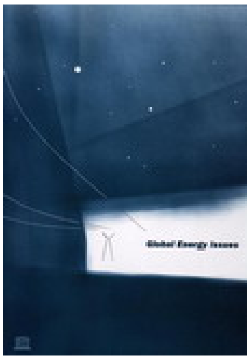
INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research



International Journal of Global Energy Issues

ISSN online: 1741-5128 - ISSN print: 0954-7118
https://www.inderscience.com/ijgei

Network security threat identification based on GNN-transformer fusion model in energy cyber systems

Yiyu Dai, Junzheng Lu, Zesen Li, Jiawei Li, Mobina Rafieipour

DOI: 10.1504/IJGEI.2026.10076525

Article History:

Received:	01 December 2025
Last revised:	17 January 2026
Accepted:	30 January 2026
Published online:	09 March 2026

Network security threat identification based on GNN-transformer fusion model in energy cyber systems

Yiyu Dai*, Junzheng Lu,
Zesen Li and Jiawei Li

Information Technology Development and Management Office,
Huaqiao University,
Xiamen, 361021, China
Email: daiyy@hqu.edu.cn
Email: ljz@hqu.edu.cn
Email: Jason.li@hqu.edu.cn
Email: lijiawei@hqu.edu.cn
*Corresponding author

Mobina Rafieipour

Electrical and Computer Engineering,
University of Victoria,
Victoria, V8P 5C2, Canada
Email: mrafieipour@hotmail.com

Abstract: At present, energy network security threat identification still faces the problem that temporal and network relationships are difficult to fuse. To address this issue, this study proposes a fusion model using Graph Neural Network (GNN) and Transformer model. This model mainly includes the following parts: using Graph Attention Network (GAN) to mine the spatial relationships between energy nodes and control entities; and using Multi-Head Self-Attention (MHSA) to extract long-range time series of energy regulation data. By combining the above two methods, the model well completes end-to-end threat detection for energy communication networks. The above research results verify that the method of joint modelling of spatial and temporal information has certain effectiveness in the field of energy network security, which provides a new idea for constructing adaptive threat identification methods in localised energy regulation networks.

Keywords: energy network security threat identification; GNN-transformer; multi-head self-attention; spatio-temporal feature fusion; intrusion detection.

Reference to this paper should be made as follows: Dai, Y., Lu, J., Li, Z., Li, J. and Rafieipour, M. (2026) 'Network security threat identification based on GNN-transformer fusion model in energy cyber systems', *Int. J. Global Energy Issues*, Vol. 48, No. 7, pp.64–84.

Biographical notes: Yiyu Dai is Senior Engineer at Information Technology Development and Management Office, Huaqiao University, China.

Junzheng Lu is Engineer at Information Technology Development and Management Office, Huaqiao University, China.

Zesen Li is Assistant Engineer at Information Technology Development and Management Office, Huaqiao University, China.

Jiawei Li is Assistant Engineer at Information Technology Development and Management Office, Huaqiao University, China.

Mobina Rafieipour is affiliated to the Electrical and Computer Engineering, University of Victoria, Canada.

1 Introduction

Nowadays, the energy network security situation has become increasingly complex. Advanced Persistent Threats (APT) attacks and multi-stage combined attacks against energy control systems are increasing, which makes the previous detection methods that only rely on fixed rules or single-angle analysis face greater challenges (Katiyar et al., 2024; Luo et al., 2023). At present, representative forms of cyberattacks include False Data Injection (FDI) attacks and ransomware attacks, among others. For instance, the Colonial Pipeline ransomware incident in 2021 resulted in a disruption of fuel supply along the eastern coast of the USA. This event highlighted the vulnerability of energy systems when confronted with coordinated and persistent attacks. Meanwhile, FDI attacks can surreptitiously and severely compromise energy dispatching and stable operation by tampering with sensor measurements or control commands. With the upgrading of attack methods, it is difficult to identify the threats hidden in normal energy regulation traffic only by feature comparison or only by observing the temporal changes of traffic (Patel, 2023). Therefore, it is necessary to propose a new idea or method that can simultaneously understand the characteristics of energy network structure and the dynamic changes of regulation traffic. However, it is precisely due to this practical demand that the application of deep learning in energy network security has been further valued (Khan et al., 2024; Ali et al., 2022). In particular, how to combine the two capabilities of understanding energy communication topology and analysing temporal dynamics has become a main direction to improve the accuracy and speed of threat identification in energy systems.

Although existing studies have made certain progress in using Graph Neural Network (GNN) and Transformer to process network data, their limitations still exist (Zhang et al., 2022). Traditional Graph Convolutional Network (GCN) can effectively extract the topological structure between network nodes, but they perform poorly in analysing temporal dynamic changes (Budžys et al., 2024). Standard Transformer models have advantages in learning long-range dependencies in sequences, but they are difficult to integrate the inherent spatial correlations between network entities (Okoli et al., 2024). In recent years, some hybrid models such as Convolutional Neural Network-Gated Recurrent Unit (CNN-GRU) have attempted to integrate spatial and temporal features. However, they still fail to fundamentally meet the two key requirements of explicit modelling of entity relationships and extraction of long-range dependencies in network data. This leads to high false positive and false negative rates in the model's detection performance when dealing with complex multi-stage attacks (Wei et al., 2022; Ashraf et al., 2022).

In view of the limitations inherent in existing threat identification methods for energy networks, this study proposes a novel GNN-Transformer fusion model. The core innovation lies in the design of a spatiotemporal joint modelling architecture tailored to the distinctive characteristics of energy networks. Unlike conventional hybrid approaches, the proposed model employs a Graph Attention Network (GAN) to construct dynamic association strengths among entities. These entities include control centres, Remote Terminal Units (RTUs), and Intelligent Electronic Devices (IEDs). This design replaces the static convolution mechanism used in traditional GCN. In addition, a Multi-Head Self-Attention (MHSA) mechanism is introduced to directly capture inter-step dependencies among dispatching commands. This approach effectively circumvents the long-term dependency attenuation commonly observed in RNN-based models. This framework is intended to address the challenge of coordinated representation of covert, multi-stage attacks in energy networks across local topological structures and global temporal dynamics. It is expected to outperform existing models in terms of threat classification accuracy, attack-stage detection rate, and cross-domain adaptability. In addition, this study provides new methodological support for the development of interpretable and adaptive threat awareness systems in energy networks.

2 Related work

In recent years, the application of deep learning to cybersecurity threat detection has exhibited an increasingly diverse development trajectory. To systematically review the technical paradigms related to spatiotemporal feature fusion that are relevant to this study, this section surveys and evaluates existing work from the following three perspectives.

2.1 GNN methods

GNN have attracted considerable attention in cybersecurity analysis due to their ability to explicitly model topological relationships among network entities. For example, Zhang et al. (2025) proposed a network security anomalous node detection approach based on GNN combined with attention mechanisms, which effectively identified potential threat nodes within networks. Guan et al. (2024) examined the applications and challenges of GNN in the domains of privacy and security. In addition, Bhandari et al. (2023) introduced a middleware framework based on distributed deep neural networks for detecting network attacks in Internet of Things (IoT) ecosystems. These studies primarily focus on extracting spatial features from network structures or static interaction relationships; however, they generally lack the capability to model the temporal evolution patterns of threat behaviours.

