Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.036033
Article

An Erebus Attack Detection Method Oriented to Blockchain Network Layer

Qianyi Dai[1,2,*], Bin Zhang[1,2], Kaiyong Xu[1,2] and Shuqin Dong[1,2]

[1]Zhengzhou Information Science and Technology Institute, Zhengzhou, 450001, Henan Province, China
[2]Henan Key Laboratory of Information Security, Zhengzhou, 450001, Henan Province, China
*Corresponding Author: Qianyi Dai. Email: qianyi.dai@mail.chzu.edu.cn
Received: 14 September 2022; Accepted: 29 January 2023

Abstract: Recently, the Erebus attack has proved to be a security threat to the blockchain network layer, and the existing research has faced challenges in detecting the Erebus attack on the blockchain network layer. The cloud-based active defense and one-sidedness detection strategies are the hindrances in detecting Erebus attacks. This study designs a detection approach by establishing a ReliefF_WMRmR-based two-stage feature selection algorithm and a deep learning-based multimodal classification detection model for Erebus attacks and responding to security threats to the blockchain network layer. The goal is to improve the performance of Erebus attack detection methods, by combining the traffic behavior with the routing status based on multimodal deep feature learning. The traffic behavior and routing status were first defined and used to describe the attack characteristics at diverse stages of **s** leak monitoring, hidden traffic overlay, and transaction identity forgery. The goal is to clarify how an Erebus attack affects the routing transfer and traffic state on the blockchain network layer. Consequently, detecting objects is expected to become more relevant and sensitive. A two-stage feature selection algorithm was designed based on ReliefF and weighted maximum relevance minimum redundancy (ReliefF_WMRmR) to alleviate the overfitting of the training model caused by redundant information and noise in multiple source features of the routing status and traffic behavior. The ReliefF algorithm was introduced to select strong correlations and highly informative features of the labeled data. According to WMRmR, a feature selection framework was defined to eliminate weakly correlated features, eliminate redundant information, and reduce the detection overhead of the model. A multimodal deep learning model was constructed based on the multilayer perceptron (MLP) to settle the high false alarm rates incurred by multisource data. Using this model, isolated inputs and deep learning were conducted on the selected routing status and traffic behavior. Redundant intermodal information was removed because of the complementarity of the multimodal network, which was followed by feature fusion and output feature representation to boost classification detection precision. The experimental results demonstrate that the proposed method can detect features, such as traffic data, at key link nodes and route messages in a real blockchain network environment. Additionally, the model can detect Erebus attacks effectively. This study provides novelty to

the existing Erebus attack detection by increasing the accuracy detection by 1.05%, the recall rate by 2.01%, and the F1-score by 2.43%.

Keywords: Blockchain network; Erebus attack; attack detection; machine learning

1 Introduction

Wide blockchain applications, such as cryptocurrency and smart contracts, have advantages such as open-source, dynamic, and decentralized. However, the blockchain network layer is vulnerable to network attacks because of its inherent defects, such as an imperfect security control mechanism, difficulty in deploying active defense strategies, and an asymmetrical attack-and-defense game. Tran et al. [1] reported that a decentralized blockchain network layer (e.g., Bitcoin and Ethereum) is characterized by a fragile authentication threatened by a new attack called Erebus [1]. This attack greatly threatens the blockchain system's hash rate balance and stability. An Erebus attack also integrates various attack patterns, such as Sybil and Eclipse, showing a long penetration cycle, high behavior imperceptibility, and a large sphere of influence. Without any doubt, the Erebus attack may severely impact the regular running and security control of autonomous systems (ASs) in the blockchain network layer [2]. In June 2021, Internet Nayana Inc. was attacked by Erebus. Consequently, masses of infiltrated transaction server data were maliciously damaged by ransomware, and several transaction websites were affected and forced to stop trading. The attacker can launch Erebus attacks to extract computing resources from blockchain network layer nodes and perform more destructive attacks (e.g., double-spend attack or 51% attack) to the hashrate [3]. Therefore, the detection of the Erebus attack is vital to the blockchain network layer.

Two primary detection approaches are vital in addressing the threatening problem of Erebus attacks. The abnormal distribution of transaction traffic information is the first approach; the second is based on the routing status on the blockchain network layer.

In the first Erebus attack detection method, the technique is to create a security authentication mechanism of traffic information and transaction identities in the blockchain network layer. The goal is to detect the attack because an Erebus attacker uses a weak identity check mechanism at the stage of identity forgery and establishes out-trades and illicit network relationships by creating several dummy Internet Protocol (IP) addresses and Sybil transaction identities. This method reinforces the blockchain network layer protocol security and improves the security strength, frequency of transaction identities, and traffic information. Network security can be enhanced [4,5] using this method. When Bitcoin networks are protected with an identity authentication mechanism oriented to the blockchain network layer, the vulnerability of 'a single IP address with multiple identities' in a peer-to-peer network is repaired, and its robustness in coping with Sybil attacks is improved [6,7]. Facing Erebus attacks, the method can create Sybil peer identity vulnerability by combining multiple different IP addresses/ports, based on a single peer node. Moreover, nodes are from the same IP address that possesses the same identity [8] and constraints traffic with abnormal identities are distributed. Fan et al. [9] proposed an active defense strategy by dynamically detecting traffic and transaction identification information in the blockchain network layer. According to Fan et al. [9], distribution detection at different traffic IP nodes, the abnormal distribution check of transaction identities, and restrictions over relevant transactions can be used to inhibit the forgery attacks of identities caused by Erebus. Apart from updating the corresponding protocol according to the network status and reducing the time for identity authentication, protocol patch uploading and network protocol

updating are performed to dynamically alter the node identification under attack and protect the Bitcoin node from being isolated [10–12]. As revealed by a study [13], Erebus attacks cannot be entirely suppressed through traffic identity authentication or protocol repair/update because this attack is a persistent threat that integrates Sybil and Eclipse. Moreover, strategies based on authentication or protocol repair and updating the passives in attack-defense games increase the overhead time for an Erebus attack. Additionally, non-licensed chains represented by Bitcoin enable a multi-identity transaction mechanism, allow Erebus attackers to generate fake IP addresses, create several malicious peer identities in AS, and gain control over the network connections of a node. Although the existing methods can improve the blockchain network layer security to some extent, such as protocol analysis, update, vulnerability detection, and repair, they still face certain deficiencies of hysteresis, poor integrity, and unsatisfactory active defense effects.

The Routing-Aware Peering (RAP) is the second method, which detects Erebus attacks through peer routing awareness of the blockchain network layer. The RAP is used to detect and suppress abnormal routing in a TorP2P network. This method can authorize partial nodes in the blockchain network layer to acquire the path information of all peer nodes. It can also be used to check peer node identities in routing and routing validity [14]. The RAP has been adopted for routing authentication for the Bitcoin network layer. However, such a method has some shortcomings. First, the routing protocol of the existing blockchain network layer must be amended. Moreover, the AS protocol extensions, which require a high deployment cost, are significantly difficult to implement. RAP is also only valid for preventing routing attacks in the blockchain network layer domain, which includes prefix hijacking and path forgery [15]. RAP performs rather poorly in the interdomain route leaking detection of Erebus attacks [2]. Third, the RAP targeting at the blockchain network layer cannot prevent the routing leaking caused by Erebus attacks, which involves the principle of node identity peering in a blockchain system. Under such a circumstance, the RAP can only discriminate the identities of attackers and incorrect routing but cannot suppress routing or implement effective defense strategies [16]. RAP also makes false-negative and erroneous decisions during attack detection. Tran et al. [12] pointed out that a high rate of missed judgments is noted in RAP detection in a permissionless blockchain network. Erebus attackers may also use such vulnerabilities to dynamically generate Sybil identities, which may serve as advantages for attackers in the asymmetric attack-and-defense game relative to the goal node.

Thus, existing Erebus attack detection faces the following challenges. First, detection objects selected using the existing methods are rather monotonous. Erebus attack detection, based on routing status or identity information, may lead to one-sidedness feature selection, making it less to comprehensively depict the core features of an Erebus attack. Consequently, the corresponding false alarm or negative rates can be rather high. Second, an Erebus attack may adopt multipath routing strategies, making the attack targets unfixed. The existing detection methods are limited in detecting certain objects, thereby causing poor dynamic awareness of Erebus attacks, low detection accuracy, and making the corresponding defense strategy less pertinent. Third, blockchain system resources are probably allocated to functional computing resources. In practice, node security defense resources are rare for the blockchain network layer. However, the detection and active defense mechanisms in the existing studies increase high network communication and node resource requirements. Thus, the existing Erebus attack detection systems must be improved, regarding their adaptability in the blockchain network layer.

This study designs a specific detection approach effective in Erebus attacks and responding to security threats to the blockchain network layer. We define the features of traffic behavior and routing status after summarizing the behavioral characteristics in a multistage Erebus attack process.

Moreover, the study establishes a ReliefF_WMRmR-based two-stage feature selection algorithm and a deep learning-based multimodal classification detection model by combining the defined features and characteristics. This study offers the following contributions:

1) We summarize the relevant studies on Erebus attack detection and defense through profound investigations of the characteristics of multistage Erebus attacks. Moreover, we raise the idea of attack detection according to the routing status and traffic behavior features to explore the influence of multistage Erebus attacks on flat routing and traffic at the blockchain network layer. Additionally, careful consideration is given to distinct characteristics in the entire process of multistage Erebus attacks, which include route leaking, traffic attacks, and identity forgery. We remove relevant defects of low detection accuracy and precision, caused by detection object monotony and one-sidedness detection. Our goal is to improve the detection pertinency, enhance the adaptability of the detection method, and decrease the detection complexity.

2) Erebus attacks are simulated in a real blockchain intranet environment. Additionally, we summarize the traffic behavior and routing status features during a multistage Erebus attack to collect and analyze the routing and traffic data. We also define various routing status features to explore how Erebus attacks affect routing structures in nodes, such as Critical AS Data (CAD), and how Erebus attacks generate abnormalities in the route update messages during route leaking. Additionally, the features of the traffic behavior are defined to elaborate its characteristics and traffic information distribution characteristics at the traffic coverage stage of an Erebus attack. Thus, attack traffic can be susceptivity found and accurately positioned, which depends on the multidimensional features. Transaction identity information in the traffic is also parsed to describe the abnormal differences in IP address and identity information distribution characteristics during identity forgery. This may contribute to the deep mining of hidden relationships in identity forgery. Erebus attack features are also described from a multidimensional perspective, thereby improving the model's perception and feature sensitivity toward the Erebus attacks. Consequently, both the accuracy and precision of the detection method are boosted.

3) Certain high dimension and heterogeneity characteristics exist, including redundant information and noise as shown in the routing status and Erebus attack traffic features on the blockchain network. Hence, the computing resource overhead and detection instantaneity increase and decrease, respectively. Thus, overfitting of the detection model occurs easily. Based on this problem, this study proposes a ReliefF_WMRmR-based two-stage feature selection algorithm to increase the information in various preselected feature subset features. Moreover, the ReliefF algorithm is used to evaluate, rate, and search the degree of importance of a mixed feature set, which comprises the routing status and traffic behavior. In this way, a low-dimensional feature subset is formed. Afterward, the preselected feature subset is input into the WMRmR algorithm. Through a designed feature selection framework, features with high conditional mutual information levels and strong correlations are selected. Based on the redundant features, the suspected network traffic is filtered to enhance the performance of an optimal feature subset. Due to the two-stage feature selection algorithm, the complexity of the feature space is reduced, assisting the detection model to learn the main feature variation and alleviate the overfitting phenomenon. Finally, the stability and robustness of the detection model are promoted.

4) Since the routing status and traffic behavior features reveal the network status, heterologous feature interaction exists between these networks. Especially, a network linking mode featuring a long cycle, but a low rate is adopted to produce traffic. However, a simple machine learning model may identify such traffic as normal. To solve the above problem, a Multilayer-Perceptron (MLP) based multimodal deep learning network is introduced. First, the routing status features at the input layer of

this model are isolated from its traffic behavior features, and the proposed deep learning model is used to learn and extract the input and core features of an Erebus attack, respectively. Afterward, feature learning is conducted via the MLP at the convergence layer to explore hidden feature relationships and prevent false alarm rates caused by heterogeneous data conflicts. Eventually, classifier outputs, indicating the probability distribution, are obtained to solve the hyperparameter sensitivity problem of the deep learning classifier and improve the accuracy of the proposed model.

The rest of the paper is organized as follows: Section 2 demonstrates the analysis and detection idea of the Erebus attack mode. This section also proposes Erebus-based attack detection by combining traffic behavior and routing status based on the experience and strategies of existing detection methods. Section 3 introduces the traffic behavior characteristics and routing state characteristics of Erebus attacks. This section also proposes the characteristics of the traffic behavior and routing state for Erebus attacks, according to the previous research. Based on the MLP-based multi-modal feature perception model, Section 4 achieves a robust feature extraction by proposing a two-stage feature selection algorithm based on ReliefF_WMRmR. On this basis, feature learning and model detection are conducted for the heterogeneous features of traffic behavior and routing state. In Section 5, the experiments and analysis are presented, in which a detection model is constructed and trained by collecting traffic and routing data in a real blockchain network layer environment. The superiority of the proposal was tested through experimental comparison and analysis. Section 6 concludes the paper and presents a future research direction.

2 Analysis of Erebus Attacks

2.1 Analyses of Characteristics of Erebus Attacks

The Erebus is a typical stealth partitioning attack or a multistep compound attack on the blockchain network layer [1]. The Erebus attack selects Eclipse as its fundamental condition to conduct rout leaking targeted at th. Then, the network topology is damaged due to the traffic coverage. Erebus attack also combines with Sybil to develop identity forgery for the AS and intranet nodes, thereby achieving the targeted subnet routing shielding and network isolation. The corresponding major attack procedures are described below:

First, an Erebus attacker launches route leaking, targets a subnet to perform network detection. The Border Gateway Protocol (BGP) tests report traffic, toward the blockchain subnet nodes under attack, which is sent to analyze the feedback information and deduce subnet architecture and routing relations. Several shadow IP addresses are generated and combined with nodes under attack and core BGP routing nodes to build a multichannel IP connection relationship. The goal is to establish a massive network and routing connections that contain the incorrect AS information for nodes during the attack [17]. Afterward, the Erebus attack enters a stage of traffic coverage, during which communication nodes, routing architecture, and blockchain transaction relationships are damaged. The victims are forced to construct erroneous routing and hinder normal blockchain nodes from transmitting legal transaction traffic after controlling the core AS nodes and sending maliciously built low-rate traffic to CAD nodes in a network channel where the attack occurs. Subsequently, the attacker executes identity forgery and gradually inserts the controlled puppet routing node information into a routing table of the targeted blockchain network layer nodes until the network connection with malicious peer nodes is fulfilled, and network links under attack are controlled by the attacker. Here, nodes under attack are isolated from the main part of the blockchain network layer. The Sybil attack mode is adopted during identity forgery to build a transaction relationship with blockchain nodes, generate several fake transaction identities, force the attacked blockchain nodes to establish

an erroneous transaction relation, create and control the network traffic of victims, and isolate the hashrate of the attacked nodes from the principal account. Fig. 1 shows the general approach of the Erebus attack.

