DOI Link: https://doi.org/10.61586/mCnhl
Vol.70, Issue.10, Part.1, Oct 2025, PP.2-28

Heterogeneous Graph Neural Network-based Network Security Situation Assessment for In-Vehicle Network

Xu Zhong[1,2], Xinglong Huang[2], Xiangyang Wang[1], Wei Luo[1], Shanpeng Lai[1], Liangyu Kang[2] and Zhenxing Dong[2],*

[1] State Key Laboratory of Intelligent Vehicle Safety Technology, Chongqing Changan Automobile Co.,Ltd., Chongqing 401133, China
[2] School of Cyberspace Security and Information Law, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Received Feb 2025 Accepted Sep 2025 Published Oct 2025

ABSTRACT

In response to the issues in current neural network-based cybersecurity situation assessment methods for In-Vehicle Network (IVN), which ignore the associations between situation entities and lack representation learning of spatial structural information in situation features—thereby limiting the accuracy of situation assessment—this paper proposes a situation assessment scheme based on heterogeneous graph neural networks. This scheme divides the graph into multiple meta-path sequences through meta-paths and uses Bidirectional Long Short-Term Memory (Bi-LSTM) networks and attention mechanisms to obtain attribute features of each node. By combining these with the structural features obtained from Metapath2Vec, node feature representations are acquired. The overall representation of the vehicle nodes is then generated using an attention mechanism for situation assessment. Both theoretical analysis and experimental results indicate that this method can accurately evaluate the level of security situational level of IVN.

KEYWORDS

Heterogeneous network, Situation assessment, Graph neural network, Meta-path.

I. INTRODUCTION

In recent years, with the rapid increase in the number of vehicles, the importance of active safety and cyber protection for automobiles has become increasingly prominent [1]. Research and development in related fields of the Internet of Vehicles (IoV) have received widespread attention. The IoV greatly enhances traffic efficiency and driving safety by interconnecting vehicles, road infrastructure, pedestrians, and other devices. However, the high degree of interconnectedness and complexity of the IoV also bring significant cybersecurity risks, such as malicious intrusions, data tampering, and privacy leaks [2]. Therefore, comprehensively understanding the overall security status of the In-Vehicle Network (IVN) is a hot issue that needs to be urgently addressed. Network Security Situation Assessment (NSSA) is one of the most common and effective solutions. It provides decision support for cybersecurity managers by real-time monitoring of cybersecurity conditions to understand the current security situation [3]. NSSA is an important part of cyber situational awareness, capable of discovering potential security risks in the network at the earliest opportunity, fully assessing the impact of these hidden dangers, helping cybersecurity managers grasp the current network situation, and taking preventive measures against threats before cyber attacks occur [4].

Significant progress has been made in research on cybersecurity situation assessment, but so far, NSSA has not formed a systematic theoretical system. Traditional methods based on mathematical logic models and knowledge reasoning models can integrate the overall situation of the network to some extent and provide decision-making recommendations for network managers [5]. However, with the advent of the big data era, some traditional methods are unable to meet the requirements of processing large amounts of network traffic and attacks, often relying too much on expert evaluation and logical reasoning. Although their accuracy is relatively high, they find it difficult to assess situations according to the real-time state of the network. Some deep learning-based models, while efficient, do not consider the relationship between various devices during the situation assessment process and lack representation learning of spatial structural information in situational features, leading to limitations in model accuracy.

With the development of Graph Neural Networks (GNNs), more and more studies have shown that GNNs exhibits superior performance in handling data with complex relational relationships [6]. GNNs can effectively represent and process graph-structured data, capturing global and local features within graphs through message passing and feature aggregation between nodes. However, traditional GNNs methods have limitations when dealing with heterogeneous

graphs and cannot effectively capture the complex relationships between different types of nodes and edges. In the task of cybersecurity situation assessment for connected vehicles, nodes within the connected car network (such as devices, attacks, channels, etc.) and edges form complex graph structures. Using heterogeneous graph neural networks based on meta-paths to aggregate information within paths and between paths layer by layer, obtaining overall semantic representations of each path, and combining Metapath2Vec to capture the structural characteristics of the entire network. In summary, this method can effectively integrate semantic features and structural information of heterogeneous networks, obtain more comprehensive node feature representations, and efficiently handle the complex relationships and dynamic changes in connected cars.

Aiming at the problems existing in current situation assessment models, such as not considering network structure and depending on prior knowledge, this paper proposes a IVN security situation assessment method based on heterogeneous graph neural networks. This method addresses two core issues in IVN security situation assessment task: a) enabling neural networks to distinguish specific attack patterns; b) extracting contextual semantic information from attack chains and learning the structural patterns of different attack chains. The main contributions of this paper are as follows:

This paper constructs the onboard network topology and various types of attacks targeting the vehicle end into a heterogeneous graph, which reflects the attack information of each node within the onboard network and attack chains. Through the method based on meta-paths, the graph can be divided, assigning vulnerability exploitation and communication attacks to corresponding meta-paths, thereby distinguishing specific attack patterns.;

This paper proposes a situation assessment method based on heterogeneous graph neural networks. In terms of attribute feature extraction, it generates path sequences based on meta-paths and uses Bidirectional Long Short-Term Memory (Bi-LSTM) networks and attention mechanisms for semantic feature extraction and aggregation. For structural feature extraction, Metapath2Vec performs random walk on the graph to acquire the structural characteristics of the graph, capturing the structural differences among various attacks. Combining the above two methods leads to more comprehensive node feature representations.

II. Related Work

NSSA is one of the key technologies for ensuring the security of network systems. Numerous NSSA methods have been proposed, including those based on

mathematical statistics, knowledge reasoning, and machine learning. Early cybersecurity situation assessments primarily relied on mathematical statistical methods such as weight analysis, analytic hierarchy process (AHP), and set pair analysis. These methods achieve an overall assessment of cybersecurity status by constructing evaluation functions that comprehensively consider various factors affecting cybersecurity. Jing et al. [7] constructed an improved threat database and developed a data logic method named Carval to automatically infer multi-stage attack paths within IVN and calculate risk values. This scheme enhanced the threat database and partially automated the risk assessment process but still needed to address issues such as asset recognition difficulty, lack of objective standards, and low automation levels. Tsai et al. [8] proposed a wireless risk assessment approach containing a risk model and assessment measures based on AHP. The model processes network dynamics across four layers, accurately assessing and distinguishing risks among different networks, but it is complex and requires high professional expertise. Li et al. [9] introduced a method combining fuzzy logic with AHP to assess the situational status in digital control rooms, particularly suitable for data-scarce scenarios. This method uses AHP to determine the weights of risk factors and employs fuzzy logic to reduce subjective judgment from experts. However, it heavily relies on accurately setting risk factor weights and rule bases and requires professional knowledge. Nevertheless, cybersecurity situation assessment methods based on mathematical statistics overly depend on prior knowledge or expert experience, are significantly influenced by human factors, and lack reasonable quantitative standards.