2.2 Time series modelling methods

To capture dynamic temporal dependencies in cybersecurity events, a variety of sequence modelling techniques have been adopted in prior studies. Wang et al. (2023) applied deep learning algorithms to urban rail transit management systems, achieving integrated intrusion detection and cybersecurity protection. Zhang and Liu (2023) developed a neural network-driven security prediction and assessment model, providing support for proactive defence systems. Traditional temporal models, including recurrent neural networks, long short-term memory networks, and temporal convolutional networks, have also been widely employed to analyse traffic sequences or log streams. Nevertheless, these approaches typically treat network entities as independent sequences and fail to effectively incorporate structural relationships among them. This limitation reduces their analytical depth in energy network scenarios characterised by complex interactions.

2.3 Hybrid GNN and time series modelling approaches

To simultaneously exploit network topology and temporal dynamics, hybrid approaches that integrate GNN with temporal models have gradually emerged as a research frontier. For instance, Xiao et al. (2022) proposed a robust graph neural network-based method for anomalous insider threat detection, which to some extent, combined entity relationships with behavioural sequences. In industrial Internet of Things scenarios, Aouedi et al. (2022) employed federated semi-supervised learning for attack detection, also taking into account both device relationships and data flows. Moreover, recent years have witnessed the development of fusion architectures such as spatiotemporal GNN and GNN-Attention models. These architectures aim to capture spatial dependencies through graph convolution while modelling temporal evolution via attention mechanisms. Notable progress has been achieved in application domains, including social network prediction and traffic flow forecasting. Nevertheless, in the context of energy network security threat identification, studies that specifically perform deep fusion modelling of communication topologies in energy control networks remain relatively scarce. Studies on the temporal characteristics of dispatching commands are also scarce. Existing hybrid models still have room for improvement in terms of explicit entity relationship modelling and the extraction of long-range dependencies (Kumari and Dhir, 2022).

The above review indicates that, although existing studies have demonstrated the effectiveness of deep learning in cybersecurity threat detection from various perspectives, most efforts focus on a single modality or specific application scenarios. They also lack deep integration of network spatial structure and temporal dynamics. Accordingly, this study proposes a GNN-Transformer fusion model that jointly learns topological associations and temporal dependencies, enabling more comprehensive and adaptive identification of cybersecurity threats.

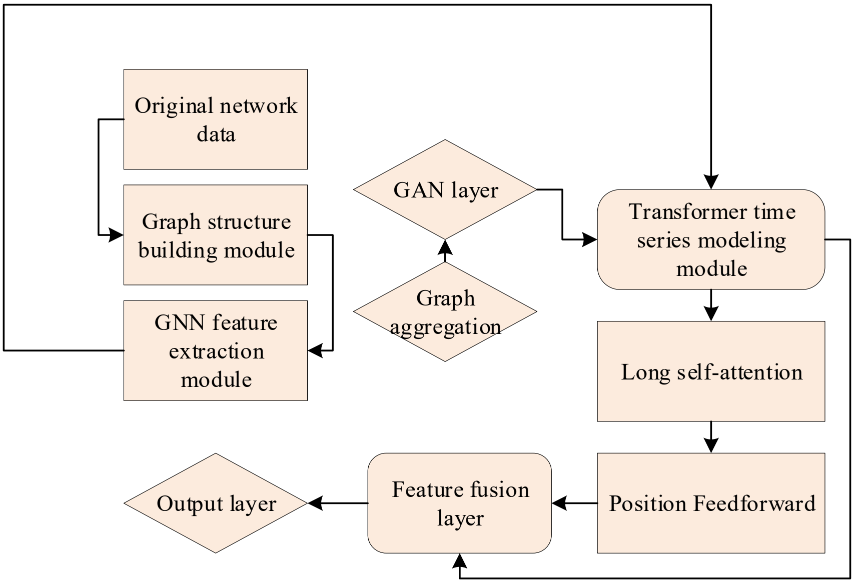
3 Network security threat identification based on GNN-transformer in energy network systems

3.1 GNN-transformer model construction

3.1.1 Overall architecture

To address the multi-dimensional identification of energy network security threats, this study proposes a GNN-Transformer fusion model. On the one hand, this model leverages the advantages of GNN in capturing complex energy network entity relationships. On the other hand, it integrates the Transformer model's capabilities in handling long-sequence dependencies and global context (Zhao and Wang, 2025; Miao et al., 2024). The overall architecture of the model is shown in Figure 1.

Figure 1 Architecture of GNN-transformer fusion model for energy network security (see online version for colours)



From Figure 1, the input of the model comes from raw energy network data. First, the regulation traffic, control log and device behaviour data are mapped into graph form through the graph structure construction module. Then, the data flows into the GNN feature extraction module, where the GAN layer is used for spatial feature learning to capture local and global topological structures of the energy communication network. Subsequently, the model inputs the extracted feature sequences into the Transformer temporal modelling module, and uses MHSA mechanism to encode temporal dynamics of energy scheduling and control commands, avoiding long-term data dependence (Kaushik and Rathore, 2023). Finally, the feature fusion layer concatenates spatial and temporal information, and uses the fully connected layer and Softmax classifier to output the probability of threat types. The entire architecture is optimised in an end-to-end manner, ensuring that the model has certain robustness and scalability for diverse threats in complex energy network environments.

3.1.2 Graph structure construction

Before identifying threat in energy networks, it is necessary to convert raw energy communication data into a graph representation first to capture the complex relationships between entities (Sunkara, 2022). Based on this, this study first constructs the graph structure in the model. In energy network security threat identification, the graph structure takes nodes and edges as core elements: nodes represent entities in the energy network, including control centres, remote terminal units (RTUs), IEDs, or energy meters. Edges represent interaction behaviours between entities, including control commands, data acquisition flows or synchronisation sessions. Among these, the control centre serves as the core dispatching node, responsible for monitoring the overall network state and issuing operational commands. RTUs are typically deployed at substations or plant sites to collect sensor data and execute control instructions. IEDs provide integrated functionalities for protection, measurement, and control. Energy meters record electricity consumption or generation. These nodes are interconnected through various interactions, including control commands, data acquisition flows, and session synchronisation, collectively forming the dynamic topology of energy communication and control networks.

Based on this, the construction process of the graph structure is as follows:

1) Given a network dataset, extract all unique entities as the node set $V = \{v_1, v_2, \dots, v_N\}$, where N represents the total number of nodes;

2) Define the edge set $E \subseteq V \times V$ based on the interaction frequency or temporal proximity between entities, and encode the connection relationship through an adjacency matrix $A \in R^{N \times N}$. Here, $A_{ij} = 1$ indicates that there is an edge between node v_i and node v_j, otherwise $A_{ij} = 0$. Meanwhile, each node is associated with a feature vector, forming a node feature matrix $X \in R^{N \times D}$, where D is the feature dimension. The features include attributes such as traffic statistics, protocol types, or behaviour logs. To enhance the expressive ability of the graph, edge weights are calculated based on interaction intensity, and the cosine similarity equation is used to quantify the correlation between nodes:

$$w_{ij} = \frac{x_i \cdot x_j}{\| x_i \| \| x_j \|} \tag{1}$$

In equation (1), x_i and x_j are the feature vectors of node v_i and node v_j respectively. Through this construction method, the graph structure can combine the spatial relationships and dynamic interactions of the network, providing a structured input for the subsequent feature extraction of GNN.

