Wiley
IET Software
Volume 2024, Article ID 7060298, 17 pages
https://doi.org/10.1049/2024/7060298

Research Article

Research and Application of Firewall Log and Intrusion Detection Log Data Visualization System

Ma Mingze

School of Computer Science and Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore

Correspondence should be addressed to Ma Mingze; mmzmmz_181@163.com

Received 26 September 2023; Revised 8 June 2024; Accepted 30 July 2024

Academic Editor: Manuel Angel Serrano

This paper tackles current challenges in network security analysis by proposing an innovative information gain-based feature selection algorithm and leveraging visualization techniques to develop a network security log data visualization system. The system's key functions include raw data collection for firewall logs and intrusion detection logs, data preprocessing, database management, data manipulation, data logic processing, and data visualization. Through statistical analysis of log data and the construction of visualization models, the system presents analysis results in diverse graphical formats while offering interactive capabilities. Seamlessly integrating data generation, processing, analysis, and display processes, the system demonstrates high accuracy, precision, recall, F1 score, and real-time performance metrics, reaching 98.3%, 92.1%, 97.5%, 98.1%, and 91.2%, respectively, in experimental evaluations. The proposed method significantly enhances real-time prediction capabilities of network security status and monitoring efficiency of network devices, providing a robust security assurance tool.

1. Introduction

With the rapid development of the Internet and the widespread adoption of big data and cloud computing technologies, network security has become an issue of critical importance to countries, societies, and individuals. Network security incidents are prevalent across various devices, including computers, mobile devices, cloud platforms, and intelligent hardware [1]. In China, the problem of network security is becoming increasingly prominent. According to a report released by the China Computer Network Emergency Response Technical Coordination Center (CNCERT or CNCERT/CC), the number of hosts infected with network viruses, the total number of websites tampered with and implanted with backdoors, the number of counterfeit web pages, and the number of information security vulnerabilities all show a general upward trend each week. As numerous online businesses continue to develop, network security issues cannot be underestimated. The economic losses caused by these issues to companies, organizations, and individuals are rising annually, making network security an unavoidable concern [2].

In recent years, despite continuous improvements in network security technology and detection standards, network attacks remain frequent. Attackers exploit various methods to obtain users' private information, causing disruptions to public network facilities and seeking illicit gains. More concerning is that as network technology becomes more widespread, the barrier to launching network attacks is lowering, leading to an annual increase in the number of attacks. Relevant statistics indicate that the threat growth rates for mobile devices, embedded systems, popular technologies, and operating systems are 12%, 27%, 18%, and 16%, respectively, with attack intensity steadily increasing. The damage from network attacks extends beyond economic losses, posing serious threats to national security [3]. Network security has become a common challenge for humanity in the information age [4].

Although network security products are currently used to protect networks, they often fail to provide complete protection against various intrusions and vulnerabilities. Network security analysts are, therefore, essential for analyzing security logs to diagnose the current network security situation and take effective measures. However, these analysts face numerous challenges when analyzing network security logs, including:

(1) The overwhelming volume of network data.
(2) Poor interactivity and weak user experience.
(3) Lack of comprehensive network visibility.
(4) Insufficient real-time analysis.

To address these issues, new methods are needed to extract hidden information from network security log data, providing robust support for analysts to identify security threats and abnormal behaviors more swiftly. To better assist network security analysts in uncovering security problems, a more effective and faster solution for network security visualization has emerged. However, existing visualization methods are often too simplistic, failing to quickly and effectively display risks.

The development of advanced visualization techniques in network security products can offer users a more intuitive, comprehensive, and convenient way to display information. These improved visualizations enable users to interact with the data, allowing them to view relevant information in a targeted manner and make informed decisions. Consequently, network security analysts can rapidly analyze and identify risks and attacks within the network.

The frequency of network security incidents continues to rise annually, rendering many current detection and protection measures inadequate against escalating attacks. To swiftly identify and counteract network threats and safeguarding against intrusion, it's imperative to develop a swift and efficient network security data visualization system capable of real-time prediction and visual representation. Numerous researchers have delved into the visualization of network security. One of the research emphases of the Visualization and Interface Design Innovation Group [5, 6], spearheaded by a distinguished leader in visualization at the University of California, Davis, is network security visualization; Zhao et al. [7], Zhang et al. [8], Ding et al. [9], Sun et al. [10, 11], Zhao et al. [12], Celenk et al. [13], Lu et al. [14], and He et al. [15] employed time series diagrams, tree diagrams, topology diagrams, star coordinates, and other 2D visualization methods to present network security data; Zhao et al. [7] introduced a top-down flow model for analyzing network traffic timing, developed and executed a multiview collaborative network traffic analysis system, and successfully realized real-time monitoring and trend prediction of network traffic; Sun et al. [10] introduced a multivariable visualization technology, representing network nodes through star coordinates. This approach aids administrators in swiftly identifying correlations between network security events, thereby enhancing the efficiency of network monitoring [10]. Zhao et al. [12] utilized X-ray maps for real-time monitoring and alerting of network traffic, effectively detecting and defending against abnormal network behaviors; Celenk et al. [13] proposed a method capable of rapidly assessing network conditions. They employ enhanced data packets to swiftly capture network traffic and analyze abnormal information, facilitating the prompt formation of the network situation [13]. Lu et al. [14] introduced a visualization method for high-dimensional network traffic data using concentric circles. This approach effectively identifies major anomalies in traffic [14]. Chen et al. [16] introduced a quasi-real-time traffic datagram mechanism. This mechanism combines a network abnormal traffic detection algorithm based on information entropy with a 3D visual display format. It facilitates the transition of data flow from abstract to concrete, thereby enhancing user awareness of the network situation [16]. Wu et al. [17] introduced a 3D visualization method utilizing spherical graphics for visualizing large-scale networks. Experimental results demonstrate that this model enhances the capacity for network anomaly analysis [17]. Fontugne et al. [18] introduced a distributed visualization tool capable of presenting network traffic data in a multidimensional spatiotemporal format and offering administrators convenient access interfaces; Wu et al. [19] conducted research and designed an IPv6 network log collection platform. They employed big data technology for data processing and analysis, constructing a visualized monitoring system supporting IPv6 network management, operation, and maintenance. This system aims to enhance the efficiency of IPv6 management, operation, and maintenance in university data centers [19]. To enhance the security of multivariate heterogeneous networks, Han [20] proposed a method for visualizing complex multidimensional data within such networks. This approach aims to reduce the dimensionality of complex multidimensional data. Various algorithms, including information entropy, were utilized to extract typical dimensional features from the heterogeneous security log data obtained from multiple sources. Additionally, symbols and tree-graph methods were employed to thoroughly explore micro details within the data. Subsequently, time series analysis was introduced to facilitate real-time prediction of network operations. Finally, image features were summarized, and attack patterns were analyzed. This comprehensive approach completed the visualization of complex multidimensional data within multivariate heterogeneous networks [20]. In response to the prevailing trend of multiple heterogeneous network security log data, Bai [21] investigated multisource heterogeneous network data using data fusion and visualization technologies. Initially, the information entropy algorithm was employed to extract typical dimensions from heterogeneous security log data, while symbol signs and tree graphs were introduced to thoroughly explore micro details within the data. Subsequently, macro predictions of network operation trends were made using time series diagrams. Finally, network attack model analysis and judgment were achieved through a combination with image feature induction [21]. Zhao [22] devised a method for visual fusion of network security data features grounded in cluster analysis. Leveraging the multisource correlation characteristics of network security data, the approach involves extracting network security data, pinpointing the location of network security data events, and utilizing cluster analysis algorithms to identify network security data characteristics. Subsequently, the reliability of network security data characteristics is computed, serving as the basis for data fusion. Through Bayesian conditional probability, the posterior probability of network security data is determined, culminating in the completion of visual fusion of network security data features [22]. Li et al. [23] introduced an enhanced method for analyzing Web threat situations. This method involves data association on multilevel mimic adjudication alarm logs, deep mining and classification of feature data information

extracted through fusion, and visual display of different types of classified data [23]. Zhang and Fu [24] introduced a method for visual fusion analysis of network security data based on time series. This approach utilizes the correlation between network data for feature selection and consistency filtering, preserving original features while eliminating redundant ones. Calculation of information entropy between network data features is proposed, followed by normalization processing using the INTERACT method. Subsequently, processed network data undergo visual fusion analysis through time series analysis. The Chebyshev method is employed to verify network packets to determine the presence of intrusion behavior, achieving visual fusion of network security data [24]. In recent years, there has been explosive growth in network security log data. However, existing visualization technologies struggle to support the analysis of high-dimensional and multigranularity NetFlow log data. To harness the potential of visualization technology, Wang and Han [25] proposed a new network security visualization framework. This framework employs 3D histograms to facilitate rapid understanding of network abnormalities depicted in NetFlow log data. It utilizes the information entropy algorithm for processing multidimensional data and employs matrix charts, bubble diagrams, and line charts to comprehensively synthesize analyzed data [25]. In order to swiftly and accurately detect internal anomalies and network security threats within vehicle networks, there was a need to comprehend the pattern of distribution of vehicle network data. Thus, real CAN bus message data was gathered by the vehicle terminal during normal driving phases. From a visualization standpoint, focusing on time and data characteristics, Qu [26] designed the time sequence scatter plot of message events and the time sequence change plot of message content. These visualizations aim to reveal and identify changing characteristics in data, such as communication frequency and content within the CAN bus network of the vehicle. Furthermore, the original message data underwent processing through methods including statistical analysis and Hamming distance calculation. Interactive views, comprising various visualization techniques and schematic diagrams, were crafted to illustrate the internal network changes [26]. As suggested by Ahmad et al. [27], in addition to pinpointing information needs, practitioners encounter numerous challenges in crafting visualization tools. Consequently, eight challenges encountered by practitioners in developing and maintaining visualization tools for software teams were identified. Furthermore, recommendations from experts proficient in developing, maintaining, and offering visualization services to software teams were outlined [27]. Frei and Rennhard [28] introduced a novel log file visualization technique termed histogram matrix (HMAT). HMAT serves to visualize the content of a log file, empowering security administrators to swiftly detect anomalies with efficiency [28]. Senthilnayaki et al. [29] introduced a feature selection algorithm that significantly reduces unwanted attributes or features. Additionally, the classification algorithm effectively identifies the type of intrusion. These algorithms are highly efficient in detecting attacks and effectively lowering the false alarm rate [29]. Ganapathy et al. [30] introduced a novel weighted fuzzy C-means clustering method based on an immune genetic algorithm (IGA-NWFCM). This method enhances the performance of existing techniques in addressing high-dimensional multiclass problems. The algorithm offers high classification accuracy, stability, and an increased probability of achieving the global optimum value [30]. Rajeswari and Kannan [31] proposed an active rule-based enhancement to the C4.5 algorithm for network intrusion detection. This enhancement aims to detect misuse behaviors of internal attackers through effective classification and decision-making in computer networks [31]. Muthurajkumar et al. [32] posit that log records frequently harbor sensitive information about the organization, which should not be exposed to the outside world. They advocate for the implementation of Temporal Secured Cloud Log Management Algorithm techniques to ensure the security of maintaining transaction history within a cloud environment over a specific time period [32].