Knowledge reasoning-based assessment methods rely on experts' knowledge and experience to construct evaluation models and use logical reasoning to analyze the overall security situation of the network. Wang et al. [10] proposed a middleware approach combining attack graphs with Hidden Markov Models (HMM) to represent network conditions using modified dependency attack graphs and input them into HMMs to evaluate attack states, employing heuristic algorithms to infer optimal strategies. This method has a certain degree of dynamic adaptability. Li et al. [11] presented a new situational assess-ment method centered around a Markov game model, utilizing Nash equilibrium points in games to judge network security, fully considering the interactions between attackers and defenders, being closer to reality, and accurately evaluating the cybersecurity situation. Poolsappasit et al. [12] introduced a cybersecurity risk management framework based on Bayesian networks aimed at addressing limitations of traditional risk models in effectively inferring causal dependencies

between network states and resource availability. This framework enables system administrators to quantify the likelihood of different levels of cyber intrusions, providing support for developing security mitigation and management plans.

Machine learning methods automatically learn patterns of sample features through algorithms and train models to evaluate the cybersecurity situation. Zhao et al. [13] proposed a comprehensive approach for cybersecurity situational awareness in big data environments by establishing a cybersecurity situational awareness index system and optimizing the multi-source data processing procedure using parallel reduction algorithms based on attribute importance matrices, while adopting particle swarm optimization algorithms to improve wave neural network parameters for calculating situational values, achieving rapid convergence and good fitting effects. Yang et al. [14] used deep autoencoders to establish the AEDNN model, combined adversarial training and undersampling weighted algorithms, although improving accuracy in large-scale wireless network environments, did not fully consider the timeliness of data. Liu et al. [15] proposed a novel cybersecurity situational assessment method based on bidirectional simplified memory units capable of effectively learning and describing long-term and short-term dependencies in time-series data, enhancing the accuracy and efficiency of cybersecurity situational assessment. Zhao et al. [16] proposed a long-term and short-term cybersecurity situation prediction method based on attention mechanisms, integrating improved bidirectional long-short term memory networks, vector autoregression, and multiscale convolution with branch attention mechanisms, achieving higher prediction accuracy than existing methods but potentially increasing computational costs due to complex model structures. Machine learning-based situational assessment methods possess advantages of being efficient and accurate without depending on expert experience, yet these models often contain numerous parameters, leading to longer training times.

Graph Neural Networks feature flexible structures and update mechanisms, capturing various dependencies within networks by aggregating features of nodes and their neighboring nodes, thereby generating new node representations that well express the structural characteristics of the network itself. Existing GNNs methods not only provide powerful tools for node representation learning but also offer new perspectives for cybersecurity situational awareness in graph analysis tasks, especially concerning how to utilize these models to handle complex network relationships. Models like GCN [17] and GAT [18], through flexible message-passing and information aggregation mechanisms, can effectively

capture node relationships in complex network structures, providing strong analytical capabilities for cybersecurity situational awareness and laying the foundation for subsequent research. RGCN [19] considers various relationships in heterogeneous networks, fusing multiple relations into node representations, ingeniously applying homogeneous graph neural networks to heterogeneous networks. However, RGCN faces the problem of excessive smoothing when dealing with various complex heterogeneous relationships, meaning node representations may lose original information, resulting in decreased model performance. Yu et al. [20] proposed a relation-aware heterogeneous graph neural network, R-HGNN, which significantly improves the quality of node representation learning through specially designed graph convolution components, cross-relation message passing modules, hierarchical relation representation learning, and semantic fusion modules. Its drawback lies in its relatively complex model structure, leading to higher computational costs during the training process and limited scalability for heterogeneous graphs. Wang et al. [21] introduced a novel heterogeneous graph neural network based on hierarchical attention mechanisms, capable of aggregating features from meta-path neighbors hierarchically by learning the importance of nodes and different meta-paths, generating node embeddings. Experimental results demonstrated the superiority of this model over existing techniques and showcased its potential for good interpretability in graph analysis, though the computational complexity of hierarchical attention mechanisms may affect the efficiency of training and inference on large-scale graph data. Zhang et al. [22] proposed the HetGNN model, sampling strongly correlated heterogeneous neighbors through a random walk restart strategy, designing a dual-module neural network architecture to encode heterogeneous content feature interactions, and aggregating attribute embeddings of different types of neighbors, ultimately surpassing state-of-the-art techniques in multiple graph mining tasks. Its limitation is the over-reliance on node features, possibly neglecting structural information. Hong et al. [23] introduced HetSANN, directly encoding the structural information of HIN without requiring meta-paths, aggregating multi-relation information through low-dimensional entity space projection and attention mechanisms along with three expansion methods, achieving richer representations. Experiments on three public datasets significantly outperformed existing solutions, though its assumption about low-dimensional entity space limits its performance in scenarios involving multilevel relationships and high-dimensional information. Yang et al. [24] proposed SeHGNN, a simplified and efficient heterogeneous graph neural

network using lightweight mean aggregators and Transformer-based semantic fusion modules to effectively utilize structural and semantic information, featuring simple structure, high accuracy, and fast training but relatively high computational costs. Dong et al. [25] proposed two scalable models, Metapath2Vec and its enhanced version Metapath2Vec++, constructing nodes' heterogeneous neighborhoods through formalized meta-paths random walk and utilizing heterogeneous skip-gram models for node embedding, further modeling both structural and semantic associations simultaneously. Results showed that random walk-based network models could effectively preserve the structural and semantic information of heterogeneous networks, but they neglected node attribute features, potentially leading to inferior performance in regression tasks.

III. Methodology

A. Overview

This paper proposes a vehicular network situation assessment method based on a heterogeneous graph neural network, with the overall architecture shown in Figure**Fig. 1.** Overall process of IVN risk assessment.. The proposed scheme consists of three core modules: the data preprocessing module, the heterogeneous graph learning module, and the result prediction module.

In the data preprocessing stage, the vehicular network topology and network attack alerts are constructed and linked to generate an attribute-free graph $G(V,E,T)$; the attribute graph $G(V,E,T,X)$ is generated using a graph attribute construction module; the attribute graph $G(V,E,T,X)$ is then serialized based on meta-paths to generate meta-path sequence data containing contextual information, aiming to extract relationship patterns between nodes.

