitoring. Through the analysis and extraction of malicious traffic features, we determine the importance and weights of different features, maintaining a balance among them. Additionally, we utilize fuzzy feature extraction methods to identify malicious traffic. During feature extraction, it is essential to assess and balance the weights of each feature to ensure the model comprehensively considers their contributions. Finally, we use clustering analysis methods to categorize malicious traffic features into different intervals. Through reasonable classification and interval assignment, we maintain a balance among features, preventing any single feature from becoming overly prominent or dominant.

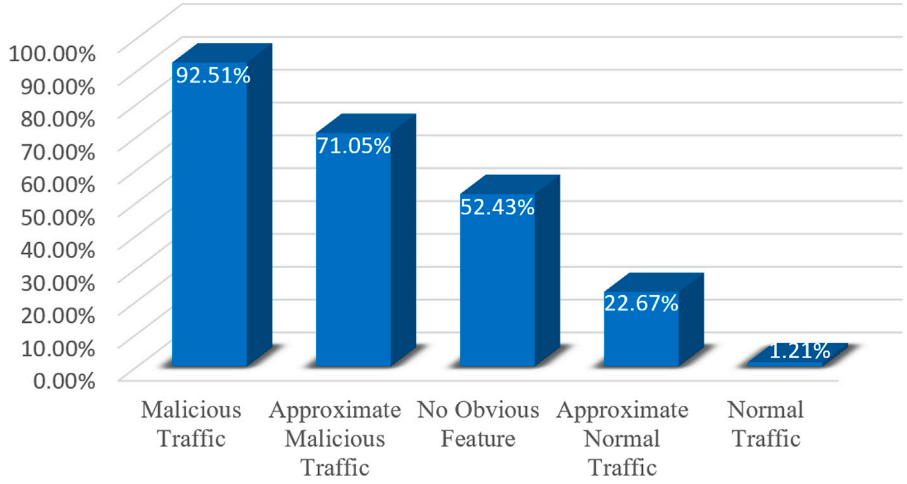


FIGURE 2
Program result graph.

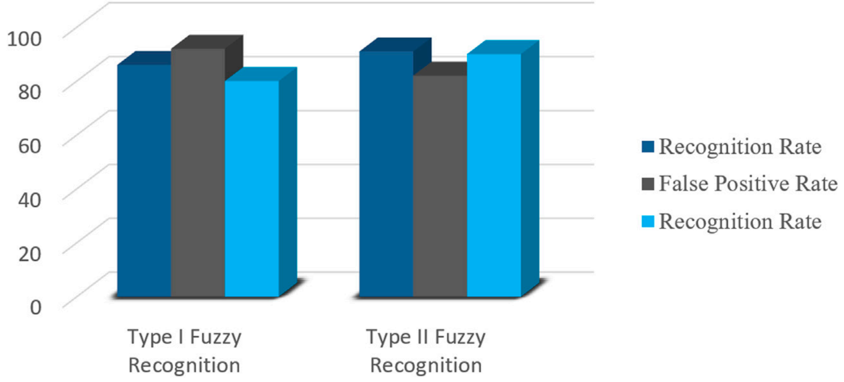


FIGURE 3
Identification scheme comparison chart.

This paper predominantly revolves around the application of the Type II fuzzy recognition theory in the context of a malware traffic detection method. The targeted scenario involves addressing the inherent uncertainty and fuzziness associated with data features. Leveraging fuzzy set theory and its conceptual and operational aspects, the paper employs fuzzy sets to describe and handle the intricacies of the problem.

While both Type I fuzzy recognition and Type II fuzzy recognition fall under the umbrella of fuzzy theory, they diverge in their approaches to problem-solving. Despite these differences, both are dedicated to addressing similar challenges. Therefore, this paper is primarily dedicated to a comparative analysis between Type I fuzzy recognition and Type II fuzzy recognition.

Type II fuzzy recognition methods adeptly handle the volatility of fuzzy features, presenting results within a range interval. Even with minor fluctuations in the results, they do not exert a significant influence on the overall judgment of malicious traffic, thus maintaining a higher level of accuracy.

As the application fields of Type II fuzzy set theory continue to expand with ongoing development, there is a need to delve into the nature and measurement methods of uncertainty within Type II fuzzy sets. Building upon an examination of the uncertainty characteristics and fuzzy entropy of Type II fuzzy sets, we propose the definition of discrete Type II fuzzy set entropy by extending the conventional fuzzy entropy definition. This endeavor opens up novel perspectives and methodologies for the application of Type II fuzzy sets in uncertain environments, as depicted in Figure 3.

It is evident that opting for Type I fuzzy recognition is more rational when dealing with messy data and intricate traffic types. On the other hand, the selection of Type II fuzzy recognition becomes more accurate in scenarios with a substantial volume of data but simpler traffic software types. Tailoring the recognition method to specific circumstances significantly impacts identification accuracy.

Deep learning techniques present a versatile approach to handling situations characterized by vast datasets and intricate data flows. However, in the context outlined in this paper, the application of deep learning typically necessitates a substantial volume of labeled data for effective training. In the domain of cybersecurity, acquiring large-scale labeled data poses challenges, particularly when dealing with labeled data pertaining to malicious traffic.

Furthermore, in the field of network security, the prompt detection of malicious traffic demands real-time responsiveness. The training and inference processes of deep learning models often entail a significant time investment, rendering them unsuitable for meeting real-time requirements. As a result, the constraints related to data availability and real-time processing pose practical challenges to the widespread application of deep learning in the described cybersecurity scenario.