Although existing network security visualizations can perform basic display functions, they are relatively simple and unable to comprehensively represent the network security situation.

This paper proposes a network security log data visualization system that utilizes diversified visual forms and real-time display technology. This system integrates complete processes, including original data acquisition, data preprocessing, database management, data manipulation, data logic, and visualization. It facilitates the rapid identification of network security issues and enhances data processing efficiency.

2. Key Technologies

The system primarily employs five key technologies: network traffic detection, raw data acquisition and analysis, preprocessing, database management, and data extraction, analysis, and processing, along with advanced visualization techniques.

2.1. Network Traffic Detection Technology. Network traffic detection technology is used to monitor, analyze, and identify traffic patterns, behaviors, and events within a network. This technology enables network security analysts to better understand network activity and to detect and respond to potential threats and issues promptly. Key network traffic detection tools include packet capture and analysis utilities such as the Library of Packet Capture, Wireshark, and tcpdump. These tools capture and analyze network packets to provide detailed insights into network traffic. The system utilizes the Library of Packet Capture for packet capture, data packet analysis, and the detection of IP and TCP traffic.

2.2. Raw Data Acquisition and Analysis, Prepreprocessing Technology. During data acquisition, raw data are obtained by capturing network packets within the local area network. This is achieved through the utilization of the Library of Packet Capture function, which is a collection of network packet capture functionalities designed for Linux/Unix systems. The process of raw data acquisition and analysis is depicted in Figure 1.

Data preprocessing techniques encompass a series of operations performed on data acquired through packet capture and parsing. These operations include auditing, cleaning,

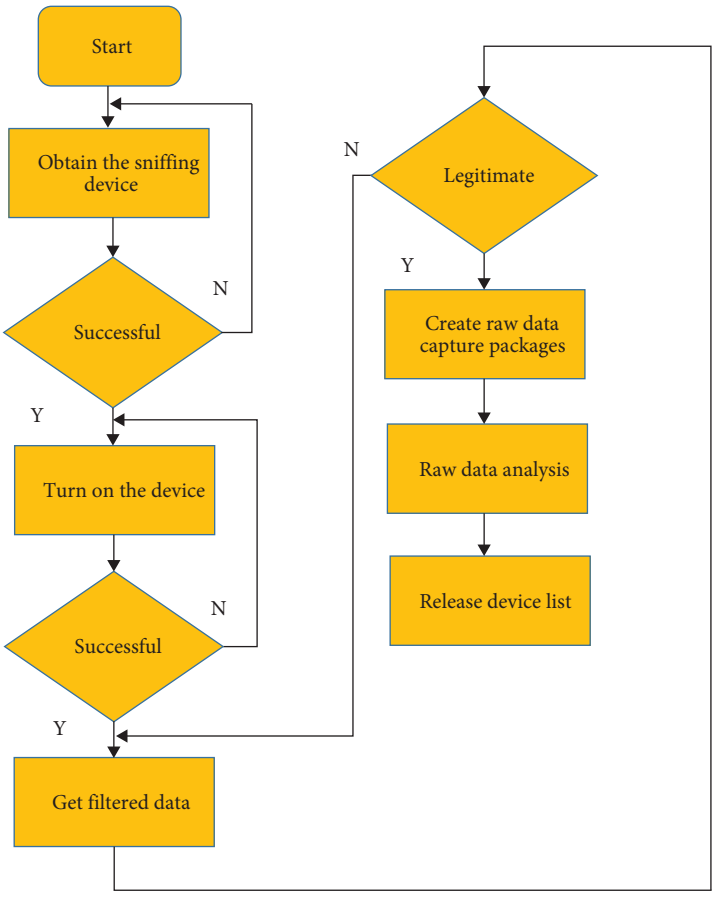

Figure 1: Process for acquiring and analyzing raw data.

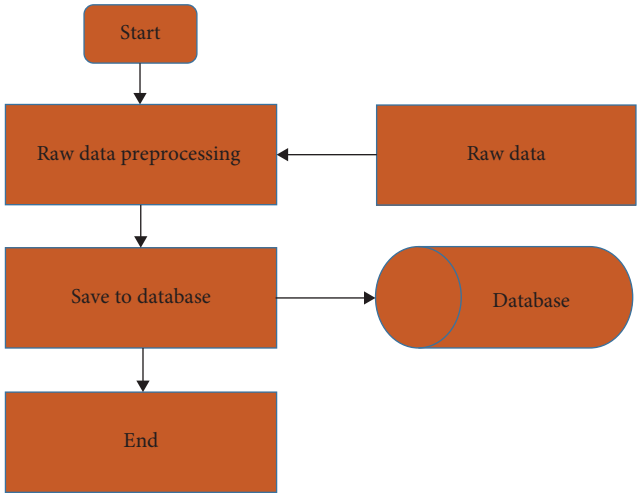

Figure 2: Data preprocessing procedure.

transforming, integrating, desensitizing, reducing, and annotating the data. This entails auditing inconsistent data, cleaning erroneous, false, invalid, missing, and duplicate data, transforming data with incompatible formats and sizes, integrating data through processing and merging, annotating data lacking necessary labeling information, and sorting unordered data. In essence, it involves standardizing the data and subsequently storing it in a database. The data preprocessing process is shown in Figure 2.

2.3. Database Technology. Once the data have been preprocessed, these are stored in the database, and the required information is displayed based on the network security log data. The database is constructed to align with several modules of the network security log data visualization system. For instance, in the case of email detection logging, the construction of each field in a database table is illustrated in Table 1.

2.4. Data Extraction, Analysis, and Processing Technology. Data stored in the database is utilized for feature selection and classification employing the Information Gain algorithm and the Random Forest classification algorithm. The Information Gain algorithm quantifies the reduction in uncertainty within the dataset attributed to a specific feature. The higher the Information Gain associated with a feature, the greater its ability to reduce uncertainty within the dataset when that feature is known, thus enhancing its performance in classifying the dataset. The calculations for Information Gain are outlined by Equations (1), (2), (3), (4), and (5).

$$H(X) = -\sum_{i=1}^{n} p_i \log p_i, \tag{1}$$

Table 1: Log record table.

Fields	Data type	Explain
RECORD_ID	INT_64	Record ID
SERVICE	INT_64	The service type refers to the protocol type of the network flow associated with this record. Service types are defined according to type 3 in the dictionary table. For instance, 31 denotes SMTP, and 32 denotes POP3
FLOW_ID	VARCHAR (512)	List of original packet flow IDs. Multiple original packet flow IDs are separated by "|"
SRC_IP	INT_32	Source IP address
DST_IP	INT_32	Destination IP address
SRC_PORT	INT_16	Source port
DST_PORT	INT_16	Destination port
CAPTURE_IP	INT_32	IP address of the front-end computer
FOUND_TIME	INT_32	Time of discovery
SERVER_ID	INT_64	The server where this log is generated: 0–32 reserved for dedicated devices; 32–64 reserved for scanning engines
USER_NAME	VARCHAR (256)	Email account name
EMAIL_FROM	VARCHAR (256)	Sender's email address
EMAIL_TO	VARCHAR (512)	All recipient email addresses, with each address appearing before and after "<>" markers
SUBJECT	VARCHAR (256)	Subject of email
EMAIL_DATE	INT_32	Email time
ATTACHES_COUNT	INT_32	Number of attachments
EMAIL_CONT_LEN	VARCHAR (1024)	Email body content
TUPLE4_LIST	VARCHAR (256)	List of related quadruples
RESERVE_1	INT_64	Reserved field 1
RESERVE_2	VARCHAR (128)	Reserved field 2

$$H(D) = -\sum_{k=1}^{k} \frac{|C_k|}{|D|} \log_2 \frac{|C_k|}{|D|}, \tag{2}$$

$$H(Y|X) = \sum_{i=1}^{n} p_i H(Y|X = x_i), \tag{3}$$

$$H(D|A) = \sum_{i=1}^{n} \frac{|D_i|}{D} H(D_i) = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} \sum_{k=1}^{k} \frac{|D_{ik}|}{|D_i|} \log_2 \frac{|D_{ik}|}{|D_i|}, \tag{4}$$

$$g(D, A) = H(D) - H(D|A). \tag{5}$$

In the formula, $H(X)$ is the entropy of the random variable X, p_i is the probability of the random variable X, D is the dataset, and $|D|$ is the number of samples in the dataset. Suppose the dataset has k classes C_k, $k = 1,2,3,\ldots, k$, and $|C_k|$ is the number of samples in the kth class C_k. Let the different values of feature A be $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$. Based on the different values of feature A, dataset D is divided into n different subsets, denoted as $D_1, D_2, \ldots D_n$, $|D_i|$ is the number of samples in D_i, D_{ik} is the subset of D_i that belongs to the kth class C_k, and $|D_{ik}|$ is the number of samples in D_{ik}. $H(D)$ is the empirical entropy of the dataset D, $H(Y|X)$ is the conditional entropy of the random variable Y given the random variable X, $H(D|A)$ is the empirical conditional entropy of the dataset D, and $g(D, A)$ is the information gain of feature A in the dataset D.