Considering the time-sensitive nature of interactions among entities in the energy network, a time decay factor and interaction frequency are introduced to adjust the edge weights. The time decay coefficient is defined as $\tau_{(ij)} = \exp\left(-\lambda \cdot \Delta t_{(ij)}\right)$, where $\Delta t_{(ij)}$ is the time difference (in minutes) between the current moment and the most recent interaction, and λ is the decay rate parameter (set to 0.05). Meanwhile, the interaction count f_{ij} between nodes v_i and v_j within a sliding time window T_w (30 minutes) is recorded. The final edge weight is computed by integrating feature similarity, interaction frequency, and temporal proximity as follows:

$$w_{ij} = \frac{x_i \cdot x_j}{\| x_i \| \| x_j \|} \cdot \frac{f_{ij}}{max(f)} \cdot \tau_{ij} \tag{2}$$

In equation (2), $max(f)$ represents the maximum interaction count among all node pairs within the sliding window and is used for normalisation. This computation ensures that control commands or data acquisition flows occurring frequently in the recent past are assigned higher connection strengths in the graph. It thereby better reflects the temporal characteristics and operational patterns of energy network traffic.

To achieve an automated mapping from raw network connection logs to a graph structure, this study defines specific rules for node and edge construction based on the characteristics of the relevant datasets. Each node in the graph corresponds to a unique source or destination IP (Internet Protocol) address. Node feature vectors are composed of statistical metrics within a given time window, including protocol type, service type, connection duration, and traffic volume in bytes for the corresponding IP. Edges are constructed based on the connection records: if communication occurs between the source and destination addresses in two records, an edge is established between the corresponding nodes. Edge weights are dynamically computed according to the number of connections, total packet volume, or temporal co-occurrence frequency. To accommodate the dynamic evolution of network topology, a sliding time window mechanism is employed to incrementally update both the adjacency matrix and node features. This approach captures the temporal evolution of entity relationships while preserving historical interaction patterns.

3.1.3 GNN feature extraction

In the GNN feature extraction module, GAN is used to learn high-level representations of nodes from the constructed graph structure to capture complex spatial dependencies between network entities (Rizvi, 2023; Saminathan et al., 2023). Given the input node feature matrix $H^{(0)} = X$, where $X \in R^{N \times D}$ represents the node feature matrix, N is the number of nodes, and D is the input feature dimension. GAN calculates the interaction weights between nodes through an attention mechanism. First, for each node pair (i, j), the attention coefficient e_{ij} is defined as:

$$e_{ij} = \text{LeakyReLU}\left(a^T \left[Wh_i \parallel Wh_j\right]\right) \tag{3}$$

In the above equation, h_i and h_j are the feature vectors of node *i* and node *j* respectively. $W \in R^{D' \times D}$ is a learnable weight matrix. $a \in R^{2D'}$ is an attention parameter vector. || denotes the vector concatenation operation. LeakyReLU is a non-linear activation function.

Then, the attention coefficient is normalised by softmax function to get the normalised attention weight:

$$\alpha_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{k \in M_i} \exp\left(e_{ik}\right)} \tag{4}$$

M_i represents the neighbour set of node *i*.

Based on the attention weight, the update feature of node *i* is calculated by weighted aggregation:

$$h_i^{'} = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} W h_j\right) \tag{5}$$

σ is the activation function.

In addition, to enhance the expression ability of the model, the study also introduces a multi-head attention (MHA) mechanism to splice the outputs of multiple attention heads:

$$h_i^{'} = \|_{k=1}^{K} \sigma\left(\sum_{j \in M_i} \alpha_{ij}^{k} W^{k} h_j \right) \tag{6}$$

K is the number of attention heads, and || represents vector splicing.

By stacking multiple GAN layers, the model can extract node features, and finally output a high-order feature matrix $H^{(C)} \in R^{N \times D''}$. Here, C represents the number of GAN layers, and D'' denotes the output feature dimension. This matrix provides a spatial structure representation for subsequent temporal modelling.

3.1.4 Transformer temporal modelling

After obtaining the spatial features extracted by GNN, the Transformer temporal modelling module is used to capture the temporal dependencies in energy network security events (Xu et al., 2025). This module takes the node feature sequence output by GNN as input, and first adjusts the feature dimension to d_{model} through linear projection to form the initial embedded representation $H_0 \in R^{L \times d_{model}}$. Here, L represents the sequence length, and d_{model} is the dimension of the model's hidden layer. To retain temporal information, sinusoidal Positional Encoding (PE) is introduced, and its calculation equation is:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}} \right) \tag{7}$$

$$PE_{(pos,2i+1)} = \cos\left(pos / 10000^{2i/d_{model}} \right) \tag{8}$$

In equations (7) and (8), pos stands for position index and *i* stands for dimension index, and the final input representation is obtained by adding the position code and the input embedding.

MHSA captures the dependencies of different subspaces by calculating multiple attention heads in parallel. The calculation process of each attention head is:

$$\text{Att} = \text{soft}\left(\frac{QJ^{T}}{\sqrt{d_k}} \right) Z \tag{9}$$

Q, *J* and *Z* are the query, key and value matrices obtained by linear transformation respectively, and d_k is the dimension of the key vector (Li et al., 2021).

The output of MHA is obtained by stitching and linear transformation:

$$\text{MHSA}(P) = \text{Concat}(p_1, \ldots p_m) W^{O} \tag{10}$$

W^{O} is the output projection matrix, and p_m is calculated as follows:

$$p_m = \text{Att}\left(HW_m^{Q}, HW_m^{J}, HW_m^{Z} \right) \tag{11}$$

The output of attention layer is further processed by layer normalisation Layer Normalisation (LN) and feed forward network Feedforward Network (FFN):

$$P' = \mathrm{LN}\left(H + \mathrm{MHSA}\left(H\right)\right) \tag{12}$$

$$P'' = \mathrm{LN}\left(H' + \mathrm{FFN}\left(H'\right)\right) \tag{13}$$

In the above calculation, the feedforward network adopts two-layer linear transformation and ReLU activation function:

$$\mathrm{FFN}\left(x\right) = max\left(0, xW_1 + b_1\right)W_2 + b_2 \tag{14}$$

By stacking multiple such encoding layers, the model can effectively learn the long-term temporal dependency patterns in network security events, providing high-level feature representations with temporal context for the final threat classification.

