TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 29/38 pp1778–1801
DOI: 10.26599/TST.2025.9010119
Volume 31, Number 3, June 2026

Research Article

A Multi-View Hierarchical Intrusion Detection Model with Integrating Spatiotemporal Feature for Industrial Control System

Ding Zhou, Liu Bo*, Shaoxun Liu, Yan Xin, Jiuxin Cao, and Xinwen Fu

Abstract: Current intrusion detection in industrial control system (ICS) typically relies on network flows or traffic packets, often neglecting differences in payloads of functional fields and their heterogeneous responses under attacks. Moreover, most methods depend on manually crafted features, limiting the utilization of raw traffic byte streams and constraining detection performance. This paper proposes a multi-view correlation intrusion detection model that incorporates spatiotemporal features to enhance detection in ICS. By integrating byte streams with parsed field data, the model leverages traffic information through multi-view collaborative modeling. A fine-grained hierarchical feature framework is developed to extract behavior patterns from each field attribute, and cross-attention mechanisms capture inter-view relationships to construct a comprehensive representation of traffic content. A spatial feature extractor based on convolutional neural network (CNN) and a temporal extractor using Transformer architecture are employed to learn deep spatiotemporal features. A focal loss function is adopted to compute anomaly scores, which support the final intrusion detection decisions. Experiments on the water distribution testbed dataset show that the proposed model achieves superior performance compared to state-of-the-art methods, enabling accurate and efficient intrusion detection in ICS environments.

Key words: industrial control system; convolutional neural network; intrusion detection; spatiotemporal feature

1 Introduction

In the era of smart manufacturing, production environments are becoming increasingly complex and diverse, making comprehensive data collection across all stages and dimensions of manufacturing crucial. Industrial control systems (ICSs) are evolving from traditional closed architectures to cloud-edge collaborative architectures. This transition breaks the isolated “island” operating environments that industrial control system once relied on, significantly enhancing industrial intelligence and production efficiency[1, 2]. However, it also introduces internet security threats into the industrial control domain. The shift to open and interconnected industrial control system has not only expanded the attack surface but also led to the diversification and surge of security risk points, giving rise to a series of unprecedented cybersecurity challenges[3]. As the central nervous system of sectors such as power, energy, and manufacturing, industrial control systems are critical to the real economy. A network attack on these systems could threaten the safe

• Ding Zhou, Liu Bo, and Shaoxun Liu are with Endogenous Security Research Center, Purple Mountain Laboratories, Nanjing 211111, China. E-mail: zhoudinghawk@163.com; bliu@seu.edu.cn; liushaoxun@pmlabs.com.cn.
• Liu Bo, Yan Xin, and Jiuxin Cao are with School of Cyber Science and Engineering, Southeast University, Nanjing 210096, China. E-mail: yanxinwork12@163.com; jx.cao@seu.edu.cn.
• Xinwen Fu is with School of Computer & Information Sciences, University of Massachusetts Lowell, Lowell, MA 210096, USA. E-mail: xinwen_fu@uml.edu.
* To whom correspondence should be addressed.
Manuscript received: 2025-06-25; revised: 2025-07-14; accepted: 2025-07-15

operation of equipment, lead to asset losses, and even pose a risk to public safety[4, 5].

The strategic importance of industrial control systems has made them a primary target for malicious actors, including hackers, and a focal point in cyber warfare[6]. In recent years, attacks on industrial control systems have become increasingly frequent, exacerbating the cybersecurity challenges they face[7]. As a result, ensuring the security of industrial control systems has become one of the most critical and significant areas of current research. Due to historical issues, the initial construction of industrial internet systems did not incorporate comprehensive security measures and relied instead on the network's closed and independent nature[8]. As industrial informatization accelerates and moves toward open and universal industrial networks, these systems are increasingly exposed to escalating security threats[9]. In the current situation, developing effective protective measures to ensure the stability of industrial production is a significant activity in the field of industrial security. A network intrusion detection system (NIDS) is a proactive security technology that can monitor network traffic in real time without disrupting normal industrial operations[10, 11]. It effectively detects various internal and external attacks, providing timely response decisions for security management personnel. This makes NIDS a novel and efficient solution for industry security[12, 13].

In the current research domain, NIDS in the industrial control field is primarily categorized based on detection techniques into misuse-based intrusion detection system (MIDS)[14] and anomaly-based intrusion detection system (AIDS)[15]. MIDS, also known as signature-based or pattern-based IDS, identifies attacks by comparing collected traffic with known attack patterns stored in a rule database[16]. Early IDS systems predominantly used misuse detection techniques[17]. Signature-based detection methods provided high accuracy and low false-positive rates. However, the reliance on known attack patterns also meant that these systems struggled to identify unknown attacks and were less adaptable to the complexities of modern network environments[18]. Therefore, researchers have proposed anomaly-based intrusion detection methods. AIDS detects intrusions by establishing a profile of normal system behavior, which allows them to identify novel and variant attacks. This approach offers good generalization and adaptability, making it a mainstream direction in the current development of IDS systems[19, 20].

Currently, significant progress has been made in industrial intrusion detection; however, much of the research has primarily adapted information technology (IT) network intrusion detection methods[21]. These approaches often lack adequate analysis and understanding of the unique characteristics of industrial control system, industrial protocol traffic, and industrial attack vectors, resulting in models that are less suitable for industrial environments[22]. To address this issue, researchers have developed specialized models and algorithms tailored for industrial environments to enhance the accuracy and efficiency of intrusion detection[23]. However, these solutions often use traffic packets or network flows as the basic unit of analysis, without fully considering the variability in payloads of different functional fields within the packets and the heterogeneous response patterns these fields exhibiting when faced with diverse network attacks[24]. This limitation restricts the model's ability to deeply explore the internal patterns of industrial traffic, thereby constraining intrusion detection performance. Additionally, most of these studies rely on manually crafted feature datasets and high-level semantic field representations of traffic, neglecting the utilization of raw byte stream data. As a result, the use of traffic information remains insufficiently explored[25].

Motivated by the above issues, this paper focuses on fully leveraging detailed information within traffic packets. We integrate features and relational data from both raw traffic and parsed field perspectives to enhance intrusion detection performance in industrial control systems. The main contributions of this paper are summarized as follows:

(1) A fine-grained hierarchical feature extraction framework is designed based on a detailed investigation of payload semantics and functional field differences in industrial traffic, as well as their varying behavioral responses under network attacks. This framework enables accurate representation of protocol-specific traffic behavior and enhances the precision of intrusion detection.

(2) To improve the expressiveness of traffic features, a traffic content modeling method is proposed that integrates both raw byte stream and parsed protocol field perspectives. By combining spatial encoding through convolutional neural network (CNN) and

global temporal modeling via Transformer architecture, the model effectively captures spatiotemporal dependencies for precise anomaly detection.

(3) The proposed approach is validated using raw traffic from the water distribution testbed (WDT), with comprehensive comparative, ablation, and parameter analysis experiments. Experimental results demonstrate superior detection performance compared to state-of-the-art methods, confirming the model's effectiveness and applicability in real-world industrial control scenarios.

2 Related Work

In the research direction of misuse detection, Ref. [26] combined semantic analysis frameworks, intrusion detection systems, and flow analysis for estimating control command execution results to achieve expert system-based intrusion detection. Given that protocols often have semantic characteristics, semantic analysis can be employed to identify anomalous states within the system. The intrusion detection system was deployed in an industrial distributed network protocol (DNP3) substation system and tested under two attack scenarios: random target attacks and fixed target attacks. The system's performance was evaluated by comparing alarm delays at different scales. Reference [27] proposed an industrial protocol state detection algorithm based on discrete-time Markov chains. This algorithm automatically constructs a normal behavior model for industrial control systems based on traffic data. The model includes behavioral information such as state events, state transitions, transition probabilities, and transition time intervals, which are used as a set of state detection rules. Misuse-based IDS essentially employs a "blacklist" mechanism by maintaining a predefined signature database that corresponds to known attacks. This approach matches input traffic data with signatures that have similar characteristics. Consequently, the system's detection capability is highly dependent on the predefined rules in the database, which means it cannot detect unknown attacks and has limited applicability, with poor transferability to new or evolving attack scenarios.

With the rise of machine learning technologies, researchers have introduced these techniques into industrial intrusion detection, proposing anomaly-based intrusion detection methods[28]. In this direction, Ref. [29] addressed the need for high precision and real-time performance in industrial network intrusion detection by leveraging the high accuracy of supervised learning algorithms with labeled datasets. They proposed an intrusion detection method for industrial control system based on LightGBM, utilizing Bayesian optimization to fine-tune the parameters of the LightGBM model. Reference [30] investigated distributed denial-of-service (DDoS) attacks on industrial control systems and proposed a novel data flow-based hybrid framework for detecting such attacks with incremental learning. The framework employs various machine learning methods, including Bayesian, random forest, decision trees, and multi-layer perceptron, for traffic detection, thereby achieving the ability to differentiate between new types of attacks.

Traditional machine learning techniques are heavily influenced by feature engineering, often resulting in relatively shallow learning of traffic data. In contrast, deep learning technologies leverage deep network architectures to explore the intrinsic patterns and hierarchical representations of data more profoundly. This capability is particularly advantageous for handling high-dimensional traffic data, thereby demonstrating significant advantages in intrusion detection. In this research direction, considering the spatial characteristics of industrial traffic due to the static nature of industrial networks, Ref. [31] designed an intrusion detection model based on CNN and applied it to the DNP3 protocol. The model transforms traffic data into a two-dimensional matrix using specific sliding window rules, which is then used for training and evaluation with CNN. On the other hand, the "polling" mechanism inherent in industrial production imparts natural periodicity and temporal characteristics to industrial traffic. To address this feature, Ref. [32] proposed a general intrusion detection framework for supervisory control and data acquisition (SCADA) systems based on long short-term memory (LSTM) networks and feedforward neural network (FNN). In this framework, LSTM is responsible for detecting anomalies with sequential dependencies, while FNN handles anomalies without sequential dependencies. Ultimately, the system integrates the results from both types of anomaly analysis to provide a more comprehensive detection conclusion. While spatial modeling and temporal modeling focus on learning the spatial and temporal features of traffic respectively, they often overlook the

overall spatiotemporal information. To address this, researchers have proposed multi-feature hybrid detection models that combine the advantages of CNNs for spatial feature analysis and recurrent neural networks (RNNs) for temporal correlation extraction. Reference [33] proposed an intrusion detection model that combines correlation information entropy with convolutional neural network and bidirectional long short-term memory (BiLSTM) networks. This model uses correlation information entropy for feature selection, reducing the dimensionality of the features, and then applies deep learning algorithms for classification. This approach not only improves accuracy but also reduces noise and computational load. Reference [8] designed an intrusion detection model named AttackNet, based on adaptive convolutional neural networks and gated recurrent units (GRUs). AttackNet is used for detecting and classifying various botnet attacks in industrial Internet of Things (IoT) environments.

However, the aforementioned research faces challenges in meeting the high detection rate requirements of real industrial environments. Existing works typically use single traffic packets or network flows as the basic unit of analysis, without considering the variability of payloads in different functional fields within packets. They also overlook the heterogeneous response patterns of these fields when confronted with diverse network attacks. For example, malicious response injection attacks may deceive monitoring devices by tampering with payload data while affecting other parts minimally. Ignoring these differences and treating the data as a whole restricts the model's ability to deeply learn and understand the internal variations of different components of the traffic. This limitation makes it challenging to uncover targeted threat signals hidden within complex network behaviors. Current research in industrial intrusion detection often relies on publicly available IT network datasets or manually designed industrial traffic feature sets. These studies typically use high-level semantic field representations of traffic while neglecting the underlying raw byte stream information. As a result, the utilization of traffic data is not sufficiently comprehensive.

Based on the above analysis, this paper addresses traffic packets from both parsed traffic and raw traffic perspectives, integrating the aggregation information and relational transformations between these perspectives to provide a more comprehensive representation of the local spatial features of traffic packets. In the parsed traffic perspective, a fine-grained hierarchical feature representation framework is constructed to explore the variation patterns within each attribute field of the traffic data. At the network flow level, temporal modeling is applied to packet sequences to capture the global temporal dependencies between packets. Finally, by integrating both spatial and temporal features of the traffic packets, a complete representation of the traffic is achieved. A focal loss (FL) function is then introduced to design the intrusion detection model.