The approach grounded in fuzzy recognition theory proves adept at addressing uncertainty and fuzzy patterns within the realm of network security. It demonstrates adaptability to the intricate characteristics and dynamic changes inherent in malicious traffic. The findings affirm that opting for a method based on fuzzy recognition theory is more fitting within the specific domain and scenario delineated in this paper.

6 Conclusion

In this paper, we introduce a malware traffic detection approach grounded in fuzzy-theory recognition, leveraging fuzzy mathematics as its theoretical foundation. Acknowledging the limitations of the certainty inherent in classical sets, we leverage the ambiguity offered by fuzzy sets to establish the variable range of characteristics for the research object, thereby extending the scope of fuzzy recognition theory. Ultimately, we use membership functions to compute the ambiguity of fuzzy features, providing an effective basis for malware detection.

This method effectively addresses uncertainty and ambiguity within the realm of cybersecurity, showcasing adaptability to the intricate characteristics and dynamic changes inherent in malicious traffic. It automatically recognizes ambiguous features, learning their ranges and weights to accommodate various types and sizes of malicious traffic. The utilization of the maximum number algorithm enhances the precision of classification results, ensuring greater accuracy.

However, it is essential to note that the method’s computational process can be complex, particularly when handling large-scale datasets. This complexity may lead to longer processing times and increased demands on computational resources. Fuzzy theoretical models typically involve parameter selection and tuning, and determining the optimal fuzzy set and affiliation function necessitates thorough validation and experimentation.

Based on the above analysis, the next steps in research should focus on the following issues:

1. A versatile and efficient method for collecting multi-source data in network environments, coupled with the rapid evolution of malware targeting backbone networks.
2. A malicious traffic detection technique grounded in the temporal and spatial characteristics of behavior has been devised, endowing it with broader applications and higher efficacy. This technology relies on behavioral patterns over time and space for effective identification of malicious network traffic.
3. The development of three-level hierarchical models encompassing traffic analysis, fuzzy feature recognition, and collaborative decision-making.
4. A collaborative-capable malicious traffic detection system has been created, providing support for multi-party cooperation, thereby comprehensively safeguarding network security. This system is designed to facilitate collaboration among various entities in order to bolster defenses against potential threats.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

WZ: Writing–original draft. JiL: Writing–review and editing. JP: Writing–review and editing. QL: Writing–review and editing. KY: Writing–review and editing.

Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. We acknowledge the financial support received for this research from the National Social Science Fund of China under grant number 20CXW016 for the project titled “Computational Communication Strategy for Chinese Technology Image on Twitter”, The authors also extend their appreciation to the support of The National Natural Science Foundation of China (Grant No. 62102049). The National Key R&D Program of China (Grant No. 2017YFB0802300). The Natural Science Foundation of Sichuan Province (Grant No. 2022NSFSC0557). The National Key Research and Development Plan of China, Key Project of Cyberspace Security Governance (No. 2022YFB3103103). The Key Research and Development Project of Chengdu (No. 2023-XT00-00002-GX). The Key Research and Development Project of Sichuan Province (No. 2022YFS0571, No. 2021YFSY0012, No. 2021YFG0332, No. 2020YFG0307).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher’s note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

1. Weiyong Y, Peng L, Jinsuo L, Yibin H. Research on data protection technologies against emerging network threats. *Electric Power* (2014) 12(5):14–5.

2. Harnish R. Cybersecurity in the world of social engineering. *Cybersecurity in Our Digital Lives* (2015) 12(5):20–1.

3. Di W, Xiang C, QiXu L, FangJiao Z. Research on Ubiquitous Botnet. *Inf Netw Security* (2017) 18(7):16–28.

4. Shuning W, Xingru C, Qiang C. Application research of AR-OSELM algorithm in network intrusion detection. *Inf Netw Security* (2017) 17(6):56–7. doi:10.3969/j.issn.1671-1122.2018.06.001

5. Lu J, Chen K, Zhuo Z, Zhang X. A temporal correlation and traffic analysis approach for APT attacks detection. *Cluster Comput* (2019) 22(Suppl 3):7347–7358. doi:10.1007/s10586-017-1256-y

6. Lu J, Lan J, Huang Y, Song M, Liu X. Anti-attack intrusion detection model based on MPNN and traffic spatiotemporal characteristics. *J Grid Computing* (2023) 21:60. doi:10.1007/s10723-023-09703-9

7. Peng L, Wuping W, Shiyong Z. Hybrid network monitoring system based on active networking technology. *Comput Eng Des* (2014) 25(9):1427–31.

8. Jun C. Network traffic management implementation via SNMP protocol. *Coal Technol* (2019) 28(8):162–5.

9. Jun L, Liang X. Distributed network traffic monitoring. *Traffic Manage* (2017) 17(7):56–8.

10. Rosenberg I, Shabtai A, Rokach L, Elovici Y. Generic black-box end-to-end attack against state classifiers. *Intrusions* (2018) 490–510. doi:10.48550/arXiv.1707.05970

11. Wang Q, Guo W, Zhang K, Ororbia A, Xing X, Liu X, et al. Adversary resistant deep neural networks with an applicatn to malware detection. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining* (2017). p. 1145–53. doi:10.1145/3097983.3098158

12. Kim JY, Bu SJ, Cho SB. Zero-day malware detection using transferred generative adversarial networks based on deep autoencoders. *Inf Sci* (2018) 460:83–102. doi:10.1016/j.ins.2018.04.092

13. Raff E, Barker J, Sylvester J, Brandon R, Catanzaro B, Nicolas C. *Malware detection by eating whole exe*. Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence (2018). p. 531–3. doi:10.48550/arXiv.1710.09435