3.1.5 Feature fusion and output

After completing spatial feature extraction and temporal modelling, the feature fusion and output layer is responsible for effectively integrating features from the two different modalities and generating the final classification results. This module receives the spatial feature matrix $H^{(C)} \in R^{N\times D''}$ from the GNN and the temporal feature sequence $P'' \in R^{L\times d_{model}}$ from the Transformer-based temporal modelling module. To unify the dimensions for subsequent fusion, the node feature matrix $H^{(C)}$ is first subjected to global average pooling along the node dimension, producing a graph-level representation $h_g \in R^{1\times D''}$. Simultaneously, the sequence features P'' from the Transformer are either max-pooled along the temporal dimension or reduced to the final time-step state, resulting in a compressed temporal representation $h_t \in R^{1\times d_{model}}$. The representations h_g and h_t are then concatenated along the feature dimension to form a unified feature representation $h_{fused} \in R^{1\times\left(D''+d_{model}\right)}$, which encapsulates both spatial topological information and temporal dynamic patterns.

To ensure the fused features have a consistent dimension, the model also incorporates a fully connected layer to project and transform the concatenated features, and enhances the feature expression ability through a non-linear activation function. Then, the fused features are further refined by a Multilayer Perceptron (MLP), which includes batch normalisation and Dropout operations to improve the model's generalisation ability. Finally, the high-dimensional features are mapped to the probability distribution of threat categories through a Softmax classifier, outputting the probability that each sample belongs to different threat types, thus completing the end-to-end network security threat identification task. This fusion strategy leverages the complementarity between spatial and temporal structures, aiming to improve the detection accuracy and robustness of the model for complex network threats.

3.2 Experimental design

3.2.1 Experimental environment and dataset

(1) *Experimental environment and model parameter configuration*

To verify the effectiveness of the proposed model, experiments are conducted on a workstation equipped with dual NVIDIA GeForce RTX 4090 GPUs (24GB memory each), an Intel Core i9-13900K processor, and 128GB DDR5 memory, with the operating system being Ubuntu 22.04 LTS. The programming language used is Python 3.9, and the deep learning framework is PyTorch 2.0, while the PyTorch Geometric library is utilised to implement operations related to GNN. The model is trained using the Adam optimiser, with an initial learning rate set to 0.001 and a batch size of 128. All experiments are repeated five times, and the average metrics were taken to eliminate the impact of randomness.

The specific experimental environment configuration and the hyperparameter setting of the model are shown in Table 1 and Table 2.

Table 1 Configuration of experimental environment

Environmental component	*Configuration specification*
Operating system	Ubuntu 22.04 LTS
CPU (Central Processing Unit)	Intel Core i9-13900K (24 cores and 32 threads)
GPU (Graphics Processing Unit)	2×NVIDIA GeForce RTX 4090(24GB GDDR6X)
Memory	128GB DDR5 4800MHz
Deep learning framework	PyTorch 2.0.1+CUDA 11.8
GNN library	PyTorch Geometric 2.3.0
Programming language	Python 3.9.16

Table 2 Hyperparameter configuration of GNN-Transformer model

Hyperparameter	*Setting value*
Graph attention head number	8
GAN hidden layer dimension	256
GAN layer number	2
Transformer encoder layer number	4
Transformer hidden layer dimension	512
Feedforward network dimension	1024
Learning rate	0.001
Batch size	128
Dropout rate	0.1
Number of training rounds	200
Weight attenuation	0.0001
Number of attention heads	8
Activation function	GELU

(2) *Dataset selection and processing*

The experiment uses the Network Security Laboratory-Knowledge Discovery and Data Mining (NSL-KDD) dataset as the main data source. This dataset is an improved version of KDDCup99, which effectively addresses the shortcomings of the original dataset by eliminating redundant records and balancing the data distribution. The NSL-KDD dataset contains 41-dimensional network connection features and 1 label feature, including normal traffic and four types of attacks: Denial-of-Service (DoS), Remote-to-Local (R2L), User-to-Root (U2R), and Probing (port scanning and probing) (Dataset link: https://www.unb.ca/cic/datasets/nsl.html).

This study extracts 140,790 network connection records from the NSL-KDD dataset, with each record containing basic features of Transmission Control Protocol (TCP) connections, connection content features, and time-based network traffic statistical features.

In the data pre-processing stage, one-hot encoding is first applied to symbolic features (such as protocol types and service types), and then Z-score standardisation is performed on numerical features to ensure the uniformity of feature scales.

In addition, the NSL-KDD dataset is randomly divided into a training set, a validation set, and a test set in a ratio of 7:2:1 for subsequent model performance verification. The specific division of the dataset is shown in Table 3.

Table 3 Statistical results of NSL-KDD dataset partition

Category	*Number of training set samples*	*Number of samples in verification set*	*Number of test set samples*
Normal	45,927	13,122	6,561
DoS	32,354	9,244	4,622
R2L	7,854	2,244	1,122
U2R	2,142	612	306
Probing	10,276	2,936	1,468

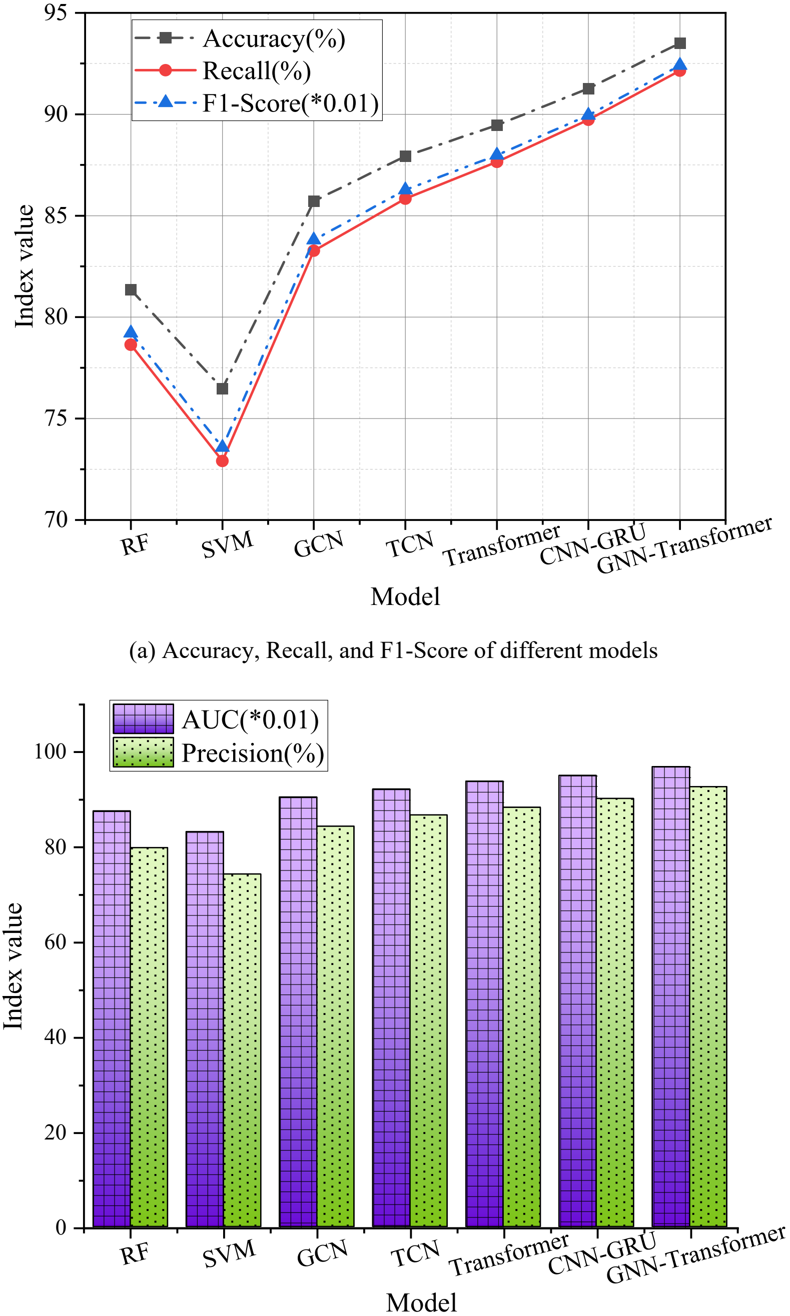
3.2.2 Evaluation index and baseline model