In the heterogeneous graph learning stage, the preprocessed serialized data is aligned and fed into a Bi-LSTM-based semantic representation module to obtain node attribute features A; the attribute-free graph $G(V,E,T)$ is input into a Metapath2Vec-based structural representation module to extract node structural features S; through the feature fusion module, the attribute features A and structural features S are integrated to obtain the complete representation F of device nodes, where F_i represents the feature of the i -th device node.

In the result prediction stage, an attention mechanism-based device feature aggregation module dynamically assigns weights to each device node feature F_i, computing the overall feature representation R of device nodes.

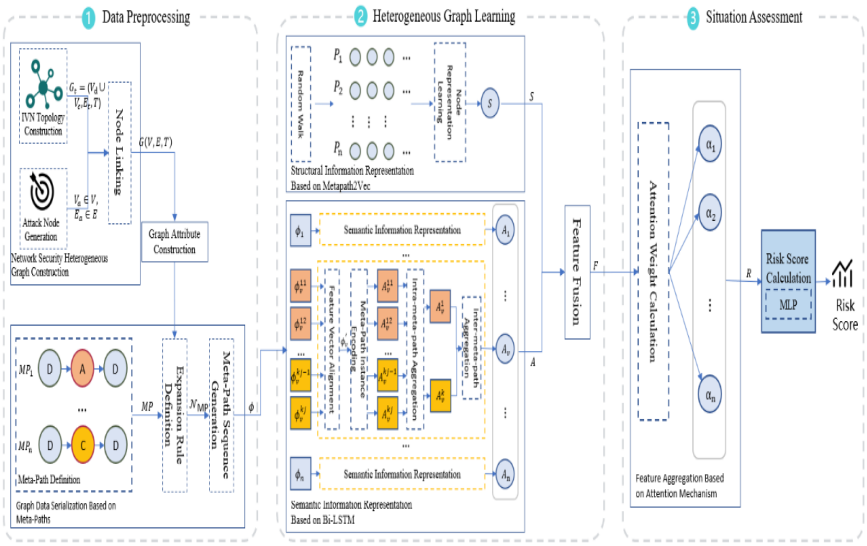

Fig. 1. Overall process of IVN risk assessment.

Finally, the overall feature representation is fed into a multilayer perceptron to predict the system risk value, thereby comprehensively assessing the current situation of the vehicular network.

B. Data Preprocessing

This section proposes a network security heterogeneous graph data construction method for generating vehicular network heterogeneous graph data. First, based on the vehicular network topology and attack alerts, the vehicular network graph data is constructed to obtain an attribute-free graph. Then, using a graph attribute construction method, an attribute graph is obtained. Finally, based on predefined meta-path templates, the attribute graph undergoes meta-path serialization processing, generating multiple meta-path sequences for each node.

1) Network Security Heterogeneous Graph Construction:

This paper proposes a vehicular network graph data construction method aimed at generating graph datasets for model training. The method is based on a central gateway architecture, abstracting devices, channels, and their interaction relationships within the vehicular network while incorporating attack alert information to describe potential network threats. Finally, a vehicular network

security heterogeneous graph is constructed through node linking. The specific construction process is as follows:

● IVN Topology Construction: By abstracting physical devices and communication channels in the network topology, a topological structure$G_t = (V_d \cup V_c, E_t, T)$is constructed to reflect the interaction relationships between devices and communication channels in the vehicular network. The physical devices in the vehicular network are abstracted as the node setV_d, and the communication channels are abstracted as the node set V_c. If a device $d_i \in V_d$interacts with a channel $c_j \in V_c$ through wireless or wired communication, an edge $e_{ij} \in E_t$ is added to the graph to represent the connection relationship. The set T is used to represent the node types.

● Attack Node Generation: Attack nodes are introduced to describe security threats in the vehicular network, ensuring that the constructed graph data not only reflects the normal network topology but also captures attack behaviors and their impact range. The attack entity is represented by the attack node set V_a, while the attack edge set E_a describes how attacks affect devices or channels. Attack node generation depends on alert data, determining affected targets based on security logs.

● Node Linking: Attack nodes are connected to the vehicular network topology to form the attribute-free graph$G(V, E, T)$. Specifically, the attack edge set E_a records how attacks impact devices or channels. If the attack target is a device, the attack node connects to the device nodeV_d; if the attack target is a communication channel, the attack node connects to the channel nodeV_c. Ultimately, the complete vehicular network security heterogeneous graph can be represented as$G = (V_d \cup V_c \cup V_a, \ E_t \cup E_a, T)$, i.e.,$G(V, E, T)$.

2) Graph Attribute Construction:

● The graph attribute construction module introduces two steps—node attribute setting and graph label construction—to compute feature values and graph labels for the attribute-free graph$G(V, E, T)$. This results in the attribute graph$G(V, E, T, X)$, which provides effective label information for the semantic representation module. The specific steps are as follows:

● Node Attribute Setting: This step processes the attribute-free graph $G(V, E, T)$ by generating node attributes to describe the feature information of different types of nodes, providing input data for model training. First, the threat level attribute $T(n_i)$ of attack nodes is calculated using the Common Vulnerability Scoring System (CVSS). Then, the node importance attribute $I(n_i)$ is assigned based on the node's functional significance in the network.

The attribute $T(n_i)$ is used as a feature for both channel and device nodes, while $I(n_i)$ is used as a feature for attack nodes. These attributes form the attribute set$X$, resulting in the attribute graph$G(V, E, T, X)$. Each node and its attributes can be represented as a node attribute set: (Node ID, Node Type, Node Importance/Node Threat Level). Meanwhile, the risk value of an individual device node is computed using (1) and (2):

$$R_{ni} = I_{ni} + T_{ni} \cdot EF_{ni} \quad (1)$$

$$EF_{ni} = 1 + \beta \cdot C_{ni-1} \quad (2)$$

where $I(n_i)$ represents the importance value of noden_i, indicating the relative significance of device and channel nodes in the graph. $T(n_i)$ denotes the total threat level of noden_i, which is the sum of the threat levels of all attack nodes linked ton_i. The enhancement factor EF(n_i) reflects the cumulative effect of the number of paths connecting the node to attack nodes. C_{n_i} represents the number of attack nodes linked to noden_i, indicating how many attack nodes are directly or indirectly affecting it.