The Random Forest classification algorithm, rooted in decision tree theory, is a classifier comprised of multiple decision trees [33]. It determines the final classification outcome by aggregating the classification results of multiple decision trees [34], with the ultimate classification being determined by the majority vote across all trees. The Random Forest classification algorithm exhibits outstanding performance in handling large-scale data with numerous features. It is characterized by rapid processing, stable performance, and high efficiency, particularly excelling in the classification of network traffic data.

2.5. Visualization Technology. As the Internet continues to evolve, visualization technology has emerged as a prominent tool. This technique draws upon various disciplines and theories, including statistics, human–computer interaction, and graphics. In an era marked by the expansion of networks and the exponential growth of network security log data, manual parsing of large-scale network logs has become increasingly burdensome and inefficient. Consequently, there is an urgent need for new methods to process massive network log data in network security analysis. Network security log data visualization technology has arisen to address this need, with its primary feature being the intuitive display of abstract data to network security analysts. This intuitive display not only enables rapid and visual comprehension of network operations but also facilitates the identification of hidden relationships within massive datasets that may elude manual scrutiny.

3. System Architecture

The primary objective of the network security log data visualization system is to comprehensively, intuitively, and reliably

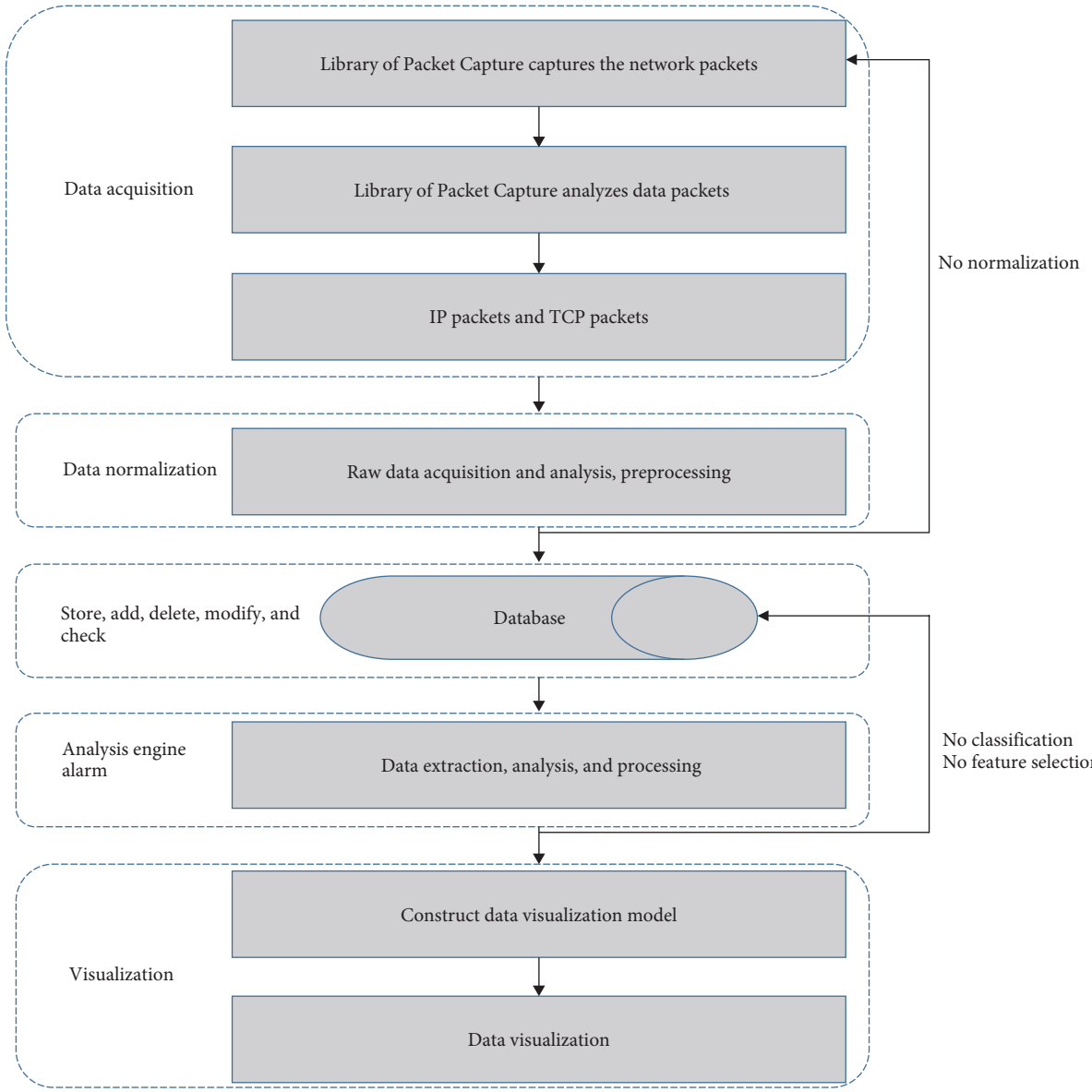

Figure 3: System architecture overview.

present network security data information in real-time. This paper outlines the overall architecture of the system, focusing on two main components: system functionality and system logic architecture. The system's overall architecture is illustrated in Figure 3.

3.1. System Functions. This paper delineates the functions of the network security log data visualization system, primarily focusing on firewall log visualization and intrusion detection log visualization. Firewall logs are primarily employed for network monitoring and protection. They safeguard the network system by documenting firewall system access and network activity. Intrusion detection logs, on the other hand, are primarily utilized to identify anomalous events within the system. They establish corresponding checkpoints at strategic locations within the network system, subsequently gathering and filtering information at these checkpoints to record, compare, and analyze various network events. Furthermore, network security analysts can utilize these logs to identify any behaviors within the current network system that deviate from security policies or detect signs of potential attacks [35].

3.2. System Logical Architecture. The logical architecture of the system, depicted in Figure 4, facilitates the visualization of a plethora of detection logs, sensitive information documents, and suspicious email data generated within the network security detection system. These associations are displayed in a visual format to enhance comprehension and analysis.

3.2.1. Raw Data. The original data are generated from detection logs generated during network traffic detection.

(1) Firewall Logs. First, it strengthens security policies by, for instance, restricting access to IP sources or port usage. Second, it comprehensively logs various activities within the

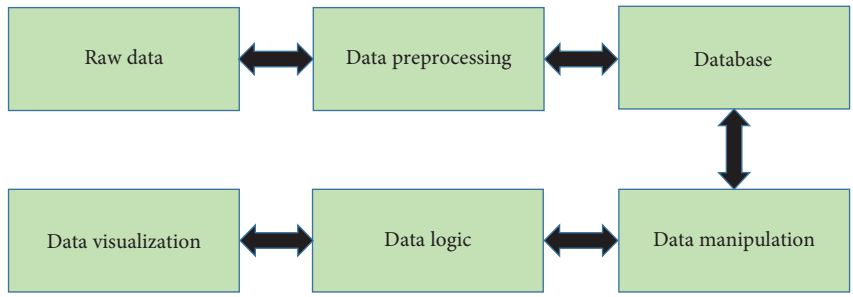

Figure 4: System logical architecture.

network system. Finally, firewall technology isolates network threats and confines the scope of malicious events. The data recorded by firewall software are termed firewall logs. These logs not only monitor and safeguard the firewall itself but also protect the entire network's data flow. Formatted in specific ways, these data constitute logs, which are commonly utilized as network security logs [36]. Firewall logs contain a wealth of network traffic information, including timestamps, source and destination IP addresses, port numbers, protocol types, and whether connections are permitted or denied. Additionally, firewall logs may encompass details regarding attack types, triggered rules, user authentication, and error messages.

(2) Intrusion Detection Logs. The data captured by an intrusion detection system are referred to as intrusion detection logs. Unlike firewall systems primarily logging for analysis purposes, intrusion detection systems are geared towards actively protecting the system and serve as an effective complement to firewall software. These logs document suspicious activities and potential threats within networks and systems. They include timestamps, source and destination IP addresses, port numbers, protocol types, triggered detection rules or signatures, alert levels, and specific attack types. Moreover, intrusion detection logs may provide detailed descriptions of intrusion events, associated attack payloads, impacted systems or files, and recommended response measures.

3.2.2. Data Prepreprocessing. Typically, a host network card only processes the network packets pertinent to the host itself. However, for the purpose of comprehensively understanding the entire network, the network card in this article is configured to accept all packets. Upon acquiring the raw data, preprocessing is essential, involving the identification and analysis of the original data.

3.2.3. Database. The database stores detection logs following data preprocessing, which include IP information detection logs and email information detection logs. For instance, after data preprocessing, email detection logs are stored in the form of a table within the database, referred to as "email records." Each entry in the email record table represents a detection log for an email. The fields in this table comprise the ID of the email record, the timestamp of email discovery, the email subject, the email body content, the length of the email body, the number of attachments, the sending time, the sender's email address, the recipient's email address, the sending IP address, the receiving IP address, the service type, the email account name, and a related quadrangle list (source IP address, destination IP address, source port, destination port), among others.