To comprehensively evaluate the performance of the GNN-Transformer fusion model, this study selects five evaluation metrics to quantify its performance. The metrics include Accuracy, Precision, Recall, F1-Score, and Area Under the ROC Curve (AUC). Among them, Accuracy measures the overall classification performance of the model, Precision evaluates the accuracy of the model’s predictions, Recall reflects the model’s ability to identify positive samples, and AUC demonstrates the overall performance of the classifier from a probability perspective.

In terms of selecting baseline models, five traditional methods are chosen, including Random Forest (RF), Support Vector Machine (SVM), GCN, Temporal Convolutional Network (TCN), and Transformer model. In addition, the recently proposed CNN-GRU hybrid model is also selected, which uses CNN to extract spatial features and obtains temporal dependencies through GRU.

All the baseline models adopted above and the GNN-Transformer fusion model proposed in this study share the same experimental configuration environment, which is used to ensure the fairness of the experiment.

Figure 2 Comparison results of multi-classification detection performance (%) (see online version for colours)



(a) Accuracy, Recall, and F1-Score of different models

(b) Precision and AUC of different models

4 Performance verification results and analysis of GNN-transformer model

4.1 Verification of the identification effect of the model

4.1.1 Multi-classification detection performance

This experiment will be used to test the classification ability of the model in complex network environments. Conducted on the NSL-KDD test set, the experiment mainly compares the proposed GNN-Transformer fusion model with various baseline methods, and then records the performance of each model in five metrics: Accuracy, Precision, Recall, F1-Score, and AUC. The specific experimental results are shown in Figure 2.

In Figure 2, on the NSL-KDD test set, different models exhibit certain differences in the detection results of complex network threats. Among them, the GNN-Transformer fusion model achieves an Accuracy of 93.51%, an F1-Score of 0.9241, and an AUC of 0.9687, which are better than those of other baseline models. For example, the Accuracy of RF is 81.35%, and that of CNN-GRU is 91.27%, both lower than the proposed model. Further analysis shows that the GNN-Transformer fusion model also performs better in terms of Precision (92.67%) and Recall (92.15%). This indicates that the model can effectively reduce classification errors and has a better attack identification ability. The main reason for this phenomenon is the joint effect of the GNN and the Transformer: The former is used to extract local traffic features, and the latter is used to model global dependencies. The above data analysis results collectively prove the effectiveness of this fusion strategy in intrusion detection.

To further evaluate the model's adaptability to different attack types, Table 4 presents the detection performance of the GNN-Transformer model on four categories of attacks in the NSL-KDD test set.

Table 4 Detection performance (%) of the GNN-transformer model against various attacks

Attack type	*Precision (%)*	*Recall (%)*	*F1-Score*
DoS	96.3	94.8	95.5
R2L	86.7	88.5	87.6
U2R	84.1	86.3	85.2
Probing	93.5	94.1	93.8

As shown in Table 4, the model achieves the highest precision (96.3%) for DoS attacks, attributable to their large traffic volume and distinctive features. The recall for Probing attacks reaches 94.1%, indicating effective identification of scanning and probing behaviours. For low-frequency and highly covert attacks such as R2L and U2R, the model maintains F1-scores of 87.6% and 85.2%, respectively, demonstrating a degree of robustness to class imbalance. The observed variations in performance across attack types indirectly reflect the model's comprehensive advantages in feature extraction and spatiotemporal modelling.

4.1.2 Effectiveness of temporal threat detection

This study constructs a dedicated temporal test set to simulate multi-step attack and persistent threat scenarios in real-world networks. It then uses this test set to explore the model's ability to identify attack chains in continuous time windows, thereby verifying the model's performance in recognising threats with temporal continuity. The experiment constructs temporal samples through a sliding window mechanism and observes the results of each model in three temporal metrics: detection latency, attack phase capture rate, and temporal false positive rate, as shown in Figure 3.

Figure 3 Comparison of temporal threat detection performance of different models (see online version for colours)

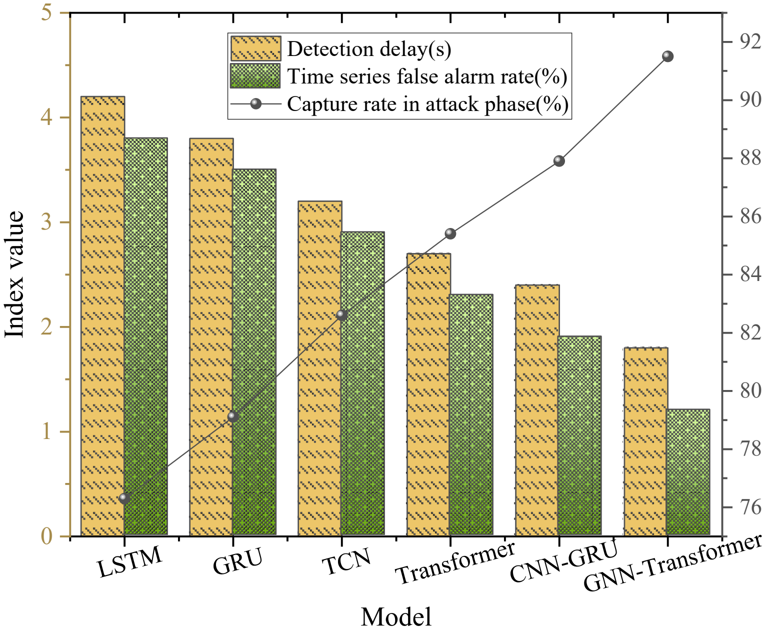
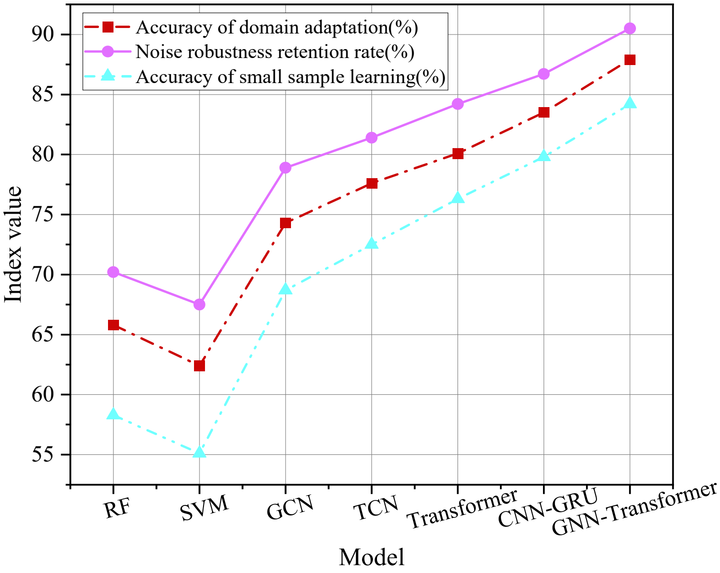