● Graph Label Construction: Based on the risk status of individual nodes, the total risk value of the entire graph is computed using (3) and (4), which serves as the graph label:

$$R_{total} = \sum_{i=1}^{N} R\,(n_i) + \alpha \cdot AB \quad (3)$$

$$AB = \sum_{n_i, n_j \in E} \left(R(n_i) + R(n_j)\right) \quad (4)$$

● where N represents the total number of nodes in the graph, $\alpha$is the additional risk coefficient between adjacent nodes, reflecting the impact of node connections on the overall risk. AB represents the additional risk contribution from adjacent nodes. The set E denotes the edges in the graph, describing the node connections, while $R(n_i)$ and $R(n_j)$ are the risk values of nodes n_i andn_j, respectively.

Using this approach, the proposed method effectively abstracts key entities

and security scenarios in the vehicular network, generating the attribute graph $G(V, E, T, X)$ as input for the semantic representation model. Moreover, the model incorporates the attack enhancement factor AB to capture the cumulative effect of different attack paths in complex attack scenarios, as defined in (3) and (4). By integrating node importance and threat levels, the model can further quantify the potential risks of individual nodes and assess the overall network vulnerability. This graph data construction method, through multi-dimensional relationship modeling, captures critical security factors in complex networks and provides a comprehensive representation of the interactions between device nodes, channel nodes, and attack-type nodes in the vehicular network environment.

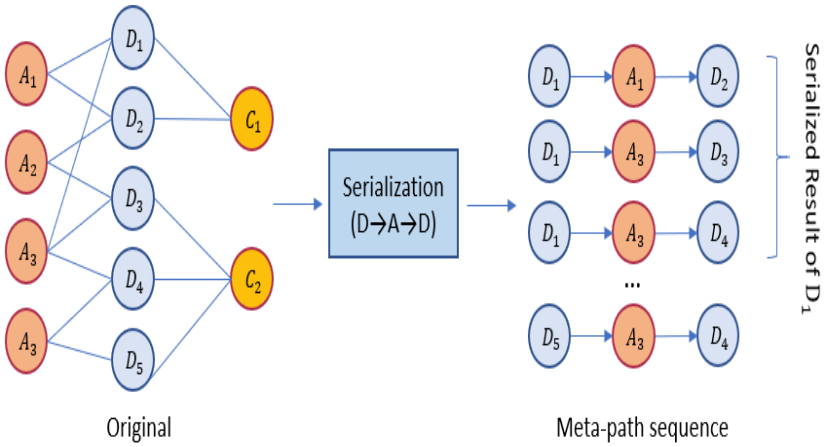

Fig. 2. Meta-path serialization.

3) Graph Data Serialization Based on Meta-Paths:

This module serializes the data to support semantic information representation, mainly involving two steps: meta-path selection and graph data serialization.

Meta-path selection is crucial for guiding the selection of neighboring nodes based on predefined complex relationships between nodes. This process helps the model capture semantic information from heterogeneous graphs and focuses on specific attack patterns to improve regression performance. The meta-paths

chosen in this study are shown in Table 2.

Based on the predefined meta-path templates, graph data serialization follows the expansion rule in (5). The process involves sequentially querying each node's adjacent nodes and filtering those that meet the meta-path requirements. This continues until the sequence fully satisfies the definition of the meta-paths, thereby generating one or more meta-path sequences with specific semantic relationships. These sequences serve as input to the Bi-LSTM model, enabling it to extract relationship patterns between nodes and capture connections between attacks.

The core of constructing meta-path sequences is to traverse the heterogeneous network $G(V, E, T, X)$ according to the defined meta-paths and generate the meta-path sequences for each node. The serialization process is illustrated in Figure 2. The detailed steps are as follows:

● Meta-Path Definition: The meta-path set $MP = \{MP_1, MP_2, \dots, MP_n\}$ is a set of path patterns defined on a heterogeneous graph. Each meta-path MP_i is represented as an ordered sequence of node types:$MP_i = (t_{i1} \to t_{i2} \to \cdots \to t_{ik})$, where t_{ij} denotes the node type at position j in the path (Device, Attack, or Channel).

● Expansion Rule Definition: Given an initial nodev_1, where$v_1 \in V_{t_1}$, the expansion starts from the initial type t_{i1} of MP_i and progressively extends to the terminal type t_{ik} according to the recursive rule in (5):

$$N_{\mathrm{MP}}(v_i) = \{ v_j \mid A[v_i, v_j] = 1, T[v_j] = t_{i+1} \} \tag{5}$$

where$v_i$represents the current node, $A[v_i, v_j] = 1$indicates the existence of an edge between v_i andv_j, and $T[v_j] = t_{i+1}$ ensures that the type of node v_j matches the next step requirement of meta-pathMP_i.

● Meta-Path Sequence Generation: Starting from the beginning of the meta-path $MP_i = (t_{i1} \to t_{i2} \to \cdots \to t_{ik})$, a complete node sequence Seq is constructed step by step under the constraint in (6):

$$Seq = \{v_1, v_2, \dots, v_k\},\ v_i \in V_{t_i},\ (v_i, v_{i+1}) \in E \tag{6}$$

where the recursive rule of the sequence is$v_{i+1} \in N_{MP}(v_i)$, and each extension step selects only the nodes that conform to the meta-path requirements. Ultimately, the meta-path sequence set ϕ_i is obtained, which represents the meta-path sequence collection of node i:$\phi_i = \{\phi_i^1, \phi_i^2, \dots, \phi_i^k\}$.

C. Heterogeneous Graph Learning

Heterogeneous Graph Learning aims to utilize heterogeneous graph data and the preprocessed meta-path sequences to obtain feature representations describing the in-vehicle device nodes through Metapath2Vec and a Bi-LSTM-based heterogeneous graph model. This module consists of three main parts: structural information representation, semantic information representation, and feature fusion. In the structural information representation, the vector representation of random walk paths is calculated based on the meta-path random walk method on the heterogeneous graph, which extracts the structural features of nodes. In the semantic information representation, the features of different types of nodes in the sequence are aligned, and contextual information is extracted and aggregated to obtain the semantic features of the nodes. Feature fusion involves concatenating the structural features and semantic features to obtain the complete feature representation of the nodes, providing support for subsequent analysis.