3 Intrusion Detection Model

3.1 Problem formulation

The set of industrial traffic is denoted as $T = \{t_1, t_2, \ldots, t_n\}$, where n represents the number of traffic packets in the set and t_i is the binary byte stream representation the i-th traffic packet. For each traffic packet, there is a label $\hat{y} \in \{0, 1, \ldots, m\}$ indicating whether it is an attack traffic or not, where m denotes the number of attack types and $\hat{y} = 0$ signifies that the packet is benign, $\hat{y} \in \{1, 2, \ldots, m\}$ indicates the specific type of attack. This paper outputs the attack type of each traffic packet in a multi-class classification format. In this paper, each traffic packet is classified into its attack type $\hat{y}$ using a multi-class classification approach.

$$f(t_i, N, P, S) \Rightarrow \hat{y} \tag{1}$$

where N denotes the set of network attributes, P denotes the set of payload attributes, and S denotes the set of statistical attributes, respectively.

For a given traffic packet t_i, the key functional field set is first parsed and constructed as $F_i = \{N_i, P_i, S_i\}$. Here, $N_i = \{n_{i,1}, n_{i,2}, \ldots, n_{i,z}\}$, where z denotes the number of network attributes, represents the network attribute fields of the i-th traffic packet, including network communication-related information such as master/slave addresses, IP addresses, and so on. $P_i = \{p_{i,1}, p_{i,2}, \ldots, p_{i,j}\}$, where j denotes the number of payload attributes, represents the payload attribute fields, including information such as payload data and execution operations, e.g., reading coil registers, writing discrete registers, etc. $S_i = \{s_{i,1}, s_{i,2}, \ldots, s_{i,k}\}$, where k denotes the number of statistical attributes, consists of the computed statistical attribute fields of the packet, such as the number of packets within a

given time interval. Next, a neural network model is constructed to extract deep feature information from the attribute fields N_i, P_i, and S_i and the raw byte stream t_i, resulting in the corresponding traffic representations N_i', P_i', S_i', and t_i'. Finally, a complete traffic packet representation is built by integrating all feature information $\{N_i', P_i', S_i', t_i'\}$, which is then used for detection.

3.2 Overall framework

The overall framework of the multi-perspective layered industrial traffic intrusion detection model that integrates spatiotemporal features, as proposed in this paper, is illustrated in Fig. 1, where DCSA stands for dimensional CNN and self-attention. The model mainly consists of four components: the traffic analysis module, the multi-perspective association based hierarchical feature extraction framework, the attention mechanism based two-level fusion strategy, and the network flow based temporal modeling and intrusion detection. The input is raw network traffic in pcap format, and the output is the specific classification of a single traffic flow.

The traffic analysis module extracts key functional fields and performs data preprocessing by analyzing and processing the raw traffic. It provides standardized input for the model while applying traffic compression rules to handle binary streams, thereby reducing computational complexity.

The multi-perspective association based hierarchical feature extraction framework uses one-dimensional (1D) CNN combined with self-attention mechanisms to capture spatial features of the traffic. It performs hierarchical learning to analyze the traffic behavior within the fields, thereby deepening the understanding of features comprehensively. The 1D CNN structure is simple, with few parameters and high computational efficiency, and has low hardware requirements, making it ideal for real-time and cost-effective scenarios. The use of rectified linear unit (ReLU) as the nonlinear activation function further enhances computational efficiency. Additionally, the pooling layer performs dimensionality reduction by highlighting primary features and downplaying secondary ones, further

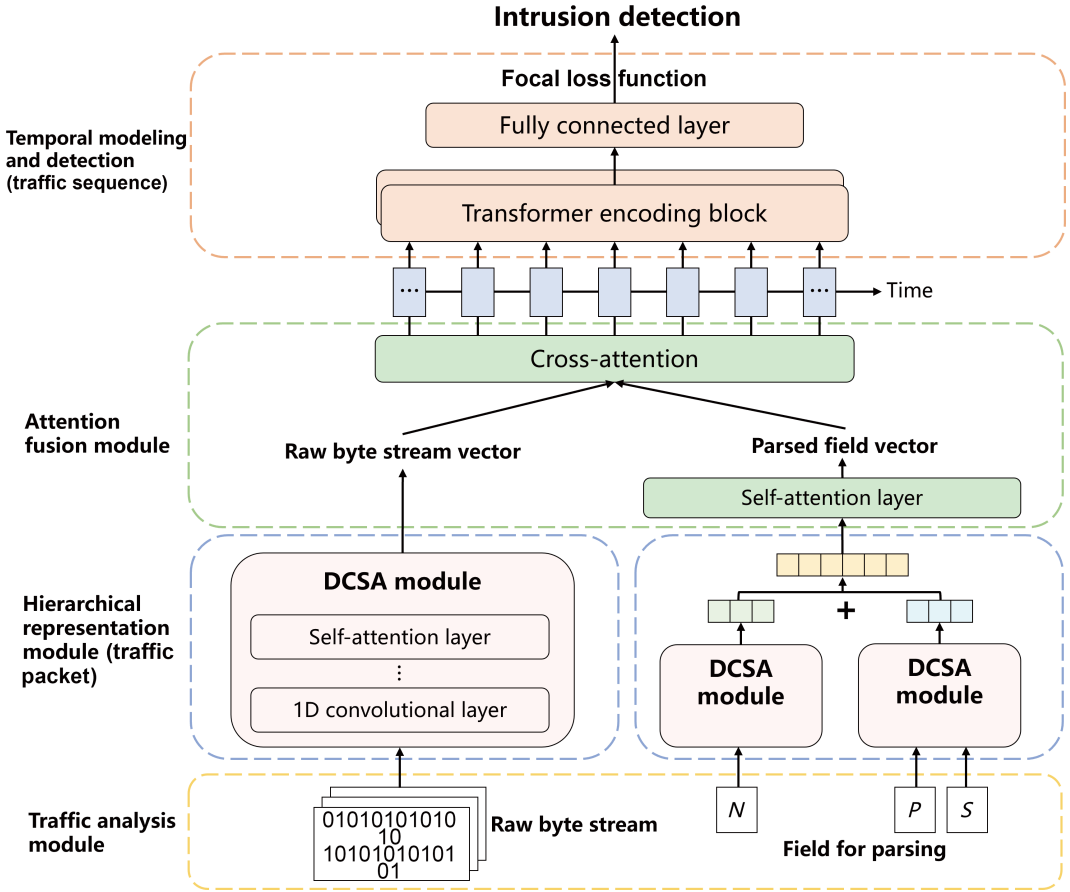

Fig. 1 Multi-perspective layered industrial traffic intrusion detection model.

reducing data dimensionality and computational complexity.

The attention mechanism based two-level fusion strategy focuses on key features through feature concatenation and self-attention mechanisms. It then integrates the perspectives of raw traffic and analyzed traffic using cross-attention mechanisms, forming a multi-perspective traffic packet representation. With the same output channel count and hidden layer dimension, direct concatenation along the output channel dimension is possible. For integrating the original traffic feature vector and the parsed traffic feature vector, simply using multi-head summation to obtain cross-attention scores is sufficient, resulting in very low computational complexity.

The network flow based temporal modeling and intrusion detection module reorganizes traffic sequences, extracts temporal dependencies using a Transformer encoder, and integrates spatiotemporal features for intrusion detection. It also applies a focal loss function to address class imbalance issues, enhancing the recognition of minority class attacks. The Transformer-encoder possesses strong generalization capabilities and scalability. It does not rely on sequential processing, which enables faster model training.

The model is specifically designed for industrial control system, which is characterized by static topology, strict real-time requirements, periodicity, and predictable traffic patterns—distinct from the dynamic and diverse traffic of general IT networks. Our model leverages tailored feature extraction and detection mechanisms suited to the structured nature of industrial traffic, emphasizing payload behaviors in protocol-specific functional fields and prioritizing features that detect deviations in periodic communications. Core parsed fields in industrial traffic are categorized into network attribute fields, load behavior features, and statistical calculation features, forming the basis of a hierarchical feature extraction framework. This framework enables detailed recognition of behavioral patterns and variations within each attribute field, facilitating precise identification of network attacks.

In contrast, the complex and irregular patterns of general IT networks may lead to false positives or negatives if our model is directly applied. Furthermore, industrial networks typically exhibit imbalanced data, with normal traffic far exceeding attack instances. We addressed this with a focal loss function to improve detection for minority attack traffic. This imbalance-oriented approach, however, may limit performance on general IT networks with more balanced traffic distributions.

In addition, we would like to emphasize that the multi-view design proposed in our model inherently enhances its robustness. Specifically, our model integrates two perspectives: the raw traffic view and the parsed traffic view. Each of these views provides different strengths in detecting certain types of attacks. Attacks that evade detection from the parsed traffic view can often be captured through the raw traffic view, as some malicious patterns might not be fully recognized when traffic is parsed and abstracted. Conversely, there are certain attacks that might evade detection in the raw traffic view but can be successfully detected from the parsed traffic perspective, as the structured data provide a clearer signal. By combining both views in a multi-view collaborative architecture, our model is able to address the weaknesses of each individual perspective. This collaboration significantly improves the robustness of the intrusion detection model against various attack strategies, including evasion attempts that target specific detection mechanisms. This approach is novel and provides an important step toward resilient intrusion detection in ICS environments.

The overview of the entire workflow, from input preprocessing to detection output, is presented in Fig. 2.

As illustrated in Fig. 2, the input raw traffic first passes through the traffic analysis module, where it is parsed into three types of attribute fields: network attribute fields, payload attribute fields, and statistical attribute fields. Each of these fields, along with the raw byte stream, is then processed by a dedicated feature extraction module, which produces high-dimensional feature representations. These features are subsequently fused using a two-level attention mechanism to construct a complete packet representation. To reduce dimensionality and prevent overfitting, an autoencoder-based compression layer is applied. The reduced representations are then grouped into network flows based on temporal order, and a Transformer-based temporal modeling module is employed to capture long-term dependencies between packets. Finally, the integrated spatiotemporal feature representations are

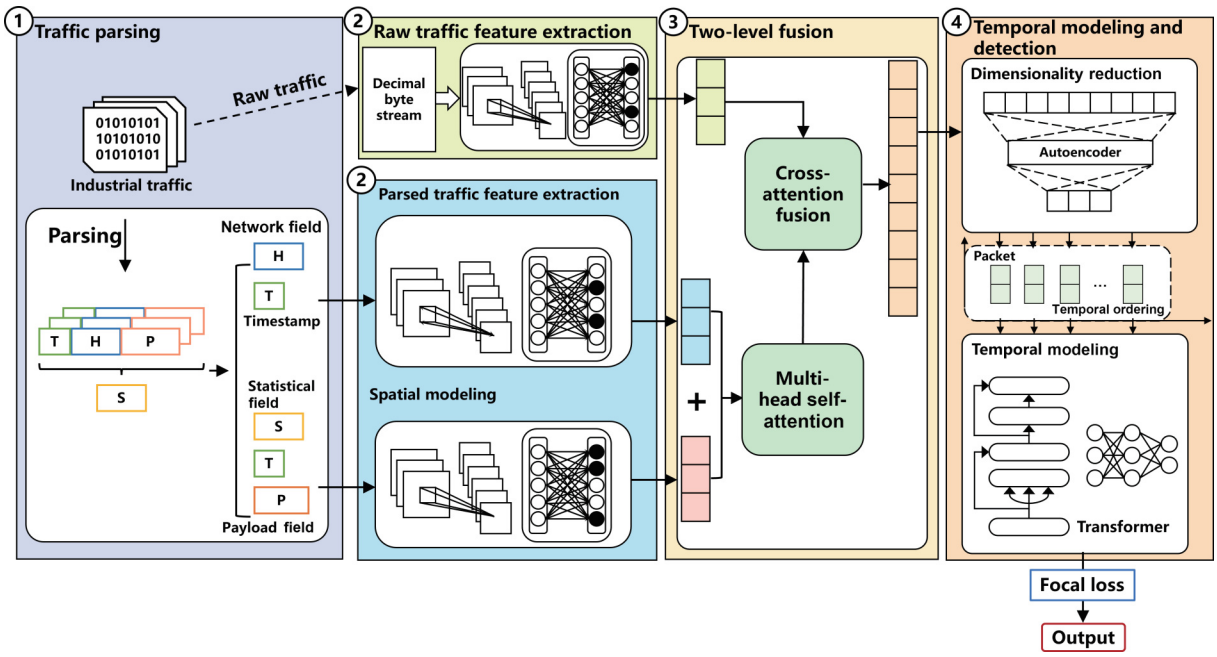

Fig. 2 Overview of the entire workflow.

passed through the detection module, where anomaly scores are computed, and each traffic packet is classified as normal or attack.

3.3 Traffic analysis module

This section explores methods for processing traffic feature information from two perspectives: raw byte streams and parsed fields, in order to transform traffic data into input vectors for the detection model.

