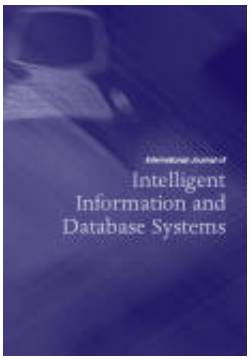


International Journal of Intelligent Information and Database Systems

ISSN online: 1751-5866 - ISSN print: 1751-5858
https://www.inderscience.com/ijiids

Power information network attack chain identification and disaster recovery early warning mechanism based on graph neural network

Yangrui Zhang, Shihui Chen, Chao Zhang, Junpeng Zhao, Kaichen Zhang, Zixun Lu

DOI: 10.1504/IJIIDS.2026.10077399

Article History:

Received:	16 September 2025
Last revised:	27 November 2025
Accepted:	03 December 2025
Published online:	05 May 2026

Power information network attack chain identification and disaster recovery early warning mechanism based on graph neural network

Yangrui Zhang

Marketing Service Centre of State Grid Hebei Electric Power Co., Ltd.,
South Wing, Central Office Building, Zhuang Sheng Plaza,
No. 10 Xuanwumenwai Street, Xicheng District, Beijing,
Shijiazhuang City, Hebei Province, 050000, China
Email: yangruizhang001@outlook.com

Shihui Chen

Marketing Business Department,
State Grid Siji Cybersecurity Technology (Beijing) Co., Ltd.,
Beijing City, 100052, China
Email: shihuichen002@outlook.com

Chao Zhang and Junpeng Zhao

Marketing Service Centre of State Grid Hebei Electric Power Co., Ltd.,
South Wing, Central Office Building, Zhuang Sheng Plaza,
No. 10 Xuanwumenwai Street, Xicheng District, Beijing,
Shijiazhuang City, Hebei Province, 050000, China
Email: chaozhang003@outlook.com
Email: junpengzhao004@outlook.com

Kaichen Zhang and Zixun Lu*

Marketing Business Department,
State Grid Siji Cybersecurity Technology (Beijing) Co., Ltd.,
Beijing City, 100052, China
Email: kaichenzhang005@outlook.com
Email: zixunlu111@outlook.com
*Corresponding author

Abstract: Advanced multi-stage cyberattacks increasingly threaten power information networks and can disrupt both communication and physical control systems. This study proposes a graph neural network (GNN)-based architecture to detect multi-stage attack paths and provide early warnings for disaster recovery. The framework models the spatio-temporal behaviour of network devices to improve resilience and support proactive cyber defence in critical power infrastructures. Traditional intrusion detection systems often fail to capture complex spatial and temporal relationships and sequential attack patterns, leading to slow detection and limited recovery capability. To address this limitation, a spatio-temporal graph neural network (ST-GNN) framework is

developed using the Kitsune network attack and HAI security datasets for comprehensive cyber-physical threat analysis. Experimental results demonstrate excellent performance with 99.98% accuracy, 99.90% precision, 99.97% recall, and an F1-score of 99.98%, with very low false positive and false negative rates. The proposed system effectively predicts multiple attack stages and significantly improves detection capability, enabling faster response and stronger protection for modern power information networks.

Keywords: spatio-temporal graph neural network; ST-GNN; power information network security; cyberattack detection; early warning system; disaster recovery planning.

Reference to this paper should be made as follows: Zhang, Y., Chen, S., Zhang, C., Zhao, J., Zhang, K. and Lu, Z. (2026) 'Power information network attack chain identification and disaster recovery early warning mechanism based on graph neural network', *Int. J. Intelligent Information and Database Systems*, Vol. 18, No. 6, pp.1–38.

Biographical notes: Yangrui Zhang holds a Master's degree. He works at State Grid Hebei Electric Power Company, in the Technology Development and Digitalization Department, as a Senior Engineer. His main research areas are: electricity metering, electricity consumption information collection, and big data analysis.

Shihui Chen works in the Marketing Business Department of State Grid Siji Network Security Technology (Beijing) Co., Ltd., as a Junior Engineer. He holds a Bachelor's degree from Peking University. His main research areas are power information system security protection and data security.

Chao Zhang received his Master's degree, and is a Senior Engineer, Marketing Service Center, State Grid Hebei Electric Power Company, Technology Development and Digitalization Department. His main research areas are electricity marketing informatisation, and electricity consumption information collection.

Junpeng Zhao received his Bachelor's degree, and is an Engineer, Marketing Service Center, State Grid Hebei Electric Power Company, Technology Development and Digitalization Department. His main research areas are power system and its automation, metering and data acquisition technology.

Kaichen Zhang works in the Marketing Business Department of State Grid Siji Cybersecurity Technology (Beijing) Co., Ltd., responsible for the top-level planning and design of power marketing business security. He holds a postgraduate degree, with a Bachelor's degree from the University of Chinese Academy of Social Sciences and Master's degree from the University of London. His main research areas are power industrial control security and data security.

Zixun Lu works in the Marketing Business Department of State Grid Siji Cybersecurity Technology (Beijing) Co., Ltd. He holds a Bachelor's degree from Oregon State University, USA. His main research area is network security protection for the state grid marketing system.

1 Introduction

Not only has the inclusion of IoT gadgets, smart metres, SCADA infrastructure, and cloud-based control frameworks raised the complexity of the contemporary power information networks, but made them an attractive cyber threat target (Halgamuge, 2024). Modern attackers employ multi-stage attack chains, as opposed to traditional attacks, where each stage of the attack is benign on its own but reveals its malicious intent when combined after a long period (Mighan and Kahani, 2021). Keeping up with this modern threat not only threatens the availability of networks but also has the potential to cause blackouts, power outages, economic meltdown, and even risks to national security (Chang et al., 2024). The interdependence of the smart grid infrastructure, both physical and cyber, requires new mechanisms for the identification of attack chains and proactive disaster recovery to maintain continuous and secure operations (Jmal et al., 2023).

Current AI and ML-based methods for anomaly and intrusion detection perform with great accuracy under laboratory settings but are severely constrained in scalability, real-time analysis, and adaptability to changing threats (Kotenko et al., 2021). Most of the models do not cope well with high-velocity and large-scale data streams, are not widely validated in real-world settings, and perform poorly in situations involving data paucity or incomplete domain knowledge (Liu et al., 2023). In addition, existing systems tend to underperform when faced with new attack patterns, advancing cyber threats, or operational uncertainty, which diminishes their performance in real-world, dynamic deployments (Sultana et al., 2022).

GNNs have already turned out to be useful tools in the context of modelling complex links in the interrelated systems (Alani and Damiani, 2023). Regarding information networks, nodes can represent either devices, processes, or network flows, and the relationships between them can be described using edges (Damaševičius et al., 2023). GNNs adoption gives an opportunity to look deeper into the network and identify anomalies at the node level and across multi-hop associations that reveal hidden attack patterns (Xu et al., 2021). The capabilities of the GNN to extract features and sequences are quite suitable to the task of attack chain reconstruction, with every malicious event being part of a carefully designed attack plan (Rehan, 2022).

In addition to identification, the strength of power information systems is determined by their capability to predict a failure and recover from it even before any catastrophic impact is experienced (Su et al., 2024). A proactive defense is made possible by an integrated disaster recovery early warning system with GNN-based detection (Tharewal et al., 2022). By forecasting possible attack paths, system operators are able to trigger automated containment, resource reallocation, and backup activation well before the attackers meet their definitive objectives (Musa et al., 2023). Early intervention in such scenarios minimises the downtime and, as a result, the financial and operational losses as well in accordance to the cyber-physical resilience of critical infrastructure (Ahmad et al., 2023).