1) Structural Information Representation Based on Metapath-2Vec:

This section uses Metapath2Vec to extract structural features of device nodes to characterize their topological relationships in the vehicular network and capture structural patterns of different types of attacks. Since the meta-path is designed as a relationship path between device nodes, Metapath2Vec learns the structural features of nodes in the graph (including different relationships defined by the meta-paths) and calculates the structural similarity between device nodes, allowing topological information to be better integrated into the situational awareness task. The specific steps are as follows:

- Node Sequence Generation Based on Random Walk: A set of node sequences is generated through random walk, reflecting the true relationships between device nodes and other nodes, thereby providing input data for node representation learning. First, the unstructured graph $G(V, E, T)$ and predefined meta-path templates are used as input. A meta-path-based random walk strategy is applied to generate multiple node visit sequences on the graph. During the random walk, the walk only proceeds along edges that conform to the meta-path patterns.
- Node Representation Learning: Using the node sequences generated by the random walk, the low-dimensional vector representation of each node is learned, thereby capturing the structural similarity and topological features of the nodes. The node sequences generated by the random walk can be regarded as a set of node vocabulary with contextual relationships. The skip-gram algorithm is used for learning, where the goal is, based on a central node v, to

predict its neighboring nodesc_t. That is, the representation of a node in the generated sequence should effectively predict its neighboring nodes in the walk path. The skip-gram model optimizes the node embedding by maximizing the objective function shown in (7). After multiple rounds of training, the skip-gram algorithm learns a low-dimensional vector representation for each device node, such that similar devices are closer in the vector space, enabling effective capture of the structural similarity of device nodes and the topological features of different attack patterns.

$$\underset{\theta}{\operatorname{argmax}} \sum_{v \in V} \sum_{t \in T_v} \sum_{c_l \in N_t(v)} log\, p(c_t \mid v; \theta) \tag{7}$$

where $N_t(v)$ represents the set of nodes, with vertex v being of type$t$, and $p(c_t \mid v; \theta)$ is the softmax function. By maximizing the above objective function, the node feature S_v is obtained, and the final output is the structural feature set of device nodes, $S = \{S_1, S_2, S_3, \ldots, S_n\}$, as part of the input for feature fusion.

2)Semantic Information Representation Based on Bi-LSTM:

This module uses Bi-LSTM to represent the semantic information of the meta-path sequences constructed in the previous section. By leveraging the ability of Bi-LSTM to handle sequential data, it models the sequential dependencies within the meta-path sequences to understand the roles of each node and their interactions in the meta-path instances. Figure 3 illustrates the overall process of semantic information representation, and the specific steps are as follows:

- Feature Vector Alignment: The feature vectors of nodes with different dimensions are mapped to a unified vector space using a vector projection method to match the input format for encoding the meta-path instances. The specific calculation method is shown in (8):

$$e_v = W_A \cdot x_v^A \tag{8}$$

where $W_A \in \mathbb{R}^{d' \times d_A}$ is the parameterized weight matrix used to project the feature vector of node type A fromd_A dimensions tod'dimensions.

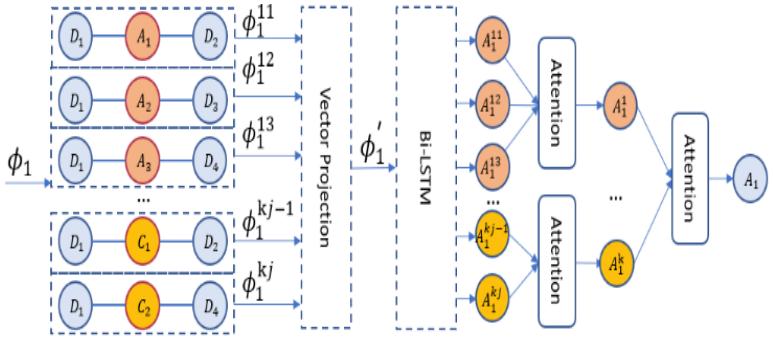

Fig. 3. Semantic information representation process based on Bi-LSTM.

● Meta-Path Instance Encoding: The Bi-LSTM is used to encode the meta-path instances, modeling the semantic information of the node sequences from both the forward and backward directions. By combining the intermediate node features, the understanding of complex relationships is enhanced, and the semantic representation of the meta-path instance is generated to provide input for path aggregation. The encoding process of Bi-LSTM is shown in (9)-(14):

$$fv_L = \sigma\left(W_f^L \cdot \left[h_{v-1}^L, e_v\right] + b_f^L\right) \tag{9}$$

$$i_L^v = \sigma\left(W_i^L \cdot \left[h_{v-1}^L, e_v\right] + b_i^L\right) \tag{10}$$

$$o_L^v = \sigma\left(W_o^L \cdot \left[h_{v-1}^L, e_v\right] + b_o^L\right) \tag{11}$$

$$C_L^v = tanh(W_C^L \cdot [h_{v-1}^L, e_v] + b_C^L) \tag{12}$$

$$C_v^L = f_v \cdot \tilde{C}_{v-1} + i_v \cdot \tilde{C}_v^L \tag{13}$$

$$h_v^L = o_v \cdot tanh(C_v^L) \tag{14}$$

where e_v represents the heterogeneous encoding of node$v$, andW_f^L, W_i^L, W_o^L, and W_C^L are weight matrices, whileb_f^L, b_i^L, b_o^L, and $b_C^L$are bias vectors. Bi-LSTM combines the forward LSTM and backward LSTM. h_v^Lrepresents the forward and backward encoding results, which are concatenated to form the

encoded representation of the meta-path instances.

● Instance Feature Aggregation within the Same Meta-Path: To integrate the semantic information of multiple instances within the same meta-path, this module uses an attention mechanism to calculate the importance weight of each instance under the same meta-path, generating a comprehensive semantic representation of the meta-path. This reduces redundant information interference and enhances the accuracy of semantic representation. The calculation process is shown in (15)-(17):

$$e_{vi} = \text{LeakyReLU}\left(a^T p_{v_i}\right) \tag{15}$$

$$\alpha_{vi} = \frac{exp(e_{vi})}{\sum_{I=1}^{i} exp(e_{vi})} \tag{16}$$

$$p_v = \sum_{i=1}^{I} \alpha_{vi} \cdot p_{v_i} \tag{17}$$

where p_v^i represents the encoding vector of the i -th meta-path instance guided by the target node$v$, a^T is the feature vector of the meta-path instance, and e_v^i represents the importance of the meta-path instance to the target node. $p_v$is the meta-path encoding vector.

● Instance Feature Aggregation across Different Meta-Paths: To fuse the semantic information of multiple meta-paths and generate a comprehensive node representation, this module uses an attention mechanism to learn the semantic contribution weights of different meta-paths to the target node. A weighted sum is performed to generate the final node semantic vector. The calculation process is shown in (18)-(20):

$$e_v^p = q^T p_v \tag{18}$$

$$\alpha_v^p = \frac{exp(e_v^p)}{\sum_{p=1}^{P} exp(e_v^p)} \tag{19}$$

$$A_v = \sum_{p=1}^{P} \alpha_v^p \cdot p_v \tag{20}$$

where$q^T$is the feature vector of the meta-path, e_v^prepre-sents the weight of

different meta-paths on the target node, and A_v is the vector representation of node v based on semantics. The semantic vector representation of the node obtained through Bi-LSTM and aggregated by the attention mechanism forms the set $A = \{A_1, A_2, \dots, A_n\}$.