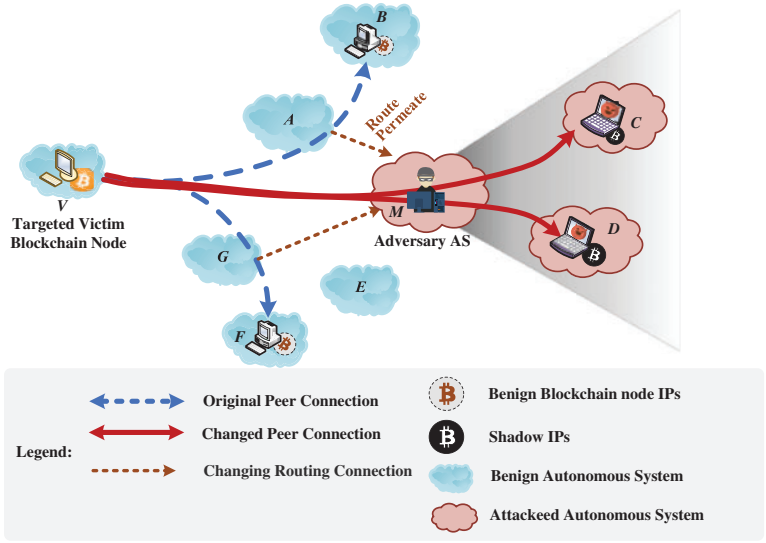

Figure 1: Erebus attack process

Unlike the typical Eclipse attack, Erebus attackers select the Internet Service Provider (ISP) as the main target and generate many 'shadow IP-fake Sybil identities' after successful infiltration in the AS. Furthermore, an erroneous blockchain transaction relationship is established, and fake transaction attacks are launched at the blockchain subnet to waste computing resources [18].

Compared with routing and traffic attacks in traditional IPv4 or IPv6 networks, Erebus attacks demonstrate the following discrepancies:

- Variations in attack target distribution: Conventional traffic and routing attacks are targeted at a single node or link; i.e., their attack targets are clear and definite, showing a distribution characteristic of 'one to one' or 'many to one.' However, Erebus is a multipath dynamic attack mode targeted at the blockchain network layer. The attacker must consider the resource and traffic constraints when selecting attack targets. The attacker must also generate several shadow IPs and fake transaction identities to hide their identities. However, Erebus attack targets vary, making the attacker dynamically adjust the relevant strategy, based on the resources to be attacked, the traffic changes, routing status, and values to be produced by such attacks. Therefore, relevant statistical information (e.g., IPs, attacker's identity and ports) is distributed in a hybrid form, such as 'one to one,' 'many to one,' and 'many-to-many.'
- Differences in attack destruction mechanisms: Traditional traffic attacks are primarily aimed at controlling the congestion mechanism of the Transmission Control Protocol (TCP)-IP protocol to exhaust core node resources. However, Erebus takes advantage of deficiencies of the abovementioned congestion control mechanism and forces the attacked node to establish a routing relation with the attacking node because of the routing update mechanism in the peer nodes of the blockchain network layer. Furthermore, the routing update mechanism can enhance attacking effects because of weak identity authentication. Thus, a routing relationship is built between the attacked and the attacking nodes to execute network isolation.

- Differences in effects caused by attacks: Traditional traffic attacks are primarily aimed at local area networks to decrease valid communication traffic with TCP and User Datagram Protocol (UDP) as transmission protocols. However, Erebus's attacking effects are mainly manifested in a varying state of repeated connection and disconnection of the AS nodes and routers in the blockchain network layer link. The goal is to damage the network architecture and affect the topological stability and reliability. As for the attacked mining nodes, some nodes are isolated from the main blockchain network because of routing screening, wasting computing resources and creating a hard fork of the bifurcation of the main network. Consequently, the resulting link to the blockchain is unique.

Fig. 2 shows the multi-stage attack process and specific attack methods of the Erebus attack.

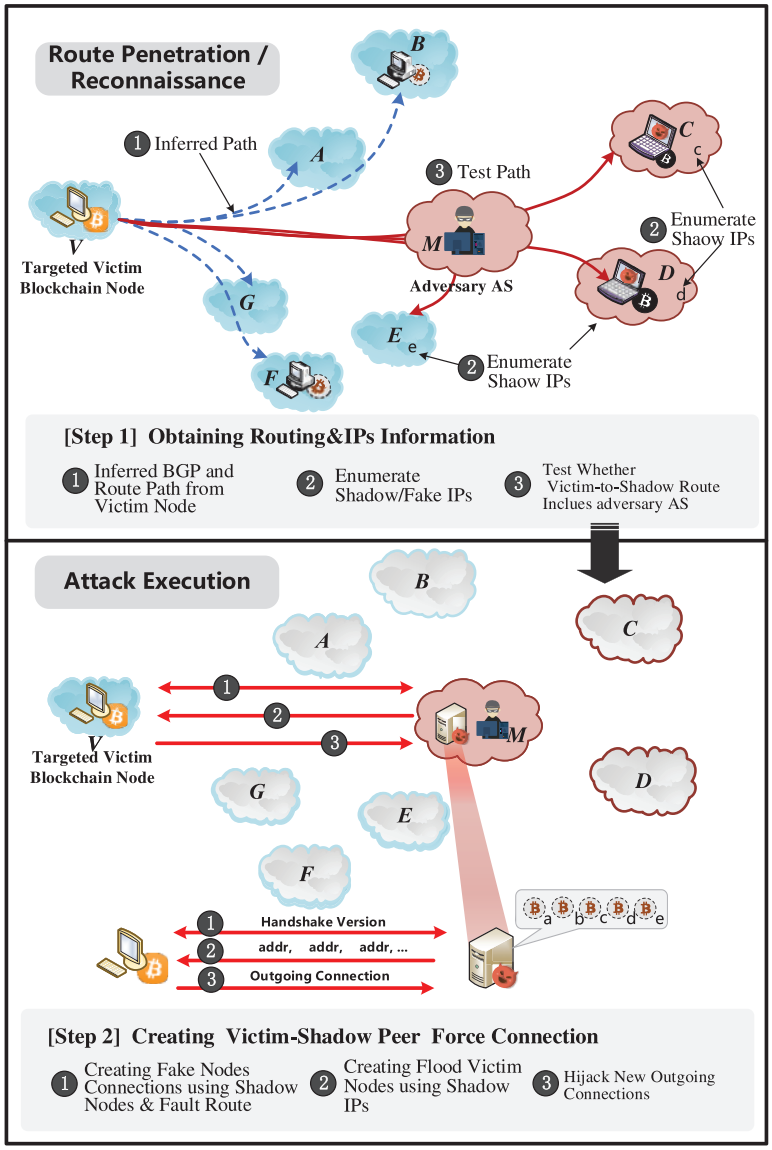

Figure 2: Multi-stage Erebus attacks

Apart from the general behavioral characteristics of the traffic and routing attacks in the existing traditional network, Erebus attacks also have some novel features in the operating mechanism of the blockchain system, summarized as follows:

- Behavior concealment: Erebus attack is a data-plane attack at the blockchain network layer, which cannot be monitored at the control level. Its attack traffic protocol shares a high similarity with normal traffic because it is characterized by long-term traffic attacks based on routing penetration and fake transactions. Thus, it is unlikely for a single index to describe the topological variations, especially when considering the routing status. Consequently, it becomes difficult for the existing defense detection to perceive the existing threats and to position abnormal links and routing. Here, the false-negative or false alarm rate is rather high.
- Address diversity: As demonstrated in the research on Erebus attacks, information in the common nodes and the primary and secondary ASs at the blockchain network layer can be easily acquired in line with route evaluation results. The AS network coverage also influences the detection results. Using wide network coverage in an interdomain topology, an Erebus attacker can effortlessly use its asymmetry advantage to generate attacking identities for diverse topologies. Consequently, statistical regularities of relevant information (e.g., attacking IPs and identities) are insignificant and cannot be expressed in a simple mathematical model.
- Attack-and-defense strategy asymmetry: Subnet vulnerabilities during an Erebus attack can be dynamically specific to a particular protocol vulnerability, causing defenders to face difficulties in repairing links or routers rapidly. A study [10] has demonstrated that the detection strategy delay may be caused by the existing method, caused by an active detection defense strategy that violates the decentralization philosophy of the blockchain, or by a nontrivial protocol update of the blockchain network layer.

2.2 Thoughts on Erebus Attack Detection

An Erebus attack unceasingly affects decisions about the AS connection and the targeted node, turning the controlled AS into a natural man-in-the-middle network for all peer-to-peer connections of the attacked nodes. The Erebus attackers may succeed in hiding their attacking identities and penetrating the attacking traffic. In this context, the concealment of the attacker's identity elevates, and the complexity of the Erebus attack detection method increases. Thus, the existing method based on a single detection object performs poorly in Erebus attack detection.

Hence, this study modifies the relevant detection ideas. Erebus attack features are comprehensively perceived from two perspectives: routing status and traffic behavior. Thus, a novel detection method was designed. Erebus attacks are detected based on the traffic behavior and the routing status because the statistical relationship between the attacking identity and traffic information distribution is mined according to traffic information, which effectively perceives the IP-attacking identity feature distribution during an Erebus attack. Moreover, an Erebus attack can be long-lasting to realize the routing coverage, thereby generating certain traffic distribution rules based on traffic statistics. Topological changes caused by an Erebus attack are perceived from the routing perspective. Machine learning models can also enhance the sensitivity features to explore the nonlinear variations at the blockchain network layer incurred by such an attack.

The fundamental assumptions of the blockchain network layer under attack proposed by one study [1] are selected herein to construct an Erebus attack detection model.

- Decentralized supervision mechanism: In a traditional peer-to-peer layer, network attackers may monitor or restrict peer-to-peer connections within the AS domain via a trusted centralized

supervision mechanism. Moreover, legal nodes may be stored based on whitelisting to establish the connections of other relay nodes [19]. However, such a method is not applicable to blockchain network layers as it violates the principle of peer-to-peer nodes of the blockchain. Therefore, a decentralized supervision mechanism is selected for the blockchain network layer detection method.

- No dynamic identity update mechanism is required: Attacking pertinency is somewhat lowered at the blockchain network layer, despite that the attacked nodes can dynamically update network identities in conformity with active defense mechanisms, such as moving targets and mimic defenses [20]. However, dynamic and frequent host identity updates at the blockchain network layer may dramatically influence the service quality of the blockchain network connection and impair hashrate stability and data uplink efficiency. Consequently, an adverse impact is applied to the blocks and transaction transmissions [21]. Therefore, the method cannot be deployed in an unlicensed decentralized blockchain network layer.
- No adoptions of cross-layer solutions: Tran et al. [1] raised a 'Smart Peer Eviction Policy', in which a consensus layer of the blockchain interacts with its network layer dynamically. Despite its practical feasibility, such a method may tremendously increase the system's complexity and introduce new systematic vulnerabilities. Thus, this study focuses on the active defense against the blockchain network layer; no cross-layer interaction strategies are adopted.

In line with the above principles, a multimodal dichotomous deep learning model is constructed based on traffic behaviors and routing status features. Moreover, a detection method was also designed.

3 Traffic Behavior and Routing Status Features of Erebus Attacks

In a machine learning model, Erebus attack samples must be learned to extract the core attack distinct features from the blockchain network layer. Thus, corresponding feature fields should be configured for model detection. Thus, it is vital to describe the distinct characteristics of this attack and routing status features during an Erebus attack to improve relevant detection accuracy.

3.1 Traffic Behavior Features

At the traffic coverage stage, routes, such as ADDR, Neighbours, and PING, are sent to the attacked subnet of the blockchain network layer to construct the command attack traffic. Such attack traffic has downstream traffic behavior and traffic attack features. Therefore, the statistical traffic features may be selected from the multiple perspectives of the traffic data at the blockchain network layer, which includes data packets and session flows. By introducing a routine traffic feature field $\{f_1, f_2, \ldots, f_{28}\}$ for the blockchain network layer, the traffic behavior features are described from the perspective of the detection window. Table 1 presents the proposed field $\{f_1, f_2, \ldots, f_{28}\}$.

Table 1: Routine traffic feature fields

No.	Name	Description
1	*duration_Time*	Duration of the data stream
2	*pkt_Count*	Quantity of packets in a single data stream
3	*byte_Count*	Number of bit packets in a single data stream
4	*pkt_Size*	Packet size
5	*access_Frequencies*	Node access frequency

(Continued)

Table 1: Continued

No.	Name	Description
6	*priority*	Array packet priority
7	*idle_Timeout*	Duration of the idle timeout
8	*cookie_ID*	Corresponding ID number of cookie
9	*actions*	Traffic action type
10	*TCP_Count*	Number of TCP session setup
11	*UDP_Count*	Number of UDP session setup
12	*TCP_Fail_Count*	Number of TCP session setup failed
13	*UDP_Fail_Count*	Number of UDP session setup failed
14	*snd_Pkt_Count*	Number of upstream packets
15	*rcv_Pkt_Count*	Number of downstream packets
16	*snd_Syn_Count*	Number of SYN packets sent
17	*rcv_Syn_Count*	Number of SYN packets received
18	*snd_Fin_Count*	Number of FIN packets sent
19	*rcv_Fin_Count*	Number of FIN packets received
20	*Snd_Len*	Upstream traffic length
21	*Rcv_Len*	Downstream traffic length
22	*connection_Per_Sec*	Number of connections corresponding to each IP
23	*TCP/UDP_rate*	Proportion of TCP packets to UDP packets
24	*server_port_Count*	Number of ports passively connected to the server
25	*Conn_Per_IP*	Number of IPs contained in each traffic connection
26	*IP_Per_Sec*	IP traffic per second
27	*protocol*	Network protocol type
28	*avg_Bits_Per_Sec*	Networked traffic rate in bytes per second

At the stage of traffic penetration of an Erebus attack, the traffic behavior features are distinguished and described by 10 representative features, such as the average session duration, proportions taken by low-rate data packets, and the distance deviation values of data packets. Based on the features of these attacks, it is vital to enhance the model's awareness of the Erebus attack behavior features. Moreover, transaction information in traffic is parsed, and the distribution features of the attackers' identities are defined based on the multiple fake identity attacks. Detailed feature definitions are as follows:

1) Average session duration: The attacker continuously occupies the link channels of the AS during the routing penetration of an Erebus attack to achieve the goal of route table coverage and route topology damages. Therefore, the attacker may extend the duration of a traffic session to link the attacking conditions. In this context, the average session duration is obtained to describe the traffic behavior of the Erebus attack shown as follows:

$$f_{T_1} : Session\_Duration\_Avg = \frac{\sum_{t=0}^{T} session\_Duration_t}{T} \tag{1}$$

where t is the number of times in statistics, and T is the total number of times of collecting data within the window.