For this study, the Kitsune network attack dataset offers a realistic and varied base for training and testing the proposed GNN-based attack chain detection model. Even though the dataset stems from IoT and IP surveillance platforms, its nine labelled attacks, including ARP MitM, fuzzing, and reconnaissance, typify several methods of genuine power information network integrations. Due to the dataset's 115 extracted statistical features and voluminous traffic captures, it is possible to model low-level packet

attributes and high-level relational dependencies, which makes it ideal for graph-based learning in cybersecurity.

The paper attempts to develop a scaled, adaptive and resilient detection system capable of operating on high accuracy, large, heterogeneous and high-speed data streams in real-time. In the proposed system, the complex feature extraction and a hybrid modelling approach will be used to deal with the damage to data unavailability, enhance coverage of domains, and offer flexibility to changing over time threats. Stringent validation shall be performed on heterogeneous real-world data and on real-time environments to ensure the robustness, scalability, and scalability to large scale implementation. The proposed study aims to apply a GNN to identify multi-stage attack chains and to apply it with an early warning system to the power information network disaster recovery. Cyberattacks that are coordinated and multi-stage are becoming more and more exposed to power information networks, jeopardising operational continuity and system stability. This paper overcomes such issues by deploying a spatiotemporal GNN architecture to identify the changing attack chains and aid in an early disaster-recovery decision-making. The main contributions are:

- A new graph-based attack model that elucidates spatial and temporal links in network traffic
- An attack chain reconstruction algorithm that converts the individual anomalies into consistent intrusion narratives,
- The early warning system module which predetermines the happenance of the disaster and the recovery steps to be taken.

This research intends to seal the fault between the detection and prevention to enhance the defense, reliability, and resiliency of the modern power information systems.

This study is segregated into five parts. Section 1 (introduction) explores the problem, the reason, and the target goals of a GNN-based solution that will be used in the detection of attack chains in power networks. Section 2 (related works) addresses the previous studies about the anomaly detection, early warning, and recovery their limitations. In section 3 (methodology), the datasets, pre-processing, graph construction, GNN model, early warning, and recovery procedures are described. Section 4 (results and Section 5 discussion) represents the performance of the model and how efficiently the recovery method works. Section 6 (conclusion and future work) summarises the lessons learned and suggests improvements to deal with real-time issues.

1.1 Novelty compared to GNN–LSTM hybrids

- Previous GNN-LSTM hybrids normally use spatial graph learning to generate a static or coarse graph embedding, which is fed through an LSTM. This sequential structure constrained the ability of the model to reason about time at the level of graph summaries and does not allow the model to spread time-conditioned messages at the granularities of nodes through more than one hop.
- The proposed ST-GNN does not follow this trend by executing joint spatio-temporal message passing: at each interaction step, node embeddings are updated through temporally-modulated neighbourhood aggregation instead of feeding an existing

graph embedding to a distinct temporal module. This enables the model to be able to compute the changing multi-hop dependencies as they occur rather than seen.

- In addition to joint propagation, ST-GNN uses continuous-time encodings (Δt-aware kernels) inside the message function so that recency and temporal density directly modulate message importance – a capability absent from standard LSTM post-processing where time is treated discretely and globally.
- The singular mathematical contribution is the time-conditioned neighbourhood aggregation operator: a message function that multiplies or gates incoming neighbour messages by a continuous temporal kernel and adaptive weights, producing time-aware, topology-sensitive updates. This operator unifies temporal recency and structural context into a single message-passing step, enabling richer temporal–structural embeddings that reconstruct cyber-physical attack progressions more accurately than separate GNN→LSTM pipelines.

2 Related works

Wu et al (2022). constructed a GNN-based anomaly detection framework designed for industrial internet of things (IIoT) use cases in smart energy, smart factories, and smart transportation (Mahi-al-rashid et al., 2022). Their framework managed to cover point, contextual, and collective anomalies and demonstrated their framework's applicability to industrial systems which evolve over time (Samah et al., 2020). Each of their experiments demonstrated the improvement of anomaly detection, showing that GNN is indeed appropriate for IIoT systems with heterogeneous data (Mahalakshmi et al., 2024). However, the framework struggled to scale to real-time large IIoT data streams. As Vitulyova et al. (2025). describe in their work, a reinforced attempt has been made to propose a hybrid cyber threat detection model using and LSTM models for attack vector reconstruction and prediction (Khan et al., 2024). With the GNN models, the structural relations of the MITRE ATT&CK framework were analysed, whereas the LSTM models were used to capture temporal attack relations (Paredes et al., 2021). Bolla et al. (2024) present GETNet, combining GNN-based relational learning with Transformer attention for high-accuracy fraud detection. Their graph–temporal integration guides our approach, inspiring similar spatiotemporal modelling to enhance anomaly detection in dynamic networked environments. Their experiments on the CICIDS2017 dataset showed an AUC of 0.99, F1-score of 0.85, and an MSE of 0.05 in the attempts to reconstruct attack paths and in risk prediction, which clearly illustrates the high accuracy of the model (Cao, 2023). Nevertheless, the model's computational complexity may restrict its implementation in real-time on a larger scale. An Chondros et al. (2021). proposed an integrated approach for the development of a coastal flood early warning system that uses the hindcast framework for the validation of high credibility numerical models, and a forecast framework to use an ANN trained on the combined multiple sea-state scenarios outputs of the numerical models (Jing et al., 2024). The artificial neural network (ANN) was able to prepare timely and accurate risk scarcification of the coastal floods using the offshore waves characteristics and sea water level elevation data only as inputs in the region of Rethymno, Crete (Dalal et al., 2023). This approach proved the possibility to use it on other different coastal zones and areas as it also eliminates the need for the long and expensive simulations used in the previous methods (Kathamuthu et al., 2022).

Nevertheless, the scarcity of the historical flood data may limit the accuracy of the model in certain locations.

Sharma et al. (2021) introduced a hybrid model of cybersecurity based on GNNs and LSTM networks to predict and reconstruct attack vectors (Huang, 2022). GNNs examined structural relations within the MITRE ATT&CK framework and LSTMs represented temporal attack patterns (Pasetti et al., 2021). On the CICIDS2017 dataset, experiments resulted in AUC = 0.99, F1-score = 0.85, and MSE = 0.05, showing high accuracy in attack path reconstruction and risk prediction (Ni et al., 2021). However, the model's computational overhead may limit real-time deployment in large-scale environments. Rahman (2022) suggested an AI/ML-based proactive system to boost supply chain resilience against pandemics, disasters, cyberattacks, and geopolitical crises (Alkadi et al., 2021). The system integrates emergency threat detection, dynamic impact simulation, adaptive response engineering, and resilience monitoring based on IoT, satellite images, social media and government alerts to obtain real-time analytics (Noorazar et al., 2021). Disruption response and resource allocation are optimised using reinforcement learning and graph neural networks and are made transparent with the help of blockchain (Elvas et al., 2021). However, it is yet to be practically applied in large-scale crises. Jung et al. (2020) created a South Korean IDMS, which does not focus on the storms, floods, and earthquakes but manages wildfires and extreme temperature events as well (Wei and Lee, 2024). The system uses big data of open APIs and AI algorithms to accelerate decision-making and increase the speed and accuracy of decisions, and includes a CNN-based fire detection subsection to track video (Gao et al., 2022). Possible integration with OSINT is proposed to detect vulnerabilities and ward off cyber-attacks. Still, the system is at a conceptual level, with minimal real-world application and testing.