3.2.4. Data Manipulation. The function of data manipulation involves extracting and selecting data from the database. This can be accomplished through various methods, such as using database query language, programing code, or graphical interface tools.

3.2.5. Data Logic. The role of data logic is to employ the Random Forest classification algorithm to analyze and process the data extracted and selected through data manipulation. This process involves constructing a data model that serves as the required structure for visually displaying the data.

3.2.6. Data Visualization. Data visualization constitutes the most critical aspect of this system. Its primary function is to present a series of analyzed and processed data in a visual format, while also offering interactive capabilities. The architecture of the data visualization component is illustrated in Figure 5.

(1) Visual Module for Email Exchange Relationship Analysis. The function of the email exchange relationship analysis visual module is to extract, analyze, and process the email detection log data stored within the network security detection log by the data logic component. It aims to display the detected sensitive email exchange relationships within a specified time period.

To manage the large volume of detected data, a time period selection mechanism is incorporated to facilitate batch processing and ensure system efficiency.

Upon selecting the designated time period, the system utilizes data operations to extract and select data from the database. Subsequently, the selected data is organized in the form of a dictionary, and the Random Forest classification algorithm is applied for analysis and processing. A corresponding matrix data model is then generated based on the email sending and receiving correspondence.

Finally, the visual module presents the relationship between sensitive email exchanges during the selected time period through an appropriate visual format. The logical architecture of the email exchange relationship analysis visual module is illustrated in Figure 6.

(2) Visual Module for Log Frequency Distribution Analysis. The function of the log frequency distribution analysis

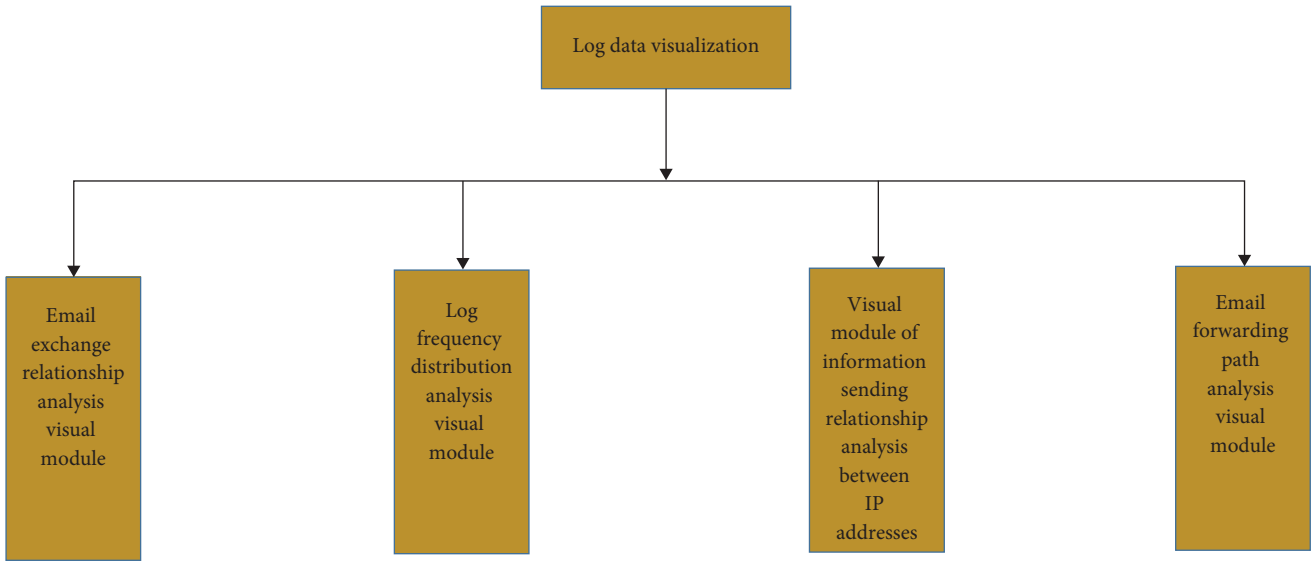

Figure 5: Data visualization architecture.

Figure 6: Logical architecture of the visual module for email exchange relationship analysis.

visualization module is to compute the count of detected logs within each time period and present them in a visually comprehensible format. This enables network security analysts to grasp the distribution of log frequencies across various time periods.

(3) Visual Module for Analysis of Information Sending Relationships Between IP Addresses. The visual module for analyzing the information-sending relationship between IP addresses aims to extract data on sensitive information transmission between IPs. It utilizes the Random Forest classification algorithm to classify, analyze, and statistically process the detection data related to inter-IP sensitive information transmission. Through a user-friendly visual interface, it displays the sensitive information sending relationships between different IP addresses over a specified time period.

The module segments and visually presents IP data according to the selected time period. It employs a scatter layout visualization mode where nodes represent entities and lines represent connections between them. The size of nodes reflects the strength of the correlation between IP information transmissions. The logical architecture of this visual module is depicted in Figure 7.

(4) Visual Module for Email Forwarding Path Analysis. The email forwarding path analysis visual module's function is to analyze and process data from the email detection log table. It counts the email forwarding relationships and displays the forwarding path of a specific email between different mailboxes. The logical architecture of this module is depicted in Figure 8.

4. System Implementation

The system is developed using the Java programing language, with a MySQL database and MyEclipse development IDE. For the front-end, JSP pages are utilized, primarily incorporating JS, jQuery, and HTML for interface development. The back-end is implemented using the SSH framework, which is known for its low coupling and ease of extension.

The system comprises six main components: original data, data preprocessing, database management, data manipulation, data logic, and data visualization. Among these, the original data component is deployed on the LAN egress route, and each host is within the LAN, while the remaining five components are installed on the server.

4.1. Implementation of Data Transmission Interface. Communication among the original data, data preprocessing, database, data manipulation, and data logic components occurs through RMI interface calls. Data transmission employs multiple threads combined with buffers, utilizing protocols such as syslog, SNMP, and NetFlow.

4.2. Implementation of Data Visualization Interface. Data logic and data visualization communicate via the RMI interface. When data logic generates new results, it notifies data visualization, which then displays the new results on the interface.

4.3. Real-Time Display of Network Security Status. Utilizing the Library of Packet Capture to capture and analyze network packets, the system extracts raw packets from IP and

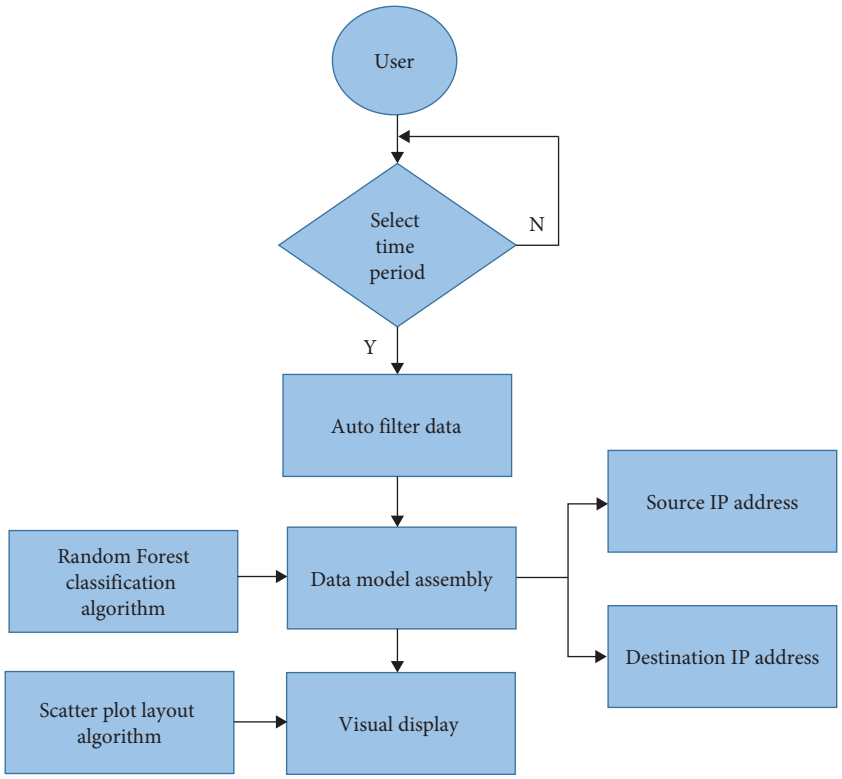

Figure 7: Logical architecture of the visual module for analyzing information-sending relationships between IP addresses.

TCP packets. These original data packets undergo analysis, preprocessing, and storage in the database. Subsequently, real-time data is displayed after extraction, analysis, and processing. The screenshot of the core code for capturing TCP packets is depicted in Figure 9.

5. System Testing and Analysis of Experimental Results

5.1. System Testing. The system's testing environment comprises both hardware and software test environments. Table 2 illustrates the hardware test environment, while Table 3 outlines the software test environment.

The main methods of system testing are "black-box testing" and "white-box testing." Black-box testing is also called functional testing. It refers to testing all functions of the system to determine whether the system has the specified functions and mainly focuses on boundary conditions, coverage conditions, and the effectiveness of error handling. The methods mainly include equivalence class partitioning, boundary value analysis, error guessing, cause–effect graph, decision table testing, orthogonal experimental design, and functional diagram method [37, 38, 39]. White-box testing, also known as performance testing, refers to testing whether the accuracy, time characteristics, and adaptability of a program and the system data meet the practical requirements [40]. Through the actual test, the system function and performance test results meet the requirements.

5.2. Analysis of Experimental Results. The firewall log and intrusion detection log data visualization system constructed in this paper is experimentally compared with existing network security visualization systems to validate the effectiveness of the proposed method.