2) Proportions of low-rate data packets: Data packet rate is the time interval from the first data packet transmission when attacking the host to the time when such a data packet arrives at the targeted node of the attack. Since a blockchain network layer is a heterogeneous network, the nodes interact at a low transmission rate to reduce communication overhead and ensure communication service quality among node traffic. At the stage of penetrating the ISP, an Erebus attack proactively emulates interactive behavior among normal node traffic to increase the hidden attacking traffic. The data packets are sent at a low rate, and the low-rate data packets are adapted to measure the traffic behavior of the Erebus attack.

$$f_{T_2} : LowRate\_Pkt\_Ratio = \frac{\sum_{t=0}^{T} LowRate\_Pkt\_Duration_t}{\sum_{t=0}^{T} Pkt\_Duration_t} \tag{2}$$

3) Average idle time of traffic: The difference in the idle time of traffic is the time interval between respective sets of traffic in a traffic session. In practice, the consensus mechanisms differ between different types of blockchain. Thus, significant differences exist between the network environment and service quality of various blockchain network layers, making the difference in traffic idle time inconsistent between normal nodes. For example, traffic is generated during the Erebus attack, producing features of periodicity and a small idle time difference. Considering this, autocorrelation coefficients were used to figure out weighted idle time difference deviation values and describe the overall spatiotemporal distribution and periodic distribution of the network traffic.

$$flow\_idle = \frac{\sum_{t=0}^{T} |flow\_idle_t|}{\sum_{t=0}^{T} flow\_Num_t} \tag{3}$$

$$f_{T_3} : flow\_var\_idle = 1/\frac{\sum_{t=0}^{T} |flow\_idle_t - flow\_idle_{avg}|}{\sum_{t=0}^{T} flow\_Num_t} \tag{4}$$

4) An average number of data packets in a session flow: The number of data packets in the Erebus attack traffic differs from that of the traffic of the normal blockchain network layer. Each session in the normal blockchain network layer traffic contains massive data packets; whereas an Erebus attacker generates fake traffic of the source IP at the traffic coverage stage and initiates a session traffic coverage targeted at the CAD nodes in a blockchain network. Such behavior may lead to a drop in the average number of data packets in each session. Therefore, the average number of data packets in a session flow is introduced here to describe the relevant traffic behavior characteristics.

$$f_{T_4} : PktNum\_Avg\_TrafficNum = \frac{\sum_{t=0}^{T} pkt\_Num_t}{\sum_{t=0}^{T} traffic\_Num_t} \tag{5}$$

5) An average number of bytes in a data packet: Interactive traffic load information at a normal blockchain network layer comprises several blockchain transaction messages; the length of such load information exceeds that of the attacking traffic. In an Erebus attack, efficient improvement is required at the traffic penetration stage. Moreover, the attacking traffic load information comprises route construction commands. This data category is generated by an attacking program or a script, in which the corresponding packet length remains comparatively constant. The byte length of the load information and the total byte length is relatively small. Thus, the average number of bytes in a data packet is introduced to describe differences in traffic behaviors.

$$f_{T_5} : pktSize\_Avg\_trafficNum = \frac{\sum_{t=0}^{T} pkt\_Size}{\sum_{t=0}^{T} traffic\_Num_t} \tag{6}$$

6) Percentage of non-peer-to-peer session flows: Peer-to-peer session flow is the IP_*src* of a session flow A serving the IP_*dest* of a session flow B, and the IP_*dest* of session flow A is the IP_*src* of session flow B. Under such circumstances, the protocols of session flows A and B are peer-to-peer because traffic is in a dynamically interactive mode in the normal blockchain network layer environments. In this process, the statistical distribution of various sessions is highly random. However, an Erebus attack aims at executing the routing coverage and the traffic penetration. In this wise, the source IP addresses must be forged, where the downstream traffic plays a major role. Additionally, the percentage of non-peer-to-peer session flows increases, this percentage is adopted to express the corresponding traffic behaviors.

$$f_{T_6} : Bidirectional\_Session\_Traffic\_Ratio = 1 - \frac{\sum_{t=0}^{T} Bidirectional\_Session\_Traffic_t}{\sum_{t=0}^{T} Session\_Traffic_t} \tag{7}$$

7) Entropy value distribution of network host IDs: Many fake IDs and Sybil attack transaction identities are randomly generated during the identity forgery of an Erebus attack. Consequently, the distribution of randomness of identity information becomes obvious in the host. The entropy values, representing the host identity distribution of the Erebus attack traffic, are above that of the normal traffic. By analyzing the transaction identification information in data packets and obtaining the frequency and entropy value distribution through statistics, the distribution of abnormal identities is more clearly defined as follows:

$$f_{T_7} : H\left(ID_{src}\right) = -\sum_{x \in ID_{src}} p\left(x\right) \log p\left(x\right) \tag{8}$$

$$f_{T_8} : H\left(ID_{dest}\right) = -\sum_{x \in ID_{dest}} p\left(x\right) \log p\left(x\right) \tag{9}$$

where $H\left(\cdot\right)$ represents the information entropy formula, and $p\left(\cdot\right)$ represents the probability of an event in a sample.

8) Entropy ratios of source IP address to host identity: The source IP attack traffic may contain traffic information of different transaction identities to conceal an Erebus attack. Thus, both the IP_src address and the transaction identity distributions are highly random in Erebus attack traffic; whereas the distribution of transaction hosts and IP_src addresses remains unchanged in a normal blockchain network layer. Moreover, the entropy ratios of the src_IP in the Erebus attacking traffic to host identities are above those of the normal attacking traffic.

$$f_{T_9} : D_{KL}\left(ID_{src} || ID_{dest}\right) = \sum_{x \in ID_{dest}} \sum_{y \in ID_{src}} p\left(x\right) \log\left(\frac{x}{y}\right) \tag{10}$$

9) Traffic variation rate at the portal: Once an Erebus attack starts building routing relationships, malicious attacking traffic is transmitted to the AS or a subnet. Consequently, more connection requests are developed from the subnet host, and the traffic rate at the network portal increases significantly within a specified statistical time window. Therefore, the traffic variation rate at the portal is based on the corresponding traffic behaviors:

$$f_{R_{10}} : Inbound\_Traffic\_\text{Rate} = \frac{\sum_{t=0}^{T} Inbound\_Traffic\_Num_{lt}}{T} \tag{11}$$

All traffic behaviors are adopted to form a feature set, denoted as $\boldsymbol{f_F} = \{f_{F1}, f_{F2}, f_{F3}, \ldots, f_{Fn}\}$, where $n = 38$.

3.2 Routing Status Features

A multistage Erebus attack is characterized by important nodes, such as the CAD within the AS of the blockchain network layer. Thus, routing status features are defined to objectively describe both the process and characteristics of the Erebus attack, which include the routing status features of the route update information and routing architecture.

1) CAD state parameter variation rate: The routing penetration of an Erebus attack is targeted at the CAD nodes of the blockchain network layer, such as ISP, routers, and switches. Moreover, an Erebus program must control many CAD nodes to execute selective isolation of the blockchain nodes attacked in the AS. Through frequent statistics and feature extraction from the data associated with the CAD nodes of adding a new router (ANNOUNCE messages), removing the old router (WITHDRAW messages), session reset, and route update messages and variation rate of the message quantity feature in unit time are calculated to describe state variations of the key route nodes:

$$f_{R_1} : \text{ANNOUNCE}_{CAD}\_Num\_\text{Ratio} = \frac{\sum_{l=0}^{L}\sum_{t=0}^{T}\text{ANNOUNCE}\_Num_{lt}}{T} \tag{12}$$

$$f_{R_2} : \text{WITHDRAW}_{CAD}\_Num\_\text{Ratio} = \frac{\sum_{l=0}^{L}\sum_{t=0}^{T}\text{WITHDRAW}\_Num_{lt}}{T} \tag{13}$$

$$f_{R_3} : \text{UPDATE}_{CAD}\_Num\_\text{Ratio} = \frac{\sum_{l=0}^{L}\sum_{t=0}^{T}\text{UPDATE}\_Num_{lt}}{T} \tag{14}$$

$$f_{R_4} : \text{RESETSESSION}_{CAD}\_Num\_\text{Ratio} = \frac{\sum_{l=0}^{L}\sum_{t=0}^{T}\text{RESETSESSION}\_Num_{lt}}{T} \tag{15}$$

where, l represents the serial numbers of routes at respective nodes, and L is the total number of subnet routes within the statistical window.

2) Routing request construction traffic of CAD nodes and links: Under an Erebus attack, the attacker may generate route request construction traffic for CAD nodes and the corresponding links. In most cases, a routing construction command comprises a program and a script, periodically distributes the command traffic. On this occasion, the routing status variations of the CAD nodes are described by the periodical distribution of the routing construction command quantities:

$$f_{R_5} : Route_{\text{CAD}}\_REQUEST\_Num = \frac{\sum_{l=0}^{L}\sum_{t=0}^{T}Route\_REQUEST\_Num_{lt}}{T} \tag{16}$$

3) Node request traffic of subnet route: At the routing coverage stage, routing update messages are extensively forwarded to all nodes in the subnet to cover the subnet routing architecture. Thus, the subnet routing status variations are expressed in the session reset and routing update messages of all subnet routing nodes in the AS as follows:

$$f_{R_6} : \text{UPDATE}_{SubNet}\_Num\_\text{Ratio} = \sum_{l=0}^{L}\sum_{t=0}^{T}\text{UPDATE}\_Num_{lt} \tag{17}$$

$$f_{R_7} : \text{RESETSESSION}_{SubNet}\_Num\_\text{Ratio} = \sum_{l=0}^{L}\sum_{t=0}^{T}\text{RESETSESSION}\_Num_{lt} \tag{18}$$

4) AS prefix variation rate: Under an Erebus attack, the prefix hijacking is adopted to execute the routing penetration, in which the attacker broadcasts that they own AS prefixes in the blockchain network layer. This may lead to route switches and network traffic diversion. In this process, the hijacked route becomes illegitimate; it continuously broadcasts among nodes of the blockchain network layer and destroys the routing architecture. Finally, the prefix information of the involved

route may change significantly. In normal blockchain network layers, communication paths between respective nodes are stable. The network distance from a node of the blockchain network layer to the targeted prefix node is also stable. Under such circumstances, the variation rates of the attribute values (e.g., AS_Path, SRC_Path, and Next_Hop in AS) in unit time are selected here to determine whether an Erebus attacker executes the routing penetration or traffic hijacking targeted at the AS [14]:

$$f_{R_8} : ASPath\_Rate = \frac{\sum_{l=0}^{L}\sum_{t=0}^{T} ASPath\_Len_{lt}}{T} \tag{19}$$

$$f_{R_9} : SRCPath\_Rate = \frac{\sum_{l=0}^{L}\sum_{t=0}^{T} SRCPath\_Len_{lt}}{T} \tag{20}$$

$$f_{R_{10}} : NextHop\_Rate = \frac{\sum_{l=0}^{L}\sum_{t=0}^{T} NextHop\_Len_{lt}}{T} \tag{21}$$

5) Subnet architecture measurement: The number of nodes in the respective subnets may change because an Erebus attack may destroy the routing structures of various subnet nodes during routing penetration. Therefore, the average number of neighbor nodes corresponding to each subnet node is obtained and used to describe the subnet node architectural variations as follows:

$$f_{R_{11}} : Route_{SubNet}\_Neighbor\_Ratio = \frac{\sum_{l=0}^{L}\sum_{t=0}^{T} Route_{SubNet}\_ASNeighbor\_Num_{t}}{T} \tag{22}$$

6) Path variation rates for routing nodes: When the CAD nodes, such as the AS, are under the routing penetration attack at the blockchain network layer, the architectural routing information, such as the ASPath, forces a change in the AS paths of nodes under attack. During an Erebus attack, the BGM in the controlled AS dynamically informs the surrounding nodes about the path accessibility and length. At the routing penetration stage, topological or AS routing strategy variations may correspond to the generation of many ASPath update messages. The topological variations of the AS under attack are described after measuring the various rates of the ASPath attributes for all paths in the blockchain network layer at different times. Here, the rate of summation path length is introduced to evaluate the differences in the total ASPath length at different times:

$$f_{R_{12}} : Route_{CAD}\_Length\_Ratio = \frac{\sum_{l=0}^{L}\sum_{t=0}^{T} Route\_ASPath\_Len_{lt}}{T} \tag{23}$$

Routing status features are collected to form a routing status feature set $\boldsymbol{f_R}$, where $\boldsymbol{f_R} = \{f_{\boldsymbol{R}1}, f_{\boldsymbol{R}2}, f_{\boldsymbol{R}3}, \ldots, f_{\boldsymbol{R}m}\}$, and $m = 12$.

4 A MLP-Based Multi-Modal Feature Awareness Model

The Erebus attack features are comprehensively perceived according to the routing status and traffic behavior. Moreover, a detection method is designed when transforming the Erebus attack detection into a machine learning-based classification and detection problem. The corresponding detection process is presented below. First, the normal traffic and the traffic under an Erebus attack are collected from a real blockchain network layer. The traffic routing status in normal cases and during an Erebus attack is also collected. Moreover, network traffic and routing status messages with the same time stamp and data stream ID are classified into the same set. Afterward, two-stage feature selection algorithms, such as ReliefF and WMRmR, are used to select and filter the features. Then, the features that are strongly correlated and less redundant are adopted. Then, features of the traffic behavior and routing status are respectively extracted from a set of traffic data and messages and inputted

into the T_2R_2C_Deep Neural Network (T_2R_2C_DNN) or deep feature learning and extraction. This way, abstract features are generated and fused. Finally, the SoftMax classifier is utilized to output dichotomous detection outcomes. Fig. 3 shows the overall operation mode of the proposed Erebus attack detection method.

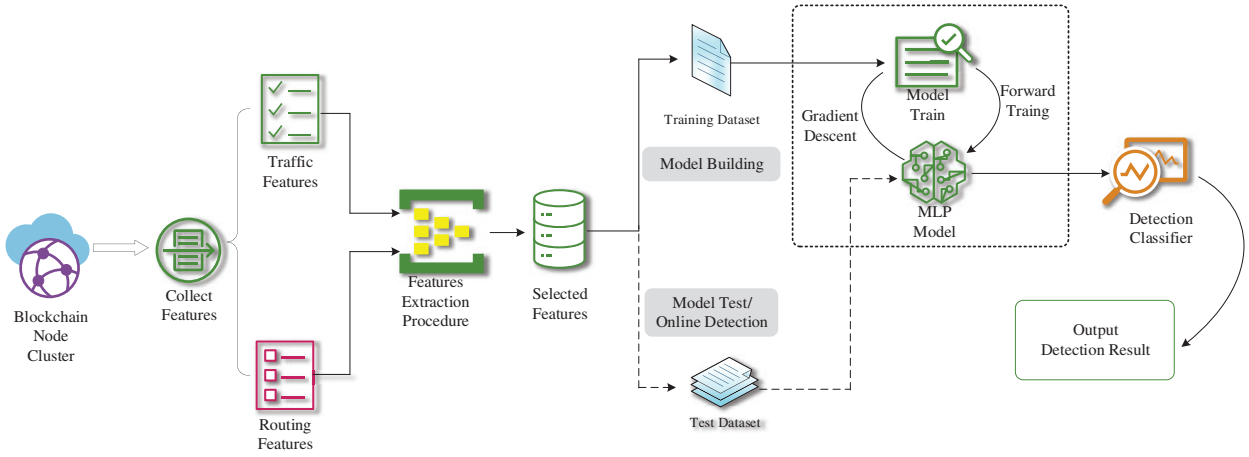

Figure 3: A flow diagram of the proposed Erebus attack detection model

4.1 Two-Stage Feature Selection Algorithm Based on ReliefF_Wmrmr

Since the routing status and traffic behavior features are derived from heterologous data, input feature dimensions are raised for the deep learning model during the heterologous feature extraction. Moreover, features containing a small amount of information and interfering variables are introduced. Consequently, model complexity and detection time overhead are elevated. Traffic behavior and routing status are features that describe the system security of the blockchain network layer. Moreover, superposed information about features exists and redundant information between these features. Data conflicts and training model overfitting are incurred during the deep learning model training, which further affects the detection accuracy and model stability. Therefore, essential heterogeneous features are selected from various features as the model input to reduce space and time overheads of detection and prevent input feature model overfitting.