By encoding and aggregating the heterogeneous information of nodes, the contextual semantic representation of nodes can be obtained, providing a foundation for subsequent node feature fusion.

3) Feature Fusion

In order to improve the completeness of node repre-sentations, this section performs feature fusion before aggre-gating the device node features. First, by concatenating the attribute features $S \in \mathbb{R}^{n \times d_s}$ and the structural features $A \in \mathbb{R}^{n \times d_A}$ of the node, the final representation of the node is obtained, with a dimension of $\mathbb{R}^{n \times (d_S + d_A)}$. Then, through a learnable feature fusion matrix $W \in \mathbb{R}^{(d_S + d_A) \times d_g}$, the features are fused into a low-dimensional vector F. The calculation process is shown in (21):

$$F = \mathrm{Concat}(S, A)W \tag{21}$$

where $F = \{F_1, F_2, \dots, F_n\}$ represents the set of feature representations of nodes, and Concat denotes the feature concatenation used to fuse the attribute and structural features of the node. Through this method, the comprehensive representation of the device node can be obtained, thereby improving the accuracy of situational assessment.

D. Situation Assessment

To predict the situation status of in-vehicular networks, this section will utilize the node feature representations obtained from the heterogeneous graph learning mentioned above through two modules: feature aggregation and situation assessment. The feature aggregation is based on an attention mechanism to aggregate device feature nodes, forming a feature representation of the entire graph. The situation assessment then uses a Multi-Layer Perceptron (MLP) to output the final risk value based on this feature representation, indicating the overall situation status of the vehicular network.

1) Feature Aggregation Based on Attention Mechanism:

To capture the global feature information of device nodes in the vehicular network, this section adopts an attention-based node feature aggregation method to fuse the node feature representation set $F = \{F_1, F_2, \dots, F_n\}$, thereby

generating the overall feature representation R of the device nodes. This method dynamically assigns attention weights to highlight the importance of key nodes while suppressing the interference of redundant information in the overall representation.

To dynamically learn the correlations between nodes, for each node$v$, the attention weight α_v between node $v$and its neighbor node u is calculated using (22):

$$\alpha_v = \frac{exp\left(\sigma\left(a^{\top} \cdot [W_h \cdot F(v) | W_h \cdot F(u)]\right)\right)}{\sum_{u \in \mathcal{N}(v)} exp\left(\sigma\left(a^{\top} \cdot [W_h \cdot F(v) | W_h \cdot F(u)]\right)\right)} \tag{22}$$

In this formula, the features of node v and its neighboring node u are first transformed and concatenated, and a learnable parameter vector a is used to compute their correlation score. Then, the activation functionσ (LeakyReLU) introduces nonlinearity, allowing the model to capture more complex feature interactions. Finally, the softmax function is applied to normalize the attention scores of node$v$, ensuring that the attention weights$\alpha_v$of neighboring nodes are comparable across the entire neighbor set.

After calculating the attention weight α_v for each node, the overall feature representation of the device nodes is generated through weighted aggregation. The calculation process is shown in (23):

$$R = \sum_{v \in V} \alpha_v \cdot F(v) \tag{23}$$

where R represents the overall feature representation of the device nodes, $V$is the set of nodes containing all nodes in the graph, and$F(v)$ is the feature representation of node v. In this formula, the feature$F(v)$ of a node is weighted and accumulated according to its attention weightα_v, resulting in the overall feature representation R of the device nodes. This process integrates global information and local structural relationships, providing input features for subsequent analysis tasks. This method not only considers the node's own features but also dynamically emphasizes its relationships with other nodes through the attention mechanism, forming a global feature representation of the node.

2) Risk Score Calculate

This section uses the graph-level feature representation R for situational awareness assessment, and this feature representation is input into the MLP, which processes it through multiple fully connected layers. The goal is to predict the continuous value y associated with the network device nodes in the vehicle, which represents the network security risk score. The calculation formula is given in (24):

$$\hat{y} = MLP(G) = \sigma_L(W_L \cdot \sigma_{L-1}(W_{L-1} \cdot \ldots \sigma_1(W_1 \cdot R + b_1) \cdots + b_{L-1}) + b_L) \quad (24)$$

where W denotes the weight matrices of each layer, b denotes the bias terms of each layer, and these parameters are continuously updated during the model's training process. The function σ represents the activation function of each layer, and L is the number of layers in the MLP.

The situational awareness task is essentially a regression task. To optimize the prediction performance, the Mean Squared Error (MSE) is used as the loss function to measure the

difference between the predicted value $\hat{y}$ and the true label y. The loss function calculation is given in (25):

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 \quad (25)$$

where $\hat{y}_i$ is the predicted value for the i -th sample, y_i is the true label value, and N is the number of training samples. By minimizing the loss function, the MLP model will optimize its weights and biases, thereby achieving the best prediction capability. During the training process, the MLP will progressively learn the relationship between the graph-level feature R and the target variable y, ultimately predicting a continuous value $\hat{y}$ as the situational awareness result.

IV. Experiments

A. Experimental Setup

The experimental environment of this paper runs on a device equipped with an i7-12650H processor, 32GB DDR5 memory, and an NVIDIA GeForce RTX 4060 Laptop GPU. The simulations are conducted using Python 3.11, PyTorch

2.11-GPU, and PyGeometric.

In terms of hyperparameter settings: The semantic information expression module uses Bi-LSTM to extract the semantic information of meta-paths and aggregates features using a multi-head attention mechanism. The attention mechanism uses 8 heads, with the attention vector dimension set to 128, and the final aggregated result has a dimension of 512. The structural information expression module uses Metapath2Vec to embed node features into a 512-dimensional space. After fusing the two types of features, an MLP is used to output the final one-dimensional feature for risk prediction. Additionally, the learning rate is set to 0.001, the weight decay coefficient is 0.001, and Dropout is used to prevent overfitting with a dropout rate of 0.5. The Adam optimizer is employed, and the early stopping strategy has a patience value set to 15 epochs to prevent overfitting. Each group of experiments is run 10 times, and the average performance metrics are reported. The maximum number of training iterations is set to 128.