Zhou et al. (2024) proposed a lightweight anomaly detection framework for IoT networks, RG-GLD, which integrates GNN and knowledge distillation principles (Morales-Molina et al., 2021). The approach reconstructs IoT communications into directed graphs, extracting structural features through GAT and traffic features through MLP for improved feature fusion (Guato Burgos et al., 2024). A local subgraph preservation mechanism and global information alignment enhance efficiency with superior classification accuracy and lower computational burden over four baseline methods (Guato Burgos et al., 2024). Yet the evaluation of the study is restricted to two public datasets, failing to test its performance in varied real-world IoT scenarios. Karthika and Senthilselvi (2023) introduced a one-dimensional DCNN for credit CCFD to alleviate the challenges of data imbalance, excessive false alarms, and dynamic fraud modes (Liu et al., 2022). The model improves on traditional CNN by integrating dilated convolutional layers to extract spatial and temporal features, balanced with imbalance prevention through under and over-sampling (Achaal et al., 2024). Tests across three databases indicated the DCNN reaching 97.39% accuracy on a small card database, beating a baseline CNN's 94.44% (Ferrag et al., 2021). But its usability on real-time streaming transaction data is yet to be tested.

Wu et al. (2022) showed that although their GNN-based anomaly detection model for IIoT performed well with several types of anomalies, it lacked scalability to real-time large IIoT data streams. Vitulyova et al. (2025) and Sharma et al. (2021) presented high accuracy in cyber threat prediction with hybrid GNN–LSTM models, but both were plagued by high computational complexity, which prevented real-time deployment in large-scale systems. Chondros et al. (2021) demonstrated that their ANN-based early

warning system for coastal floods could be extended to other areas but that its accuracy was limited by the paucity of historical flood data. Rahman et al. (2022) designed an AI/ML-based proactive supply chain resilience framework but in the absence of real-world validation in large-scale crisis situations. Jung et al. (2020) proposed an IDMS for South Korea with enhanced disaster coverage, although it was still largely conceptual with very little field testing. Zhou et al. (2024) also proposed the RG-GLD framework for light-weight IoT anomaly detection, but its performance was only tested using two public datasets, which limited generalisability to real-world scenarios. Lastly, Karthika and Senthilselvi (2023) enhanced CCFD accuracy using a DCNN but the performance of the model in real-time streaming transactions is unknown. According to the existing research of the problem of power grid security, there is a necessity of sophisticated anomaly detection systems, but most of them are incapable of capturing the multi-hop and time-based attack development. Recent GNN-based models are more sophisticated in terms of structural modelling, but without incorporated early-warning systems and recovery functions on cyber-physical power networks.

3 Methodology

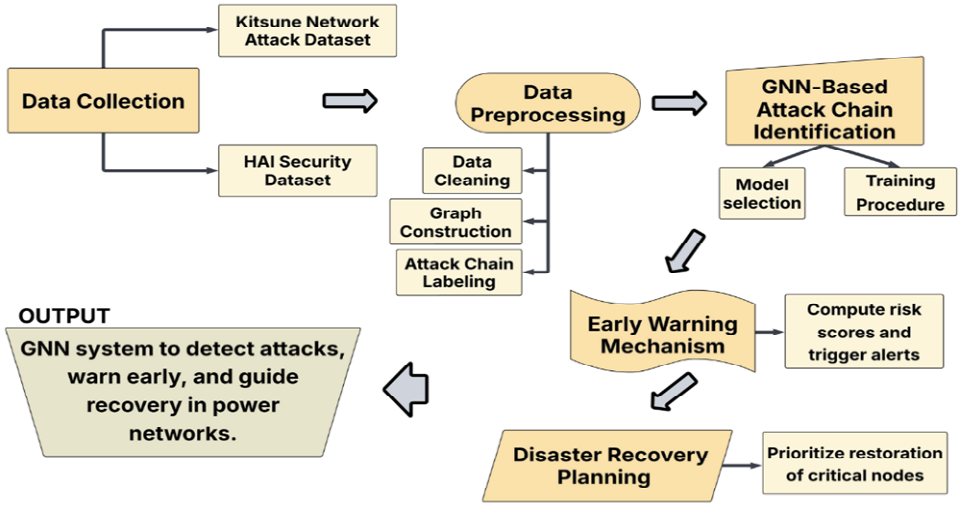
This research formulates a method using the HAI Security and Kitsune datasets in order to identify and respond to cyber threats that target electrical networks. The methodology develops spatiotemporal graphs based on power network traffic and SCADA data, which allows the ST-GNN to support dynamic patterns of attacks between devices. The model then creates risk scores of early warning and recovery actions based on node criticality and attack-chain prediction. The first step is pre-processing the data and converting it into network graphs that showcase attack sequences. Afterwards, a GNN model is developed with the purpose of understanding the cyberattack sequences in the spatial and temporal domains. An early warning system is then developed and, based on this model, it is able to compute risk scores and send out warnings. In the end, a recovery plan is implemented to ensure that the critical nodes are recovered as fast as possible, thus reducing the recovery time and enhancing the network security.

This diagram depicts the flow beginning with data collection and pre-processing in order to conduct an attack chain identification. It further depicts the early warning and disaster recovery planning. The framework uses a GNN model in order to detect attacks issued for alerts and recovery actions prioritisation in power networks as shown in Figure 1.

3.1 Data collection

The Kitsune dataset was gathered from observing network traffic in two settings: an IP-based commercial monitoring system and a network filled with IoT devices. Data was captured at the level of the routers, at points where an intrusion detection system might realistically be placed. For every one of the nine attack cases, clean traffic was captured for the first million packets to serve as a baseline, followed by a cyberattack being conducted. The raw PCAP as well as the processed features – 115 real-time statistical network features extracted using the AfterImage feature extractor – were stored along with related attack labels for supervised machine learning (Ymirsky, 2020).

Figure 1 Flow diagram of GNN-based power network attack detection and recovery framework (see online version for colours)



The HAI dataset was produced from a realistic ICS testbed with an added HIL simulator modelling steam-turbine power generation and pumped-storage hydropower. The testbed connects physical control systems (boiler, turbine, and water treatment) with simulated processes using OPC-UA gateways to enable synchronised cyber-physical interaction. Normal operation and attack scenarios were run over lengthy continuous runs, and measurements were taken from standard SCADA sensors and controllers at regular time intervals. Various datasets were generated with normal and attack duration marked with signs, over a wide range of ICS operating conditions and the types of cyber-attacks ('HAI Security Dataset', 2023).

The concurrent application of the HAI and Kitsune data is deliberate, given that this way, the suggested ST-GNN structure will be able to train on complementary features of the cyber-physical threat behaviour. Kitsune offers finer-grained patterns of network traffic, detailed packet-level variations of statistics, and the variety of types of attack that is a common feature of the IT and IoT worlds. Conversely, the HAI dataset measures process-level dynamics and physical-state deviations in an industrial control system that is representative of the impacts of cyber intrusions on the operational variables in power systems. Despite the difference between the domains, the similarity between the two datasets is in the fact that they have similar underlying temporal-relational structures that can be used in graph modelling. Their joint heterogeneity enables the model to see how the patterns of attack occur both in pure network traffic and also in the cyber-physical interactions. The ST-GNN achieves more resilient spatio-temporal representations by learning on data with topological, device, and attack semantic variations, and does not overfit to a specific environment. Such a mixed domain exposure is beneficial to the generalisation capacity of the model and to its capacity to identify multi-staged attacks in real power networks of operation, where cyber and physical processes are skewed closely together.