Figure 3 shows that in the temporal threat detection experiment, the proposed GNN-Transformer model achieves better performance. Specifically, the model has a detection latency of 1.8 seconds, which is lower than all other comparison models (e.g., the LSTM model has a latency of 4.2 seconds). Meanwhile, the model achieves an attack phase capture rate of 91.5%, which is a 15% improvement compared to the LSTM model. This indicates that it can more accurately track multi-step attack chains. In addition, the model has a temporal false positive rate of 1.2%, and this result reflects its higher robustness in complex attack scenarios. All the above results collectively verify that the architecture integrating GNN and temporal attention mechanisms can better capture hidden relationships in persistent threats, providing a certain technical foundation for dynamic network security defence.

4.1.3 Robustness and generalisation of model

Since the proposed model is mainly used for network security threat identification, it is necessary to examine the model's adaptability in diverse network environments. To this end, datasets with different distributions are used in the training and testing phases to simulate the domain adaptation problem in real-world networks. The experiment adopts a transfer learning framework: the model is pre-trained on the NSL-KDD dataset, and then fine-tuned and tested on the unknown attack categories of the CIC-IDS2017 (Canadian Institute for Cybersecurity-Intrusion Detection Systems 2017) dataset (Access link: https://www.unb.ca/cic/datasets/ids-2017.html). This is to verify whether the model's performance remains stable under the condition of data distribution changes. In addition, considering the differences between the two datasets in feature dimensionality, distribution, and attack definitions, a lightweight feature alignment layer is introduced during the fine-tuning stage. This layer maps the high-dimensional features of CIC-IDS2017 into a latent space compatible with the pre-trained NSL-KDD model. Simultaneously, an attack-semantic-based class mapping strategy is employed to align the attack types in CIC-IDS2017 with the four attack categories defined in NSL-KDD, thereby mitigating the impact of label inconsistencies on model adaptability. During this alignment process, only the parameters of the alignment layer and the classifier are updated, while the backbone network weights remain largely unchanged, enabling effective cross-domain knowledge transfer under limited labelled data.

Figure 4 Test results of model robustness and generalisation (see online version for colours)



From Figure 4, the proposed GNN-Transformer model exhibits a certain level of cross-domain adaptability. Its domain adaptation accuracy reaches 87.9%, which is higher than that of the traditional RF (65.8%) and the standard Transformer (80.1). This indicates that the model can effectively cope with the data distribution change from NSL-KDD to CIC-IDS2017. At the same time, the model achieves a noise robustness retention rate of 90.5% and a few-shot learning accuracy of 84.2%, both outperforming other baseline models. This verifies that the fusion of GNN and attention mechanism can enhance the model's strong generalisation ability for unknown attack patterns. In addition, since the CIC-IDS2017 dataset used in the experiment contains multi-stage attack traffic and diverse protocol interaction information in real network environments. Its complex scenarios such as Brute Force and infiltration attacks provide a near-practical verification environment for the model. The above data further illustrate that the model has high practical application value in dynamic network security defence.

4.2 Ablation experiment

4.2.1 Module contribution analysis

The purpose of this experiment is to investigate the contribution of each module in the GNN-Transformer fusion model to the system performance through methods such as sequential removal and replacement. Conducted on the NSL-KDD test set, the experiment verifies the performance of using GNN alone, using only the Transformer module, and combining the two in different ways. The obtained results are shown in Figure 5.

Figure 5 Module contribution analysis ablation experimental results (%) (see online version for colours)

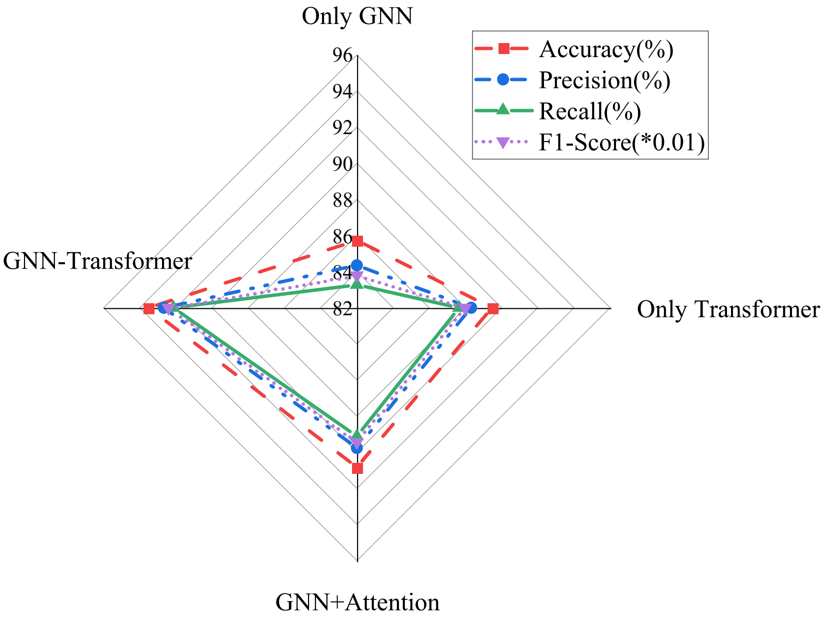
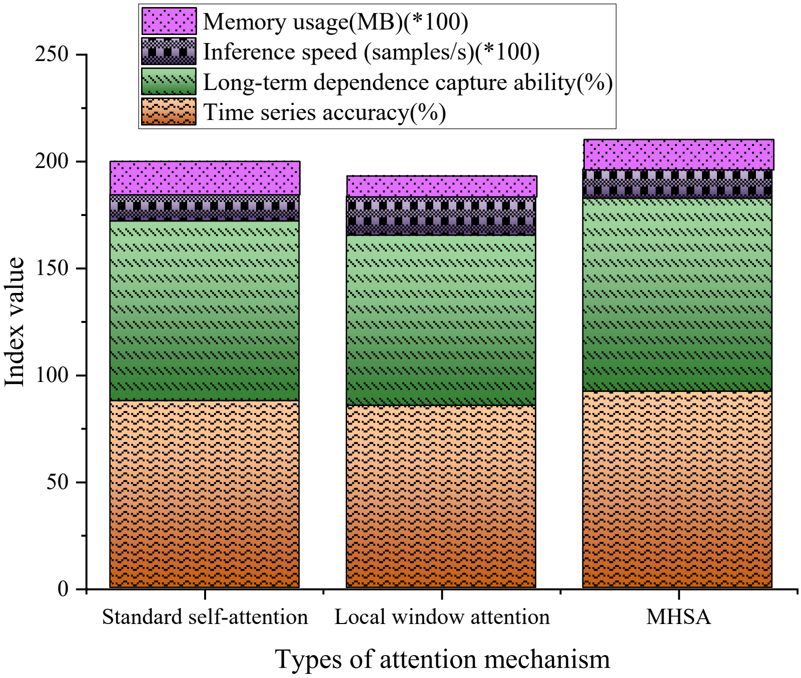