B. Dataset

Using deep learning methods to predict vehicular network security situational awareness is a new research area, and there is currently no publicly available dataset to evaluate the model proposed in this paper. In this experiment, an automated data generation tool was developed to simulate data related to vehicular network security. Combined with vehicular network security threat data and CVSS scores obtained from cutting-edge papers, the tool simulated and constructed thousands of graph data. According to the scheme shown in Figure 3-1, relationships between network entities can be extracted through meta-path semantics and structural analysis. The situational security analysis graph dataset constructed in this paper includes 31,131 network threat entities and 61,018 links.

C. Metrics

To evaluate the model proposed above, three widely used metrics are chosen, as described below:

Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{\epsilon}\sum_{i=1}^{\epsilon}\left(\hat{Y}_i - Y_i\right)^2} \tag{26}$$

Mean Absolute Error (MAE):

$$MAE = \frac{1}{\epsilon}\sum_{i=1}^{\epsilon}\left|\widehat{Y}_i - Y_i\right| \tag{27}$$

Mean Absolute Percent Error (MAPE):

$$MAPE = \frac{1}{\epsilon}\sum_{i=1}^{\epsilon}\left|\frac{\widehat{Y}_i - Y_i}{Y_i}\right| \times 100\% \tag{28}$$

where ϵ represents the number of test samples, $\widehat{Y}_i$ and Y_i represent the predicted value and the true value of the i -th element.

D. Experimental Result

1) Baseline

In this section, the model proposed in this paper is compared with different types of graph embedding models, including classical attribute-based graph embedding models and non-attribute-based graph embedding models. The baseline models are as follows:

- **HAN** [21] is a heterogeneous GNN that learns path-specific node embeddings from different homogeneous graphs based on meta-paths and combines them into a single vector representation for each node using an attention mechanism.
- **Metapath2Vec** [25] is a traditional heterogeneous model that generates node embeddings by providing meta-path guided random walk to the skip-gram model. The model relies on user-specified meta-paths.
- **GCN** [18] is a GNN on homogeneous graphs, where the model performs convolution in the graph Fourier domain. In this paper, node types are treated as features, and the feature dimensions are aligned for testing the GCN.
- **GAT** [17] is a GNN on homogeneous graphs that performs convolution in the graph spatial domain and incorporates an attention mechanism. Similarly, node types are treated as features, and the feature dimensions are aligned for testing the GAT.

As shown in Table I, the experimental results demonstrate that the proposed method outperforms other models in risk assessment, proving its effectiveness and

superiority.

Compared to HAN, the proposed method better utilizes the structural information of heterogeneous graphs and overcomes the issue of HAN ignoring intermediate nodes, improving risk assessment accuracy. Additionally, Metapath2Vec, which relies solely on topological structure without integrating node attribute information, performs poorly in risk evaluation. Homogeneous graph models like GCN and GAT struggle to model multiple node types and relationships, limiting their effectiveness in heterogeneous scenarios. In contrast, the proposed method combines multi-relation modeling with attention mechanisms, enhancing the identification of key risk factors and improving risk assessment accuracy.

TABLE I. COMPARATIVE EXPERIMENT

Method	MAE	RMSE	MAPE
HAN	0.0569	0.0748	33.2907%
Metapath2vec	0.1651	0.1965	145.7664%
GCN	0.0678	0.0842	41.2461%
GAT	0.0601	0.0892	41.4673%
Ours	0.0441	0.0695	28.9332%

2) Encoder Analysis

To validate the impact of different meta-path instance encoders on model performance, this paper designs an ablation experiment. The aim of this experiment is to explore the performance differences of the Bi-LSTM encoder, the average encoder, and the linear encoder in capturing the features of meta-path instances and to determine whether the Bi-LSTM encoder has a significant advantage in handling the complex relationships in heterogeneous graphs, further assessing its contribution to the overall model performance.

The experiment compares three encoders: The Bi-LSTM encoder utilizes sequence modeling to capture the sequential dependencies in the meta-path instances and generates context-aware representations; the average encoder generates representations by calculating the element-wise mean of the node vectors in the meta-path instances; the linear encoder adds a linear transformation on top of the average encoder to enhance expressive power. By comparing the performance of these three encoders under the same experimental setup, this paper can comprehensively evaluate their strengths and weaknesses in graph regression

tasks. The specific experimental results are shown in Table II.

TABLE II. ENCODER ANALYSIS EXPERIMENT

Encoder	MAE	RMSE	MAPE
Bi-LSTM	0.0569	0.0748	33.2907%
Linear	0.1651	0.1965	145.7664%
Average	0.0678	0.0842	41.2461%

The results show that the Bi-LSTM encoder performs best in terms of MAE, RMSE, and MAPE, with values of 0.0472, 0.0648, and 30.3512%, respectively, significantly outperform-ing both the linear encoder and the average encoder. This is mainly due to the strong sequence modeling capability of Bi-LSTM, which effectively captures the sequential dependencies and contextual information in the meta-path instances, generating richer feature representations. In contrast, the average encoder disregards node order, and the linear encoder, though enhancing expressiveness, is still constrained by linear transformations and cannot fully exploit complex relationships. Therefore, Bi-LSTM shows a significant advantage in handling the complex structures of heterogeneous graphs.

3) Ablation Study

To validate the impact of the Metapath2Vec module on model performance, this paper compares the performance of the complete model, which includes this module, with the simplified model after removing the module under the same experimental conditions. The comparison of evaluation metrics is shown in Table III. After removing the Metapath2Vec module, the stability of the model decreases. The comparison between the predicted values and actual values before and after removing the Metapath2Vec module is shown in Figure 4. As can be seen from the figure, when processing some data, the deviation between the predicted values (orange) and the actual values significantly increases after removing the Metapath2Vec module, especially at sample indices 5, 10, 15, where the predicted values fluctuate greatly and deviate far from the actual values. In contrast, the predicted values (green) of the complete model are closer to the actual values (blue), capturing the data features more accurately. This demonstrates that the Metapath2Vec module plays a crucial role in extracting path information and enhancing prediction capabilities. This result indicates that the module is essential for the overall performance of the model, and its functionality

cannot be easily replaced, further highlighting its value in complex network analysis tasks.