(1) Parsed traffic processing. The payload in industrial communication protocols consists of different parts that exhibit significant variations based on their business functions. These differences are further amplified under different types of network attacks, leading to distinct changes. For instance, a malicious response injection attack alters device states by tampering with the data payload, while minimally affecting other components[34, 35]. Reconnaissance attacks often cause abrupt changes in fixed IP and media access control (MAC) address sets. Distributed denial-of-service attacks typically result in significant changes in statistical properties, such as transaction response intervals. The functional payloads of various traffic components help differentiate between attacks, but existing research typically treats them as a whole, limiting the model's ability to learn specific attack features.

To build industrial traffic features, we referenced real industrial control network environments as well as relevant literature[36–38], focusing on features common to industrial protocols based on the transmission control protocol (TCP). Industrial traffic features are extracted by considering several key aspects: network throughput, IP address-port matching, packet length, time intervals, communication path consistency, communication protocols, IP-MAC mapping, number of connections, TCP flags, function codes, and payload values. The constructed features are categorized into three types: network attribute fields, payload attribute fields, and statistical attribute fields, as detailed in Tables 1–3.

Network attribute field: These include metadata related to industrial network communication, such as IP addresses and ports, primarily found in the transport and network layers of the TCP/IP stack. Industrial

Table 1 Network attribute field.

Number	Feature	Description
1	Time	Timestamp
2	Protocol	Protocol type
3	Ether_src	Ethernet source address (MAC)
4	Ether_dst	Ethernet destination address (MAC)
5	IP_len	Packet length
6	IP_src	Source IP address
7	IP_dst	Destination IP address
8	TCP_sport	Source port number
9	TCP_dport	Destination port number
10	TCP_flags	Packet flag

Table 2 Payload attribute field.

Number	Feature	Description
1	Protocol_transId	Transaction ID
2	Protocol_protoId	Protocol identifier
3	Protocol_len	Message length
4	Protocol_unitId	Unit identifier
5	Protocol_funcCode	Function code
6	Protocol_error	Exception code
7	Protocol_payload	Request or response identifier
8	Protocol_addr	Starting address of the data block
9	Protocol_quantity	Quantity of data block
10	Protocol_byteCount	Read data block size
11	Protocol_value	Value of the read data block

Table 3 Statistical attribute field.

Number	Feature	Description
1	Second_Time	Timestamp (s)
2	DataCountPerSec	Packets per second
3	n_pkt_src	Number of packet from the same source address in 2 s
4	n_pkt_dst	Number of packet arriving at the same destination address in 2 s

networks are highly static, with pre-planned network devices and connections, resulting in stable topology and configurations. By analyzing IP-port pairs, protocol types, and other network field information, it is possible to effectively identify unauthorized connections or unexpected protocol packets, which aids in detecting scanning attacks.

Payload attribute field: These carry specific operational details relevant to industrial processes, such as read/write operations and data block values, which are mainly located in the application layer of the TCP/IP architecture. Industrial operations follow a polling pattern with a complex and fixed behavioral logic. Thus, analyzing behavior patterns within payload fields can effectively reveal man-in-the-middle (MITM) attacks and other malicious interventions.

Statistical attribute field: These are computed features derived from packet properties, such as the number of packets per second, reflecting real-time changes in network status. Statistical attributes help detect high-traffic events that impact network stability, such as DoS attacks.

For industrial control protocols based on the TCP transport protocol, the network attribute fields and statistical attribute fields relate to communication behaviors and statistical data on communication processes, making them applicable to various industrial control protocols. For payload attribute fields, we selected features that are common across most industrial protocols. Different industrial control protocols can parse out the necessary core functional fields according to the format of the respective protocol. These fields typically include payload data, IP addresses, ports, read/write operations, data block values, function codes, and error codes.

(2) Raw traffic processing. The binary form of raw industrial traffic exhibits issues such as excessive data dimensions and sparse information, which can impose significant computational burdens if processed directly. To reduce the computational load for subsequent models, this paper designs a binary stream mapping rule. It uses bytes as the basic unit to convert the raw traffic, mapping continuous 8 binary bits to corresponding decimal values to obtain a dense and effective traffic representation. The calculation process is shown in Eq. (2), where b_i represents the i-th binary bit, starting from the highest bit.

$$\text{Result} = \sum_{i=n}^{n+7} b_i \times 2^{n+7-i} \tag{2}$$

In addition, the byte stream of network packets has variable lengths, making it difficult to use directly as input for neural network models. Therefore, this paper analyzes the distribution of packet lengths in the dataset, selects a scheme that covers most of the byte stream lengths, and adjusts the byte stream sequence structure using truncation or zero-padding to fully utilize the data and fit the model's input interface. The proposed traffic parsing algorithm is shown in Algorithm 1.

3.4 Hierarchical spatial feature extraction model based on multi-view correlation

This section focuses on mining and learning deep-level feature information about the intrinsic nature and behavioral patterns of traffic packets. First, to effectively capture the spatial characteristics of industrial traffic, 1D CNN integrated with a self-attention mechanism is designed for spatial feature extraction. Then, traffic spatial information is mined separately from the perspectives of the raw byte stream and the parsed fields.

3.4.1 Spatial feature extraction model

To ensure the stability of industrial production, the devices and topology of industrial control system are meticulously planned and configured. The

Algorithm 1 Traffic analysis and dataset construction

Input: Raw traffic pcap file packets = {packet$_1$, packet$_2$, …, packet$_i$,…, packet$_N$} and CSV file storing traffic labels labels = {label$_1$, label$_2$, …, label$_N$}, where packet$_i$ = {b_1, b_2, …, b_j, …, b_m}, b_j denotes binary bytes of the raw traffic.

Output: Constructed traffic fields and raw byte stream set T.

for $i = 0, 1, 2, N$ **do**
  Create d_i and add (label$_i$, d_i)
  $o_i = \sum_{i=n}^{n+7} b_i \times 2^{n+7-i}$
  Add(reshape(o_i), d_i)
  if Ether_flags in t_i **then**
    Extract Ethernet layer information and add it to the field set d_i
  end if
  if TCP_flags in t_i **then**
    Extract transport layer-related information and add it to the field set d_i
  end if
  if Protocol_flags in t_i **then**
    if operation_count > 1 **then**
      Extract application layer operation information values
      Split operation information according to the number of operations values = split(operation_count)
      for j = 0, 1, 2, …, operation_count **do**
        d_i' = copy(d_i)
        Add values$_j$ to the field set d_i'
        Add the extracted information d_i' to the dataset T
      end for
    end if
  end if
end for
Return generated traffic dataset T

communication patterns between devices are fixed, and the traffic exhibits a certain regularity and predictability in the physical space. Based on the aforementioned characteristics, a traffic packet spatial feature extraction model is constructed, as shown in Fig. 3. The core structure is composed of 1D CNN and self-attention mechanism (DCSA). The model leverages the regularity of traffic in physical space, utilizing the CNN to extract local spatial structural information from the traffic, while the self-attention mechanism is introduced to capture the global dependencies within the traffic and enhance the relational information in the features. In the basic CNN unit, a max-pooling layer and a batch normalization

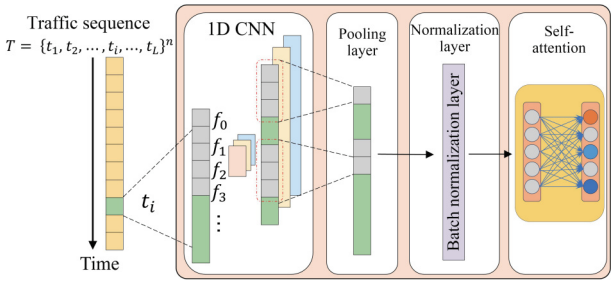

Fig. 3 Spatial feature extraction model.

layer are integrated to perform dimensionality reduction and standardization of features, enhancing the model's robustness and generalization capability. The design philosophy of the entire module aims to fully leverage the local perceptual ability of CNNs in the spatial domain and the global relational capability of the self-attention mechanism.

The industrial traffic is collected and recorded in a single stream, making it one-dimensional sequential data. Therefore, this section selects a 1D CNN, which is adept at handling one-dimensional sequences, as the basic architecture for the traffic spatial feature extraction model. The one-dimensional representation vector of a traffic packet is defined as $t = \{f_0, f_1, f_2, \ldots, f_{n_f}\}$, where n_f denotes the number of features in the traffic report message. The convolution window size is m, with a stride of 1 and a padding length of 0. The number of output channels of the CNN is T_f. When the convolution kernel performs a dot product operation along the traffic feature direction, the one-dimensional convolution process for a feature f_i is described as

$$x_{i,j} = \mathrm{ReLU}\left(\sum_{k=0}^{m-1} w_{i,k} \cdot f_{j+k} + b_i'\right) \tag{3}$$

where $x_{i,j}$ represents the j-th element of the output tensor in the i-th channel, $w_{i,k}$ represents the k-th weight of the i-th convolution kernel, f_{j+k} represents the $(j+k)$-th element of the input sequence, and b_i' is the bias term for the i-th channel. Consequently, the traffic packet matrix obtained after 1D CNN convolution is represented as $X = \{x_0, x_1, \ldots, x_i, \ldots, x_T\}$, where $X \in \mathbb{R}^{T \times d}$. Here, x_i is a one-dimensional vector of length d in the i-th output channel, with $x_i \in \mathbb{R}^{1 \times d}$ and $d = n - m + 1$.

After convolution, the packet vector contains rich high-dimensional spatial information, leading to a significant computational burden and the risk of overfitting. Therefore, this paper introduces a max-

pooling layer after convolution. By selecting the maximum value within the pooling window as the representative feature, the method reduces data dimensionality and computational complexity, while also mitigating the risk of overfitting in the model. To stabilize the model training process and reduce internal covariate shift between layers, a batch normalization layer is designed to standardize the feature vectors. The specific process includes calculating the mean μ_B and variance ${\sigma_B}^2$ of the batch, followed by standardization. This is then linearly transformed using learnable scaling factor γ and shift β, as shown in Eqs. (4) and (5). Here, ε is a small constant used for numerical stability.

$$\hat{x}_{i,j} = \frac{x_{i,j} - \mu_B}{\sqrt{{\sigma_B}^2 + \varepsilon}} \tag{4}$$

$$z_{i,j} = \gamma \hat{x}_{i,j} + \beta \tag{5}$$

After the above operations, the traffic packet is transformed into a feature matrix $Z = \{z_1, z_2, \ldots, z_i, \ldots, z_T\}$ that contains deep spatial information, where $z_i \in \mathbb{R}^{1\times d}$ is the feature vector of length d in the i-th channel, and $Z \in \mathbb{R}^{T\times d}$ represents the feature matrix. To uncover the hidden states and dependencies between elements at different positions in the traffic packet, the model perceives and understands the packet's structure and semantic information from a global perspective. Additionally, it performs soft feature selection within the packet to focus on more discriminative and fine-grained features. After convolution, the multi-head scaled dot-product attention mechanism (MHSA) is used to compute attention weights for the input sequence in parallel. Each head captures feature information from different aspects of the input sequence within specific subspaces. For each feature matrix Z, the MHSA calculates the query matrix Q, key matrix K, and value matrix V through three fully connected layers. Then, Q, K, and V are split along the feature dimension according to the number of heads. In the i-th head, attention weight matrix $Q_i K_i^{\mathrm{T}}$ is computed based on the split query matrix Q_i and key matrix K_i, followed by scaling with the factor $\sqrt{d_k}$ for dimensional scaling. The results are then normalized using softmax, producing the final output of head_i, as shown in Eq. (6). Finally, the outputs of all heads are concatenated and linearly transformed into a unified output, as shown in Eq. (7). Here, head_i represents the output of the i-th attention head, concat denotes the concatenation operation of the multi-head attention scores, and W^0 is the weight matrix for the linear transformation.

$$\mathrm{head}_i = \mathrm{softmax}\left(\frac{Q_i K_i^{\mathrm{T}}}{\sqrt{d_k}}\right) V_i \tag{6}$$

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{concat}(\mathrm{head}_0, \mathrm{head}_1, \ldots, \mathrm{head}_m) W^0 \tag{7}$$

The spatial feature extraction algorithm for traffic packets is shown in Algorithm 2, where out_channels denotes the number of output channels in the 1D CNN, which is equal to the number of convolution kernels, with each kernel capturing different features.