The existing feature selection algorithms are classified into two. The first depends on the classification, such as a filter or wrapper-based feature selection; the second is independent of classification, which includes the embedded-based feature selection. However, the wrapper-and embedded-based feature selection methods do not measure the combined effect of features. Thus, their performance in selecting redundant mutual information among multisource input features is rather poor, and the selected features may contain redundant feature information. A two-stage feature selection algorithm based on ReliefF_WMRmR is introduced herein to suppress redundant features, select features with high mutual information from multisource features, and reduce feature dimensions. First, feature selection is performed using ReliefF built on the original routing status and traffic behavior features during Erebus attack detection. Then, correlations between the original features and label information are calculated. Next, a subset of weight features that significantly influence classification results is selected. Hence, the neural network learns the main feature variation during training. Thereafter, the WMRmR algorithm is used to filter the redundant features in a feature subset, simplify the feature

subsets, improve the Erebus attack feature dataset detection effects of classification based on the model, and enhance the stability and robustness of the model. Fig. 4 shows the corresponding process.

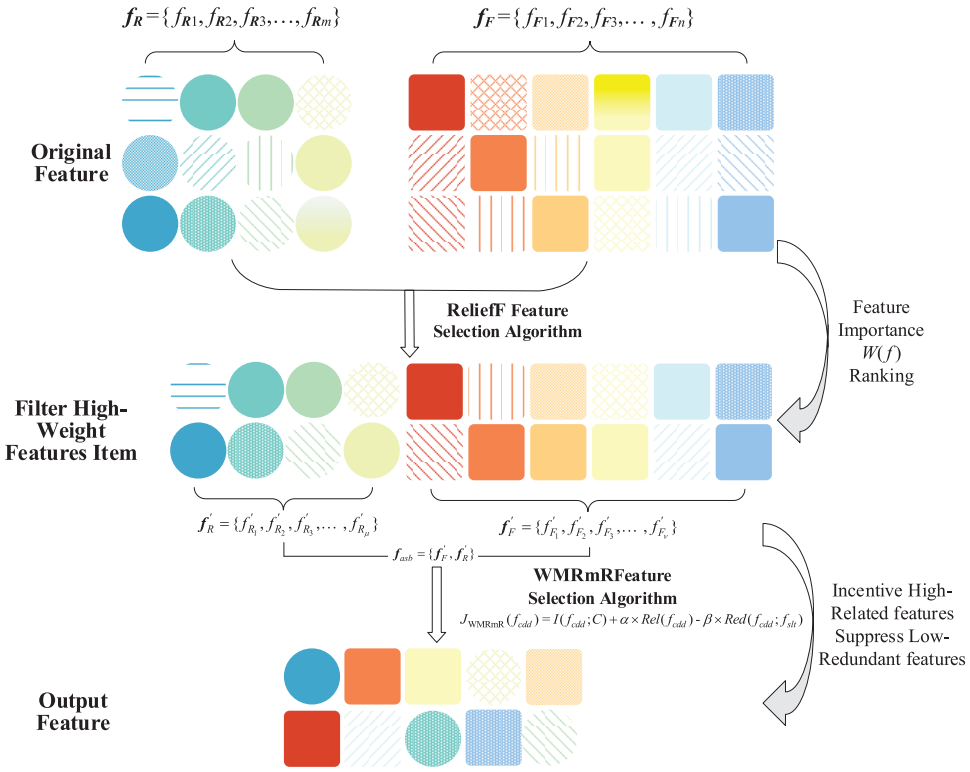

Figure 4: A flow diagram of feature selection based on ReliefF_WMRmR

ReliefF is a filter-based feature selection method of continuous eigenvalues that can effectively perceive insensitive feature items among various features and select feature items strongly correlated with the classification labels. This method is adopted to select a subset of the dimension reduction features from the original features of the routing status and traffic behavior during the preprocessing stage. Moreover, the features are sorted according to the feature correlation weights [22]. The main steps are as follows: a sample R is randomly extracted each time from the original samples. Afterward, k neighboring samples (*i.e.*, H_j ($j = 1, 2, \ldots, k$)) are searched in the same class as sample R, and k neighboring samples are denoted as M in other classes. Then, we determine the weights representing correlations between respective feature items A and the classification label. The features are ranked in line with the feature importance W, and highly important features are selected. Thus, such a process should be repeated m times, and the corresponding correlation weights are expressed as follows [23]:

$$W(A) = W(A) - \sum_{j=1}^{k} \text{diff}(A, W, H)/(m \cdot k) + \sum_{T \notin R} \left[\frac{p(Class(T))}{1 - p(Class(R))} \right] \sum_{j=1}^{k} \text{diff}\left(A, W, M_j(T)\right)/(m \cdot k) \tag{24}$$

where $W(A)$ is the correlation weight of feature item A. The higher the value of $W(A)$, the better the performance of feature item A in distinguishing samples. Moreover, diff (A, R_1, R_2) is a correlation weight residual of sample R_1 relative to R_2. In addition, $M_j(T)$ is the j^{th} nearest sample in class T. The

corresponding equation of diff (A, R_1, R_2) is given as follows:

$$\text{diff}(A, R_1, R_2) = \begin{cases} \dfrac{|R_1[A] - R_2[A]|}{\max(A) - \min(A)} & \text{if } A \textit{ is continuous} \\ 0 & \text{if } A \textit{ is discrete and } R_1[A] = R_2[A] \\ 1 & \text{if } A \textit{ is discrete and } R_1[A] \neq R_2[A] \end{cases} \tag{25}$$

The process of the ReliefF-based feature selection algorithm is described as follows:

Algorithm 1: ReliefF based feature selection algorithm

Input: **Raw dataset** D; Class label set C; Input feature set ***f;*** selecting threshold δ of ReliefF; frequency in sampling *m;*

Output: A feature set ***f'*** after extraction

Start

```
Weight item W for respective features is set at 0;
f'←Φ;
 for i = 1 to m do:
      A sample R is randomly extracted from D;
      To search k nearest samples denoted by H_j in the same class of R;
      To search k nearest samples denoted by M_j(T) of R in other classes;
   for i = 1 to N All features do:
      To calculate W(f_i) according to the equation:
      if W(f_i) > δ
          f_i is added to f'
   End For
```

End

The routing status feature $\boldsymbol{f}_R$ and the traffic behavior feature $\boldsymbol{f}_F$ are substituted into algorithm 1, which outputs the sorted and selected eigenvectors $\boldsymbol{f}'_R$ and $\boldsymbol{f}'_F$, respectively, thereby forming an eigenvector set $\boldsymbol{f}_{asb}$.

$$\boldsymbol{f}'_R = \{f'_{R_1}, f'_{R_2}, f'_{R_3}, \ldots, f'_{R_\mu}\} \tag{26}$$

$$\boldsymbol{f}'_F = \{f'_{F_1}, f'_{F_2}, f'_{F_3}, \ldots, f'_{F_v}\} \tag{27}$$

$$\boldsymbol{f}_{asb} = \{\boldsymbol{f}'_F, \boldsymbol{f}'_R\} \tag{28}$$

A failure of the ReliefF in measuring the combined effect of multisource feature subsets enables the model to generate similar redundant features. Herein, the feature selection framework and evaluation criteria, proposed by Brown et al. [24] and Zhang et al. [25], respectively, were employed to raise a WMRmR-based feature evaluation standard J_{WMRmR} and to design a detection algorithm. A feature subset was generated in combination with ReliefF to select strongly correlated features of much conditional mutual information. Suppressing the redundant features is vital to improving the selection quality of the optimal feature subset. Here, the standard J_{WMRmR} is expressed as follows:

$$J_{\text{WMRmR}}(f_{cdd}) = I(f_{cdd}; C) + \alpha \times Rel(f_{cdd}) - \beta \times Red(f_{cdd}; f_{slt}) \tag{29}$$

$$Rel(f_{cdd}) = \sum_{f_{slt} \in f_s} I(f_{slt}; f_{cdd}) \tag{30}$$

$$Red\,(f_{cdd};f_{slt}) = Rel\,(f_{slt}) - \frac{H(f_{slt}|f_{cdd})}{H(f_{slt})} Rel\,(f_{slt}) \tag{31}$$

where $J(\cdot)$ is the feature evaluation standard, and $I(\cdot)$ is the mutual information formula [26]. More particularly, f_{cdd} and f_{slt} are the features to be selected and selected features, respectively, where $f_{slt} \in \boldsymbol{f}_s$ and $\boldsymbol{f}_{cdd} \in \boldsymbol{f}_{asb} - \boldsymbol{f}_s$. Moreover, $\boldsymbol{f}_S$ is the optimal feature subset ($\boldsymbol{S} \subset \boldsymbol{f}_{asb}$); $|\boldsymbol{f}_{asb}|$ represents the number of features outputted by ReliefF from the feature subset; C is a label set, and $|C|$ represents the number of labels. According to J_{WMRmR}, $I\,(f_{cdd};C)$ is the correlation between f_{cdd} and C. Through the $Rel\,(f_{cdd})$ maximization approach, the amount of independent mutual information lost is minimized. Through the $Red\,(f_{cdd};f_{slt})$ minimization approach, conditional correlation and redundant information of f_{cdd} and f_{slt} are decreased. J_{WMRmR} is designed by referring to a feature selection algorithm standard of minimum redundancy and maximum relevance. Considering the correlations of features and labels, this process influences important features on the overall datasets and labels and the performance of mutual information during feature selection.

We also weigh the impact of the redundant information and independent mutual information of features on the J_{WMRmR} standard design to establish the weighted values α and β. In this case, a standard deviation is introduced to calculate their weights. Considering that standard deviations can measure the system stability, the relative importance between the redundant and independent mutual information is dynamically and adaptively balanced. As a result, the weights of respective items are denoted as follows:

$$\mu_\alpha = \frac{1}{|f_s|}\sum_{i=1}^{|fs|} I\,(C;f_{cdd}|f_{slt}) \tag{32}$$

$$\alpha = \sqrt{\frac{1}{|f_s|}\sum_{i=1}^{|fs|} (I(C;f_{cdd}|f_{slt}) - \mu_\alpha)^2} \tag{33}$$

$$\mu_\beta = \frac{1}{|f_s|}\sum\nolimits_{i=1}^{|fs|} I\,(C;f_{slt}|f_{cdd}) \tag{34}$$

$$\beta = \sqrt{\frac{1}{|f_s|}\sum_{i=1}^{|fs|} (I\,(C;f_{slt}|f_{cdd}) - \mu_\beta)^2} \tag{35}$$

where $|\boldsymbol{f}_s|$ represents the number of features in an optimal feature subset. The two-stage feature selection algorithm based on ReliefF and J_{WMRmR} is described.

Algorithm 2: two-stage feature selection algorithm based on ReliefF and J_{WMRmR}

Input: **Dataset** D; Class label set C; Routing status feature set $\boldsymbol{f}_R$; traffic behavior feature set $\boldsymbol{f}_F$; threshold κ of J_{WMRmR};

Output: Optimal feature set $\boldsymbol{f}_s$

Start

1: Initialization

2: $\boldsymbol{f}_R^{'} \leftarrow \Phi$

3: $\boldsymbol{f}_F^{'} \leftarrow \Phi$

(Continued)

Algorithm 2: Continued

4: $\boldsymbol{f}_s \leftarrow \Phi$
5: $set\_JMRmR \leftarrow \Phi$
6: To select features from $\boldsymbol{f}_R$ according to Algorithm 1, and generate $\boldsymbol{f}'_R$
7: To select features from $\boldsymbol{f}_F$ according to Algorithm 1, and generate $\boldsymbol{f}'_F$
8: $\boldsymbol{f}_{asb} = \{\boldsymbol{f}'_F, \boldsymbol{f}'_R\}$
9: for each $\forall f_i \in \{\boldsymbol{f}_{asb}\}$ do:
10: To calculate $I(f_i; C)$, and input it into the set_JMRmR set;
11: $f_{\max} = \text{argmax}\ (set\_JMRmR)$
12: Set $\boldsymbol{f}_{asb} \leftarrow \boldsymbol{f}_{asb} \backslash \{f_{\max}\}$
13: Set $\boldsymbol{f}_s \leftarrow \{f_{\max}\}$
14: while $\boldsymbol{f}_{asb}$ is not a null set && $cdd < \kappa$:
15: for each $f_{cdd} \in \boldsymbol{f}_{asb}$ do:
16: To calculate parameter items $\mu_\alpha, \alpha, \mu_\beta, \beta$ based on
17: To calculate $J_{\text{WMRmR}}(f_{cdd})$ according to Equation
18: To search the optimal candidate feature f_{cdd} according to $J_{\text{WMRmR}}\ (f_{cdd})$
19: Set $\boldsymbol{f}_{asb} \leftarrow \boldsymbol{f}_{asb} \backslash \{f_{cdd}\}$
20: Set $\boldsymbol{f}_s \leftarrow \{f_{cdd}\}$
21: $cdd = cdd + 1$
22: end while

End

In algorithm 2, the optimal feature set $\boldsymbol{f}_s$ and the relevant parameter sets are initialized in steps 1–5. From steps 6–8, the preliminary selection of $\boldsymbol{f}'_R$ and $\boldsymbol{f}'_F$ are conducted under ReliefF, generating a rough eigenvector set $\boldsymbol{f}_{asb}$ and requiring further selection. In steps 9–13, the feature items, correlated to labels, are preliminarily selected from $\boldsymbol{f}_{asb}$. Moreover, the maximum correlated feature item is pre-set; then, this is eliminated from $\boldsymbol{f}_{asb}$ and added to $\boldsymbol{f}_s$. Finally, in steps 14–22, the greedy algorithm with a forward-looking search strategy is used to calculate the weights of items in J_{WMRmR} based on the equation, thereby producing assessed values denoted by $J_{\text{WMRmR}}\ (f_{cdd})$ for candidate feature items f_{cdd}. Moreover, features exceeding the optimal feature items are selected from $J_{\text{WMRmR}}\ (f_{cdd})$, deleted from $\boldsymbol{f}_{asb}$, and eventually added to $\boldsymbol{f}_s$.

4.2 A Multi-Modal Deep Learning Model Based on MLP

The routing status and traffic behavior fall into the heterogeneous data category. A multimodal deep neural network was selected for abstract expression, dichotomous detection of the routing status, and processed traffic behavior features to prevent a decline in detection accuracy caused by source data conflicts. The fusion of heterogeneous feature data comprehensively improves the model's ability to detect Erebus attack samples.