5.2.1. Analysis of Performance Evaluation Metrics. The performance evaluation metrics employed in the experiments primarily include accuracy, precision, recall, F1 score, and real-time performance. Accuracy measures the system's ability to correctly differentiate between normal and attack traffic, identifying any potential false positives or false negatives. Precision evaluates the system's accuracy in identifying malicious behavior, minimizing the negative impact of false positives. Recall assesses the system's ability to detect actual attacks, gauging its sensitivity across various attack scenarios. The F1 score considers both accuracy and recall, providing insight into the system's performance in balancing false positives and false negatives. Real-time performance assesses the system's capability to detect and respond to threats in real-time, ensuring prompt and effective decision-making.

5.2.2. Comparative Analysis of Experimental Results. The types of intrusions and attacks addressed in this paper primarily include denial of service (DoS) attacks, remote-to-local (R2L) attacks, user-to-root (U2R) attacks, probe attacks, and others. A DoS attack disrupts the normal operation of a target system by inundating it with a high volume of requests, leading to service interruption and preventing legitimate users from accessing the system or service. R2L attacks involve attempting to gain unauthorized access to a local system from a remote location, often by exploiting system vulnerabilities or weak passwords. U2R attacks involve attempting to

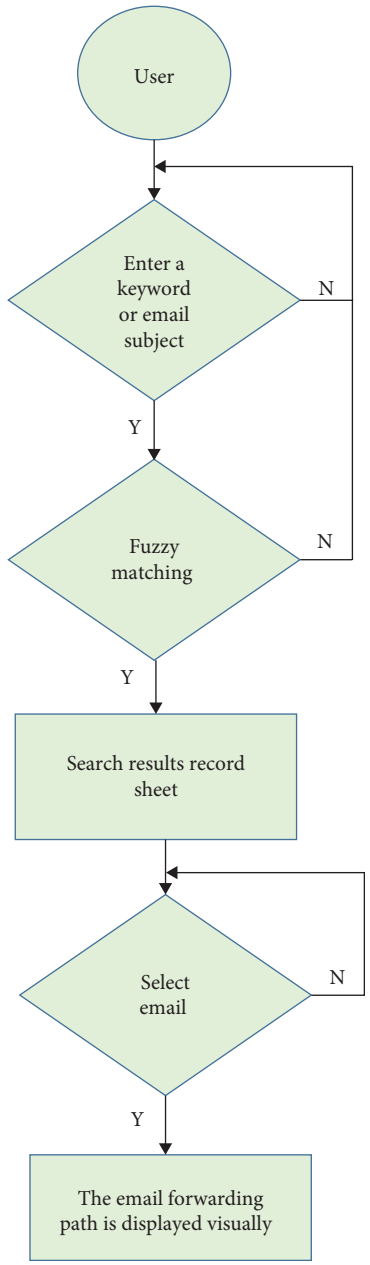

Figure 8: Logical architecture of the visual module for email forwarding path analysis.

escalate privileges from a regular user to a system administrator, typically by exploiting system vulnerabilities or other security flaws to gain complete control over the system. Probe attacks entail scanning target networks or systems to identify weaknesses and vulnerabilities for the purpose of gathering information.

This paper employs the KDD Cup'99 dataset, renowned as a classic network intrusion detection dataset. It comprises ~4.9 million network connection records sourced from the DARPA 1998 dataset. Each record encompasses 41 features, encompassing fundamental attributes like source IP address, destination IP address, port number, protocol type, as well as content-based error status features, time-based traffic features, and host-based traffic features. The dataset is classified into normal traffic and four attack types: DOS, R2L, U2R, and Probe. It serves as the training and testing dataset for evaluating the performance of intrusion detection systems. The comparative results are presented in Table 4.

Table 4 illustrates the comparative outcomes between the developed visualization system and existing counterparts, focusing on metrics, including accuracy, precision, recall, F1 score, and real-time performance. The constructed visualization system attains an accuracy of 98.3%, surpassing that of existing systems, indicating superior capability in distinguishing normal and attack traffic with minimal false positives or false negatives. Its precision stands at 92.1%, also outperforming existing systems, showcasing its effectiveness in identifying malicious behavior with minimal negative impact.

```
TCPPacket tcpPacket =(TCPPacket) packet;
if(tcpPacket.dst_ip.getHostAddress ().equals( getIp())) {
    EthernetPacket ethernetPacket =(EthernetPacket)packet.datalink;
    srcIp =tcpPacket.src_ip ;// The IP address of the source host obtained
    dstIp =tcpPacket.dst_ip ;// The IP address of the destination host is obtained
    srcPort =tcpPacket.src_port ;// Obtain the source host port
    dstPort =tcpPacket.dst_port;// Obtain the destination host port
    srcAddress =ethernetPacket.getSourceAddress ();// The MAC address of the source host is
    obtained
    dstAddress =ethernetPacket.getDestinationAddress ();// The MAC address of the destination host
    is obtained
    protocol =tcpPacket.protocol;// Get protocol name
}
```

Figure 9: Captures the core code of TCP packets.

Table 2: Hardware testing environment.

Equipment	Hardware environment requirements
Server	
Server	IBM X series, HP ProLiant series, or PC servers based on X86 architecture
Central processing unit	Dual-core 3.0 GHz or Quad-core 2.0 GHz or above
Main memory	Above 4 GB
Auxiliary memory	More than 146 GB of free space
Network interface card	100 M/1,000 M adaptive network interface card
Host	
Computer	The system can run on desktop computers
Central processing unit	Above 2.5 GB
Main memory	Above 2 GB
Video adapter	Video adapter memory more than 1 GB
Image resolution	More than 1,024 × 768 pixels
Auxiliary memory	More than 1 GB of free space
Network interface card	10 M/100 M adaptive network interface card

Table 3: Software testing environment.

System environment	Software environment requirements
Server	
Operating system	Linux
Database	Microsoft SQL Server 2014
Program development language	Java
Host	
Operating system	Microsoft Windows series
Browser version	IE7.0 or above

Table 4: Comparison of evaluation metrics.

Visualization system	Accuracy (%)	Precision (%)	Recall rate (%)	F1 score (%)	Real-time performance (%)
Constructed visualization system	98.3	92.1	97.5	98.1	91.2
Existing visualization system	84.2	79.3	85.1	83.7	78.5

With a recall rate of 97.5%, the constructed system outperforms its counterparts, reflecting its higher detection rate of actual attacks and sensitivity to real threats. Achieving an F1 score of 98.10%, the constructed system excels in minimizing false positives and false negatives compared to existing systems. Furthermore, with a real-time performance of 91.2%, the constructed visualization system demonstrates greater responsiveness to real-time threats and faster response rates.

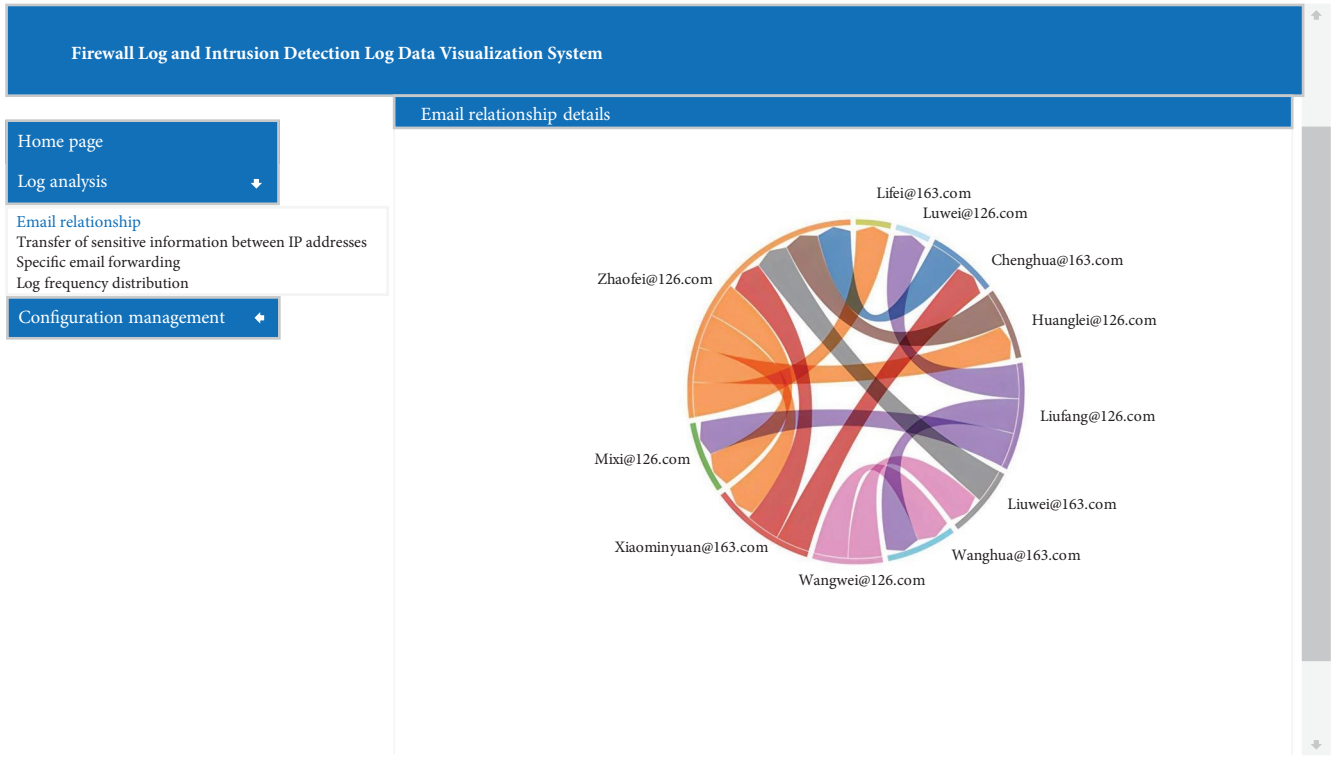

Figure 10: Visualization of sensitive data exchanges among email accounts.