3.4.2 Hierarchical feature extraction framework

To enhance the richness of traffic representation and improve intrusion detection performance, this paper introduces raw traffic information in addition to parsed traffic features. By collaborating from two perspectives, the framework comprehensively captures both the low-level details and high-level semantics of traffic communication. Moreover, to investigate the differences in payloads across different functional fields and their change patterns after an attack, the paper performs spatial modeling and feature extraction based on field divisions during the parsing process. This fine-grained learning of specific behavior patterns within each attribute field aims to more effectively distinguish between various types of network attacks. The hierarchical feature representation framework designed in this paper is illustrated in Fig. 4. Since the number of statistical attribute features is relatively small and contains limited information, and because

Algorithm 2 Spatial feature extraction for traffic packet

Input: Traffic packet $t_i = \{f_0, f_1, f_2, \ldots, f_{n_f}\}$.
Output: Feature-extracted representation of the traffic packets, denoted as T'.

1: **for** $i = 0, 1, 2, n$ **do**
2: $x_{i,j} = \mathrm{ReLU}\left(\sum_{k=0}^{m-1} w_{i,k} \cdot f_{j+k} + b_i\right)$
3: $y_{i,j} = \mathrm{MaxPooling}\left(x_{i,j}\right)$
4: $\hat{x}_{i,j} = \frac{y_{i,j} - \mu_B}{\sqrt{{\sigma_B}^2 + \varepsilon}}$
5: $z_{i,j} = \gamma \hat{x}_{i,j} + \beta$
6: $a_{i,j} = \mathrm{softmax}\left(z_{i,j}\right)$
7: $t'_{i,j} = a_{i,j} \times z_{i,j}$
8: **end for**
9: **Return** $T' = \{t'_1, t'_2, \ldots, t'_{\text{out\_channels}}\}$

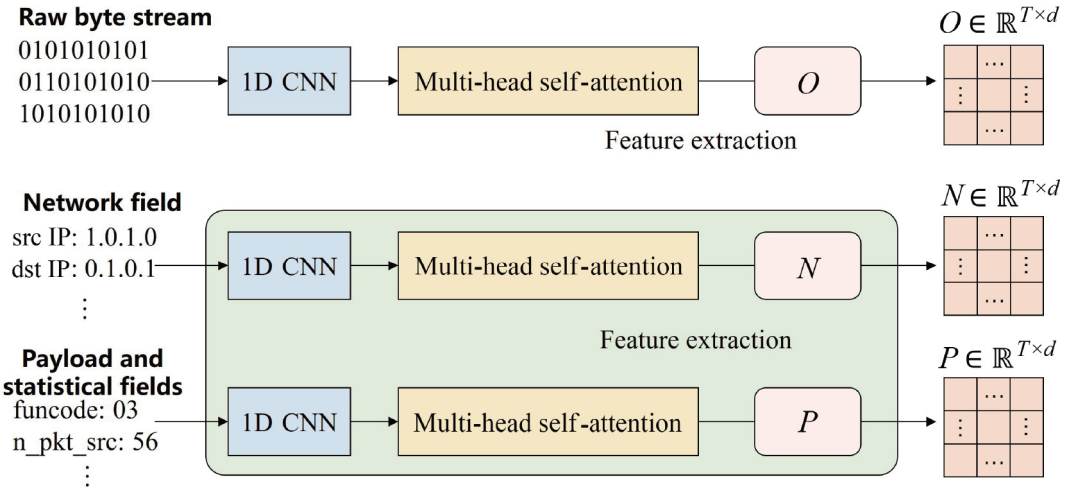

Fig. 4 Hierarchical feature representation framework.

statistical attributes are similar to payload attributes in terms of payload types and include real-time data related to industrial production and network status, the paper concatenates statistical attribute features with payload attributes. A spatial model is then used for feature extraction to reduce the dimensionality of the data after feature extraction and accelerate model computation speed.

3.5 Two-level fusion strategy for traffic feature

Considering the different perspectives and richness of information extracted at each layer, the raw traffic feature vector contains complete low-level traffic information from the perspective of network transmission, while the attribute field vectors focus on high-level abstract local information related to their specific functions. This section presents a two-stage attention-based fusion strategy, as illustrated in Fig. 5. In the first stage, the attribute field vectors are fused, and the self-attention mechanism is employed to adaptively focus on key feature information, constructing a complete traffic packet feature representation. In the second stage, a cross-attention mechanism is used to fuse the raw traffic and parsed traffic feature vectors, dynamically capturing the relational transformation between the low-level byte stream and high-level semantic features.

3.5.1 First-stage self-attention fusion

To maximize the retention of these attribute feature information, a cascade strategy is adopted to concatenate the attribute feature matrix vectors along the output channel direction, constructing a complete traffic packet feature representation. Since the 1D convolution layer is configured with the same number

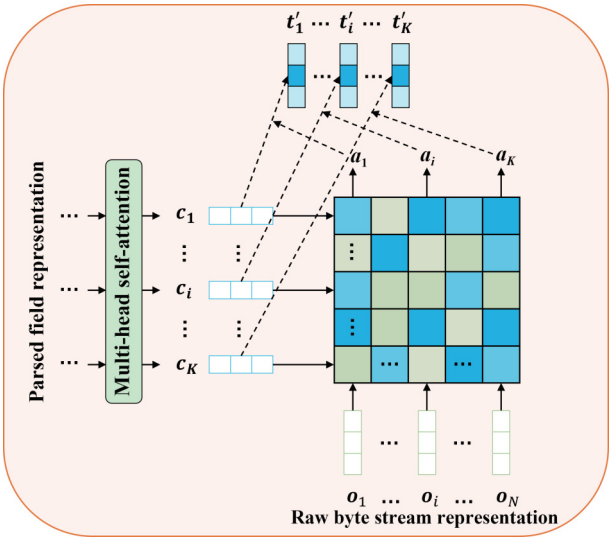

Fig. 5 Two-stage attention-based fusion strategy.

of output channels and the self-attention representation layer is set with the same hidden layer dimensions, the final weighted aggregated vector matrix is consistent in the spatial dimension, allowing it to be directly concatenated along the output channel dimension. The network attribute feature matrix is denoted as $N = \{n_1, n_2, \ldots, n_i, \ldots, n_T\}$, where T is the number of output channels after convolution and $n_i \in \mathbb{R}^d$ represents the network attribute feature vector of length d for the i-th channel. Thus, $N \in \mathbb{R}^{T\times d}$. The payload and statistical attribute feature matrix is denoted as $P = \{p_1, p_2, \ldots, p_i, \ldots, p_{T_f}\}$, where $p_i \in \mathbb{R}^d$ represents the payload and statistical attribute feature vector of length d for the i-th channel. Therefore, $P \in \mathbb{R}^{T\times d}$. The formal description of the above process is shown in Eq. (8), where $\|$ denotes the concatenation operation.

$$\text{ConcatenatedFeature} = \text{MultiHead}(N||P) \tag{8}$$

3.5.2 Second-stage cross-attention fusion

The raw traffic and parsed traffic represent different perspectives of the same data flow. In this paper, cross-attention fusion is used to extract features from both the raw traffic and parsed traffic, leveraging multiple perspectives to uncover deeper characteristics of communication traffic. Cross-attention is a variant of the attention mechanism that establishes relationships between two different sequences, enabling cross-layer information interaction. Cross-attention allows the model to learn relational dependencies between semantic and raw features. When an attack targets one feature space (e.g., parsed fields), the model can still leverage signals from the unaffected view (e.g., raw bytes) to flag suspicious behavior. This multi-perspective comparison provides a built-in form of redundancy that mitigates the risk of successful evasion. In this paper, the parsed field representation vector is used to generate the query vector, while the raw traffic vector is used to generate the key and value vectors. This approach is based on guiding global information aggregation with local insights. Let the raw traffic vector matrix, after the hierarchical extraction framework, be denoted as $O \in \mathbb{R}^{N\times d}$, $N = T$. The fusion process is given by

$$T' = \text{CrossAttention}(Q^c, K^o, V^o) \tag{9}$$

3.6 Temporal feature extraction model and intrusion detection

This section presents the constructed temporal feature extraction model, as shown in Fig. 6. The model consists of the following components: a dimensionality reduction layer, a temporal embedding layer, a temporal modeling layer, and a detection layer. The polling mechanism in industrial production causes industrial network traffic to exhibit periodic fluctuation patterns and distinct temporal characteristics. Consequently, after extracting spatial features from individual packets, this paper utilizes a Transformer model to perform temporal modeling on the traffic sequence, aiming to capture the temporal dependencies between traffic flows. Since an output sequence is not required, only the Transformer-encoder portion is used. The Transformer encoder is adopted for temporal modeling due to its strength in capturing long-range dependencies and global patterns in traffic sequences features common in industrial control systems with

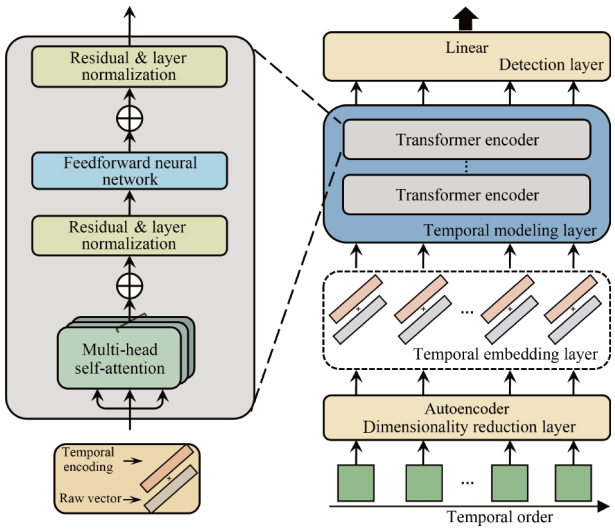

Fig. 6 Temporal model based on traffic sequence.

their periodic communication behavior. Unlike sequential RNN-based models such as LSTM or GRU, the Transformer supports parallel computation and attention-driven global modeling, improving both accuracy and efficiency. This makes it well-suited for real-time intrusion detection in ICS environments.

3.6.1 Dimensionality reduction layer

After fusion and flattening, the traffic packet vector representation t' becomes a high-dimensional vector. Directly using these high-dimensional data may lead to issues such as overfitting. Therefore, this paper applies the encoder part of an autoencoder to compress and reduce the dimensionality of the traffic packet t'. The ReLU activation function is used during this process to enhance non-linearity and improve model performance.

$$\begin{aligned} h &= f'(W_e t' + b), \\ t' &= g(W_d h + b') \end{aligned} \tag{10}$$

where f' denotes the encoder, g denotes the decoder, W_e and W_d represent the learnable weight matrices in the encoder and decoder, respectively. b and b' are the bias vectors corresponding to the encoding and decoding stages. After dimensionality reduction, the traffic packet vector is represented as $h = \{h_1, h_2, \ldots, h_i, \ldots, h_d\}$, where h is the compressed representation of the original high-dimensional vector, capturing the most relevant features while reducing redundancy and complexity.

3.6.2 Temporal embedding layer

The Transformer encoder processes input sequences in parallel, making it unable to automatically capture the temporal position of packets within the sequence. Therefore, to integrate the time order between traffic

packets, this paper uses sine and cosine functions to generate fixed positional encodings, embedding temporal information into each packet vector. These encodings are written as

$$\mathrm{PE}_{(i,2j)} = \sin\left(\frac{i}{10\,000^{\frac{2j}{d_{\mathrm{model}}}}}\right) \tag{11}$$

$$\mathrm{PE}_{(i,2j+1)} = \cos\left(\frac{i}{10\,000^{\frac{2j}{d_{\mathrm{model}}}}}\right) \tag{12}$$

where j represents the input dimension, and $d_{\mathrm{model}} = D$ refers to the hidden unit dimension. The term $\mathrm{PE}_{(i,2j)}$ corresponds to the $2j$-th component of the traffic packet feature vector, while $\mathrm{PE}_{(i,2j+1)}$ refers to the $(2j+1)$-th component of the traffic packet feature vector. After calculating the positional vector PE_i, it is added element-wise to the packet vector h_i to inject temporal information into the model.

3.6.3 Temporal modeling layer

For each data packet after temporal embedding, it is sequentially passed through the multi-head self-attention layer and the FNN layer, followed by residual connections and layer normalization. This process is repeated multiple times. To alleviate the degradation issues in neural network models, the Transformer introduces residual connections and layer normalization after the attention calculations. Residual connections add the input directly to the output of the network, enabling information transfer across layers and alleviating vanishing or exploding gradient issues. Layer normalization stabilizes the training process by reducing internal covariate shift, making the model more robust and easier to train. Since the self-attention mechanism processes sequences with linear transformations, to better capture the complex patterns and features within traffic sequences, a feedforward neural network introduces non-linear transformations. The feedforward neural network used in this paper consists of two linear layers, with an ReLU function for non-linear mapping in between, which is developed as follows:

$$\mathrm{FNN}(h) = \max(0, hW_1 + b_1)W_2 + b_2 \tag{13}$$

where W_i and b_i represent the weights and biases, respectively. After the FNN layer, the output will pass through another residual connection and normalization, resulting in a traffic sequence that integrates temporal context information.