MLP is a classification model that can construct a hyperplane in the space of multiple heterogeneous samples. Relevant data are classified within the hyperplane after minimizing the distance from a misclassification point to a separating hyperplane. Herein, the proposed multimodal deep learning network model is portrayed in Fig. 5. The model comprises five DNNs, i.e., two DNNs of the traffic behavior (i.e., Traffic-DNN_1 and Traffic-DNN_2), two DNNS of the routing status (i.e., Routing-DNN_1 and Routing-DNN_2), and a DNN of the processing units and the convergence layer

(i.e., Converge-DNN). The model is named T_2R_2C_DNN, considering the architecture. First, Traffic-DNN_1 and Routing-DNN_1 form no direct correlations independent mutually extracting abstract features from two of the traffic behavior and routing status features, respectively. The goal is to effectively prevent data conflicts between traffic behavior and routing status features. Subsequently, Traffic-DNN_2 and Routing-DNN_2 recode the generated features. By increasing the deep learning model's depth, the proposed model is boosted to express the intermediately distinguishing feature information reconstruction. Finally, the Converge-DNN performs feature extraction and fusion specific to abstract features extracted by Traffic-DNN_2 and Routing-DNN_2, thereby producing fusion features of the output data.

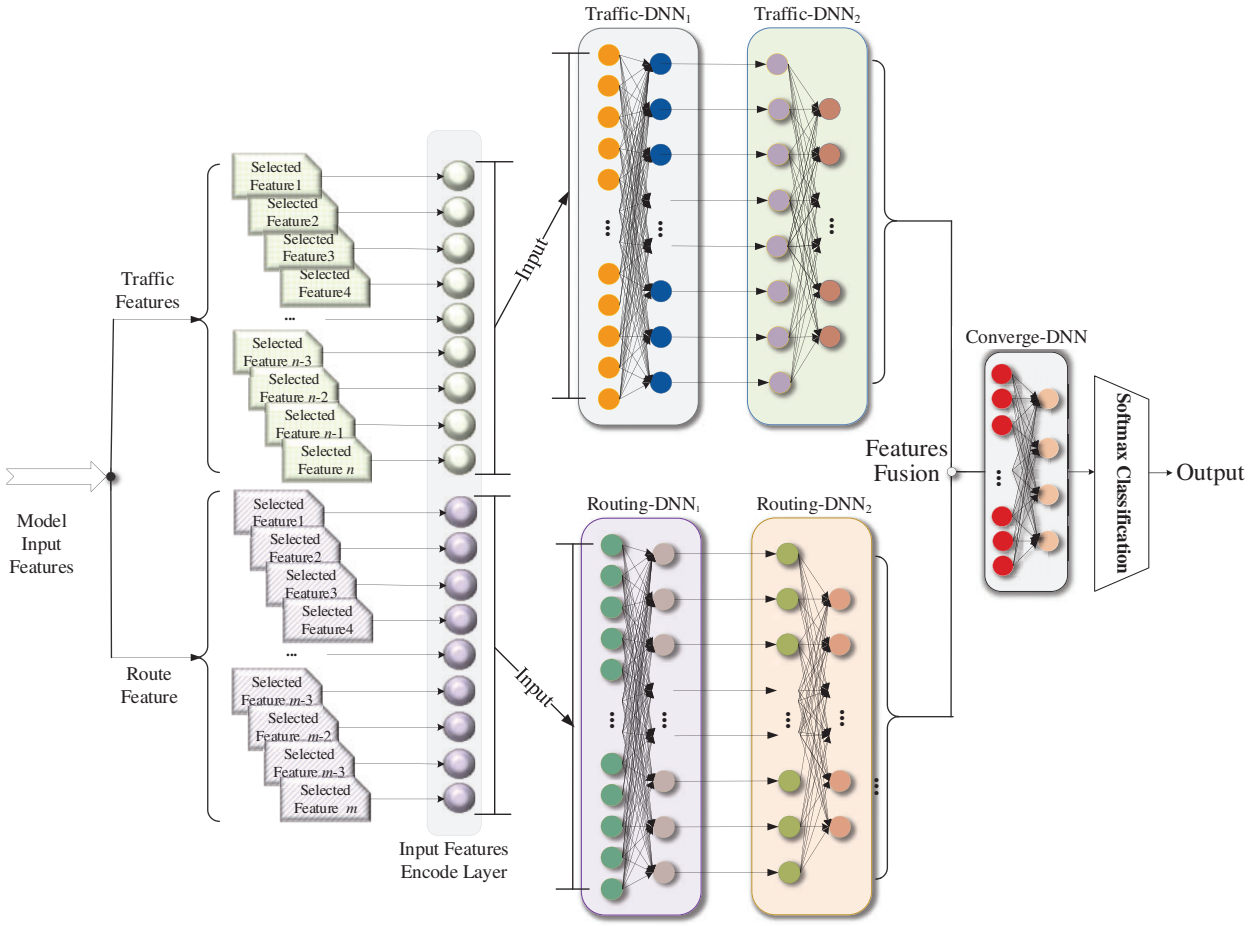

Figure 5: T2R2C_DNN model architecture

The T_2R_2C_DNN detection process is as follows: x^T and x^R refer to the eigenvectors of the traffic behavior and routing status, respectively. Vector set $\{x^T, x^R\}$ serves as the input of the multimodal deep neural network, and the preliminary feature extraction is performed by Traffic-DNN_1 and Routing-DNN_1. Considering that Traffic-DNN_1, Traffic-DNN_2, Routing-DNN_1, and Routing-DNN_2 have the same DNN model structure, only the Traffic-DNN_1 structure is described. Moreover, l_T represents the serial numbers of the respective layers in Traffic-DNN_1; $\boldsymbol{h}^{lT}$represents the output value of layer l_T; $\boldsymbol{W}^{lT}$ is the weight coefficient of layer l_T, and $\boldsymbol{b}^{lT}$ is the offset of layer l_T. The swish activation function is

denoted by $f(x)$ [27], expressed as follows:

$$\sigma(x) = x \cdot (1 + e^{-x})^{-1} \tag{36}$$

In Traffic-DNN, the following equation is written for the output $\boldsymbol{h}^{lT}$ of the respective layers in this neural network:

$$\boldsymbol{h}^{lT} = \begin{cases} \sigma(\boldsymbol{x}^T \cdot \boldsymbol{W}^{lT} + \boldsymbol{b}^{lT}), & l_t = 0 \\ \sigma(\boldsymbol{h}^{lT-1} \cdot \boldsymbol{W}^{lT} + \boldsymbol{b}^{lT}), & l_t = 1, 2, ..., n \end{cases} \tag{37}$$

The traffic behavior and routing status eigenvectors output the abstract eigenvector set $\{\boldsymbol{y}^T, \boldsymbol{y}^R\}$ after feature extraction through Traffic-DNN_{12} and Routing-DNN_{12}. By executing this combination, an output vector of convergence is obtained and written as follows:

$$\boldsymbol{y}^{Converge} = \boldsymbol{y}^T \oplus \boldsymbol{y}^R \tag{38}$$

Furthermore, $\boldsymbol{y}^{Converge}$ is used as the Converge-DNN input to execute feature extraction the feature fusion. Similar to Traffic-DNN_1, $\boldsymbol{l}_M$ represents the serial numbers of various layers in the Converge-DNN. Moreover, $\boldsymbol{h}^{lM}$, $\boldsymbol{W}^{lM}$, and $\boldsymbol{b}^{lM}$ refer to the output value, the weight coefficient, and the offset of layer l_M, respectively. The Sigmoid activation function is selected for the last layer of the Converge-DNN, expressed as follows:

$$g(x) = (1 + e^{-x})^{-1} \tag{39}$$

A layer $\boldsymbol{h}^{lM}$ of the Converge-DNN is also written as follows:

$$\boldsymbol{h}^{lM} = \begin{cases} \sigma(\boldsymbol{y}^{Converge} \cdot \boldsymbol{W}^{lM} + \boldsymbol{b}^{lM}), & l_M = 0 \\ \sigma(\boldsymbol{h}^{lM-1} \cdot \boldsymbol{W}^{lM} + \boldsymbol{b}^{lM}), & l_M = 1, 2, ..., m-1 \\ g(\boldsymbol{h}^{lM-1} \cdot \boldsymbol{W}^{lM} + \boldsymbol{b}^{lM}), & l_M = m \end{cases} \tag{40}$$

The SoftMax classifier is used to normalize the output values z^M of Converge-DNN and the output classification results from k types of traffic. According to the probability values, the SoftMax classifier may map the output of multiple neurons into [0, 1); and the sum of output values is 1, expressed as follows:

$$S\left(z^M\right) = \frac{e^{z_i^M}}{\sum_k e^{z_i^M}} \tag{41}$$

The SoftMax function maps the model output to [0, 1), with the cumulative sum of the outputs as 1. As a result, the output value of the SoftMax function conforms to the probability distribution. Moreover, the maximum output probability is selected as the output result of the detection model during the output results, or the probability that the input detection sample is an Erebus attack. Under this circumstance, dichotomous detection, T_2R_2C_DNN, k, is set as 2. When the output value of $S(z^M)$ exceeds 0.5, it signifies that an Erebus attack occurs at the blockchain network layer.

5 Experimental Analyses

The proposed model was tested in a real blockchain network environment to simulate the actual transaction behavior and Erebus attacks. The proposed model is comprehensively evaluated based on the following: First, the T_2R_2C_DNN model was trained to verify the validity of the proposed model. Second, the ReliefF_WMRmR-based two-stage feature selection algorithm was tested by analyzing the rationality of the traffic behavior and routing status features associated with Erebus attacks.

Additionally, the optimal feature number in a set was identified. Finally, the performance of the existing detection methods was experimentally compared with that of the classical machine learning algorithm to clarify the advantages and disadvantages of the proposed Erebus attack detection model.

5.1 Experimental Configurations and Evaluation Indexes

Model training and contrast experiments are performed in the Windows 10 operating system with the following system requirements: a CPU of Intel Core i7-9750, a memory of 32.0 GB RAM, a GPU of NVIDIA GeForce GTX 2060, and 8 G video memory. A deep learning framework PyTorch 1.5 of Anaconda Python and the JetBrain PyCharm 2020.3 software operating environment were used to implement the proposed model.

Indexes of accuracy (Acc), such as precision, false alarm rate, F_1-score, and the Area Under Curve (AUC), are configured to evaluate its detection performance.

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{42}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{43}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{44}$$

$$\text{FAR} = \frac{\text{FP}}{\text{TN} + \text{FP}} \tag{45}$$

$$F_1\text{-score} = 2 \times \frac{\text{Recall} \times \textit{Precision}}{\text{Recall} + \textit{Precision}} \tag{46}$$

$$AUC = 1 - \frac{1}{m^+ m^-} \sum_{x^+ \in D^+} \sum_{x^- \in D^-} (F((f\left(x^+\right) < f\left(x^-\right)) + \frac{1}{2} F\left((f\left(x^+\right) = f\left(x^-\right)\right)\right) \tag{47}$$

where TP represents True Positive to correctly classify the attacked samples; FP represents False Positive, which is the erroneously classified attacked samples; TN represents True Negative, which is the normally classified normal samples; and FN represents False Positive, which is erroneously the normally classified samples. AUC is the area under the Receiver Operating Characteristic (ROC) curve. Moreover, the vertical and horizontal axes represent the True Positive Rates and False Positive Rates (TPR and FPR), respectively.

$$FPR = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{48}$$

$$TPR = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{49}$$

In (47), m^+, and m^- represent the number of obtained positive and negative examples, respectively, and D^+ and D^- are the positive and negative example sets, respectively. Additionally, $f(x^+)$ is the probability of detecting positive samples, and $F(x)$ is the performance index function. If x is true, the value assigned to $F(x)$ is 1. To test unevenly distributed datasets, higher ACC values, F_1-score, G-Means, and AUC shows that the detection performance of the proposed model is better than the conventional detection methods.

Particularly, an evaluation index of the Average Effective Classification Information (AECI) [28], expressed in (50), was introduced to test the performance of the proposed RelifF_WMRmR-based

two-stage feature selection algorithm in extracting strongly correlated features and filtering redundant features. Moreover, the evaluation index compares its performance with that of other typical feature selection algorithms.

$$AECI\left(\boldsymbol{f}_{slt}\right) = \frac{1}{|\boldsymbol{f}_{slt}|}\sum_{x_i \in \boldsymbol{f}_{slt}} I\left(C; x_i\right) - \frac{2}{|\boldsymbol{f}_{slt}|(|\boldsymbol{f}_{slt}|-1)}\sum_{x_i, xy \in \boldsymbol{f}_{slt}} I\left(C; x_i; x_j\right) \tag{50}$$

In (50), the preceding item is the average correlation of features with the label class; whereas the subsequent item represents the average redundancy of the features. AECI value is high when features have rich information.

5.2 Data Pre-Processing

Data pre-processing comprises three manipulations: data validation, symbolic attribute feature numeralization, and data normalization.

Data preprocessing comprises three main manipulations: data validation, symbolic attribute feature numeralization, and data normalization.

① Data validation

Some feature data collected in a real network environment are default. Default items in attribute values, and features with all values equal were removed and replaced with zero for the validity and integrity of model input vectors.

② Symbolic attribute feature numeralization

Through attribute mapping, symbolic features were transformed into binary numeric features, and eigenvalues in a hexadecimal system were converted into decimal eigenvalues. Additionally, non-numeric address-type attribute values are substituted by the occurrence frequency of such values in the entire dataset.

③ Data normalization

Numeric features were normalized to eliminate the influence of such differences in the proposed model because a significant value field difference exists between different features in a dataset. Moreover, relevant eigenvalues were mapped in a range of [0, 1] to ensure that the quantitative attributes are at the same magnitude. The equation is written as follows:

$$x_i' = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{51}$$

where x'_i represents a normalization result of the i^{th} eigenvalue, and $\min(x)$ and $\max(x)$ represent the minimum and maximum values of the attribute feature, respectively.

5.3 Experimental Environment and Data

The experimental traffic data were collected from the traffic behavior, and the routing status features were collected when operating the blockchain system. The corresponding topology environment comprises a node cluster of the normal blockchain network layer, Erebus-attacked nodes, and CAD nodes represented by ISP. These topologies were used to simulate and acquire traffic and routing features in a running environment during an Erebus attack on a normal blockchain network layer. The experimental traffic data was derived from the real UDP and TCP traffic samples, collected by Wireshark, and generated pcap files. Simultaneously, the dissector plug-in of the Ethereum Devp2p

protocol was added to Wireshark to parse the information in Ethereum packets. As noted by a study [29], the routing features were extracted by RouteViews [30]. During the experiment, 20 detection nodes were designed. Along with the Visualroute tool, the routing information of normal and attacked blockchain network layers were captured to perform feature analyses.

For the normal blockchain network layer nodes, the transaction traffic of Ethereum 2.0 and the operating traffic of Hyperledger Fabric1.4 were selected to simulate the traffic in multiple types at the blockchain network layer. The Ethereum 2.0 environment was formed by five virtual hosts. In this environment, the core-Geth was manipulated on Ubuntu 18.04 to run transactions among blockchain mining rigs (IP address segment: 192.168.127.0/24). Hyperledger Fabric1.4 comprises eight virtual hosts operating a blockchain program in a Docker container (IP segment: 192.168.103.0/24).