Combining SCADA sensor data with HAI dataset increases the temporality of the model through the addition of continuous data on the behaviour of physical processes, including pressure, flow, valve state, and actuator position. These variables contribute to

a complementary signal of time to the bare network traffic characteristics, which makes the model track how cyber activities can cause changes in physical states over time. The ST-GNN is capable of capturing physically related anomalies that occur at many stages of attack, including when an intrusion initially affects communication patterns, and then later an intrusion modifies sensor or actuator behaviour. Such fusion process enhances the temporal embeddings, which establish the cause-and-effect associations between the cyber activities and the reactions of the physical systems, so that the model can discover the abnormal evolution patterns, which would not be seen when using traffic-based data alone. As a result, the inclusion of HAI sensor streams strengthens the model's ability to detect coordinated cyber–physical attack chains.

3.2 Data pre-processing

- Notation: raw packet stream: $P = \{p_k\}_{k=1}^{N}$, each packet p_k has fields: timestamp t_k' source IP s_k' dest IP d_k' src MAC, dst MAC, protocol π_k', length l_k, and the Kitsune 115-dim feature vector $f_k \in R^{115}$ if using pre-processed CSV, time window length: W (seconds). Slide (stride): S (seconds), graph snapshot index: t (discrete windows), window time interval $[T_t, T_t + W)$, set of unique hosts (IP or MAC) in window t: V_t, edges in window t: E_t, node feature matrix for window t: $X_t \in R^{|V_t| \times d}$, adjacency/edge-weight matrix for window t: $A_t \in R^{|V_t| \times |V_t|}$, attack labels per packet: $l_k \in L$ (e.g., clean, scan, DoS, etc.). Attack stage mapping function: S: $L \rightarrow \{1, \ldots, C\}$ (e.g., 1 = recon, 2 = initial intrusion, 3 = privilege escalation, 4 = impact), node label at time t: $y_{v,t}$.

3.2.1 Data cleaning

- Duplicate removal: if packets have identical 5-tuples and timestamps, remove duplicates as shown in equation (1)

$$Keep\ p_k\ if\ f\ \nexists j < k s.t. \left(t_j = t_k \wedge s_j = s_k \wedge d_j = d_k \wedge \pi_j = \pi_k \wedge l_j = l_k\right) \tag{1}$$

- Missing packets/interpolation: If packet times for a given flow are sparse but you need continuous flow rate features, compute perhost/flow packet counts per bin and interpolate missing bins with zero or linear interpolation.

  Let $C_v(t)$ be packet count for host v in second t. If a continuous sequence has missing seconds, fill as shown in equation (2):

$$\tilde{C}_v(t) = \{C_v(t)\ if\ exists\ 0\ (or\ linear\ interp)\ otherwise \tag{2}$$

- Outlier handling on features: for each raw feature dimension i, apply winsorisation or z-score clipping as shown in equation (3):

$$\mu_i = mean\left(\{f_{k,i}\}\right), \sigma_i = std\left(\{f_{k,i}\}\right) \tag{3}$$

- Clip:

$$f_{k,i} \leftarrow clip\left(f_{k,i}, \mu_i - \alpha\sigma_i, \mu_i + \alpha\sigma_i\right) \tag{4}$$

where α (e.g., 3) is chosen as shown in equation (4).

- Normalisation/scaling: for numerical stability, scale features per host or global as shown in equations (5) and (6):
- Min-max:

$$f'_{k,i} = \frac{f_{k,i} - \min_i}{\max_i - \min_i} \tag{5}$$

- Standard:

$$f'_{k,i} = \frac{f_{k,i} - \mu_i}{\sigma_i} \tag{6}$$

In this work, min–max normalisation and z-score clipping were adopted to ensure stable and consistent feature behaviour during GNN training. The network-traffic features in both Kitsune and HAI datasets show significant variation in scale, with certain packet-level statistics reaching disproportionately high values during attack bursts. Min–max normalisation compresses all dimensions into a uniform range, preventing features with large numeric magnitudes from dominating the graph convolution operations and improving gradient stability during aggregation. Z-score clipping further prevents extreme outliers common in network spikes, malformed packets, and flooding attacks from distorting the statistical distribution of node and edge attributes. By limiting values to a reasonable range of standard deviations, the model avoids gradient explosions and learns smoother decision boundaries. Together, these pre-processing techniques promote faster and more stable convergence of the ST-GNN, reduce noise sensitivity, and enhance robustness across multi-stage attack scenarios.

3.2.2 Graph construction

We produce a temporal sequence of graphs $\{G_t\}_{t=1}^{T}$ where each snapshot shown in equation (7).

$$G_t = (V_t, E_t, X_t, A_t) \tag{7}$$

represents network interactions during window $[T_t, T_t + W)$.

3.2.2.1 Node definition

Nodes represent network entities (choose IP or MAC, or both as bipartite nodes). Let $V_t = \{v_1, \ldots, v_{n_t}\}$.

Mapping function from packet to node index as shown in equation (8):

$$map\_node(p_k) = \{idx(s_k), idx(d_k)\} \tag{8}$$

3.2.2.2 Edge definition

Define an (undirected or directed) edge (u, v) if there is at least one packet between u and v in window t as shown in equation (9):

$$E_t = \left\{(u,v) \mid \exists p_k : s_k = u \wedge d_k = v \wedge t_k \in \left[T_t, T_t + W\right)\right\} \tag{9}$$

3.2.2.3 Edge attributes and weights

Compute edge features for (u, v) in window t as shown in equations (10) and (11):

- Packet count (frequency):

$$count_{u,v}^{(t)} = \sum\nolimits_{k:s_k=u,d_k=v,t_k\in[T_t,T_t+W)} 1 \tag{10}$$

- Average packet size:

$$\bar{l}_{u,v}^{(t)} = \frac{1}{count_{u,v}^{(t)}} \sum\nolimits_k l_k \tag{11}$$

- Protocol distribution vector $\pi_{u,v}^{(t)}$ (one-hot or frequency for common protocols).

  Combine into an edge attribute vector $e_{u,v}^{(t)}$. To get scalar weight for adjacency as shown in equation (12):

$$A_t[u,v] = w_{freq} \frac{count_{u,v}^{(t)}}{\max\limits_{(a,b)} count_{a,b}^{(t)}} + w_{size} \frac{\bar{l}_{u,v}^{(t)}}{\max\limits_{(a,b)} \bar{l}_{a,b}^{(t)}} \tag{12}$$

  with $w_{freq} + w_{size} = 1$ (hyperparameters).

  Alternatively, use mutual information between hosts' packet sequences for edge weight as shown in equation (13):

$$A_t[u,v] = I\left(X_u, X_v\right) \ (\textit{estimated from packet time series}) \tag{13}$$

3.2.2.4 Node attributes (aggregating Kitsune features)

Kitsune provides per-packet feature vector f_k. Aggregate per host v in window t to create $x_{v,t}$ of dimension d.