3.6.4 Detection layer

The stability requirements of industrial production mean that industrial control systems are typically in a steady state, with normal traffic volumes far exceeding those of various attack traffic types. System anomalies are therefore rare, representing low-probability events. To address the class imbalance issue of industrial traffic and enhance the loss weight of minority attack classes, this paper uses the focal loss function to optimize loss calculation.

$$\mathrm{FL}(p_t) = \sum -\alpha_t \times h \times (1-p_t)^{\gamma} \log(p_t) \tag{14}$$

where α_t denotes the weight factor, γ represents an additional factor used to reduce the weight of easily classified samples that contribute less to model improvement, p_t represents the output distribution, and h is the one-hot vector of the current sample, respectively.

Finally, after N Transformer encodings, the output, which contains spatiotemporal information, will pass through a fully connected layer to compute the final class scores, indicating whether the traffic is benign or attack traffic. The scores will be normalized using softmax to produce the final classification vector, completing the entire intrusion detection process, which is written as

$$\hat{y} = \mathrm{softmax}(W_F h_i + b_F) \tag{15}$$

where h_i denotes the vector representation of a single traffic packet, W_F and b_F represent the weights and biases of the fully connected layer, respectively.

3.7 Time complexity analysis

The time complexity of intrusion detection model is $O(m \cdot c \cdot F^2) + O(n^2 \cdot d \cdot h_{\mathrm{a}})$, primarily encompassing the hierarchical representation module, attention fusion module, and temporal modeling and detection module. Specifically, the temporal modeling and detection module incorporates $|r|$ DCSA models, resulting in a time complexity of $|r| \cdot (O(m \cdot c \cdot F^2) + O(n^2 \cdot d \cdot h_{\mathrm{a}}))$, where m is the input size, c is the number of input channels, and F^2 represents the kernel size. Additionally, n is the input sequence length, d is the input dimension, and h_{a} is the hidden dimension of the multi-head attention mechanism. Both the attention fusion module and hierarchical representation module have a time complexity of $O(n^2 \cdot d \cdot h_{\mathrm{a}})$, where n, d, and h_{a} are defined as above. Thus, the overall time complexity is approximately $|r| \cdot O(m \cdot c \cdot F^2) + (|r|+2)O(n^2 \cdot d \cdot h_{\mathrm{a}})$. Employ parallel computing in the model, where $|r| = 1$, which simplifies

to $O(m \cdot c \cdot F^2) + 2 \cdot O(n^2 \cdot d \cdot h_a)$. Therefore, with proper allocation of computational resources, the complexity of our model can be effectively reduced, making it suitable for real-world applications.

3.8 Enhanced robustness with dynamic heterogeneous redundancy (DHR) framework

To enhance the robustness of the model, we adopt the dynamic heterogeneous redundancy concept to further strengthen the robustness of our intrusion detection model. The core idea behind the dynamic heterogeneous redundancy architecture is to deploy multiple models with the same functionality but different implementations in a resource pool. These models work collaboratively to detect network traffic anomalies. The heterogeneity of the models makes it much more difficult for attackers to successfully evade detection, as adversarial techniques are often designed to bypass specific detection mechanisms. By using diverse models, the likelihood of an attack being successful across all models is drastically reduced. The system employs a multi-model decision mechanism to assess the consistency of detection results across different models. In case of inconsistencies, the system triggers an alert and activates a feedback control loop to cleanse or replace the anomalous models. This approach, using heterogeneous detection models and a decision-making mechanism, significantly increases the cost for attackers attempting to exploit specific intrusion detection model weaknesses, while also boosting overall system robustness. Based on this concept, we designed the following architecture, as shown in Fig. 7, using the MHST model we proposed as the primary detection model to meet the time and performance requirements of industrial control scenarios.

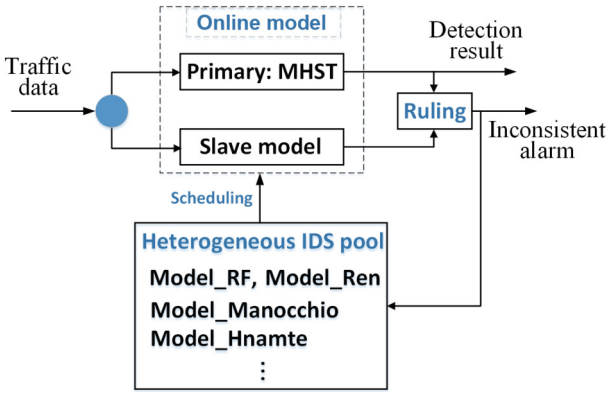

Fig. 7 Robust DHR intrusion detection model.

We incorporate intrusion detection models proposed by other researchers as heterogeneous intrusion detection resources. During operation, the primary intrusion detection model (MHST) and the backup models simultaneously receive traffic data for detection. Since the detection times and performance vary between models, and the decision-making of the adjudicator can introduce delays, using multiple models for cooperative detection may impact real-time performance. Therefore, we designed the system to rely on the primary MHST model for real-time detection, providing immediate results. The main role of the backup models and adjudicator is to align the detection results in terms of timing and ensure consistency in their decisions. If there is any inconsistency, an alert is triggered. Although this may introduce some delay compared to MHST, it does not affect detection performance. Due to the use of heterogeneous detection models, evasion techniques and countermeasures employed by attackers are typically designed to target specific detection models. It is difficult for attackers to design evasion strategies or countermeasures that work against multiple models simultaneously. This mechanism provides an additional layer of defense against sophisticated evasion techniques, ensuring that no single model can be bypassed easily, and making adversarial attacks more costly and difficult. Additionally, through the use of heterogeneous models and adjudication mechanism, the robustness of the intrusion detection system is enhanced.

4 Experiment and Analysis

This section validates the effectiveness of the proposed intrusion detection model. Most existing publicly available industrial traffic datasets are based on manually designed feature sets that represent traffic using high-level semantic fields, and the majority rely on the Modbus protocol[39–41]. The only dataset providing both raw traffic data and packet-level annotations is the water distribution testbed dataset, published by Faramondi et al.[34] in 2021.

4.1 Dataset

WDT is a Modbus TCP traffic dataset collected from a hardware-in-the-loop water distribution testbed. The WDT dataset contains two hours of industrial communication traffic, storing over 20 million traffic records. In this paper, the Attack 2 subset, which

includes the richest variety of attack types, is selected as the experimental dataset. The Attack 2 subset contains 5 159 469 network traffic records, with detailed attack descriptions provided in Table 4.

To accelerate model training and reduce computational resource costs, this study randomly selected 10% of the data from the dataset to construct the experimental dataset. The dataset was then split into training, validation, and test sets in a 6:2:2 ratio, as shown in Table 5. The training set is used to train the model's parameters, capturing the features and patterns within the experimental data. The validation set helps adjust the model's hyperparameters and structure during training, preventing overfitting and ensuring good generalization. The test set is used after training to evaluate the model's practical performance.

4.2 Evaluation methodology and model parameter

Intrusion detection can essentially be categorized as a multi-class classification problem. In this section, a confusion matrix is introduced to construct quantitative evaluation metrics.

True positive (TP): Positive samples correctly predicted as positive by the model.

False positive (FP): Negative samples incorrectly predicted as positive by the model.

True negative (TN): Negative samples correctly predicted as negative by the model.

False negative (FN): Positive samples incorrectly predicted as negative by the model. Due to the extreme class imbalance in industrial datasets, where the majority of normal samples significantly outnumber the minority of attack samples, there is a risk of "fraud detection" effect. Therefore, this paper selects macro precision, macro recall, macro F1-score, macro false positive rate (Macro FPR), and time-consuming as evaluation metrics for the intrusion detection model[42, 43]. The parameters for the intrusion detection model and training are shown in Table 6.

4.3 Experimental result and analysis

4.3.1 Comparative experiment

To validate the effectiveness and superiority of the proposed multi-view hierarchical intrusion detection model with spatiotemporal features, advanced models from recent years are selected for comparison based on the publicly available industrial traffic dataset WDT. The performance of the models is quantitatively evaluated according to the established metrics. For fairness, all models use the focal loss function as the training loss. The experiments were conducted on a system with the following hardware specifications: an Intel(R) Xeon(R) Platinum 8352V CPU @ 2.10 GHz, three NVIDIA GeForce RTX 3090 GPUs, and 125 GB of DDR4 3200 MHz RAM. The software environment included JetBrains PyCharm ×64 and IntelliJ IDEA ×64, providing a robust platform for model implementation and testing. The selected comparison models are as follows:

(1) Model_RF: Random forest is a classic machine learning model that builds multiple decision trees to independently learn and make decisions. Final decisions are made based on a majority voting mechanism, which has shown excellent performance in industrial intrusion detection tasks[44].

(2) Model_Ren (2023): Reference [45] proposed a hierarchical convolutional attention network. This model extracts local spatiotemporal features of traffic packets by combining convolutional neural networks and attention mechanisms. It uses a three-layer stacked learning approach to capture multi-level spatiotemporal feature information.

(3) Model_Hnamte (2023): Reference [46] designed a two-stage ensemble model that combines long short-term memory networks and autoencoders. The LSTM network is used to encode the traffic data, and the encoded vectors are weighted using an ensemble attention mechanism. The vectors are then decoded and detected using another LSTM network.

Table 4 Description in the Attack 2 dataset.

Attack domain	Definition	Attack type	Traffic labeling	Proportion
Cyber domain	Attack on network, host, or communication link	Man-in-the-middle attack	MITM	4.2100%
Cyber domain		Denial-of-service attack	DoS	11.0800%
Cyber domain		Scanning attack	Scan	0.0006%
Physical domain	Attack on physical device such as sensor or device failure	Leakage/sensor/pump failure	Physical fault	5.3700%
Others	–	Normal traffic	Normal	79.3300%
Others	–	Unspecified labeling reason	Anomaly	0.0020%

Table 5 Dataset splitting.

Category	Training set	Validation set	Test set
Normal traffic	250 872	80 886	78 843
DoS attack	29 696	12 288	14 366
MITM attack	13 312	6144	6144
Physical fault	15 360	4096	4096
Scan attack	18	6	6

Table 6 Model parameter setting.

Module	Parameter	Value
Overall model training	Epoch	50
	Batch size	1024
	Optimizer	AdamW
	Learning rate	1×10^{-3}
	Dropout	0.5
Hierarchical feature representation module	out_channels	64
	kernel_size	64
	DCSA head	4
Two-level fusion strategy	Self head	4
	Cross head	8
Temporal modeling and intrusion detection module	auto_hidden_dim	2048
	auto_output_dim	512
	Transformer head	2
	num_layers	2
	Alpha	0.65
	Gamma	1

(4) Model_Li (2022): Reference [47] designed a hybrid intrusion detection model that combines convolutional neural networks and bidirectional long short-term memory networks. The hierarchical structure of the CNN is used to extract spatial features of the traffic, while the BiLSTM network captures the temporal features.

(5) Model_Manocchio (2024): Reference [48] proposed an intrusion detection framework based on the Transformer architecture. The core of this model leverages the Transformer's strength in modeling long sequences to capture complex temporal correlations in traffic data.

Table 7 presents a comparison of our MHST model with existing models based on the metrics $\text{Precision}_{\text{macro}}$, $\text{Recall}_{\text{macro}}$, $F1_{\text{macro}}$, and $\text{FPR}_{\text{macro}}$. Higher values of $\text{Precision}_{\text{macro}}$, $\text{Recall}_{\text{macro}}$, and F1_{macro} indicate better model performance, while lower values of $\text{FPR}_{\text{macro}}$ also signify better performance. Table 7 shows that the proposed model outperforms all other baseline models across all metrics. Compared to the second-best model (Model_Li), the proposed model improves $\text{Precision}_{\text{macro}}$ by 1.91%, $\text{Recall}_{\text{macro}}$ by 0.81%, and F1_{macro} by 1.34%, and decreases $\text{FPR}_{\text{macro}}$ by 0.41%.

Table 7 Comparison experimental result.