Three nodes were selected for the Erebus-attacked nodes, to practically simulate the multistage Erebus attack scenarios and generate the corresponding traffic and routing information. First, the attack tool, Yersinia, was used to deceive neighboring nodes by forging particular information and data in a protocol through vulnerability mining based on the network protocol, thus fulfilling routing penetration and routing topology damages. Second, Scapy-based attack scripts were written on Python 3.9 to periodically generate Erebus traffic attack commands of ping, pong, findnode, neighbors, and ADDR construction. The traffic nodes were attacked through multiplexing. During each nodal attack, the corresponding traffic pulse was set at 0.1–1 Mbps to add the routing information of the node into the attacked AS. Regarding the attacking script, the above command was continuously executed periodically to attack subnet nodes and routers, forcing them to construct erroneous routing and shield subnet nodal routing under attack. Third, Netwox was selected for the blockchain network to generate fake node messages, simulate interactions between the normal transaction traffic of the regular and attacked nodes at the blockchain network layer, and send routing status information of nodes to neighbor nodes and in AS regularly. The IP segment address was set as 192.168.27.0/24 in the Erebus attack network environment. Fig. 6 shows the experimental topology.

RouteViews [31] and Wireshark were adopted to collect the routing status and traffic behavior data at each CAD and blockchain network layer for link node acquisition. This experiment aims to generate a feature dataset that can be directly used for model detection. The routing status and traffic behavior data, collected on February 7–16, 2022 subject to experimental conditions, were used for the experiment to analyze variations in the system security of the blockchain network layer during this period. Moreover, 47,000 pieces of normal traffic and 12,000 sets of routing data acquired on 7–12, 2022 served as the normal dataset. Fig. 7 shows the average network traffic and total routing node path length in the normal blockchain network layer state on February 7. Fig. 8 shows the average network traffic and total path length of routing nodes in the state of the blockchain network layer in the attack state on February 14.

5.4 Model Training Results and Model Validity Verification

The training of the T_2R_2C_DNN model was performed in a mini-batch mode. After sample preprocessing, independent random sampling was conducted to generate multiple new datasets, which were further divided into the training and data sets. The training dataset forward underwent training in line with the model structure. Based on the model error, backward parameter training was conducted for the parameter optimization to obtain an optimal parameter set. Additionally, a testing dataset was adopted to verify the model performance. During the minibatch model training, Bootstrap iteration was conducted for samples after the training was repeated 50 times. During each iteration, 5000 data samples were extracted (ratio of normal and Erebus attack samples = 1:1). For unbiased experimental

results, the datasets experienced independently repeated experiments, and 10-fold cross-validation was performed among testing datasets. Moreover, the detection results of each dataset were averaged during the experiment.

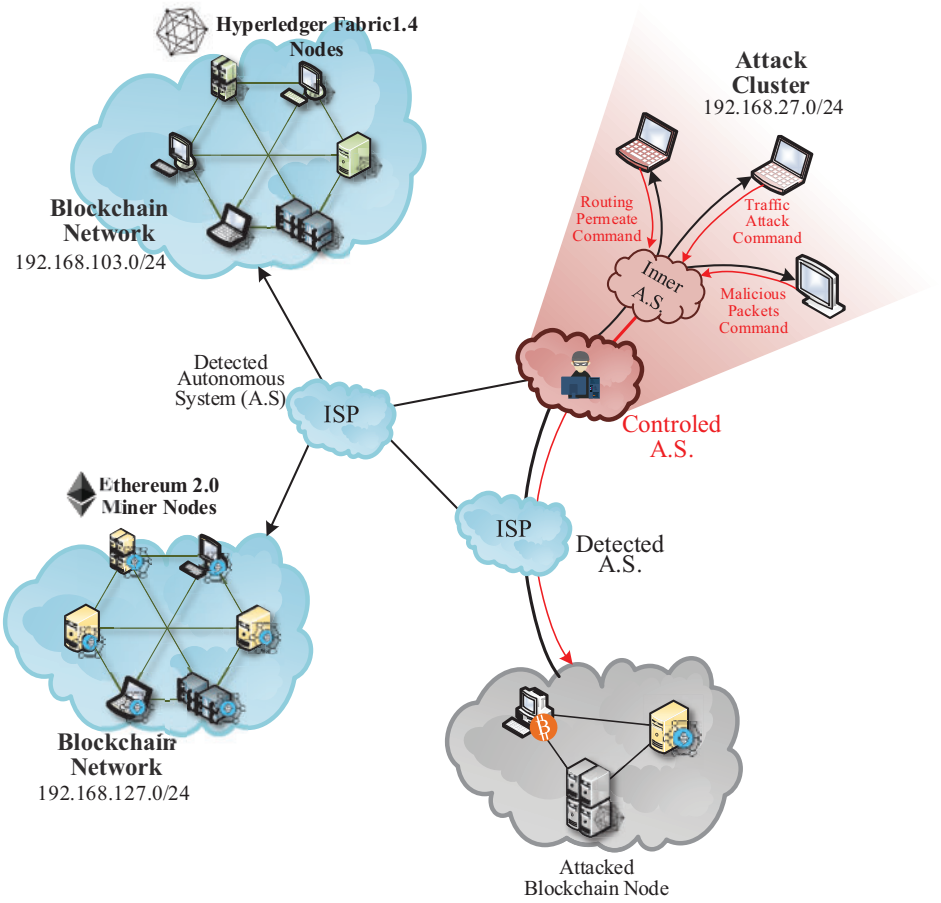

Figure 6: Experimental topology

Loss variations of classical deep learning models (*e.g.*, $T_2R_2C\_DNN$, 3-layer CNN (Convolution Neural Network), 5-layer CNN, and Long short-term memory (LSTM)) were configured under the same experimental conditions. The goal is to verify the validity of the proposed $T_2R_2C\_DNN$ model under the Erebus-attacked sample detection and verify its convergence rate in the same experimental conditions.

The training and testing datasets were subjected to 60 iterations, i.e., a total of 3000 rounds of training. The relevant parameters were adjusted according to the accuracy variations in different rounds (Fig. 9). At 20 rounds of the model training iteration, the average training accuracy exceeds 90% and 93.3% for the training and testing datasets, respectively. Thus, the training level of the proposed T2R2C_DNN model is satisfactory and suitable for experiments and tests in a real blockchain network layer environment.

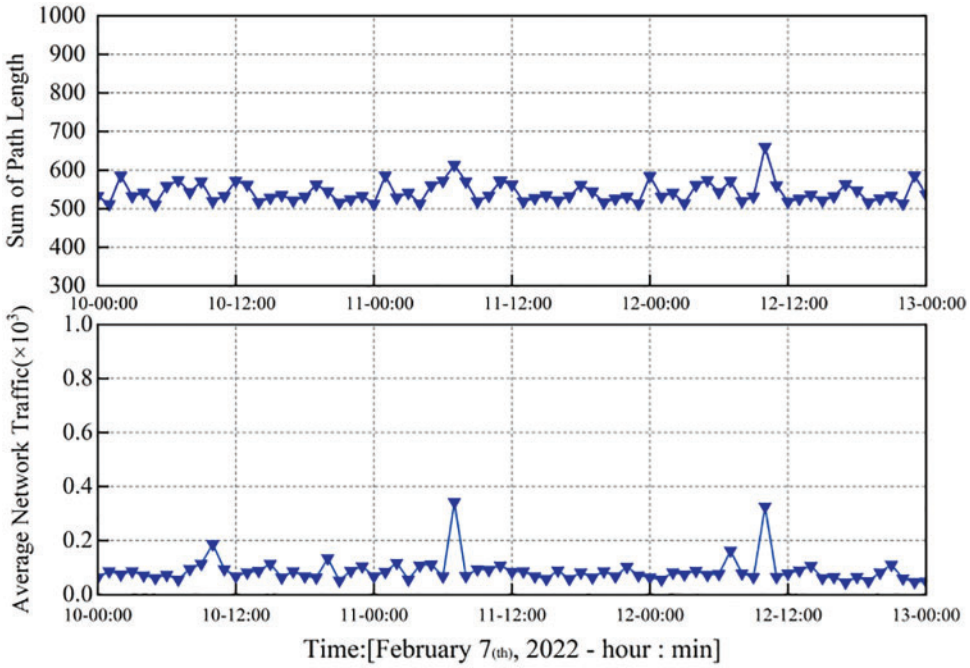

Figure 7: Average network traffic and the sum of path length at the blockchain network layer in a normal network state

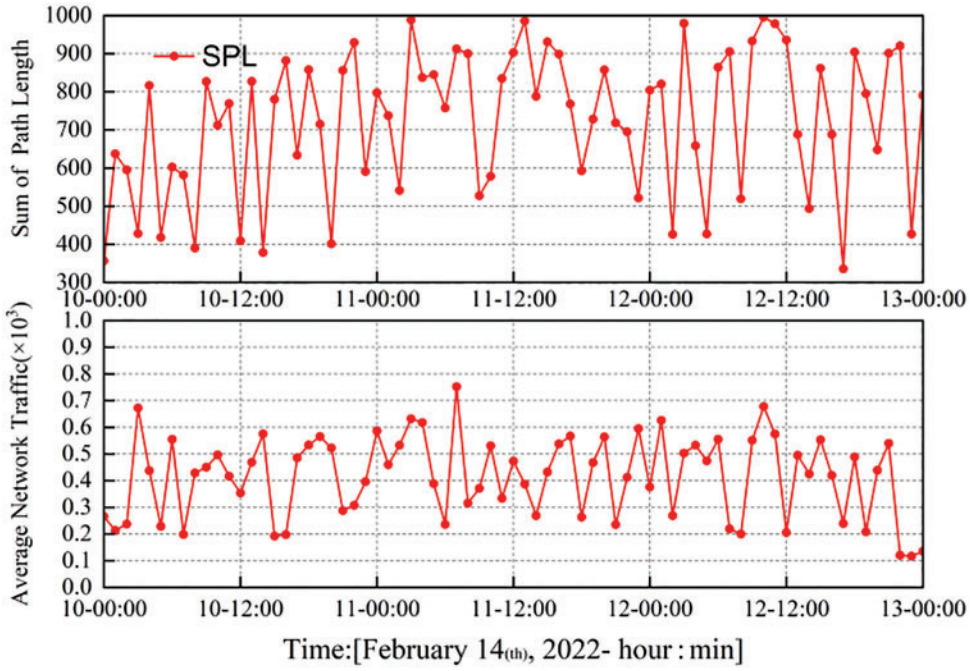

Figure 8: Average network traffic and the sum of path length at the blockchain network layer during an Erebus attack

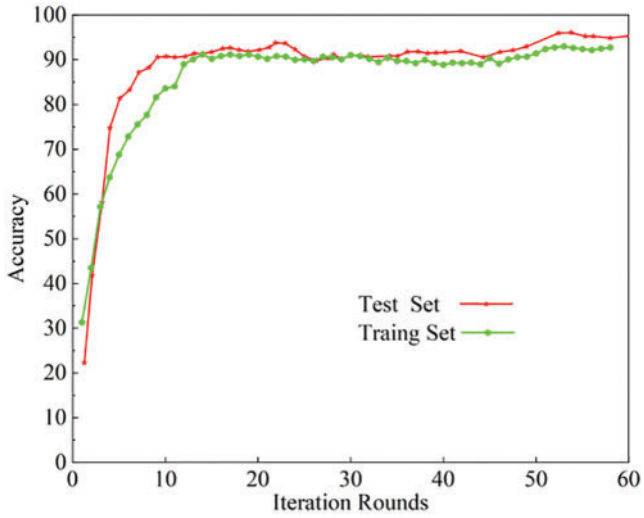

Figure 9: Accuracy variations of training and testing datasets

Fig. 10 shows the loss variations of the four models in different rounds of training. The experimental results show that when the number of training iterations is low, the T_2R_2C_DNN model shows some high loss value and a low convergence rate at the early phase for the following two reasons: (1) the model cannot sufficiently train sample features, and (2) the high complexity of hybrid features formed by traffic behavior and routing status features increase the training model complexity. Within the first 20 rounds of training, the T_2R_2C_DNN has a significant loss value, exceeding the other three algorithms. However, as the number of iterations increases, its loss value gradually decreases to < that of the other three algorithms. Additionally, the convergence rate of the T_2R_2C_DNN model is inclined to be stable, which signifies that the feature awareness of this model is rapidly enhanced, and model convergence is eventually realized. After 75 rounds of training, the loss value of the T_2R_2C_DNN model falls below 0.05, revealing that the proposed model has a strong capability of Erebus attack awareness.

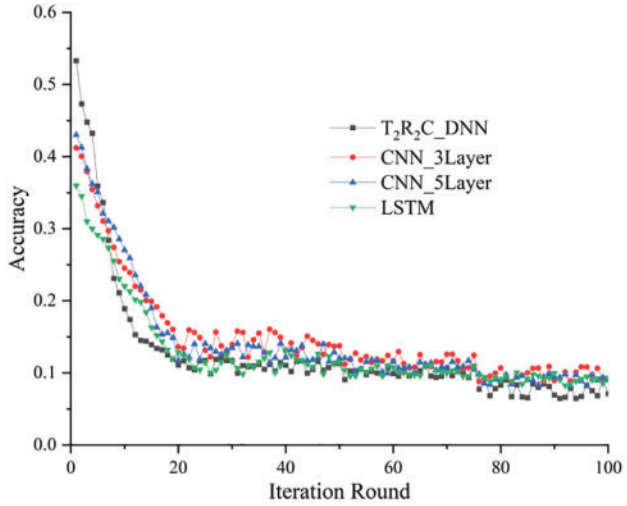

Figure 10: Loss values of the model in different iteration rounds

As shown in Fig. 11, the iteration rounds and detection accuracy of the training dataset vary in the same experimental conditions based on T_2R_2C_DNN, 3-layer CNN, 5-layer CNN, and LSTM. Regarding the T_2R_2C_DNN, the average detection accuracy of the corresponding training dataset reaches 95.25% in 10 rounds; whereas those of the other three models are >93.5%. Thus, T_2R_2C_DNN outperforms 3-layer CNN, 5-layer CNN, and LSTM in detection accuracy based on the experimental results. As the iteration round increases, the accuracy of the proposed T_2R_2C_DNN model exceeds that of the other models considerably, indicating that the T_2R_2C_DNN can effectively mine the hybrid feature formed by traffic behavior and routing status and boost the model's capability in detecting Erebus-attacked samples.

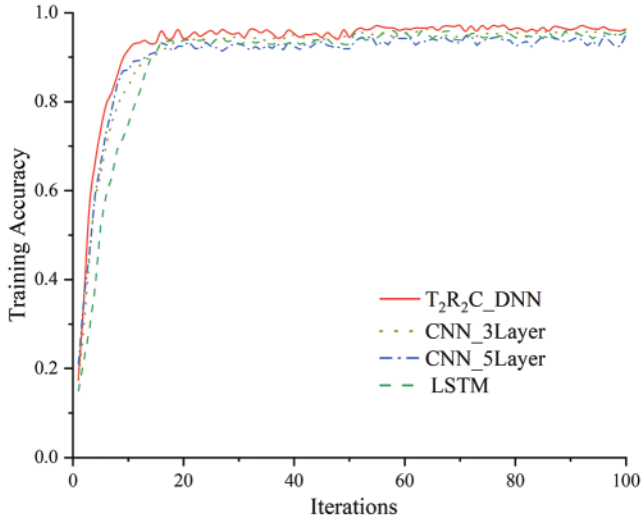

Figure 11: Training accuracy variations of deep learning models in a condition of different iteration rounds

5.5 Feature Selection Results

The AECI values of the feature items were calculated and visualized in a thermodynamic chart to verify whether the proposed Erebus attack features were reasonable. This way, the importance of feature attributes in Erebus attack detection is explored. Classical machine learning models with a strong feature awareness capability are selected for horizontal comparison, which includes support vector machine (SVM), C4.5 decision tree, 3-layer CNN, 5-layer CNN, and LSTM.