Figure 5 shows that each module in the GNN-Transformer model has different impacts on the overall model performance. With the single GNN module, the model's performance metrics are average (Accuracy: 85.72%, F1-Score: 0.8380), indicating that a single module has limited ability to describe complex correlations. When only the Transformer module is used, the model's performance is improved to a certain extent (Accuracy: 89.46%, F1-Score: 0.8798), which demonstrates the effectiveness of the Transformer's sequence modelling capability. After combining the two modules, the model's Accuracy and F1-Score reach 90.83% and 0.8944 respectively. This benefit comes from the GNN's in-depth exploration of network topology and the Transformer's accurate capture of long-range dependencies, which together drive the model's performance in intrusion detection tasks.

4.2.2 Transformer module function verification

To better understand the impact of attention mechanisms on temporal model construction, this study focuses on investigating the identification performance of standard self-attention, local window attention, and MHSA in capturing long-range dependencies through a comparative approach. The results are shown in Figure 6.

Figure 6 Performance comparison of different attention mechanisms (%) (see online version for colours)



In Figure 6, the proposed MHA mechanism enables the temporal accuracy to reach 92.15% and the long-range capture capability to reach 90.47%, both outperforming other attention mechanisms. Although local window attention demonstrates significant advantages in terms of long-range capture capability, its processing speed is relatively reduced. The above research results indicate that the proposed model is expected to provide strong theoretical support for the temporal analysis of network security threats.

4.2.3 Comparative analysis of different fusion strategies

To assess the optimality of the current GNN-Transformer fusion approach, this study further compares the effects of three common feature fusion strategies on model performance. These include serial fusion, in which GNN outputs are fed directly into the Transformer, and parallel fusion, where inputs are processed independently through the GNN and Transformer and their outputs are concatenated. The third strategy is weighted fusion, which introduces learnable weights to dynamically combine the two feature streams. The experimental results are summarised in Table 5.

Table 5 Performance comparison of different feature fusion strategies on the NSL-KDD test set (%)

Fusion strategy	*Accuracy*	*F1-Score*	*AUC*	*Training time/epoch (s)*
Serial Fusion	90.12	0.8876	0.953	38.4
Parallel Fusion	93.51	0.9241	0.969	42.3
Weighted Fusion	92.07	0.9085	0.961	44.1

As shown in Table 5, the currently adopted parallel concatenation strategy achieves superior performance in terms of accuracy, F1-score, and AUC. This indicates that the strategy more effectively preserves the complementary information of spatiotemporal features, minimising information loss or redundancy.

4.2.4 Fine-grained ablation analysis of internal model components

To further validate the effectiveness of key internal components, a series of fine-grained ablation experiments was conducted to investigate how different design choices impact overall performance. The specific configurations include varying the number of graph attention heads (4, 8, 12) and replacing or removing the sinusoidal positional encoding in the Transformer. They also include removing the MLP operation in the feature fusion layer. The results are presented in Table 6.

Table 6 Fine-grained ablation results of internal model components (based on NSL-KDD test set)

Experimental configuration	*Accuracy*	*F1-Score*
GAT head count: 4	92.36%	0.9112
GAT head count: 8	93.51%	0.9241
GAT head count: 12	92.87%	0.9168
No position encoding	90.31%	0.8893
Learnable position encoding	93.12%	0.9195
Feature fusion layer without MLP	91.82%	0.9057
Feature fusion layer without Dropout	92.05%	0.9082
Complete model	93.51%	0.9241

The results in Table 6 indicate that the model achieves optimal performance with 8 attention heads; too few or too many heads either fail to capture sufficient information or introduce noise. Positional encoding plays a critical role in temporal modelling, as its removal leads to a noticeable drop in accuracy. Additionally, the MLP and Dropout operations within the feature fusion layer positively contribute to generalisation, with the removal of either component resulting in a decline in F1-score.

4.3 Analysis of training efficiency and resource consumption

To further evaluate the practicality of the model, the training-time consumption and memory usage of the GNN-Transformer fusion model were recorded. Experiments were conducted on the full NSL-KDD training set over 200 epochs to track the average training time per epoch, peak GPU memory usage, and CPU utilisation. The results are summarised in Table 7. For comparative analysis, training efficiency metrics of other mainstream baseline models under the same conditions are also presented.

Table 7 Comparison of training efficiency and resource consumption across models

Model	*Avg. training time per epoch (s)*	*Peak GPU memory (GB)*	*Average CPU utilisation (%)*	*Total training time (min)*
GNN-Transformer	42.3	9.8	78.5	141.0
CNN-GRU	38.7	7.2	72.1	129.0
Transformer	35.1	6.5	68.3	117.0
GCN	25.6	4.9	65.4	85.3
RF	-	-	89.2	12.5 (CPU Training)

As shown in Table 7, the GNN-Transformer fusion model requires an average of 42.3 seconds per epoch and a peak GPU memory usage of 9.8 GB during training. Its overall efficiency is comparable to similar hybrid models, such as CNN-GRU, while providing superior spatiotemporal feature fusion capabilities. In contrast, traditional machine learning approaches like Random Forest offer faster training times but exhibit weaker detection performance and generalisation ability. Although the proposed model incurs slightly higher resource consumption compared to single-architecture models, it exhibits comprehensive advantages in threat detection accuracy, latency, and cross-domain adaptability. These advantages indicate its practical feasibility for deployment in real-world energy network environments.

5 Conclusion

To enhance the fusion modelling effect of spatial structure and temporal features in energy network security threat identification, this study proposes a combined model integrating a GNN and a Transformer. On one hand, the model uses a GAN to extract complex topological relationships between energy network entities. On the other hand, it adopts a MHSA mechanism to analyse temporal dependency patterns in threat behaviours targeting energy control processes. Experimental analysis shows that the proposed model achieves an Accuracy of 93.51% and an F1-Score of 0.9241 in the multi-classification task on the NSL-KDD dataset. In the temporal threat detection task, it has a latency of 1.8 seconds and an attack phase capture rate of 91.5%. Meanwhile, in cross-dataset tests, the model reaches a domain adaptation accuracy of 87.9%, outperforming traditional machine learning and single deep learning baseline models. However, this study still has limitations. For example, the model is highly dependent on labelled data, and there is room for further optimisation in real-time inference speed in ultra-large-scale energy networks. In practical energy network environments, obtaining large-scale, high-quality labelled threat data is often costly and time-consuming due to security policy restrictions and operational privacy considerations. This limitation constrains the rapid deployment and iterative improvement of supervised learning models. To address this challenge, future research will further explore learning paradigms based on self-supervised pre-training or unlabelled data augmentation. Approaches may include generating noisy labelled samples using generative adversarial networks. They may also employ contrastive learning to extract transferable feature representations from massive amounts of unlabelled industrial control traffic. This thereby maintains the model's detection effectiveness under low-label resource conditions. Additionally, it will explore model lightweighting and edge deployment solutions to enhance its applicability and extensibility in real energy network environments.