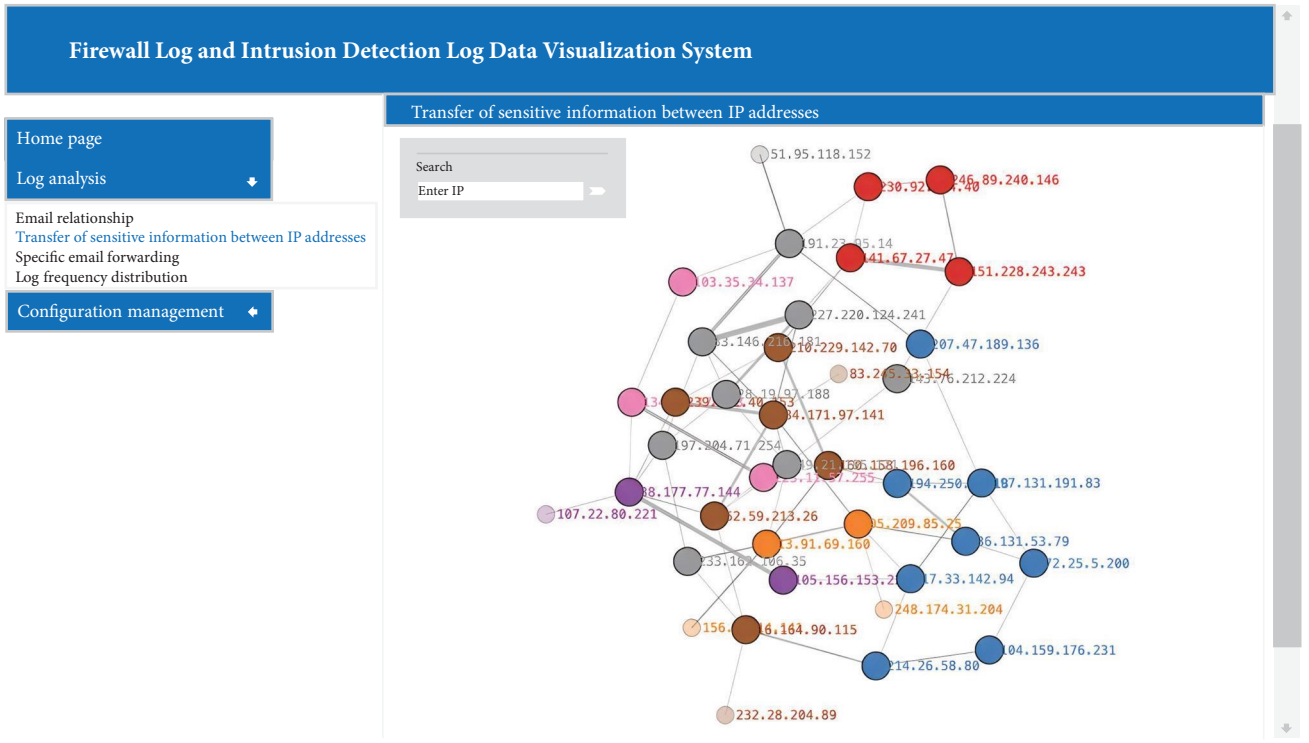

Figure 11: Visualization of sensitive information transmission relationships between IP addresses.

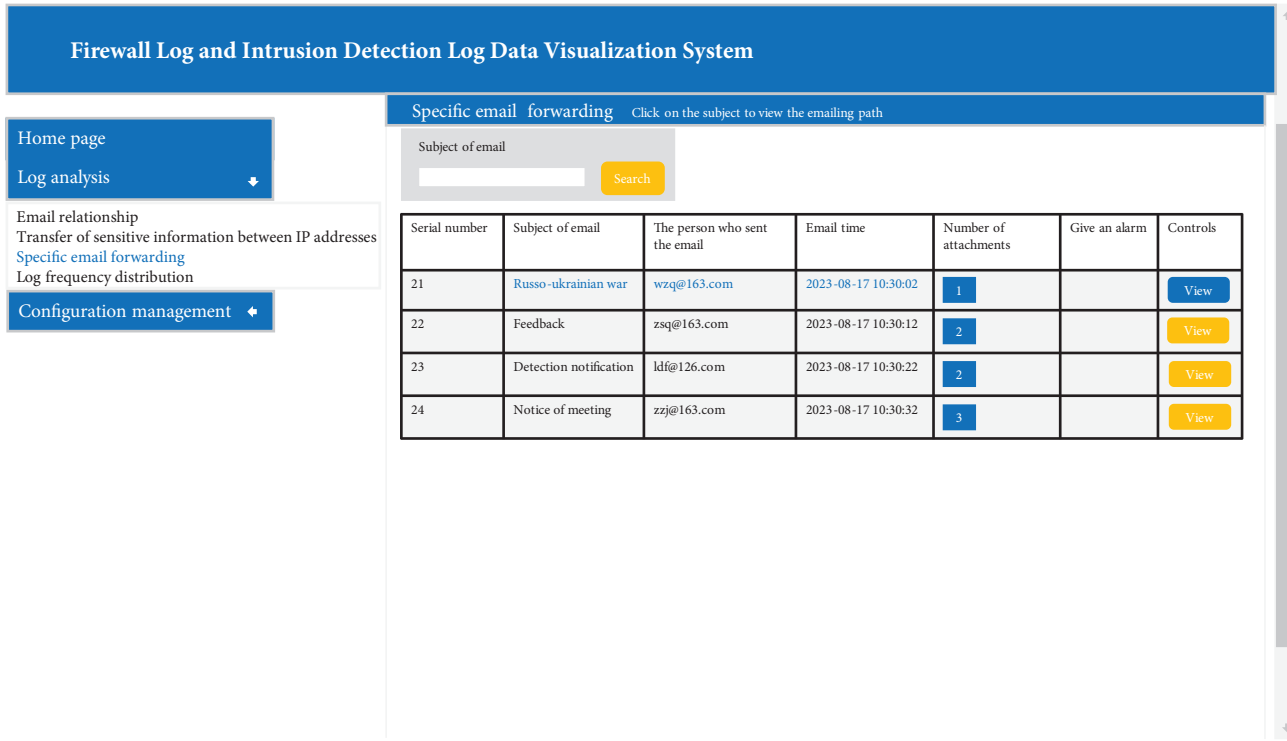

Figure 12: Visualization of sensitive email forwarding pathways.

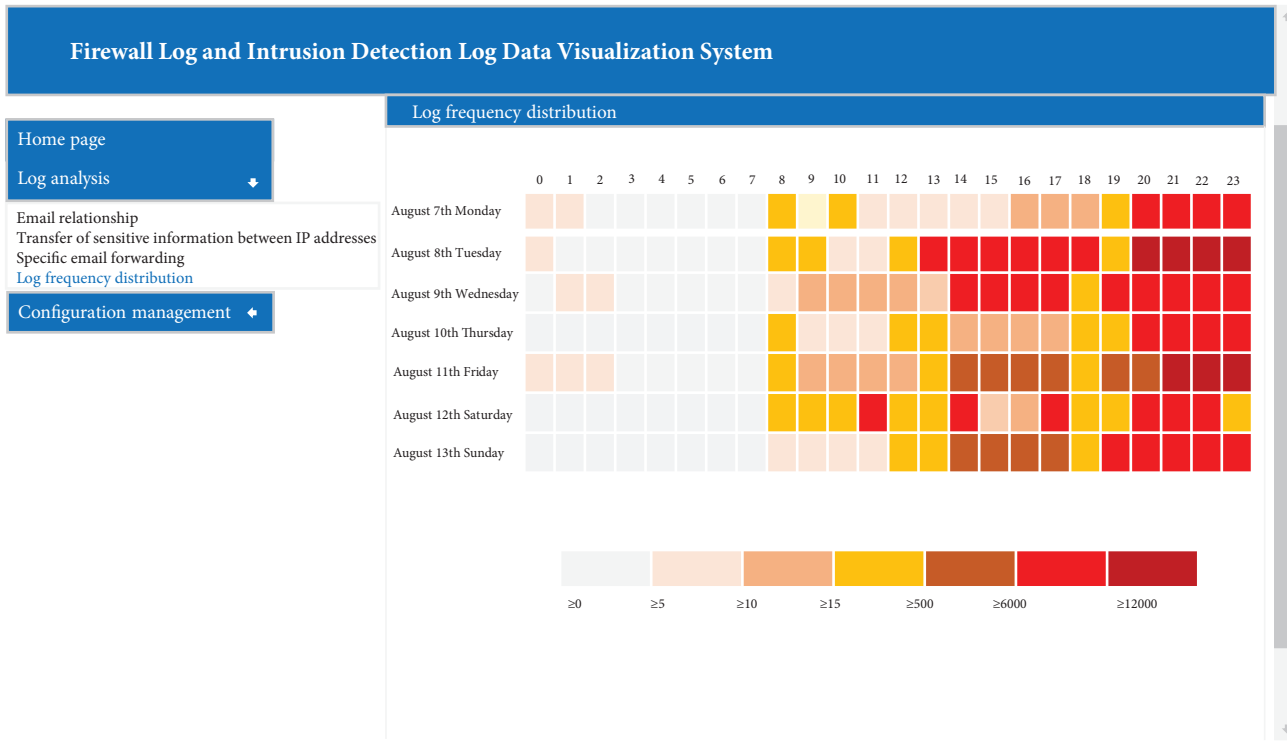

Figure 13: Visualization of the distribution of log event counts.