Fig. 12 shows the results, in which the *y*-axis refers to different detection methods; whereas the *x*-axis represents the AECI values of different Erebus feature items. Blue and red signify that the corresponding feature item is less important and highly important, respectively. Moreover, the AECI values of the SVM, 3-layer CNN, 5-layer CNN, and LSTM are similar to those of the proposed T2R2C_DNN model. Items 5, 6, 9, 22, 23, 27, 28, 29, 30, 31, 32, 33, 34, 35, 37, 38, 39, 40, 41, 43, 45, 47, 48, and 49 are of high heat degrees, manifesting the importance of these items. From these results, the proposed Erebus attack features can preferably describe the structural characteristics of Eclipse attacks, indicating the rationality of the feature items.

The detection accuracy of diverse feature selection algorithms was experimentally compared under the different feature data sizes. The goal is to validate the rationality of the proposed Relief_WMRmR-based two-stage feature selection algorithm in feature selection from Erebus attack samples. Typical

feature selection algorithms of RelifF, JMMC, IG, MRmR, and DEAFS with a strong feature selection capability were selected to perform the horizontal comparison. Features selected through the above six algorithms were inputted into the T_2R_2C_DNN model for training and for clarifying variations in model accuracy. Fig. 13 compares the results. Thus, the feature selection results of these algorithms are distributed in a centralized way. When the number of features ranges from 23 to 27, maximum accuracy beyond 85% is reached in all cases. This manifests that feature selection results of the abovementioned six algorithms remain consistent. Therefore, the number of features selected was 24 for the Relief_WMRmR-based method.

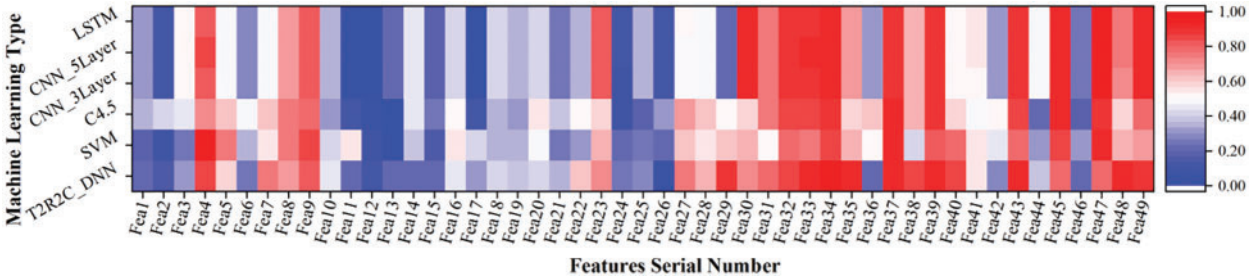

Figure 12: AECI values of the defined feature items

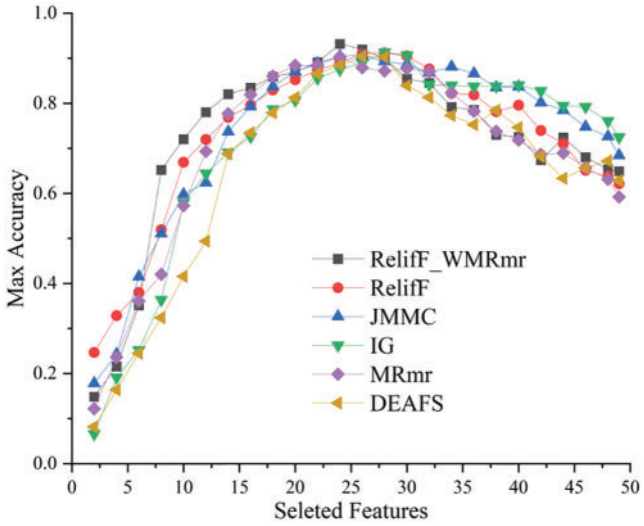

Figure 13: Feature selection results of different algorithms

Although the AECI values are selected to demonstrate the effective classification information of the selected features, the following defects still exist. The strongly correlated low redundant feature items of AECI values are the same as those of the weakly correlated highly redundant features. Considering this, the AECI fails to express correlation information of the selected features. For this reason, a thermodynamic correlation diagram was selected for the experiment to visualize the correlation information. Moreover, the Pearson correlation demonstrates the correlation of the features (Fig. 14). The correlations of features are symmetrically distributed diagonally. Red and blue represent high and low correlations, respectively. Thus, the experimental results show that the routing status features are strongly correlated to other primary features, signifying that the routing status defined here still shows a strong correlation of features after feature selection by the Relief_WMRmR

algorithm. The routing status during an Erebus attack remains rather sensitive to variations on relevant features. This indirectly manifests that routing status features are closely associated with other features during an Erebus attack, and the variability of routing status features depends on variations in other features.

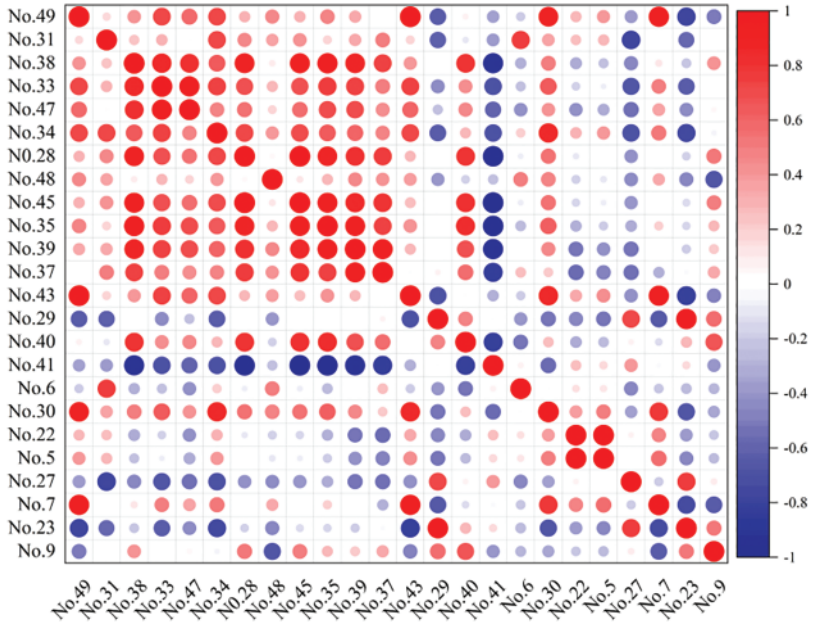

Figure 14: Correlation information values of the features selected according to Relief_WMRmR

5.6 Detection Performance Comparison Through Experiments

The same datasets were adopted, and relevant results were compared with those of classical machine learning algorithms and the existing detection approach. The goal was to prove the superiority of the proposed detection model in the Erebus attack to other existing detection approaches.

First, the experiment was designed to simulate the dichotomous detection of the Erebus attacks on a real blockchain network layer within different time windows based on different ratios of attack traffic. The network traffic was extracted from the simulated blockchain network layer in real-time, and the actual Erebus attack behavior was simulated. The density of the Erebus attack traffic, the routing request link rate, and the attack injection cycle were set at 1000 times/s, 200/s, and 10 s, respectively. The quantities of Ethereum and HpyerLedger users for simulation were 100 separately, and the visit view was 10 times/s. The average traffic rate at the blockchain network layer was 1 MB/s. The interval between acquisition time windows was selected to be 60, 120, 180, 240, or 300 min. Within diverse time windows, differences in recall and accuracy between the proposed method and a classical machine learning model were dynamically tested. Figs. 15 and 16 shows the relevant results.

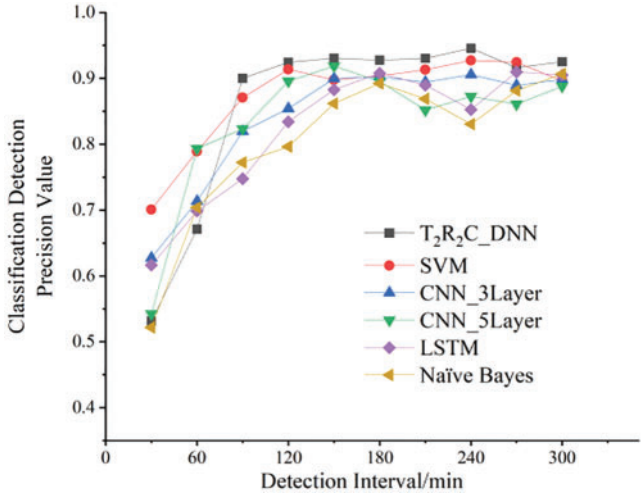

Figure 15: Erebus attack detection precision comparison of various machine learning models in a condition of different time windows

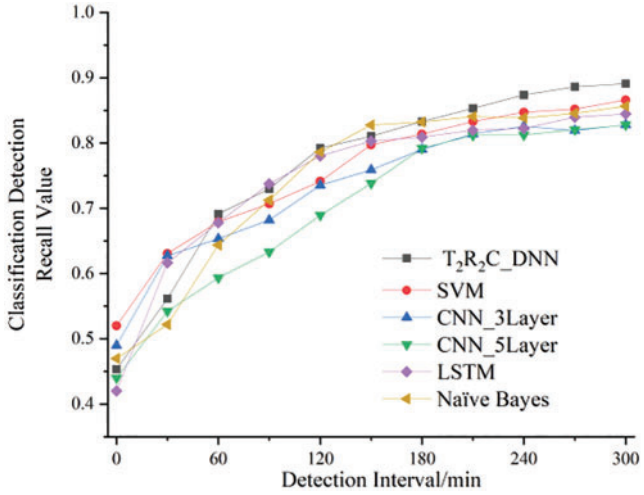

Figure 16: Erebus attack detection recall comparison of various machine learning models in a condition of different time windows

Second, the selected features were compared using 3-layer CNN, 5-layer CNN, and LSTM following the same experimental conditions based on their TPRs, FPRs, and AUCs. This way, the intertype detection performance characteristics of the deep learning models were validated. The AUC of the proposed method is proved to be 0.880, which exceeds that of the other three methods (Fig. 17). This indicates the superiority of the proposed model over the classical deep learning model in detecting Erebus attacks.

Third, the blockchain network layer was attacked by Erebus in a condition of different traffic rates of attacks to verify the detection precision and recall the variations of the proposed model subject to different Erebus attack scenarios at the blockchain network layer. Moreover, the experiment adopted a box plot to realize visualization (Figs. 18 and 19). In such two figures, the horizontal axes reflect different Erebus attack traffic rates within a range of 5%–50%, whereas the vertical axes correspond

to detection precision and recall. Additionally, each box in the figures represents the statistical results of six traffic volumes (i.e., 1, 2, 5, 10, 20, and 50 MB/s) in a condition of the same attack traffic rate. The analysis of the relevant results shows that the detection precision and recall of the model increase steadily as the attack traffic rate increases. Furthermore, the average values of the detected precision and recall exceed 70%. However, the detection precision range gradually decreases at a particular attack traffic rate when the detection ratio is raised. Here, the range of the detected precision results is gradually extended.

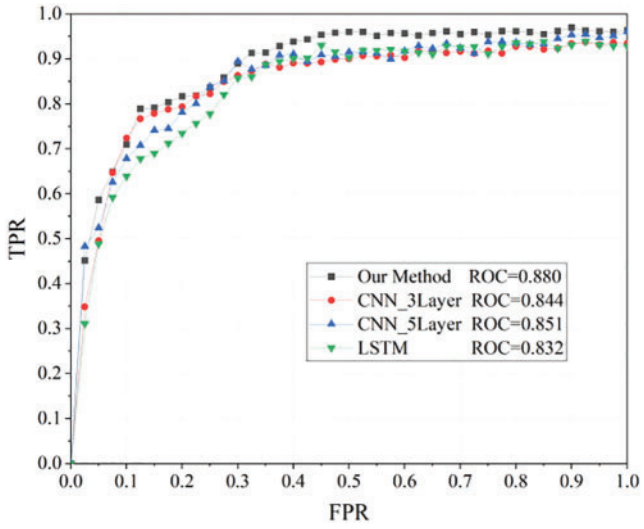

Figure 17: ROC comparison between the proposed method and the existing deep learning models

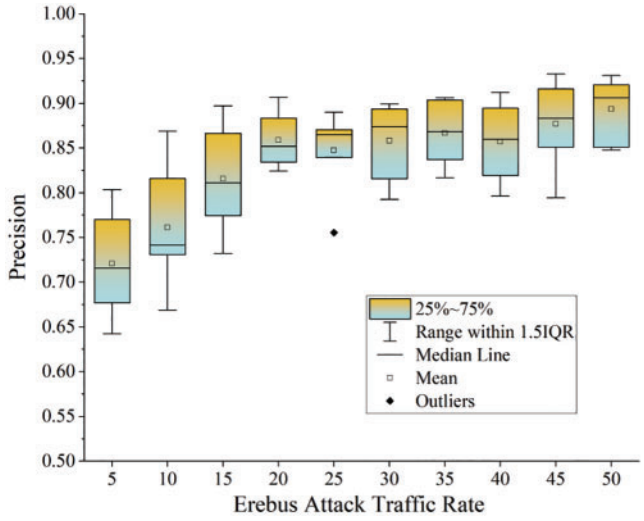

Figure 18: Statistical results of detection precision in a condition of different attack traffic rates

Identical datasets were used to validate the superiority of the proposed mode in the attack detection performance. The results were compared with those described by Fan et al. [9], Tran et al. [12], Maria et al. [19], Additionally, the accuracy, F_1-score, *Recall,* and AUCs reveal the advantages of the proposed model.