Model	$\text{Precision}_{\text{macro}}$	$\text{Recall}_{\text{macro}}$	F1_{macro}	$\text{FPR}_{\text{macro}}$
Model_RF	0.9128	0.8568	0.8568	0.0277
Model_Ren (2023)	0.9576	0.9323	0.9441	0.0174
Model_Hnamte (2023)	0.9529	0.9551	0.9540	0.0130
Model_Li (2022)	0.9640	0.9643	0.9642	0.9642
Model_Manocchio (2024)	0.9362	0.9638	0.9493	0.0119
MHST (ours)	0.9831	0.9724	0.9776	0.0058

Table 8 presents a comparison of the proposed MHST model with existing models in terms of minimum time consuming, maximum time consuming, and average time consuming. The test set contained 103 455 samples, and multiple detection time tests are conducted on five baseline models and our MHST model. For our MHST model, the detection time results are as follows: a minimum of 1445.786 ms, a maximum of 1532.452 ms, and an average of 1482.393 ms. As shown in Table 8, our MHST model outperformed the Model_RF, Model_Ren, Model_Li, and Model_Manocchio models in terms of average time consuming, underscoring the efficiency of our approach in handling response time performance. Although our model's average time consuming was 259.000 ms higher than that of Model_Hnamte, Model_Hnamte demonstrated lower detection accuracy. Additionally, it is worth noting that our model required an average of 1482.393 ms to process 103 455 samples. This corresponds to an average inference time of 0.0143 ms per packet, which meets the real-time performance requirements of practical

Table 8 Comparison of time-consuming experimental result.

Model	Minimum time consuming (ms)	Maximum time consuming (ms)	Average time consuming (ms)
Model_RF	1430.177	1538.272	1529.195
Model_Ren (2023)	1644.959	1731.519	1657.905
Model_Hnamte (2023)	1214.306	1240.043	1223.005
Model_Li (2022)	1660.206	1723.900	1672.387
Model_Manocchio (2024)	1752.951	1839.825	1792.139
MHST (ours)	1445.786	1532.452	1482.393

industrial control scenarios.

Among the models mentioned, only Model_RF is a traditional machine learning model, with a $\text{Precision}_{\text{macro}}$ of 0.9128 and a $\text{Recall}_{\text{macro}}$ of 0.8568, which is relatively lower compared to the performance of deep learning models. This is because deep learning models can more effectively handle high-dimensional traffic data and identify complex patterns and nonlinear relationships within the traffic for intrusion detection tasks. Model_Ren, Model_Hnamte, and Model_Manocchio identify characteristics of traffic in either the spatial or temporal dimensions and select corresponding models to extract relevant information based on these characteristics. While these models achieve good detection performance, they are limited to analyzing a single aspect of traffic-either focusing on the temporal dimension or the spatial dimension. This unidimensional analysis fails to fully and comprehensively extract the multidimensional spatiotemporal features and the complex interaction patterns between these features. Model_Li developed an ensemble model that integrates CNN and BiLSTM, taking into account the multidimensional characteristics of traffic in both spatial and temporal dimensions, thereby improving intrusion detection performance. In contrast to the aforementioned models, the proposed MHST conducts a deeper analysis of the differences in functionality fields and payloads of traffic, as well as the mechanisms of network attacks, based on a comprehensive understanding of spatiotemporal characteristics. The designed hierarchical feature representation framework can finely mine concentrated features and behavior patterns of different attribute fields. Additionally, this paper incorporates raw byte stream information of traffic and models traffic representation based on multi-view correlations, further enriching the feature representation of traffic. This enables the model to gain a deeper understanding of traffic behavior patterns during network transmission, thus enhancing the effectiveness of intrusion detection.

Figures 8 and 9 respectively display the confusion matrices of the proposed MHST and the suboptimal model (Model_Li). In these matrices, the vertical axis represents the true class of the samples, while the horizontal axis represents the predicted class. Each element in the matrix indicates the proportion of samples from the row (i.e., true label) that were predicted as a specific class. The values along the main

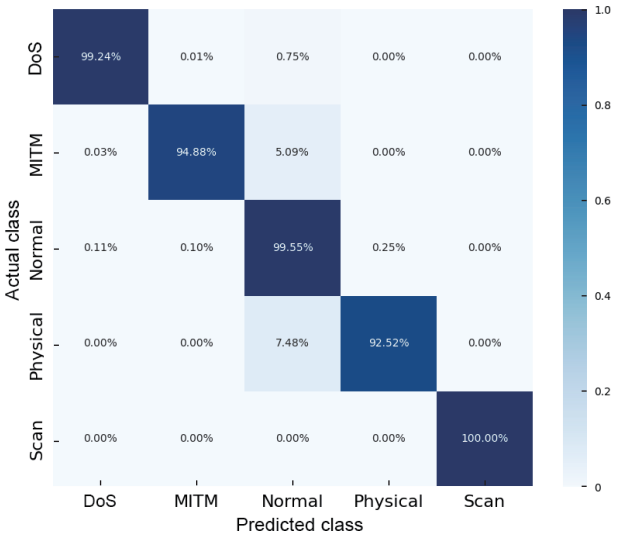

Fig. 8 Confusion matrix of the proposed MHST model.

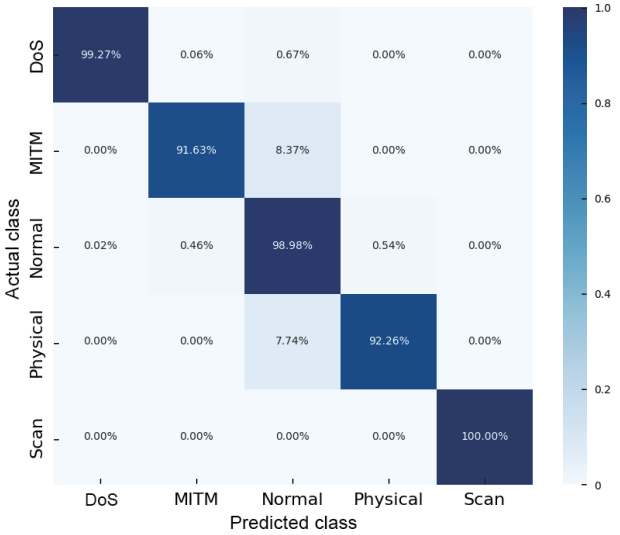

Fig. 9 Confusion matrix of the Model_Li.

diagonal represent the proportion of correctly predicted samples for each class relative to the total number of actual samples in that class, indicating the model's detection rate for that class of traffic. It can be observed that the proposed MHST demonstrates strong classification capability across all types of traffic. Compared to the suboptimal model, MHST exhibits optimal detection performance in all categories except for DoS attacks. The most significant improvement of the MHST is in the identification of MITM attacks, with a 3.25% increase in detection rate compared to the suboptimal model. Analysis reveals that MITM attacks primarily interfere with the effective payload of communication traffic, while having limited impact on other components of the traffic. Other models, which

focus on overall traffic feature learning and representation, struggle to capture the specific patterns of influence caused by the attack. This makes it difficult for them to detect the targeted threat information within the traffic.

4.3.2 Ablation experiment

To further verify the validity of the proposed MHST model, this section conducts an ablation experiment to analyze the contribution of each module to the overall performance of MHST, demonstrating the necessity of each component. The core modules of MHST include: the hierarchical feature extraction framework, the raw traffic feature extraction module, the attention-based two-stage fusion strategy, and the cost-sensitive loss function. In the following, we systematically assess the impact on intrusion detection capability by removing each module individually and in combination.

MHST without hierarchical: The hierarchical feature extraction framework is removed, and all parsed fields and raw traffic representations are concatenated. A single-layer model is used to extract traffic features, aiming to validate the effectiveness of fine-grained traffic feature extraction.

MHST without parsed, hierarchical: The parsed feature extraction module and the hierarchical feature extraction framework are removed, leaving only the raw feature extraction module. This configuration aims to validate the effectiveness of parsed field information in enhancing the representation of traffic features.

MHST without raw: The raw traffic feature extraction module is removed, retaining only the parsed feature hierarchical extraction module. This setup aims to validate the effect of raw traffic information on enhancing the representation of traffic features.

MHST without cross: The cross-attention fusion from the two-stage fusion strategy is removed to validate the effectiveness of cross-attention in multi-view feature integration.

MHST without FL: Cost sensitivity is removed from the network, replacing the focal loss function with a cross-entropy loss function. This setup aims to validate the effectiveness of cost-sensitive learning in addressing the issue of imbalanced traffic distribution.

MHST without payload field: Both the raw traffic and parsed field perspectives are retained, but the payload attribute fields in the parsed field perspective are removed. This experiment aims to verify the accuracy of intrusion detection without using payload attribute fields.

Based on the experimental dataset WDT, the relevant parameters continue to follow the settings established in the experimental design. The final results of the ablation study are presented in Table 9.

Table 9 shows that in Experiment 2, $\text{Precision}_{\text{macro}}$, $\text{Recall}_{\text{macro}}$, and F1_{macro} decrease by 1.74%, 1.26%, and 1.49%, respectively, demonstrating the effectiveness of the layered feature extraction framework. Theoretically, this hierarchical framework enables the model to capture traffic behavior patterns and variations within different attribute fields at a finer granularity, allowing for more in-depth traffic analysis. Additionally, each layer within the framework can focus on key features relevant to specific attack types, facilitating accurate identification of different network attacks and ultimately improving intrusion detection performance.

Experiments 3 and 4 indicate that the model's performance declines when either the raw byte stream features or parsed field features are removed. Notably, the removal of parsed field feature extraction results in a significant drop in detection performance, with $\text{Precision}_{\text{macro}}$, $\text{Recall}_{\text{macro}}$, and F1_{macro} decreasing by 3.15%, 2.90%, and 3.02%, respectively, underscoring the importance of high-level semantic features in traffic representation. Similarly, when raw byte stream feature extraction is removed, detection performance declines slightly, with reductions of 0.88%, 1.03%, and 0.95%

Table 9 Ablation experimental result.

Experiment No.	Model	$\text{Precision}_{\text{macro}}$	$\text{Recall}_{\text{macro}}$	F1_{macro}
1	MHST	0.9831	0.9724	0.9776
2	MHST without hierarchical	0.9657	0.9598	0.9627
3	MHST without parsed, hierarchical	0.9516	0.9434	0.9474
4	MHST without raw	0.9743	0.9621	0.9681
5	MHST without cross	0.9798	0.9634	0.9715
6	MHST without FL	0.9698	0.9636	0.9667
7	MHST without payload field	0.9683	0.9665	0.9655

in these metrics. These results suggest that combining multiple perspectives for traffic content modeling enriches the feature representation, thus enhancing the model's detection performance.

In Experiment 5, the model's performance shows a slight decrease without the cross-attention fusion process: $\text{Precision}_{\text{macro}}$ decreases by 0.33%, $\text{Recall}_{\text{macro}}$ by 0.90%, and F1_{macro} by 0.61%. This suggests that while establishing correlations and transformations between the two perspectives can enrich traffic message representation, the features in the raw byte stream and parsed fields are already sufficient to express traffic characteristics, so the additional cross-attention fusion offers only minor improvement.

In Experiment 6, cost sensitivity is removed by replacing the focal loss function with the commonly used cross-entropy loss function for multiclass classification. This lead to decreases of 1.33%, 0.88%, and 1.09% in $\text{Precision}_{\text{macro}}$, $\text{Recall}_{\text{macro}}$, and F1_{macro}, respectively, underscoring the importance of balancing strategies for handling imbalanced traffic data.

In Experiment 7, when payload attribute fields are removed, detection performances on these three metrics are 0.9683, 0.9665, and 0.9655, decrease by 1.50%, 0.60%, and 1.20%, suggesting that payload attribute fields enhance the feature representation of traffic data, and removing them results in some decline in detection performance.

For encrypted protocols, our model extracts features directly from the raw byte stream, eliminating the need for decrypted payload data. From the parse traffic perspective, the network attribute fields and statistical attribute fields capture communication behaviors and statistical properties of communication processes, without requiring direct access to encrypted payload data. These indirect features have proven effective in detecting abnormal behaviors associated with intrusions or attacks, even in the absence of payload visibility. From Experiment 7, the detection performances of MHST without payload fields, including $\text{Precision}_{\text{macro}}$, $\text{Recall}_{\text{macro}}$, and F1_{score}, are 0.9683, 0.9665, and 0.9655, respectively. Compared to the results in Table 7, removing the payload attribute fields still results in slightly better detection performance than the five baseline models. These results indicate that our model can support intrusion detection for encrypted traffic using the raw traffic and parsed field perspective excluding payload attribute fields.

In summary, the proposed MHST model outperforms other variants across $\text{Precision}_{\text{macro}}$, $\text{Recall}_{\text{macro}}$, and F1_{macro} metrics, indicating that the coordination of MHST's four modules is well-designed, with each module working in synergy to achieve optimal intrusion detection performance.

4.3.3 Parameter analysis experiment

To explore the optimal settings of the model, this section analyzes important parameters, mainly including the number of heads in the first stage self-attention layer of the two-level fusion strategy and the number of heads in the second stage cross-attention, as well as the dimensionality of traffic representation after dimensionality reduction by the autoencoder.