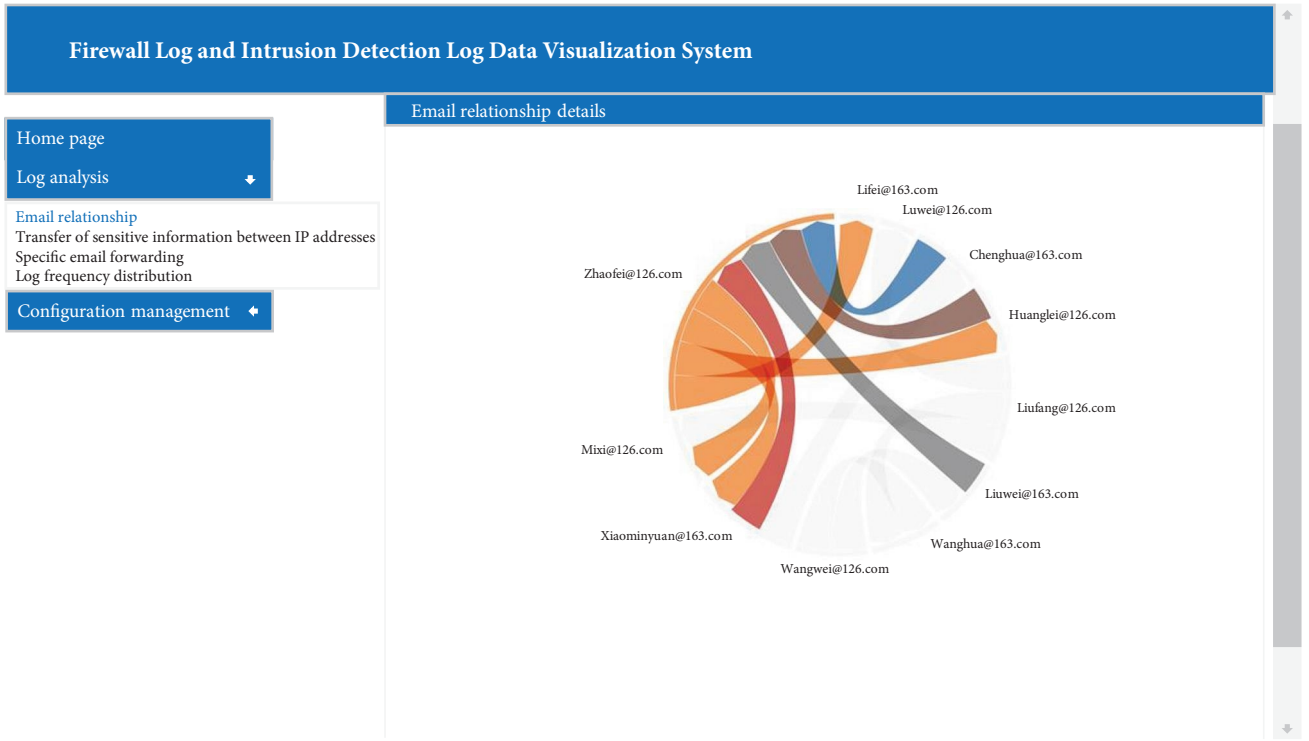

FIGURE 14: Human–computer interaction visualization test for sensitive information exchanges among email accounts.

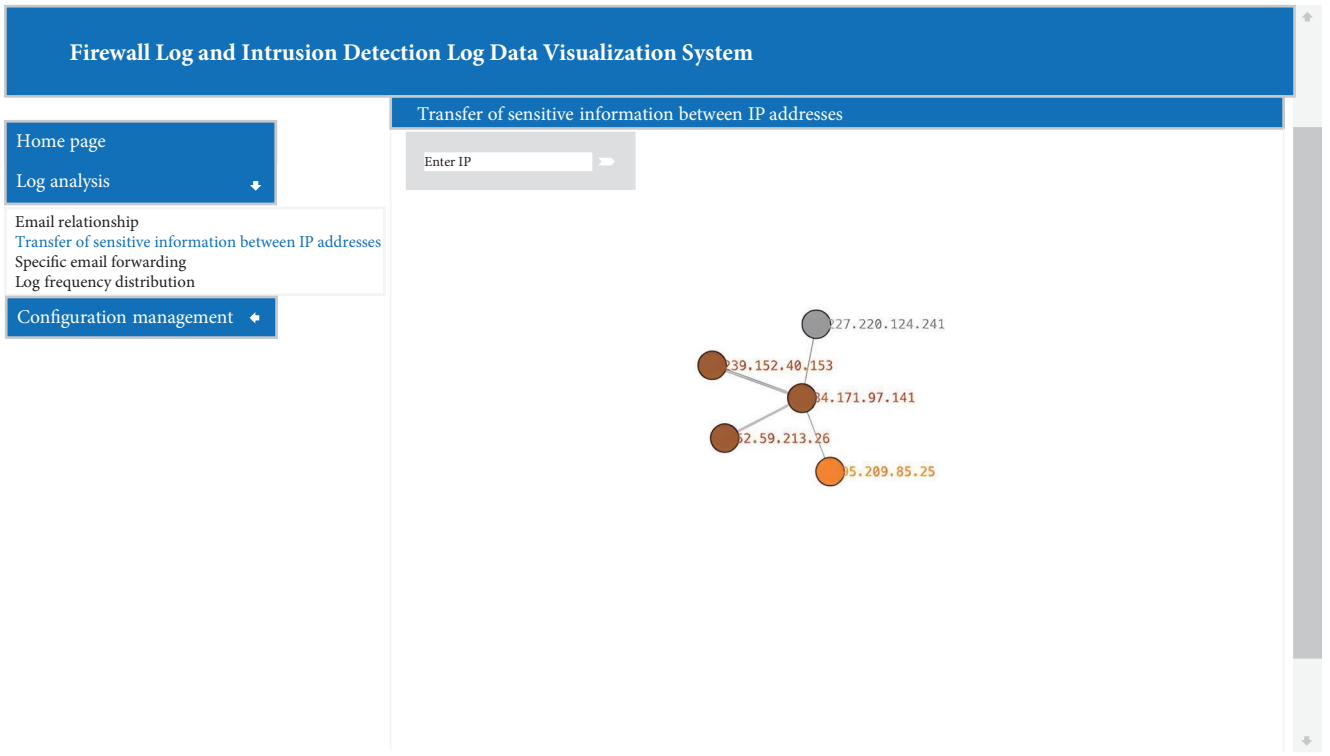

FIGURE 15: Human–computer interaction visualization test for sensitive information transmissions between Ips.

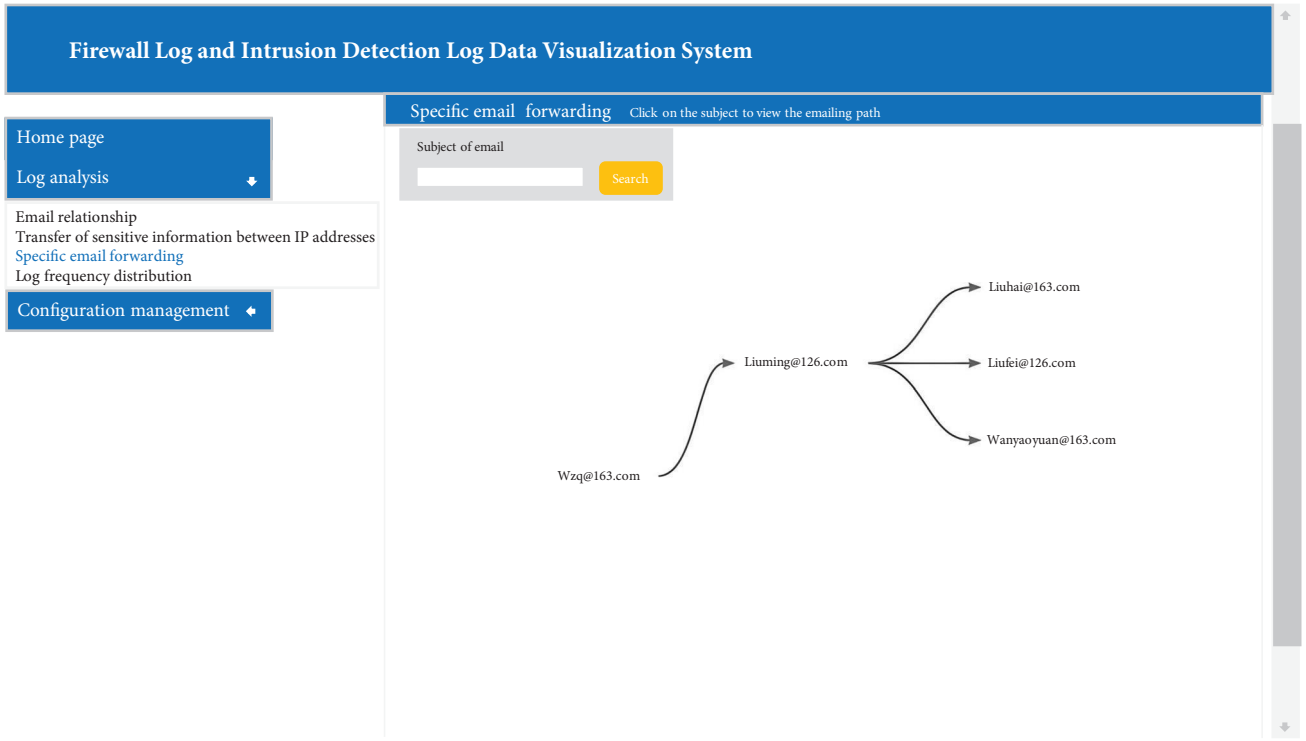

Figure 16: Human–computer interaction visualization test for sensitive email forwarding pathways.

The results from Table 4 clearly indicate that the developed visualization system outperforms existing ones across accuracy, precision, recall, F1 score, and real-time performance metrics.