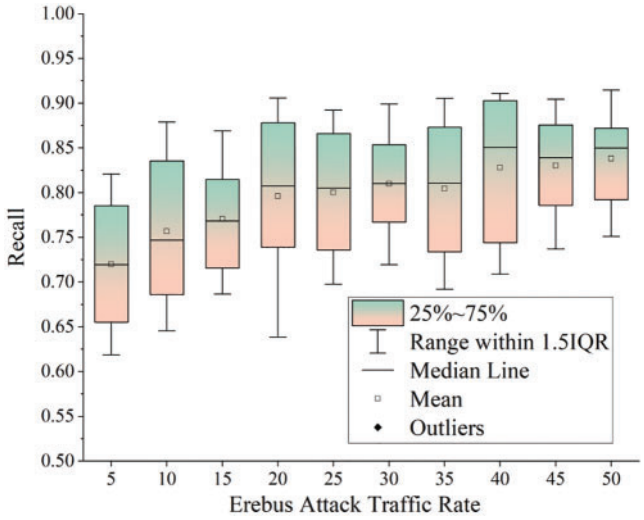

Figure 19: Statistical results of detection recall in a condition of different attack traffic rates

Initially, 20% of the daily sample data from February 7 to February 10 were extracted as the validation set for the experiment. Then, the comprehensive detection capabilities of the four detection methods for Erebus attacks were analyzed by comparing the AUC value performance with the separate inputs of Tran et al. [12], Maria et al. [19] and Hildrum et al. [8], and the proposed model. The AUC values of the detection methods proposed by Tran et al. [12] and Maria et al. [19] greatly varied in the detection period and were poorly stabilized, indicating that the two detection methods had insignificant advantages in the detection performance, based on the results shown in Fig. 20. The proposed Erebus attack detection method and the detection method proposed by Fan et al. [9] have detection AUC values higher than 93.15% with a stable detection performance, indicating that the two methods have robust comprehensive performance. Furthermore, the average AUC value of the proposed Erebus attack detection method is better than those of the other three detection methods, showing that the proposed model is more advantageous in detection performance.

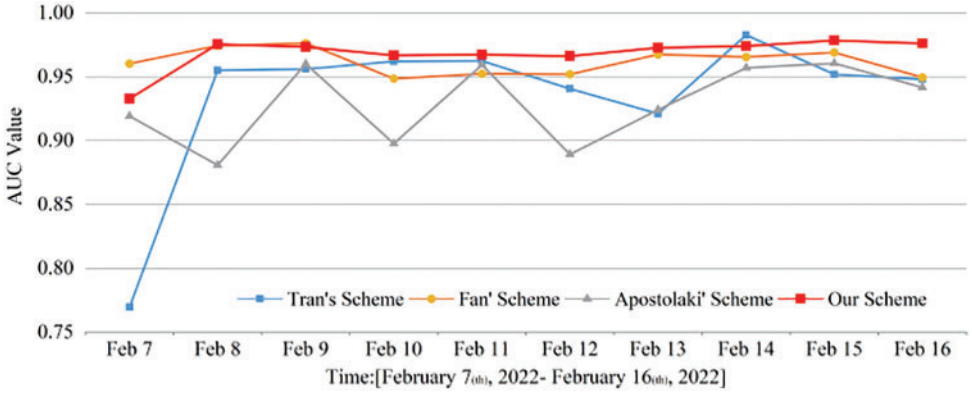

Figure 20: Changes in the AUC values of the four detection methods in the detection period

The stabilized models proposed by Tran et al. [12], Maria et al. [19], and Hildrum et al. [8], and the proposed detection model were sampled for testing of 15,000 sets randomly selected from the overall

dataset. The ROC curves and changes in the detection accuracy of the four detection methods for different sample numbers are shown in Figs. 21 and 22, respectively. The proposed detection model has achieved satisfactory detection results for the samples, with the area under the ROC curve being larger than those of the other three detection methods, as shown in Fig. 21. This finding indicates that the proposed detection model is more superior to other methods in comprehensive detection performance. For the detection sample with a small data volume (less than 2000), a small number of outlier samples have a certain impact on the accuracy performance of the four detection methods since the dataset is sparsely distributed, as shown in Fig. 22. Hence, the accuracy rates of the four methods are below 80%. As the number of samples increases, the influence of a few outlier samples on the accuracy rate weakens. Moreover, the accuracy rates of the four detection methods reach more than 85%. Thus, the proposed method is highly advantageous as its average accuracy reaches up to 94.43%. Furthermore, the proposed model has strong detection adaptability over large-scale samples by effectively separating the Erebus attack traffic.

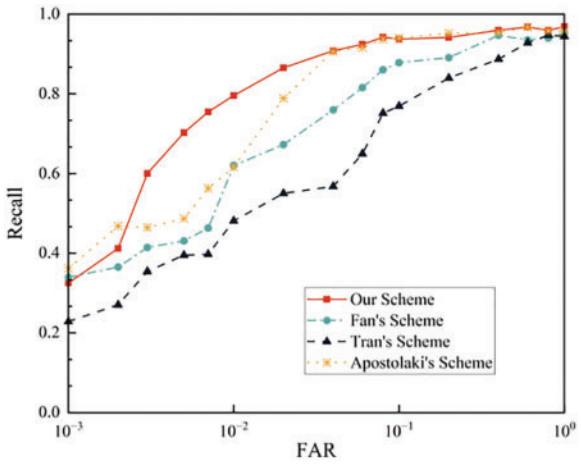

Figure 21: ROC curves of the four detection methods in random samples

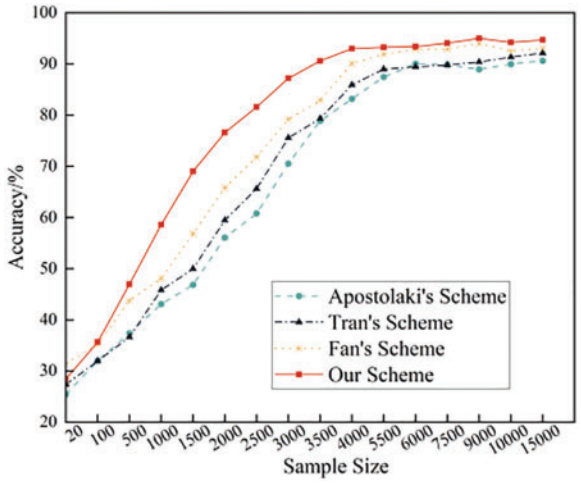

Figure 22: Changes in the accuracy of the four detection methods in random samples

The proposed model performed excellently according to these three indexes. As shown in Fig. 23, the average performance of the proposed model overmatched the other two detection approaches based on two primary reasons. First, the proposed T2R2C_DNN model can describe the influence of the traffic behavior and routing status features on Erebus attacks, perceiving the correlation information about such features and fulfilling classified output and expressions of the attack features. Second, the proposed features can describe the core features of an Erebus attack. By combining the ReliefF and WMRmR feature selection algorithms, the redundant feature information is effectively removed, which can enhance the model's feature mining and awareness capability.

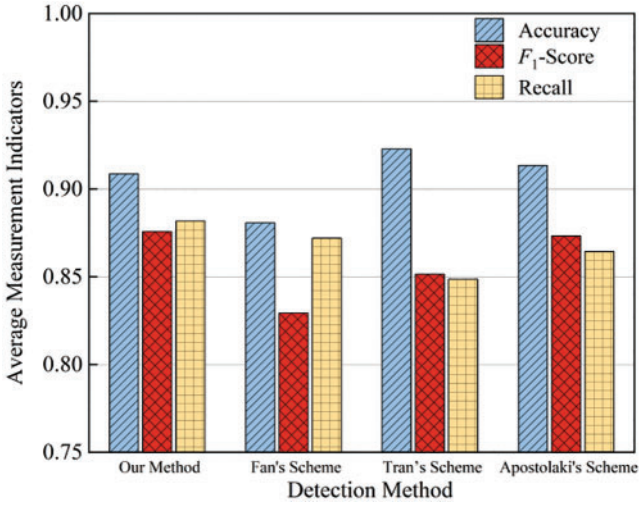

Figure 23: Average detection performance comparison between the proposed method and the existing detection approaches

6 Conclusions

A multimodal deep learning algorithm was proposed based on the routing status and traffic behavior specific to multistage Erebus attacks on the blockchain network layer. Core traffic behavior and routing status features are first defined for multistage Erebus attacks to describe their distinguishing characteristics accurately. Then, a RelieF_WMRmR-based two-stage feature selection algorithm is designed to extract the features that contain much information, eliminate redundant information, and improve the quality of the selected features. Finally, an MLP-based multimodal neural network was created to extract core features from heterologous features, thereby boosting the detection performance of the model. As demonstrated by the experimental results, the proposed model can effectively identify the Erebus attack behavior on the blockchain network layer and offer high detection performance. Built on the proposed detection model, Erebus attack-defense schemes will be further investigated to design Erebus attack-defense strategies and dynamically customize defense frameworks. It is expected to make reasonable decisions in further suppressing Erebus attacks and adapting dynamically to a blockchain network layer environment with a dynamically varying topology.

Acknowledgement: The authors appreciate the anonymous reviewers for their valuable comments.

Funding Statement: This work is funded by Open Fund Project of Information Assurance Technology Key Laboratory (No. KJ-15-109) and Zhengzhou Science and Technology Talents (131PLKRC644).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. Tran, I. Choi, G. J. Moon, A. V. Vu and M. S. Kang, "A stealthier partitioning attack against bitcoin peer-to-peer network," in *2020 IEEE Symp. on Security and Privacy (SP)*, San Francisco, CA, USA, pp. 894–909, 2020.

[2] A. Biryukov and I. Pustogarov, "Bitcoin over Tor isn't a good idea," in *2015 IEEE Symp. on Security and Privacy*, San Jose, CA, USA, pp. 122–134, 2015.

[3] L. Lai, T. Zhou, Z. Cai, Z. Liang and H. Bai, "A survey on security threats and solutions of bitcoin," *Journal of Cybersecurity*, vol. 3, no. 1, pp. 29, 2021.

[4] P. Swathi, C. Modi and D. Patel, "Preventing sybil attack in blockchain using distributed behavior monitoring of miners," in *2019 10th Int. Conf. on Computing, Communication and Networking Technologies (ICCCNT)*, Kanpur, India, pp. 1–6, 2019.

[5] S. Zhang and J. H. Lee, "Mitigations on sybil-based double-spend attacks in bitcoin," *IEEE Consumer Electronics Magazine*, vol. 10, no. 5, pp. 23–28, 2020.

[6] A. Pandey and H. Chaouchi, "Employing SABRE relay network for country-wide blockchain network," in *2020 Second Int. Conf. on Blockchain Computing and Applications (BCCA)*, Antalya, Turkey, pp. 2–8, 2020.

[7] E. Heilman, "net: Add test-before-evict discipline to addrman," https://github.com/bitcoin/bitcoin/pull/9037, 2019.

[8] K. Hildrum and J. Kubiatowicz, "Asymptotically efficient approaches to fault-tolerance in peer-to-peer networks," in *Int. Symp. on Distributed Computing*, Sorrento, Italy, pp. 321–336, 2003.

[9] W. Fan, S. -Y. Chang, X. Zhou and S. Xu, "Conman: A connection manipulation-based attack against bitcoin networking," in *2021 IEEE Conf. on Communications and Network Security (CNS)*, Tempe, AZ, USA, pp. 101–109, 2021.

[10] G. Naumenko, "p2p: Supplying and using asmap to improve IP bucketing in addrman," 2020.

[11] K. Otsuki, Y. Aoki, R. Banno and K. Shudo, "Effects of a simple relay network on the bitcoin network," in *Proc. of the Asian Internet Engineering Conf.*, Phuket, Thailand, pp. 41–46, 2019.

[12] M. Tran A. Shenoi and M. S. Kang, "On the {Routing-aware} peering against {Network-Eclipse} attacks in bitcoin," in *30th USENIX Security Symp. (USENIX Security 21)*, Vancouver, B.C., Canada, pp. 1253–1270, 2021.

[13] S. Baek, H. Nam, Y. Oh, M. Tran and M. S. Kang, "On the claims of weak block synchronization in bitcoin," *Cryptology ePrint Archive*, pp. 2105.08159, 2021.

[14] R. Nithyanand, O. Starov, A. Zair, P. Gill and M. Schapira, "Measuring and mitigating AS-level adversaries against Tor," arXiv preprint arXiv, pp.1505.05173, 2016.

[15] I. Pustogarov, "Deanonymisation techniques for Tor and bitcoin," Ph.D. dissertation, University of Luxembourg, Luxembourg, 2015.

[16] Y. Sun, A. Edmundson, N. Feamster, M. Chiang and P. Mittal, "Counter-RAPTOR: Safeguarding Tor against active routing attacks," in *2017 IEEE Symp. on Security and Privacy (SP)*, San Jose, CA, pp. 977–992, 2017.

[17] A. Maria, Z. Aviv and V. Laurent, "Hijacking bitcoin: Routing attacks on cryptocurrencies," in *2017 IEEE Symp. on Security and Privacy (SP)*, San Jose, CA, USA, pp. 375–392, 2017.

[18] S. S. Kushwaha, S. Joshi, D. Singh, M. Kaur and H. N. Lee, "Systematic review of security vulnerabilities in ethereum blockchain smart contract," *IEEE Access*, vol. 10, no. 7, pp. 6605–6621, 2022.

[19] A. Maria, M. Gian, M. Jan and V. Laurent, "SABRE: Protecting bitcoin against routing attacks," arXiv preprint arXiv, pp. 1808.06254, 2018.

[20] S. Jajodia, A. K. Ghosh, V. Swarup, C. Wang and X. S. Wang, *Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats*, vol. 54. Springer Science & Business Media: V.A., USA, 2011.

[21] M. A. Imtiaz, D. Starobinski, A. Trachtenberg and N. Younis, "Churn in the bitcoin network: Characterization and impact," in *2019 IEEE Int. Conf. on Blockchain and Cryptocurrency (ICBC)*, Seoul, Korea, pp. 431–439, 2019.
[22] I. Kononenko, "Estimating attributes: Analysis and extensions of RELIEF," in *European Conf. on Machine Learning*, Berlin, Heidelberg, pp. 171–182, 1994.
[23] X. Wen and Z. Xu, "Wind turbine fault diagnosis based on ReliefF-PCA and DNN," *Expert Systems with Applications*, vol. 8, no. 178, pp. 115016, 2021.
[24] G. Brown, A. Pocock, M. Zhao and M. Lujan, "Conditional likelihood maximisation: A unifying framework for information theoretic feature selection," *The Journal of Machine Learning Research*, vol. 13, pp. 27–66, 2012.
[25] Y. Zhang, A. Yang, C. Xiong, T. Wang and Z. Zhang, "Feature selection using data envelopment analysis," *Knowledge-based Systems*, vol. 64, pp. 70–80, 2014.
[26] W. Qian, S. Yu, J. Yang, Y. Wang and J. Zhang, "Multi-label feature selection based on information entropy fusion in multi-source decision system," *Evolutionary Intelligence*, vol. 13, no. 2, pp. 255–268, 2020.
[27] P. Ramachandran, B. Zoph and Q. V. Le, "Searching for activation functions," arXiv preprint arXiv: 1710.05941, 2017.
[28] P. E. Meyer and G. Bontempi, "On the use of variable complementarity for feature selection in cancer classification," in *Workshops on Applications of Evolutionary Computation*, Budapest, Hungary, pp. 91–102, 2006.
[29] C. Orsini, A. King and D. Giordano, "BGPStream: A software framework for live and historical BGP data analysis," in *Proc. of the 2016 Internet Measurement Conf.*, Santa Monica, California, USA, pp. 429–444, 2016.
[30] B. Huffaker, D. Plummer, D. Moore and K. Claffy, "Topology discovery by active probing," in *Proc. 2002 Symp. on Applications and the Internet (SAINT) Workshops IEEE*, Nara, Japan, pp. 90–96, 2002.
[31] Y. Guo, Z. Wang, S. Luo and Y. Wang, "A cascading failure model for interdomain routing system," *International Journal of Communication Systems*, vol. 25, pp. 1068–1076, 2012.