(1) Hyperparameter analysis of two-level fusion strategy based on attention.

In the two-stage fusion strategy based on the attention mechanism, multi-head self-attention and cross-attention mechanisms are used for feature fusion in two stages. Below is an analysis of the number of heads in the attention mechanisms for each stage of fusion.

Figure 10 illustrates the impact of the number of attention heads on model performance in each stage of the attention-based two-stage fusion strategy. In Fig. 10a, the parameter analysis results of the multi-head self-attention layer are presented. The F1_{macro} score is used as the evaluation metric, and four different candidate head numbers are tested: 1, 2, 4, and 6. When fewer attention heads are set (1 and 2), the model performance is lower, as the limited number of heads restricts the model's ability to capture diverse features from the traffic data. When the number of attention heads increases to 4, the model achieves optimal performance. However, as the number of heads continues to increase to 6, the model exhibits a phenomenon of diminishing returns, where the additional heads provide progressively less benefit. Figure 10b shows the parameter analysis results for the cross-attention layer, with the F1_{macro} score as the evaluation metric. Five different candidate head numbers are tested: 2, 4, 8, 12, and 16. When the number of heads is set to 8, the model achieves optimal performance. With fewer heads, the model is unable to fully capture the complex feature information in the traffic data. However, when the number of heads exceeds 8, the excessive number of heads causes the attention mechanism to become dispersed, reducing the amount of information captured by each head, leading

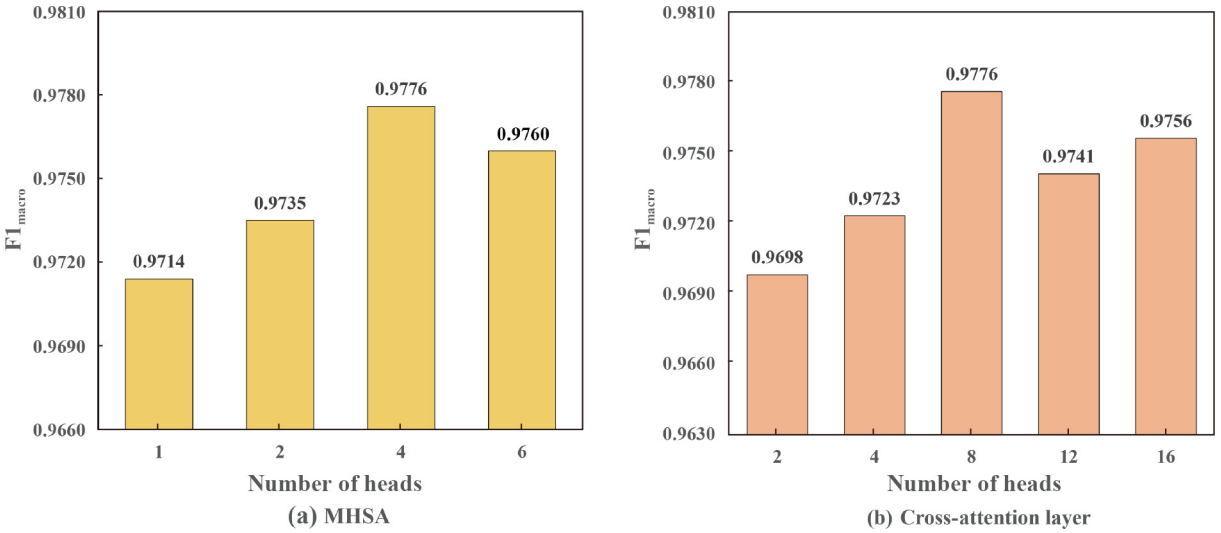

Fig. 10 Parameter analysis of the number of heads.

to a decline in model performance.

(2) Hyperparameter analysis of dimensionality reduction using autoencoder.

In the dimensionality reduction process of high-dimensional traffic packets, the encoding dimension of the encoder is directly related to the representation capability and information density of the reduced traffic data, thus impacting the performance of intrusion detection. In this section, the $F1_{macro}$ score is ussic, and the candidate encoding dimensions selected are 128, 256, 384, 512, 640, 768, and 1024. The experimental results are shown in Fig. 11.

As shown in Fig. 11, the $F1_{macro}$ score generally exhibits a trend of first increasing and then decreasing with the increase in encoding dimension, reaching optimal performance at 512 and 768 dimensions. Analysis indicates that in the lower range of encoding dimensions, as the dimensionality increases, the encoder model can more effectively capture feature information and complex patterns within the traffic data. However, when the dimensionality increases to 1024, model performance begins to decline. This is because a higher dimensionality retains more information while also introducing unnecessary noise and redundant features, leading to overfitting and diminishing the model's generalization capability.

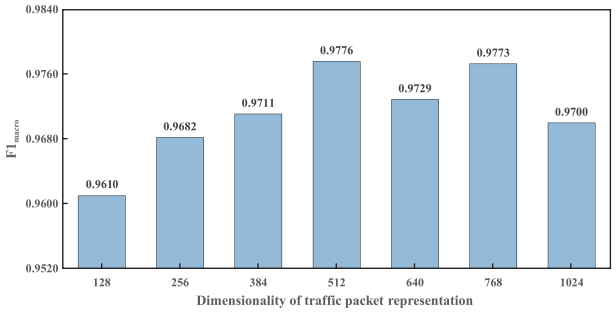

Fig. 11 Parameter analysis of the encoding dimension.

(3) Analysis of variant of the two-stage fusion strategy based on the attention mechanism.

This paper utilizes the cross-attention mechanism for cross-attention fusion (Cross F) to analyze high-dimensional features of traffic alongside the original high-dimensional features. To validate the rationality of the designed fusion strategy, this section conducts comparative experiments on the fusion strategies, including feature concatenation fusion (FCF) and dual self-attention fusion (DSAF). The experimental results are shown in Fig. 12.

As shown in Fig. 12, the cross-attention two-stage fusion strategy proposed in this paper achieves the best detection performance. The DSAF fusion strategy ranks the second, with average decreases of 0.20%, 0.68%, and 0.44% in $Precision_{macro}$, $Recall_{macro}$, and

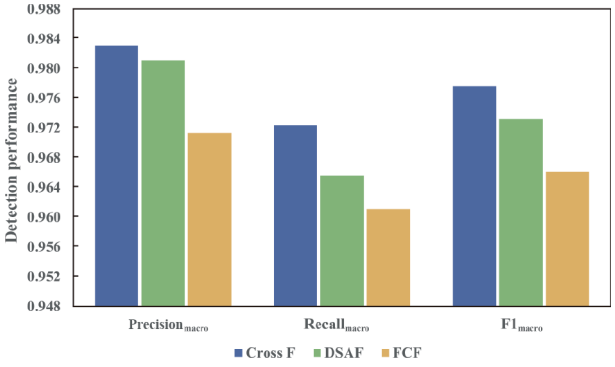

Fig. 12 Comparative experiment on feature fusion strategy.

$F1_{macro}$, respectively, compared to the proposed method. This is because the self-attention mechanism primarily focuses on internal correlations within feature vectors, failing to fully exploit complementary and relational information between different perspectives. The FCF method performs the worst, with average decreases of 1.08%, 1.53%, and 1.30% in $Precision_{macro}$, $Recall_{macro}$, and $F1_{macro}$, respectively. This is because simple feature concatenation struggles to capture complex relationships between features, and the indiscriminate combination introduces substantial redundancy and noise, reducing model performance.

4.3.4 Robustness validation experiment

We select the Attack 2 subset from the WDT dataset, which includes the widest variety of attack types, as our experimental dataset. In the training set, we add noise to the input data to serve as perturbation data, in order to validate the robustness of the intrusion detection model designed using the DHR architecture. We take the proposed MHST model as the primary model, and choose Model_Hnamte and Model_ Manocchio as slave models respectively to verify the robustness of the proposed DHR architecture intrusion detection method. The experimental results are presented in Table 10.

From the experimental results, it can be observed that when using a single MHST model, the system is able to accurately detect most anomalous traffic under normal conditions. However, when the DHR architecture with heterogeneous intrusion detection models is employed, precision becomes very high. This is due to the higher detection accuracy of the DHR architecture's models, which enhances detection capabilities by integrating heterogeneous models, offering a multi-perspective approach to traffic feature analysis. For example, certain attacks based on protocol mutations may be difficult to detect with the MHST model but can be effectively identified by other models, such as Model_Hnamte and Model_Manocchio. By combining the detection results from these different models, the system achieves higher robustness.

Table 10 Robustness experiment with DHR framework.

Experiment No.	Model	$Precision_{macro}$	$Recall_{macro}$	$F1_{macro}$
1	MHST	0.9831	0.9724	0.9776
2	MHST+ Hnamte	0.9981	0.9452	0.9709
3	MHST+ Manocchio	0.9919	0.9531	0.9721

5 Conclusion

This paper proposes a multi-view hierarchical intrusion detection model that integrates spatial and temporal features. The model constructs a fine-grained feature extraction framework at the functional field level of traffic data, enabling in-depth and detailed mining of traffic information. By integrating traffic features and relational information from both the raw traffic perspective and the parsed field perspective, the model achieves multi-view collaborative traffic modeling, further enhancing the feature representation of the traffic data. A local spatial feature extraction model based on convolutional neural networks and a global temporal feature extraction model based on the Transformer architecture are designed to capture the spatial and temporal features of the traffic. A cost-sensitive focal loss function is introduced to train the model, achieving high-precision intrusion detection. Experiments are conducted on a real industrial intrusion detection dataset to validate the model, comparing it with several advanced models. The results demonstrate that the proposed model achieves superior performance in terms of $Precision_{macro}$, $Recall_{macro}$, $F1_{macro}$, and FPR metrics. Additionally, through ablation experiments and parameter analysis experiments, the rationality of the proposed model and the optimal parameter settings are further verified, providing an effective solution for intrusion detection in industrial control system. Currently, our model is primarily designed for industrial network protocols based on the TCP transport protocol. We plan to expand and refine this approach to improve its versatility, enabling support for a broader range of industrial protocols to better adapt to diverse industrial network environments.

Acknowledgment

This work was supported by the National Key Research and Development Program of China (No. 2022YFB3104300).

References

[1] D. Benka, D. Horváth, L. Špendla, G. Gašpar, and M. Strémy, Machine learning-based detection of anomalies, intrusions, and threats in industrial control systems, *IEEE Access*, vol. 13, pp. 12502–12514, 2025.

[2] H. Luo and L. Wan, A recombination generative adversarial network for intrusion detection, *Int. J. Appl. Math. Comput. Sci.*, vol. 34, no. 2, pp. 323–334, 2024.

[3] M. Asiri, N. Saxena, R. Gjomemo, and P. Burnap, Understanding indicators of compromise against cyber-attacks in industrial control systems: A security perspective, *ACM Trans. Cyber-Phys. Syst.*, vol. 7, no. 2, p. 15, 2023.

[4] M. Imran, H. U. R. Siddiqui, A. Raza, M. A. Raza, F. Rustam, and I. Ashraf, A performance overview of machine learning-based defense strategies for advanced persistent threats in industrial control systems, *Comput. Secur.*, vol. 134, p. 103445, 2023.

[5] D. Zhang, M. Wang, Y. Bu, J. Yu, and L. Yang, PdGAT-ID: An intrusion detection method for industrial control systems based on periodic extraction and spatiotemporal graph attention, *Comput. Secur.*, vol. 149, p. 104210, 2025.

[6] D. Zhou, L. He, Z. Cao, A. Zhang, and X. Han, Event-triggered fixed-time resilient control for mobile sensor networks with a Sybil attack and input delay, *Int. J. Appl. Math. Comput. Sci.*, vol. 35, no. 1, pp. 129–142, 2025.

[7] T. Alladi, V. Chamola, and S. Zeadally, Industrial control systems: Cyberattack trends and countermeasures, *Comput. Commun.*, vol. 155, pp. 1–8, 2020.

[8] H. Nandanwar and R. Katarya, Deep learning enabled intrusion detection system for industrial IOT environment, *Expert Syst. Appl.*, vol. 249, p. 123808, 2024.

[9] M. Al-Zewairi, S. Almajali, M. Ayyash, M. Rahouti, F. Martinez, and N. Quadar, Multi-stage enhanced zero trust intrusion detection system for unknown attack detection in internet of things and traditional networks, *ACM Trans. Priv. Secur.*, vol. 28, no. 3, p. 30, 2025.

[10] C. Jia, Research on industrial internet intrusion detection based on deep learning algorithms, *Appl. Computat. Eng.*, vol. 133, pp. 32–36, 2025.