Common aggregations as shown in equations (14), (15), (16), (17), (18), (19):

- Count:

$$cnt_v^{(t)} = \sum\nolimits_{k:s_k=vord_k=v} 1 \tag{14}$$

- Mean of feature dimension i:

$$\mu_{v,i}^{(t)} = \frac{1}{cnt_v^{(t)}} \sum\nolimits_{k:v\in\{s_k,d_k\}} f_{k,i} \tag{15}$$

- Variance:

$$\sigma_{v,i}^{2(t)} = \frac{1}{cnt_v^{(t)}} \sum_k \left(f_{k,i} - \mu_{v,i}^{(t)} \right)^2 \tag{16}$$

- Flow rate (pkts/sec):

$$r_v^{(t)} = \frac{cnt_v^{(t)}}{W} \tag{17}$$

- Entropy of destination distribution:

$$H_{dst,v}^{(t)} = -\sum_z p_{v\to z}^{(t)} log p_{v\to z}^{(t)}, p_{v\to z}^{(t)} = \frac{count_{v\to z}^{(t)}}{cnt_v^{(t)}} \tag{18}$$

- Concatenate to form node vector (example):

$$x_{v,t} = \left[\mu_{v,1}^{(t)}, \ldots, \mu_{v,115}^{(t)}, \sigma_{v,1}^{(t)}, \ldots, \sigma_{v,115}^{(t)}, r_v^{(t)}, H_{dst,v}^{(t)}, role_v \right] \in R^d \tag{19}$$

3.2.2.5 Temporal graph snapshots

For sliding window index t with start $T_t = T_0 + (t–1)\cdot S$, build snapshot G_t as above. The sequence $\{G_1, \ldots, G_T\}$ is input into temporal GNN models.

3.2.3 Attack chain labelling (mapping Kitsune types → stages)

You want to convert Kitsune attack-type labels into attack stages and produce node-level stage labels for supervised/semi-supervised training.

3.2.3.1 Define stage mapping

Let Kitsune label set $L = \{l^{(1)}, \ldots, l^{(m)}\}$. Define mapping as shown in equation (20).

$$S : L \to \{1, \ldots, C\} \tag{20}$$

Example mapping (illustrative):

- Reconnaissance (1): portscan, service scan, OS scan ⇒ $S(l) = 1$
- Initial Intrusion (2): exploit, injection ⇒ $S(l) = 2$
- Privilege escalation/lateral movement (3): ARP spoofing, man-in-the-middle, credential abuse ⇒ $S(l) = 3$
- Impact/DoS/data exfil (4): DoS, data exfiltration, disruption ⇒ $S(l) = 4$.

  You store mapping as a dictionary in code.

3.2.3.2 Window-level node label assignment (deterministic)

For node v in window t, collect packet labels:

$$L_{v,t} = \left\{ l_k \middle| \left(s_k = v \; or \; d_k = v \right), t_k \in \left[T_t, T_t + W \right) \right\} \tag{21}$$

If $L_{v,t} = \varnothing$, label as clean $S = 0$. Otherwise, define node stage label $y_{v,t}$ as shown in equation (21):

- Max-stage rule (conservative, captures most severe seen) as shown in equation (22):

$$y_{v,t} = \max_{l \in L_{v,t}} S(l) \tag{22}$$

- Most-frequent stage as shown in equation (23):

$$y_{v,t} = \arg\max_{c} \left\{ l \in L_{v,t} : S(l) = c \right\} \tag{23}$$

- Multi-label: represent as one-hot vector $y_{v,t} \in \{0, 1\}^C$ as shown in equation (23).

$$y_{v,t}[c] = 1\left\{ \exists l \in L_{v,t} \;\; s.t. \; S(l) = c \right\} \tag{24}$$

3.2.3.3 Transition/chain labelling (for chain reconstruction)

If you want labelled attack chains (i.e., sequences), create event list per host ordered by time as shown in equation (25):

$$E_v = \left(\left(t_{v,1}, c_{v,1} \right), \left(t_{v,2}, c_{v,2} \right), \ldots \right), c_{v,1} = S\left(l_{v,i} \right) \tag{25}$$

A multi-host chain is a sequence of stage-labelled host events across the network ordered by timestamp as shown in equation (26):

$$C = \left(\left(v_1, c_1, \tau_1 \right), \left(v_2, c_2, \tau_2 \right), \ldots, \left(v_m, c_m, \tau_m \right) \right), \tau_i < \tau_{i+1} \tag{26}$$

You can generate chains by linking events where edge interactions exist within a temporal proximity Δ as shown in equation (27):

$$Link\left(v_i, \tau_i \right) \to \left(v_j, \tau_j \right) \Longleftrightarrow \exists \left(v_i, v_i \right) \in E_{\lfloor \pi / W \rfloor} \wedge \left(\tau_j - \tau_i \right) \le \Delta \tag{27}$$

3.2.3.4 Empirical transition probabilities (Markov model)

Estimate stage transition probabilities from data as shown in equation (28):

$$\hat{P}\left(c_{t+1} = j \mid c_t = i \right) = \frac{\mid \#observed \;\; transitions \;\; i \to j}{\sum_k \mid \#observed \;\; transitions \;\; i \to k} \tag{28}$$

For node-to-node stage transitions, estimate conditional probabilities as shown in equation (29):

$$\hat{P}\left(\left(v_j, c_j, \tau_j \right) \middle| \left(v_i, c_i, \tau_i \right) \right) \approx \frac{\#\left\{ i \to j \;\; events \;\; within \;\; \Delta \right\}}{\#\left\{ i \;\; events \right\}} \tag{29}$$

3.2.3.5 Chain probability score

Given a candidate chain $C = (c_1, \ldots, c_m)$, compute chain probability under Markov assumption:

$$P(C) = \pi_{c_1} \prod_{i=1}^{m-1} Pn\left(c_{i+1} \middle| c_i\right) \tag{30}$$

where π_{c_1} is empirical prior probability of starting in stage c_1 as shown in equation (30).

You can threshold $P(C)$ to select likely attack chains for supervised training.

3.2.3.6 Additional useful computations/equations

Risk score for node v at time t

Use model output or heuristic:

$$R_{v,t} = \sigma\left(w^{\top} h_{v,t} + b\right) \tag{31}$$

where $h_{v,t}$ is node embedding from GNN and σ is sigmoid. Trigger alert if $R_{v,t} > \tau$ as shown in equation (31).

Cross-layer edge creation (if integrating HAI later)

Map network device v to ICS node u via function ϕ: $V_{net} \rightarrow V_{ICS}$. Create cross edges (v, u) with weight proportional to control impact.