TABLE III. ABLATION STUDY

Method	MAE	RMSE	MAPE
Non-Metapath2Vec	0.0618	0.0770	44.5098%
Ours	0.0435	0.0580	25.0122%

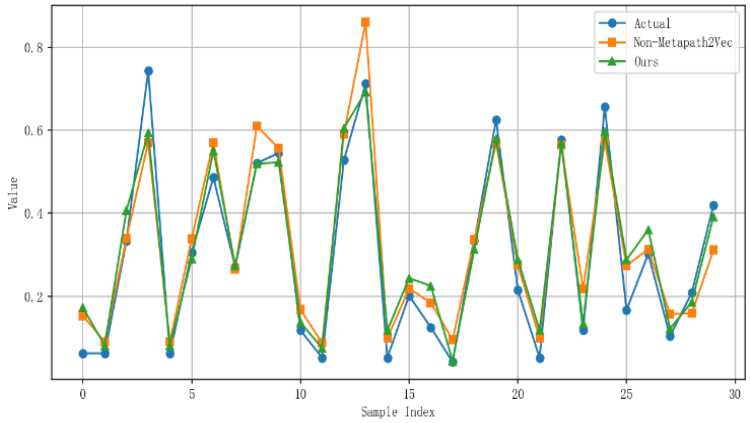

Fig. 4. Ablation experiment results.

V. CONCLUSION

This paper analyzes the background and necessity of IVN situation assessment and summarizes the characteristics of situation assessment in the context of IVN attack scenarios. By constructing a heterogeneous information network for vehicular network attacks, this paper models vulnerability exploitation and channel attack behaviors, providing data support for situation assessment. This paper proposes a IVN security situation assessment method based on heterogeneous graph neural networks. This method can perform feature extraction and node embedding on heterogeneous networks by using Bi-LSTM and Metapath2Vec to extract attribute and structural features. It also combines an attention mechanism to aggregate node features for subsequent graph regression tasks. By constructing serialized data through meta-paths, extracting its semantic information, and incorporating the structural information of heterogeneous graphs

for graph regression tasks, the model can effectively predict the situation value of complex attacks. Experimental results demonstrate that the feature vectors obtained by the proposed model show good prediction performance in the situation assessment task.

VI. Author contributions

Conceptualization, X.W., X.Z. and X.H.; methodology, X.W., X.H. and X.Z.; investigation, X.H.; writing—original draft, X.Z, X.H. and X.W.; writing—review and editing, X.Z, X.W., W.L., S.L.and L.K.; supervision, W.L. and S.L.; funding acquisition, X.W. All authors have read and agreed to the published version of the manuscript.

VII. Funding

This work was supported by the State Key Laboratory of Intelligent Vehicle Safety Technology (Grant No. IVSTSKL-202315).

REFERENCR

[1] Omar H A, Lu N, Zhuang W. Wireless access technologies for vehicular network safety applications[J]. IEEE Network, 2016, 30(4): 22-26.

[2] Sharma S, Kaushik B. A survey on internet of vehicles: Applications, security issues & solutions[J]. Vehicular Communications, 2019, 20: 100182.

[3] Liu X, Yu J, Lv W, et al. Network security situation: From awareness to awareness-control[J]. Journal of Network and Computer Applications, 2019, 139: 15-30.

[4] Zhang J, Feng H, Liu B, et al. Survey of technology in network security situation awareness[J]. Sensors, 2023, 23(5): 2608.

[5] Leau Y B, Manickam S, Chong Y W. Network security situation assessment: a review and discussion[J]. Information Science and Applications, 2015: 407-414.

[6] Zhang Z, Cui P, Zhu W. Deep learning on graphs: A survey[J]. IEEE Transactions on Knowledge and Data Engineering, 2020, 34(1): 249-270.

[7] Jing P, Cai Z, Cao Y, et al. Revisiting Automotive Attack Surfaces: a Practitioners' Perspective[C]//2024 IEEE Symposium on Security and Privacy (SP). IEEE Computer Society, 2023: 80-80.

[8] Tsai H Y, Huang Y L. An analytic hierarchy process-based risk assessment method for wireless networks[J]. IEEE Transactions on Reliability, 2011, 60(4): 801-816.

[9] Li P, Zhang L, Dai L, et al. An assessment method of operator’s situation awareness reliability based on fuzzy logic-AHP[J]. Safety Science, 2019, 119: 330-343.

[10] Wang S, Zhang Z, Kadobayashi Y. Exploring attack graph for cost-benefit security hardening: A probabilistic approach[J]. Computers & security, 2013, 32: 158-169.

[11] Li X, Lu Y, Liu S, et al. Network security situation assessment method based on Markov game model[J]. KSII Transactions on Internet and Information Systems (TIIS), 2018, 12(5): 2414-2428.

[12] Poolsappasit N, Dewri R, Ray I. Dynamic security risk management using bayesian attack graphs[J]. IEEE Transactions on Dependable and Secure Computing, 2011, 9(1): 61-74.

[13] Zhao D, Liu J. Study on network security situation awareness based on particle swarm optimization algorithm[J]. Computers & Industrial Engineering, 2018, 125: 764-775.

[14] Yang H, Zeng R, Xu G, et al. A network security situation assessment method based on adversarial deep learning[J]. Applied Soft Computing, 2021, 102: 107096.

[15]Liu Z, Yang C, Liu Y, et al. A BIPMU-based network security situation assessment method for wireless network[J]. Computer Standards & Interfaces, 2023, 83: 103661.

[16]Zhao D, Shen P, Zeng S. ALSNAP: Attention-based long and short-period network security situation prediction[J]. Ad Hoc Networks, 2023, 150: 103279.

[17]Veličković P, Cucurull G, Casanova A, et al. Graph attention networks[J]. arXiv preprint arXiv:1710.10903, 2017.

[18]Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.

[19]Schlichtkrull M, Kipf T N, Bloem P, et al. Modeling relational data with graph convolutional networks[C]//The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15. Springer International Publishing, 2018: 593-607.

[20]Yu L, Sun L, Du B, et al. Heterogeneous graph representation learning with relation awareness[J]. IEEE Transactions on Knowledge and Data Engineering, 2022, 35(6): 5935-5947.

[21]Wang X, Ji H, Shi C, et al. Heterogeneous graph attention network[C]//The world wide web conference. 2019: 2022-2032.

[22]Zhang C, Song D, Huang C, et al. Heterogeneous graph neural network[C]//Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019: 793-803.

[23]Hong H, Guo H, Lin Y, et al. An attention-based graph neural network for heterogeneous structural learning[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(04): 4132-4139.

[24]Yang X, Yan M, Pan S, et al. Simple and efficient heterogeneous graph neural network[C]//Proceedings of the AAAI conference on artificial intelligence. 2023, 37(9): 10816-10824.

[25]Dong Y, Chawla N V, Swami A. metapath2vec: Scalable representation learning for heterogeneous networks[C]//Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. 2017: 135-144.