5.3. Visualization System Performance Display. Figure 10 depicts the visualization of sensitive information exchanges between email accounts. Figure 11 illustrates the visualization of sensitive information transmissions between IPs. Figure 12 showcases the visualization of sensitive email forwarding paths. Finally, Figure 13 presents the visualization of the distribution of log event counts.

The human–computer interaction visualization test for sensitive information exchanges between email accounts is depicted in Figure 14. Figure 15 showcases the human–computer interaction visualization test for sensitive information transmissions between IPs. Additionally, Figure 16 displays the human–computer interaction visualization test for sensitive email forwarding paths.

6. Conclusion

Visualizing network security log data represents an important future development direction and a research hotspot within network security monitoring. The visualization system for firewall logs and intrusion detection logs developed in this paper introduces an innovative feature selection algorithm based on information gained through the analysis of network security data. The system presents network security data using diverse visual formats such as chord diagrams, point-line diagrams, tree diagrams, heat maps, and time series diagrams. This approach not only overcomes the constraints of existing network security visualization methods, which often rely on single forms and lack the ability to swiftly and effectively display risks, but it also showcases robust interactive capabilities.

The application results demonstrate that the designed and implemented visualization system for firewall logs and intrusion detection logs renders the network status more intuitive, thereby enhancing the efficiency of network security analysts.

In the future, artificial intelligence technology will be a pivotal focus of research and development in network security monitoring, necessitating further in-depth studies.

Data Availability

The data used to support the findings of this study are currently under embargo, while the research findings are commercialized. Requests for data, (6/12 months) after publication of this article, will be considered by the corresponding author.

Conflicts of Interest

The author declares that he has no competing interests.

Authors' Contributions

The author confirms contribution to the manuscript as follows: research conception and design, data collection, analysis and interpretation of results, and draft manuscript preparation were done by Mingze Ma. Mingze Ma reviewed the results and approved the final version of the manuscript.

Acknowledgments

This manuscript is completed under the friendly care and modest guidance of Professor Vicky Ray. Here, I would like to express my heartfelt thanks to Professor Vicky Ray for his modest help and guidance.

References

[1] Tencent Security, *2017 Annual Internet Safety Report*, Tencent, ShenZhen, GuangDong, China, 2018.

[2] L. Liangfu, "*Research on DDoS Attacks Detection and Related Network Security Visualization Techniques*," Ph.D. dissertation, school of computer, Tianjin University, Tianjin, China, 2008.

[3] B. Yuan, D. Zou, and H. Jin, "Network security visualization: a survey," *Journal of Cyber Security*, vol. 1, no. 3, pp. 10–20, 2016.

[4] J. Zhang, "Design of campus network security system based on network information security," in *2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, pp. 1194–1197, IEEE, Dalian, China, April 2022.

[5] M. A. Kwan-Liu, "Homepage," 2010, http://www.cs.ucdavis.edu/ma/.

[6] "Vidi Homepage," 2014, http://vidi.cs.ucdavis.edu/new.

[7] Y. Zhao, Q. Wang, Y. Z. Huang, Q. Wu, and S. Zhang, "Collaborative visual analytics for network traffic time-series data with multiple views," *Journal of Software*, vol. 27, no. 5, pp. 1188–1198, 2016.

[8] S. Zhang, R. Shi, and Y. Zhao, "Visual fusion and analysis for multivariate heterogeneous network security data," *Journal of Computer Applications*, vol. 35, no. 5, pp. 1379–1384, 2015.

[9] L. Ding, Z. Miao, Z. Pan, G. Ni, and G. Hu, "Data visualization for one-class classification algorithm," *Journal of Data Acquisition & Processing*, vol. 23, no. 5, pp. 600–603, 2008.

[10] Y. Sun, X. Zhao, J.-Y. Tang, D.-Q. Tang, and W.-D. Xiao, "Multivariate network visualization paradigm," *Journal of Software*, vol. 21, no. 9, pp. 2250–2261, 2010.

[11] Y. Sun, H. Zhang, and H. He, "Study on visualization algorithm in large-scale network security precaution," *Computer Engineering and Applications*, vol. 43, no. 21, pp. 115–117, 2007.

[12] Y. Zhao, F. Zhou, and R. Shi, "NetSecRadar: a real-time visualization system for network security—VAST 2012 mini challenge. Award: honorable mention for interesting use of radial visualization technique," in *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 281-282, IEEE, Seattle, WA, USA, October 2012.

[13] M. Celenk, T. Conley, J. Willis, and J. Graham, "Predictive network anomaly detection and visualization," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 288–299, 2010.

[14] L. F. Lu, J. W. Zhang, M. L. Huang, and L. Fu, "A new concentric-circle visualization of multi-dimensional data and its application in network security," *Journal of Visual Languages and Computing*, vol. 21, no. 4, pp. 194–208, 2010.

[15] H. He, G. Fan, J. Ye, and W. Zhang, "A topology visualization early warning distribution algorithm for large-scale network security incidents," *The Scientific World Journal*, vol. 2013, Article ID 827376, 7 pages, 2013.

[16] P. Chen, J. Si, Z. Yu, and W. Wang, "Flow abnormity supervision based on information entropy and 3D visualization," *Computer Engineering and Applications*, vol. 51, no. 12, pp. 88–93, 2010.

[17] Y. D. Wu, H. Y. Jiang, S. R. Zhao, and B. Li, "3D visualization method for network security data," *Journal of University of Electronic Science and Technology of China*, vol. 44, no. 4, pp. 594–598,604, 2015.

[18] R. Fontugne, T. Hirotsu, and K. Fukuda, "A visualization tool for exploring multi-scale network traffic animalies," *Journal of Networks*, vol. 6, no. 4, pp. 577–586, 2011.

[19] Y. Wu, L. Xing, and X. Qin, "Design and implementation of IPv6 network log collection platform in data enter," *Applied Science and Technology*, vol. 49, no. 3, pp. 76–83, 2022.

[20] L. Han, "Visualization method of complex multidimensional data in multiple heterogeneous networks," *Computer Simulation*, vol. 37, no. 11, pp. 299–303, 2020.

[21] B. Bai, "Analysis of data security protection in heterogeneous network based on visualization and data fusion technology," *Electronic Design Engineering*, vol. 28, no. 13, pp. 137–140,146, 2020.

[22] L. Zhao, "The research on visual fusion of network security data features based on cluster analysis," *Journal of Changchun Institute of Technology (Natural Science Edition)*, vol. 21, no. 2, pp. 94–97, 2020.

[23] W. Li, Z. Zhang, L. Wang, Z. Liu, and H. Liu, "A web threat situation analysis method for mimic structure," *Computer Engineering*, vol. 45, no. 8, pp. 1–6, 2019.

[24] J. Zhang and W. Fu, "Analysis method of visual fusion of network security data," *Microelectronics & Computer*, vol. 36, no. 6, pp. 101–104, 2019.

[25] Q. Wang and X. Han, "Cyber security visualization analysis based on netflow," *Computer Systems & Applications*, vol. 28, no. 4, pp. 1–8, 2019.

[26] J. Qu, "Design and implementation of vehicle network security data visualization," *Journal of Xiamen University of Technology*, vol. 27, no. 1, pp. 53–59, 2019.

[27] A. Ahmad, O. Leifler, and K. Sandahl, "Data visualisation in continuous integration and delivery: information needs, challenges, and recommendations," *IET Software*, vol. 16, no. 3, pp. 331–349, 2022.

[28] A. Frei and M. Rennhard, "Histogram matrix: log file visualization for anomaly detection," in *2008 Third International Conference on Availability, Reliability and Security*, pp. 610–617, IEEE, Barcelona, Spain, March 2008.

[29] B. Senthilnayaki, K. Venkatalakshmi, and K. Arputharaj, "Intrusion detection system using fuzzy rough set feature selection and modified KNN classifier," *International Arab Journal of Information Technology*, vol. 16, no. 4, pp. 746–753, 2019.

[30] S. Ganapathy, K. Kulothungan, P. Yogesh, and A. Kannan, "A novel weighted fuzzy C–means clustering based on immune genetic algorithm for intrusion detection," *Procedia Engineering*, vol. 38, pp. 1750–1757, 2012.

[31] L. P. Rajeswari and A. Kannan, "An active rule approach for network intrusion detection with enhanced C4.5 algorithm," *I. J. Communications, Network and System Sciences*, vol. 1, no. 4, pp. 314–321, 2008.

[32] S. Muthurajkumar, S. Ganapathy, M. Vijayalakshmi, and A. Kannan, "Secured temporal log management techniques for cloud," *Procedia Computer Science*, vol. 46, pp. 589–595, 2015.

[33] P. N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*, Posts & Telecom Press, Beijin, China, 2011.

[34] L. Han, "*Pedestrian detection technology research based on the random forests*," M.S. thesis, School of Transportation Engineering, North China University of Technology, Beijing, China, 2014.

[35] W. Liang, "*Design and implementation of distributed web security monitor system*," M.S. thesis, School of Information Science and Engineering, Central South University, Changsha, China, 2013.
[36] K. Qian, "*Desige and implementation of the audit and mangement center in log-based AuditSystem*," M.S. thesis, School of Information and Electronic Engineering, ZheJiang GongShang University, Hangzhou, China, 2009.
[37] W. Lei, "*Research of the algorithm for super-quadratic particles in two dimensions DEM*," M.S. thesis, Software School, Jilin University, Changchun, China, 2010.
[38] G. Gao, "*The research on the logistics management information system for Dong An San Ling*," M.S. thesis, School of Economics and Management, Harbin Engineering University, Harbin, China, 2008.
[39] L. Dai, Z. Zhang, and F. Li, "Analysis of traditional software testing methods," *Technology Wind*, vol. 16, pp. 136-137, 2011.
[40] J. Dong, "Third party logistics management information system analysis and design," *Decision-Making & Consultancy*, vol. 5, pp. 29–32, 2010.