Loss function for stage classification + chain prediction

- Node stage classification loss (cross-entropy) as shown in equation (32):

$$L_{stage} = -\frac{1}{\sum_t |V_t|} \sum_t \sum_{v \in V_t} \sum_{c=1}^{C} y_{v,t,c} log\ \hat{y}_{v,t,c} \tag{32}$$

- Chain prediction (next-stage) loss (cross-entropy over predicted next stage given current embedding) as shown in equations (33) and (34):

$$L_{next} = -\sum_{(v,t)} \sum_{c'} z_{v,t+1,c'} log \hat{z}_{v,t+1?v,t,c'} \tag{33}$$

- Full loss:

$$L = \lambda_1 L_{stage} + \lambda_2 L_{next} + \lambda_3 L_{reg} \tag{34}$$

Pseudocode for data pre-processing

Step 1. *Load dataset*

Graph snapshots G_t (*nodes* V, *edges* E_t)

Temporal features (*flow stats, packet metadata, attack indicators*)

Step 2. *Initialise model*

Spatial encoder: GCN or GraphSAGE

Temporal encoder: TGN or TGAT

Step 3. For each training epoch:

For each time step t:

a. Extract spatial features:

h_{struct} = *SpatialEncoder*(G_t)

b. Extract temporal features:

z = TemporalEncoder(h_struct, Δt, temporal_context)

c. Predict:

Current attack stage

Next probable attack stage

d. Compute losses:

L_cls = CrossEntropy(y_true_current, y_pred_current)

L_seq = CrossEntropy(y_true_next, y_pred_next)

e. Combine losses:

*L_total = λ1 * L_cls + λ2 * L_seq*

f. Backpropagate and update weights

Step 4. Output:

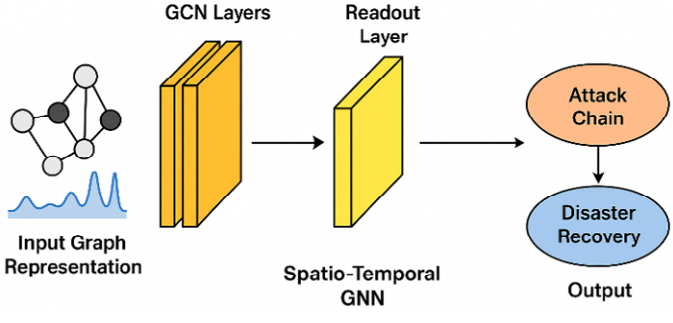
Predicted attack stage for each node

Predicted next attack stage in the chain

3.3 GNN-based attack chain identification

The diagram shows how to identify sequences of attacks in a power information network through the use of GNNs. This identification then aids in suggesting appropriate disaster recovery measures, which may include the isolation of certain nodes and the prioritisation of their restoration as shown in Figure 2.

Figure 2 Architecture for GNN-based power network attack chain identification and recovery (see online version for colours)



3.3.1 Model selection

Employ a ST-GNN to capture both the structural topology of the network and the temporal evolution of attack patterns.

3.3.1.1 Graph topology learning

GCN or GraphSAGE is used to extract spatial features from the network graph:

$$h_v^{(l+1)} = \sigma\left(W^{(l)} \cdot AGG\left(\left\{h_v^{(l)}\right\} \cup \left\{h_u^{(l)}, u \in N(v)\right\}\right)\right) \tag{35}$$

where

$h_v^{(l)}$ embedding of node v at layer l

$N(v)$ neighbours of node v

AGG aggregation function (mean, max, sum)

$W^{(l)}$ learnable weight matrix

σ activation function (ReLU) as shown in equation (35).

3.3.1.2 Temporal pattern learning

Temporal graph network (TGN) or temporal graph attention network (TGAT) models the time evolving attack behaviour:

$$z_v(t) = f_\theta\left(h_v^{struct},\ \Delta t, context_v(t)\right) \tag{36}$$

where

h_v^{struct} structural embedding from GCN/GraphSAGE

Δt time since last interaction

$context_v(t)$ neighbourhood temporal context as shown in equation (36).

If the dataset distinguishes between device types (loT, router, camera, etc.) and attack stages, we use a HetGNN to learn type-specific embeddings.

Unlike traditional intrusion detection models such as WNN, CIC-IDS2017, and BiGGCN, the proposed ST-GNN is designed to capture multi-hop and higher-order dependencies that naturally arise during multi-stage cyberattacks. In power information networks, malicious behaviour does not remain localised to a single device; instead, it propagates across several hops through lateral movement, privilege escalation, and coordinated interactions. The spatial aspect of the ST-GNN recursively combines information on many layers, allowing every node to receive a signal in context of its immediate neighbours, but also get a signal about more distant nodes several hops away. This implies a device can learn the impact of remote nodes of which their behaviour adds to a previous step in the attack chain. This is further supported by the temporal encoder that maintains the changing pattern of interactions and the model is able to trace the evolution of the abnormal patterns through time and propagation throughout the network topology. NN-based approaches like WNN use local statistical characteristics mostly and do not model multi-hop propagation, whereas CIC-IDS2017 uses only packet-based signatures without directly considering network topology. Even BiGGCN, which is graph based, is restricted to comparatively shallow structural neighbourhoods and fails to incorporate temporal dependencies. Through joint modelling of spatial relations and temporal dynamics, the ST-GNN is able to efficiently reconstruct higher-order, long-range attack patterns and much finer interactions not noticed by traditional models, leading to higher reconstruction of multi-stage attack sequences.

The reasons behind the use of aggregation function in GraphSAGE is also significant as it directly influences the accuracy of the learning of spatial features over several hops of neighbourhoods. The mean aggregator is chosen in this work since it gives a balanced and scale-invariant representation of all the neighbouring nodes, which is essential in the

power information networks where the intensity of traffic and connectivity between the devices are highly disparate. Sum aggregation will inflate feature values on high-degree nodes and will tend to overemphasise local anomalies whereas max aggregation only maintains the strongest signal and can ignore smaller but significant deviations, which are indicative of early-phase intrusions. The mean aggregator strikes a balance between the two extremes by normalising neighbourhood data and generating less spikey and more trustworthy embeddings that capture the behaviour of collective nodes as opposed to isolated spikes. This results in an improved convergence in the training process and helps the model to capture multi-hop relationships that are learned over time as the attack propagation takes place. Consequently, mean aggregation leads to better spatial representation accuracy and better overall tenacity of the ST-GNN to identify multi-stage cyberattacks.

The ST-GNN model is selected due to the inherent integration of spatial dependencies and time evolution, which enables the model to discover how the behaviour of the attack can propagate within the network by multiple network hops over time. Even though HetGNN is capable of the heterogeneous type of nodes, it lacks explicit models of fine-grained temporal transitions that are critical to the detection of sequential attack phases in power communication networks. Equally, graph attention networks boost neighbour weighting, but more focus on spatial attention not long-terminal patterns. Conversely, the ST-GNN employed in this study concurrently represents structural topology with time-varying interactions, which, in turn, allows it to detect subtle multi-hop dependencies and rebuild attack chains that evolve better. This renders ST-GNN to be more appropriate and precise in the context of spatio-temporal modelling of cyberattacks in power information networks.

3.3.1.3 Temporal attention mechanism

TGAT temporal attention mechanism plays an important role in modelling temporal changes in attacks. TGAT models time difference Δt between events using a time-continuous kernel which allows the model to more strongly emphasise recent interactions at the same time as being able to capture the long-range dependencies of multi-stage intrusions. This coding makes the difference between sudden outbreaks of maliciousness and the gradually formed attack behaviours in the model. By combining Δt with neighbourhood context, TGAT aggregates temporally adjacent messages from connected devices, allowing it to track how abnormal patterns propagate across the network. This joint treatment of temporal proximity and structural influence produces richer spatio-temporal embeddings that significantly improve the prediction of the next probable attack stage. As a result, TGAT supports the ST-GNN in reconstructing fine-grained progression patterns within evolving attack chains.

3.3.2 Training procedure

Input

- Graph snapshots: $G_t = (V, E_t)$, where V is the set of devices and E_t is the set of connections at time t.
- Temporal features: flow statistics, packet metadata, and extracted attack indicators.

Output

- Predicted attack stage for each node.
- Predicted probable next attack stage in the chain.