Declarations

All authors declare that they have no conflicts of interest.

References

Ali, M.H., Jaber, M.M., Abd, S.K., Rehman, A., Awan, M.J., Damaševičius, R. and Bahaj, S.A. (2022) 'Threat analysis and distributed denial of service (DDoS) attack recognition in the internet of things (IoT)', *Electronics*, Vol. 11, No. 3, p.494.

Aouedi, O., Piamrat, K., Muller, G. and Singh, K. (2022) 'Federated semisupervised learning for attack detection in industrial internet of things', *IEEE Transactions on Industrial Informatics*, Vol. 19, No. 1, pp.286–295.

Ashraf, I., Narra, M., Umer, M., Majeed, R., Sadiq, S., Javaid, F. and Rasool, N. (2022) 'A deep learning-based smart framework for cyber-physical and satellite system security threats detection', *Electronics*, Vol. 11, No. 4, p.667.

Bhandari, G., Lyth, A., Shalaginov, A. and Grønli, T.M. (2023) 'Distributed deep neural-network-based middleware for cyber-attacks detection in smart IoT ecosystem: a novel framework and performance evaluation approach', *Electronics*, Vol. 12, No. 2, p.298.

Budžys, A., Kurasova, O. and Medvedev, V. (2024) 'Deep learning-based authentication for insider threat detection in critical infrastructure', *Artificial Intelligence Review*, Vol. 57, No. 10, p.272.

Guan, F., Zhu, T., Zhou, W. and Choo, K.K.R. (2024) 'Graph neural networks: a survey on the links between privacy and security', *Artificial Intelligence Review*, Vol. 57, No. 2, p.40.

Katiyar, N., Tripathi, M.S., Kumar, M.P., Verma, M.S., Sahu, A.K. and Saxena, S. (2024) 'AI and cyber-security: enhancing threat detection and response with machine learning', *Educational Administration: Theory and Practice*, Vol. 30, No. 4, pp.6273–6282.

Kaushik, P. and Rathore, S.P.S. (2023) 'Deep learning multi-agent model for phishing cyber-attack detection', *International Journal on Recent and Innovation Trends in Computing and Communication*, Vol. 11(9s), pp.680–686.

Khan, S., Khan, M.A. and Alnazzawi, N. (2024) 'Artificial neural network-based mechanism to detect security threats in wireless sensor networks', *Sensors*, Vol. 24, No. 5, p.1641.

Kumari, S. (2022) 'Cybersecurity in digital transformation: using AI to automate threat detection and response in multi-cloud infrastructures', *J. Computational Intel. & Robotics*, Vol. 2, No. 2, pp.9–27.

Li, Z., Cheng, X., Sun, L., Zhang, J. and Chen, B. (2021) 'A hierarchical approach for advanced persistent threat detection with attention-based graph neural networks', *Security and Communication Networks*, Vol. 2021, No. 1, 9961342.

Luo, N., Yu, H., You, Z., Li, Y., Zhou, T., Jiao, Y., Han, N., Liu, C., Jiang, Z. and Qiao, S. (2023) 'Fuzzy logic and neural network-based risk assessment model for import and export enterprises: a review', *Journal of Data Science and Intelligent Systems*, Vol. 1, No. 1, pp.2–11.

Miao, W., Zhao, X., Wang, C., Chen, S., Gao, P. and Li, Q. (2024) 'A GNN-enhanced ant colony optimization for security strategy orchestration', *Symmetry*, Vol. 16, No. 9, p.1183.

Okoli, U.I., Obi, O.C., Adewusi, A.O. and Abrahams, T.O. (2024) 'Machine learning in cybersecurity: a review of threat detection and defense mechanisms', *World Journal of Advanced Research and Reviews*, Vol. 21, No. 1, pp.2286–2295.

Patel, S.K. (2023) 'Attack detection and mitigation scheme through novel authentication model enabled optimized neural network in smart healthcare', *Computer Methods in Biomechanics and Biomedical Engineering*, Vol. 26, No. 1, pp.38–64.

Rizvi, M. (2023) 'Enhancing cybersecurity: The power of artificial intelligence in threat detection and prevention', *International Journal of Advanced Engineering Research and Science*, Vol. 10, No. 5, pp.055–060.

Saminathan, K., Mulka, S.T.R., Damodharan, S., Maheswar, R. and Lorincz, J. (2023) 'An artificial neural network autoencoder for insider cyber security threat detection', *Future Internet*, Vol. 15, No. 12, p.373.

Sunkara, G. (2022) 'AI-driven cybersecurity: advancing intelligent threat detection and adaptive network security in the era of sophisticated cyber attacks', *Well Testing Journal*, Vol. 31, No. 1, pp.185–198.

Wang, Z., Xie, X., Chen, L., Song, S. and Wang, Z. (2023) 'Intrusion detection and network information security based on deep learning algorithm in urban rail transit management system', *IEEE Transactions on Intelligent Transportation Systems*, Vol. 24, No. 2, pp.2135–2143.

Wei, Y., Machica, I.K.D., Dumdumaya, C.E., Arroyo, J.C.T. and Delima, A.P. (2022) 'Liveness detection based on improved convolutional neural network for face recognition security', *Int J Emerg Technol Adv Eng*, Vol. 12, No. 8, pp.45–53.

Xiao, J., Yang, L., Zhong, F., Wang, X., Chen, H. and Li, D. (2022) 'Robust anomaly-based insider threat detection using graph neural network', *IEEE Transactions on Network and Service Management*, Vol. 20, No. 3, pp.3717–3733.

Xu, S., Shi, Y., Shi, L. and Zhang, H. (2025) 'Efficient network defense policies via GNN-enhanced reinforcement learning', *The Journal of Supercomputing*, Vol. 81, No. 8, p.968.

Zhang, D., Wang, J., Gao, H., Ni, Z.W. and Zhang, H. (2025) 'Network security anomaly node detection based on graph neural network and attention mechanism', *Journal of Network and Systems Management*, Vol. 33, No. 4, p.87.

Zhang, L. and Liu, Y. (2023) 'Network security prediction and situational assessment using neural network-based method', *Journal of Cyber Security and Mobility*, Vol. 12, No. 4, pp.547–568.

Zhang, W., Li, Y., Li, X., Shao, M., Mi, Y., Zhang, H. and Zhi, G. (2022) 'Deep neural network-based SQL injection detection method', *Security and Communication Networks*, Vol. 2022, No. 1, 4836289.

Zhao, Z. and Wang, Y. (2025) 'Soft actor-critic algorithm and improved GNN model in secure access control of disaggregated optical networks', *Scientific Reports*, Vol. 15, No. 1, 29358.