[11] M. Nuaimi, L. C. Fourati, and B. Ben Hamed, Intelligent approaches toward intrusion detection systems for industrial internet of things: A systematic comprehensive review, *J. Network Comput. Appl.*, vol. 215, p. 103637, 2023.

[12] H. Kheddar, Y. Himeur, and A. I. Awad, Deep transfer learning for intrusion detection in industrial control networks: A comprehensive review, *J. Network Comput. Appl.*, vol. 220, p. 103760, 2023.

[13] D. Chaudhary, D. Shekhawat, S. Gupta, A. Kalwar, N. Mishra, and M. Nawal, Enhancing cybersecurity using optimized anti-interference dynamic integral neural network-based intrusion detection system, *Knowledge Inf. Syst.*, vol. 67, no. 6, pp. 5413–5435, 2025.

[14] Z. Liu, B. Xu, B. Cheng, X. Hu, and M. Darbandi, Intrusion detection systems in the cloud computing: A comprehensive and deep literature review, *Concurr. Comput.*: *Pract. Exper.*, vol. 34, no. 4, p. e6646, 2022.

[15] S. Srivastav, A. K. Shukla, S. Kumar, and P. K. Muhuri, HYRIDE: HYbrid and robust intrusion DEtection approach for enhancing cybersecurity in industry 4.0, *Internet Things*, vol. 30, p. 101492, 2025.

[16] X. Yang, F. Tong, F. Jiang, and G. Cheng, A lightweight and dynamic open-set intrusion detection for industrial internet of things, *IEEE Trans. Inf. Forensics Secur.*, vol. 20, pp. 2930–2943, 2025.

[17] A. Heidari and M. A. Jabraeil Jamali, Internet of Things intrusion detection systems: A comprehensive review and future directions, *Cluster Comput.*, vol. 26, no. 6, pp. 3753–3780, 2023.

[18] A. Chahal, P. Gulia, N. S. Gill, and D. Rani, Design of a federated ensemble model for intrusion detection in distributed IIoT networks for enhancing cybersecurity, *J. Ind. Inf. Integr.*, vol. 44, p. 100800, 2025.

[19] R. Saadouni, C. Gherbi, Z. Aliouat, Y. Harbi, A. Khacha, and H. Mabed, Securing smart agriculture networks using bio-inspired feature selection and transfer learning for effective image-based intrusion detection, *Internet Things*, vol. 29, p. 101422, 2025.

[20] M. Shahin, M. Maghanaki, A. Hosseinzadeh, and F. F. Chen, Advancing network security in industrial IoT: A deep dive into AI-enabled intrusion detection systems, *Adv. Eng. Inf.*, vol. 62, p. 102685, 2024.

[21] S. Krishnaveni, S. Sivamohan, B. Jothi, T. M. Chen, and M. Sathiyanarayanan, TwinSec-IDS: An enhanced intrusion detection system in SDN-digital-twin-based industrial cyber-physical systems, *Concurr. Comput.*: *Pract. Exper.*, vol. 37, no. 3, p. e8334, 2025.

[22] B. Lampe and W. Meng, Intrusion detection in the automotive domain: A comprehensive review, *IEEE Commun. Surv. Tutorials*, vol. 25, no. 4, pp. 2356–2426, 2023.

[23] T. Al-Shurbaji, M. Anbar, S. Manickam, I. H. Hasbullah, N. Alfriehat, B. A. Alabsi, A. R. Alzighaibi, and H. Hashim, Deep learning-based intrusion detection system for detecting IoT botnet attacks: A review, *IEEE Access*, vol. 13, pp. 11792–11822, 2025.

[24] K. Rahim, Z. U. I. Nasir, N. Ikram, and H. K. Qureshi, Integrating contextual intelligence with mixture of experts for signature and anomaly-based intrusion detection in CPS security, *Neural Comput. Appl.*, vol. 37, no. 8, pp. 5991–6007, 2025.

[25] S. Mishra and V. K. Chaurasiya, Secure transactions in IoT network using PINN-based intrusion detection system and HDPoA blockchain protocol, *Trans. Emerg. Telecommun. Technol.*, vol. 36, no. 4, p. e70116, 2025.

[26] H. Lin, A. Slagell, Z. T. Kalbarczyk, P. W. Sauer, and R. K. Iyer, Runtime semantic security analysis to detect and mitigate control-related attacks in power grids, *IEEE Trans. Smart Grid*, vol. 9, no. 1, pp. 163–178, 2018.

[27] J. Men, Z. Lv, X. Zhou, Z. Han, H. Xian, and Y. Song, Machine learning methods for industrial protocol security analysis: Issues, taxonomy, and directions, *IEEE Access*, vol. 8, pp. 83842–83857, 2020.

[28] D. Huderek, S. Szczęsny, P. Pietrzak, R. Rato, and Ł. Przyborowski, A spiking neural network based on thalamo-cortical neurons for self-learning agent applications, *Int. J. Appl. Math. Comput. Sci.*, vol. 34, no. 3, pp. 467–483, 2024.

[29] W. Tong, B. Liu, Z. Li, J. Lin, and X. Jin, Intrusion detection method of industrial control network based on lightgbm, in *Proc. 2021 Int. Conf. Communications, Information System and Computer Engineering*, Beijing,

China, 2021, pp. 631–635.
[30] S. Hosseini and M. Azizi, The hybrid technique for DDoS detection with supervised learning algorithms, *Comput. Network.*, vol. 158, pp. 35–45, 2019.
[31] S. Y. Diaba, T. Anafo, L. A. Tetteh, M. A. Oyibo, A. A. Alola, M. Shafie-Khah, and M. Elmusrati, SCADA securing system using deep learning to prevent cyber infiltration, *Neural Network.*, vol. 165, pp. 321–332, 2023.
[32] J. Gao, L. Gan, F. Buschendorf, L. Zhang, H. Liu, P. Li, X. Dong, and T. Lu, Omni SCADA intrusion detection using deep learning algorithms, *IEEE Internet Things J.*, vol. 8, no. 2, pp. 951–961, 2021.
[33] D. Shou, C. Li, Z. Wang, S. Cheng, X. Hu, K. Zhang, M. Wen, and Y. Wang, An intrusion detection method based on attention mechanism to improve CNN-BiLSTM model, *Comput. J.*, vol. 67, no. 5, pp. 1851–1865, 2024.
[34] L. Faramondi, F. Flammini, S. Guarino, and R. Setola, A hardware-in-the-loop water distribution testbed dataset for cyber-physical security testing, *IEEE Access*, vol. 9, pp. 122385–122396, 2021.
[35] P. Gogoi, M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, Packet and flow based network intrusion dataset, in *Proc. 5th Int. Conf. Contemporary Computing*, Noida, India, 2012, pp. 322–334.
[36] A. Ayodeji, Y. K. Liu, N. Chao, and L. Q. Yang, A new perspective towards the development of robust data-driven intrusion detection for industrial control systems, *Nucl. Eng. Technol.*, vol. 52, no. 12, pp. 2687–2698, 2020.
[37] S. Kriaa, L. Pietre-Cambacedes, M. Bouissou, and Y. Halgand, A survey of approaches combining safety and security for industrial control systems, *Reliab. Eng. Syst. Saf.*, vol. 139, pp. 156–178, 2015.
[38] M. Mantere, M. Sailio, and S. Noponen, Network traffic features for anomaly detection in specific industrial control system network, *Future Internet*, vol. 5, no. 4, pp. 460–473, 2013.
[39] Y. Fang, Y. Yao, X. Lin, J. Wang, and H. Zhai, A feature selection based on genetic algorithm for intrusion detection of industrial control systems, *Comput. Secur.*, vol. 139, p. 103675, 2024.
[40] M. Conti, D. Donadel, and F. Turrin, A survey on industrial control system testbeds and datasets for security research, *IEEE Commun. Surv. Tutorials*, vol. 23, no. 4, pp. 2248–2294, 2021.
[41] A. Dehlaghi-Ghadim, M. H. Moghadam, A. Balador, and H. Hansson, Anomaly detection dataset for industrial control systems, *IEEE Access*, vol. 11, pp. 107982–107996, 2023.
[42] J. Shao, G. Zeng, K. Lu, G. Geng, and J. Weng, Automated federated learning for intrusion detection of industrial control systems based on evolutionary neural architecture search, *Comput. Secur.*, vol. 143, p. 103910, 2024.
[43] J. A. Alzubi, O. A. Alzubi, I. Qiqieh, and A. Singh, A blended deep learning intrusion detection framework for consumable edge-centric IoMT industry, *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 2049–2057, 2024.
[44] N. Farnaaz and M. A. Jabbar, Random forest modeling for network intrusion detection system, *Proc. Comput. Sci.*, vol. 89, pp. 213–217, 2016.
[45] K. Ren, S. Yuan, C. Zhang, Y. Shi, and Z. Huang, CANET: A hierarchical CNN-attention model for network intrusion detection, *Comput. Commun.*, vol. 205, pp. 170–181, 2023.
[46] V. Hnamte, H. Nhung-Nguyen, J. Hussain, and Y. Hwa-Kim, A novel two-stage deep learning model for network intrusion detection: LSTM-AE, *IEEE Access*, vol. 11, pp. 37131–37148, 2023.
[47] A. Li and S. Yi, Intelligent intrusion detection method of industrial internet of things based on CNN-BiLSTM, *Secur. Commun. Network.*, vol. 2022, p. 5448647, 2022.
[48] L. D. Manocchio, S. Layeghy, W. W. Lo, G. K. Kulatilleke, M. Sarhan, and M. Portmann, FlowTransformer: A transformer framework for flow-based network intrusion detection systems, *Expert Syst. Appl.*, vol. 241, p. 122564, 2024.

Ding Zhou received the BS, MS, and PhD degrees in control engineering, systems engineering, and aeronautical and astronautical science and technology from Northwestern Polytechnical University, Xi'an, China, in 2014, 2017, and 2020, respectively. He is currently a system architecture research fellow at Endogenous Security Research Center, Purple Mountain Laboratories, Nanjing, China. He has published more than 15 papers in journals and conferences, including *IET Control Theory & Applications*, *International Journal of Control, Automation and Systems*, and *Alexandria Engineering Journal*. He has received two provincial and ministerial-level awards for scientific and technological progress. His research interests include the security control of cyber-physical systems, industrial control network, mobile sensor network, and robust control strategies for multi-agent system.

Bo Liu received the PhD degree from Southeast University, China, in 2007. She is currently a full professor at Endogenous Security Research Center, Purple Mountain Laboratories, Nanjing, China. She is the principal of several national projects in China. She has published over 80 papers in major journals and international conferences, such as Annual Meeting of the Association for Computational Linguistics, International World Wide Web Conference (WWW), ACM Conference on Knowledge Discovery and Data Mining, *IEEE/ACM Transactions on Networking*, *IEEE Transactions on Neural Networks and Learning Systems*, etc. Her research interests include network anomaly detection, network behavior analysis, user behavior analysis in social network, trustworthiness of large language model, and social influence.

Jiuxin Cao received the PhD degree from Xi'an Jiaotong University, China, in 2003. He is currently a full professor at School of Cyber Science and Engineering, Southeast University, China. He is the leader of the Jiangsu Provincial Key Laboratory of Computer Network Technology. He is the principal of several national projects in China. He has published more than 100 papers and most of them have been published in prestigious journals and conferences including International Joint Conference on Artificial Intelligence, ACM International Conference on Multimedia, WWW, *IEEE Transactions on Computational Social Systems*, etc. His research interests include network security, computational society, and cross-modality media fusion.

Xinwen Fu received the BS degree from Xi'an Jiaotong University, China, in 1995, the MS degree in electrical engineering from University of Science and Technology of China, China, in 1998, and the PhD degree in computer engineering from Texas A&M University, USA, in 2005. He is currently a professor at School of Computer & Information Sciences, University of Massachusetts Lowell, Lowell, MA, USA. He has published at prestigious venues including the top four cybersecurity conferences. His research was reported by various media such as Wired and aired on Cable News Network and CCTV 10. His research interests include computer security and privacy and digital forensics.

Shaoxun Liu received the BS degree in automation from Henan University of Technology, China, in 2005 and the MS degree in measuring and testing technologies and instruments from Yanshan University, China, in 2008. He is currently a project leader and senior engineer at Endogenous Security Research Center, Purple Mountain Laboratories, China. He has authored 5 journal papers in *Computer Science*, *Journal of Electronics & Information Technology*, and *Internet of Things Technologies*, and holds 15 invention patents. His research interests include cyber-physical system security, resilience-empowered unmanned systems, and embodied intelligence cybersecurity.

Xin Yan received the BS degree in computer engineering from Northwestern University, China, in 2021. She is currently pursuing the MS degree in cyber security at Southeast University, China. Her research interests include industrial control network.