3.3.2.1 Loss functions

- Node-level attack stage classification loss
  a Cross-entropy loss:

$$L_{cls} = -\frac{1}{|V|}\sum_{v\in V}\sum_{k=1}^{K} y_{v,k} log\hat{y}_{v,k} \tag{37}$$

where:

K number of attack stages

$y_{v,k}$ ground truth (one-hot) label for node v

$\hat{y}_{v,k}$ predicted probability for stage k as shown in equation (37).

- Sequence prediction loss
  a Predicting the next attack stage as a sequence modelling problem:

$$L_{seq} = -\sum_{t=1}^{T-1}\sum_{k=1}^{K} y_{t+1,k} log\hat{y}_{t+1,k} \tag{38}$$

where $y_{t+1,k}$ and $\hat{y}_{t+1,k}$ are the true and predicted next-stage probabilities as shown in equation (38).

- Total loss
  a Combined objective:

$$L_{total} = \lambda_1 L_{els} + \lambda_2 L_{seq} \tag{39}$$

where λ_1, λ_2 are balancing weights as shown in equation (39).

Pseudocode for GNN-based attack chain identification process

Step 1. Initialise:

Model parameters (*GCN/GraphSAGE + TGN/TGAT*)

Node memory for temporal updates

Loss weights λ_1, λ_2

Step 2. For each training epoch

a. For each time step t in dataset

i. Load graph snapshot $G_t = (V, E_t)$

Load node features $X_v(t)$ *and labels* $y_v(t)$

ii. Structural encoding:

Use GCN or GraphSAGE to compute h_v^{struct}

$h\_v\text{^}struct = \sigma(W*AGG(\{h_v\} \cup \{h_u : u \in N(v)\}))$

iii. Temporal Encoding:

Use TGN/TGAT to update memory based on Δt *and context*

iv. Predictions:

Current stage: $\hat{y}\_v(t) = Softmax(W_{cls} * z_v(t))$

Next stage: $\hat{y}\_v(t+1) = Softmax(W_{seq} * z_v(t))$

v. Loss Calculation:

$L\_cls = CrossEntropy(y_v(t), \hat{y}_v(t))$ *# eq. (37)*

$L\_seq = CrossEntropy(y_v(t+1), \hat{y}_v(t+1))$ *# eq. (38)*

$L\_total = \lambda_1 * L_{cls} + \lambda_2 * L_{seq}$ *# eq. (39)*

vi. Backpropagation:

Update model parameters to minimise L_{total}

Step 3. After training

For each node, predict attack stages over time

Build attack chains by linking predicted stages in temporal order

Step 4. Output:

Predicted attack stage for each node at each time

Probable next attack stage

Attack chain sequences for compromised nodes

During training, the balancing weights λ_1 and λ_2 are applied directly during backpropagation so that the gradients from the stage-classification loss and the sequence-prediction loss contribute in the intended proportion. This prevents one objective from dominating the optimisation process. The Adam optimiser is used because it handles variations in graph neighbourhood size and feature scales efficiently, which stabilises gradient updates. A gradual learning-rate schedule is also applied to speed up convergence in the initial epochs and slow it down as the model approaches a minimum. The combination of these settings makes the joint loss smooth and well-converging to ST – GNN.

In the model, there is an early-warning calibration step that is used to make sure that the softmax output has meaningful probability scores and not raw confidence spikes. This scaling corrects the logits using temperature scaling so that early-stage attack indicators give a smoother and easier to interpret probabilities that can be used to make timely alerts. Simultaneously, severity weighting is used when calculating the loss to deal with the disproportion between frequent benign or minor incidents and infrequent but high impact phases of cyber-physical attacks. The training process ensures that the model does not underestimate the critical transition in an attack chain by increasing the importance of these low-frequency, high-severity states. Combined, early-warning calibration and severity weighting assist the ST-GNN to produce credible probability prediction, and to be sensitive to discriminative, yet uncommon phases of developing multi-stage intrusions.

3.3.3 Model hyperparameters

In order to allow the proposed ST-GNN model to be reproduced, the entire hyperparameter setting of the present study is explained. The network is given a spatial component which uses two layers of GCN/GraphSAGE, and the hidden representation is increased to 256 dimensions. The use of neighbour feature fusion is done on a mean aggregation strategy with ReLU activation and the dropout rate is 0.2 and layer

normalisation is used to stabilise the updates. The Xavier uniform scheme is used to initialise all weights in order to keep the training consistent. In the case of temporal modelling, the TGN/TGAT encoder will be set to a 256-dimensional memory, four attention heads, and a message function takes the form of a two-layer MLP with 128 units per layer and ReLU activation. Temporal information is encoded using a 16-dimensional sinusoidal time embedding, and attention dropout is fixed at 0.1. A GRU-based memory updater processes sequential interactions, and the temporal look-back window incorporates the five most recent graph snapshots.

The training process uses the Adam optimiser with a learning rate of 0.001 and a weight decay of 1e-5. Experiments are conducted with a batch size of 32 and across 40 epochs, while the combined loss is controlled by two balancing factors, assigning 0.7 to the stage-classification loss and 0.3 to the sequence-prediction loss. During graph construction, each snapshot is generated using a sliding window of two seconds with a one-second stride, and edge weights are computed by combining packet-frequency and packet-size contributions with coefficients of 0.6 and 0.4 respectively. Providing these hyperparameters establishes a complete and reproducible configuration for multi-stage attack chain detection using the proposed ST-GNN model.

3.4 Early warning mechanism

Notation

- $G_t = (V_t, E_t)$: network graph at time t
- $h_v(t) \in R^d$: temporal GNN embedding for node v
- $S = \{1, \ldots, K\}$: ordered attack stages (low → high severity)
- $p_{v,s}(t) = Pr(\text{ stage} = s \mid \text{data up to } t)$: calibrated class probabilities
- $w_s > 0$: severity weight for stage s
- $f_v(t) \in [0, 1]$: historical attack frequency estimate (EMA)
- $\beta \in [0, 1]$: propagation factor, $\gamma_{v,u}(t)$ normalised edge influence, $\tau_{on} > \tau_{off}$ hysteresis thresholds as shown in equations (40) to (47).

3.4.1 From embeddings to calibrated probabilities

$$h_v(t) = T - GNN(\cdots),\ \tilde{p}_v(t) = softmax\left(W_p h_v(t) + b_p\right) \tag{40}$$

Calibrate $\tilde{p}_v(t)$ (temperature scaling/isotonic regression) to obtain $p_{v,s}(t)$.

3.4.2 Severity score

$$sev_v(t) = \sum_{s=1}^{K} w_s p_{v,s}(t) \tag{41}$$

(Select w_s so later stages have distinctly larger weight; normalise during experiments.)

3.4.3 Historical prior (streaming)

Exponential moving average (EMA) for robustness:

$$f_v(t) = \alpha \cdot 1\{attack\ observed\ at\ v\ at\ t\} + (1-\alpha) f_v(t-\delta) \tag{42}$$

with $\alpha \in (0, 1)$.

3.4.4 Composite risk (canonical form)

$$R_v(t) = sev_v(t) \cdot \left(f_v(t) + \in \right) \tag{43}$$

where $\in > 0$ is a small constant to prevent zeroing (e.g., 10^{-6}).

3.4.5 Propagated network risk

$$\tilde{R}_u(t) = R_u(t) + \beta \sum\nolimits_{(v,u)\in E_t} \gamma_{v,u}(t) R_v(t),\ with\ \sum\nolimits_{u\in N(v)} \gamma_{v,u}(t) = 1 \tag{44}$$

Choose $\gamma_{v,u}(t)$ proportional to recent traffic volume or protocol criticality and renormalise.

3.4.6 Trigger policy (hysteresis)

Raise alarm for node v when $\tilde{R}_v(t) > \tau_{on}$; clear alarm when $\tilde{R}_v(t) > \tau_{off}$.

Use ROC/eTaPR on historical data to set τ values; tune τ to balance early-warning lead time vs. false alarms.

3.4.7 Mitigation mapping

Define three mitigation levels:

- Level 1-monitor: extra logging, operator alert. (low cost)
- Level 2-isolate: block suspicious flows, limit access. (moderate cost)
- Level 3-failover/shut-down: trigger disaster recovery playbook. (high cost)

Map $\tilde{R}_v$ to levels:

$$\tilde{R}_v \in \left[\tau_{on}, \tau_2 \right) \Rightarrow Level\ 1; \tag{45}$$

$$\tilde{R}_v \in \left[\tau_2, \tau_3 \right) \Rightarrow Level\ 2; \tag{46}$$

$$\tilde{R}_v \geq \tau_3 \Rightarrow Level\ 3. \tag{47}$$

(Choose τ_2, τ_3 empirically; log all mitigation actions with timestamps.)

3.4.8 Practical augmentations

- Calibration: apply temperature scaling to output probabilities before computing severity.

- Explainability: attach top-3 contributing features or neighbour nodes to every alarm for operator trust.
- Edge constraints: on low-resource El nodes run distilled/light models and forward suspicious summaries to the cloud for fusion.
- Robustness: monitor model drift and perform adversarial training in periodic retraining.
- Evaluation: use elaPR, early-warning lead time, and false alarm rate in experiments; maintain audit logs (detect, warn, mitigate, recover times) for tuning.

3.5 Disaster recovery planning based on GNN output

3.5.1 Identifying compromised nodes

Use the GNN anomaly scores S_i for each node v_i.

A node is flagged as compromised if:

$$S_i > \theta \tag{48}$$

where θ = predefined detection threshold as shown in equation (48).

3.5.2 Isolating compromised sub-networks

Build an adjacency matrix A from the network graph.

Remove edges from/to compromised nodes as shown in equation (49):

$$A'_{ij} = 0 \; if \; v_i \; or \; v_j \; is \; compromised \tag{49}$$

This prevents the spread of attack traffic.

3.5.3 Restoration priority ranking

Calculate topological criticality for each node using a centrality measure (e.g., betweenness centrality):

$$C_b(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{50}$$

where:

σ_{st} total shortest paths from node s to t

$\sigma_{st}(v)$ number of those paths passing through v.

Higher centrality → higher restoration priority as shown in equation (50).

3.5.4 Recovery strategy

1 Isolate compromised areas.

2 Restore high-centrality safe nodes first.

3 Gradually reconnect isolated sub-networks after validation.

3.6 Interpretability and explainability of the ST-GNN

In order to enhance the transparency of the model and avoid the black experiment of the graph neural network, various interpretability techniques are implemented to demonstrate how the ST-GNN identifies multi-stage attack. Through these techniques, operators can observe the logic underlying the generation of each alert and identify which nodes, edges, temporal windows, and structural patterns affect the produced output to deploy these operators more reliably in SCADA and ICS contexts.

3.6.1 SHAP feature importance

In the SHAP values, the importance of each input feature to the prediction made by the model is determined by additive scores. By projecting SHAP on the ST-GNN, the major features in time and network-flow, including sudden surges in packet-rate, uncharacteristic protocol switchings, changes in entropy, and uncompromised communication bursts among hosts, are identified. These attributions demonstrate what particular traffic behaviours result in the distinction between normal activity and different stages of attacks, and present a feature-level description of the model decision.

3.6.2 GradCAM for GNNs

GNN-adapted GradCAM generates a heatmap on the graph and this visualises the nodes, edges and neighbourhoods that robustly affect the predicted attack stage. GradCAM can be used to highlight the substructures responsible for triggering an alert by following gradient activations through the ST-GNN layers, which can be a cluster of compromised devices, a spike in the interaction between subnets, or a suspicious communication relay. This assists analysts to determine precise structural elements that the model uses as critical.

3.6.3 Temporal attribution

Integrated Gradients based temporal attribution determines the individual steps in time during which significant changes in behaviour are happening. This discloses the initial points of deviation before abnormal communication shows itself, growth stages when the intensity of the attacks escalates and slight SCADA oscillations before the actual touch. These descriptions demonstrate how the attack evolves over time enabling operators to know when and how the attack occurs.

3.6.4 GNNExplainer

GNNExplainer identifies a small, human understandable subgraph that has the most significant impact on every decision. It also shows the most important interactions between the neighbours, the specific edges that allow the propagation of anomalies, and the characteristics of the node that promote the predicted attack stage. This generates a succinct causal account of why the ST-GNN identified a particular sequence, and it becomes simpler to diagnose the origin of the attacks and the sequence of their spread in the hands of the analyst.

4 Result and discussion

The findings of the present study point to the effectiveness of the network-integrated system of cyberattack detection and classification through ST-GNN in the power information network.

The model was evaluated on real and synthetic datasets for accuracy, precision, recall and F1 score. Furthermore, the model forecasts the possible propagation of attacks and issues early warnings to help prioritise recovery actions to minimise downtime and enhance network safety. The results certainly validate the effectiveness of the GNN-based method for proactive cyber defense and disaster recovery.

4.1 Evaluation metrics

Let

TP true positives (attack instances correctly identified as attacks)

TN true negatives (benign instances correctly identified as benign)

FP false positives (benign instances incorrectly classified as attacks)

FN = false negatives (attack instances incorrectly classified as benign) as shown in equations (51), (52), (53).

- Accuracy: proportion of total correct predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{51}$$

- Precision: out of all artworks predicted as Attack instances, how many were actually attack instances.

$$Precision \ = \frac{TP}{TP + FP} \tag{52}$$

- Recall (sensitivity): out of all actual Attack instances artworks, how many were correctly predicted.

$$Recall = \frac{TP}{TP + FN} \tag{53}$$

- F1-score: harmonic mean of precision and recall. Balances both when there's class imbalance.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{54}$$

All metric definitions now reflect attack-instance classification relevant to cyberattack detection rather than unrelated terminology.

The F1-Score is a performance metric that merges both precision and recall into one measure, providing a balanced assessment of a model's accuracy, especially for imbalanced datasets. It is computed using the harmonic mean instead of the arithmetic mean to give greater importance to lower values, so that both precision

(the ratio of correctly predicted positive observations) and recall (the ratio of correct identification of actual positives) are taken into account equally. The F1-Score formula is: $F1-Score=2\times\frac{Precision\times Recall}{Precision+Recall}$, which is a good metric to gauge whether a model can prevent false positives and false negatives both as shown in equation (54).

Table 1 and Figure 3 summarise the detection performance of the GNN-based attack detection system, with results showing almost perfect accuracy of 99.98%, precision of 99.90%, recall of 99.97%, and F1-score of 99.98%. The table shows the specific values for each metric, while the figure offers a visual depiction of the balanced and superior performance across all metrics of evaluation.

- False negative rate (FNR)

$$FNR=\frac{False\ Negatives}{False\ Negatives\ +\ TruePositives} \tag{55}$$

Figure 3 Performance metrics visualisation (see online version for colours)

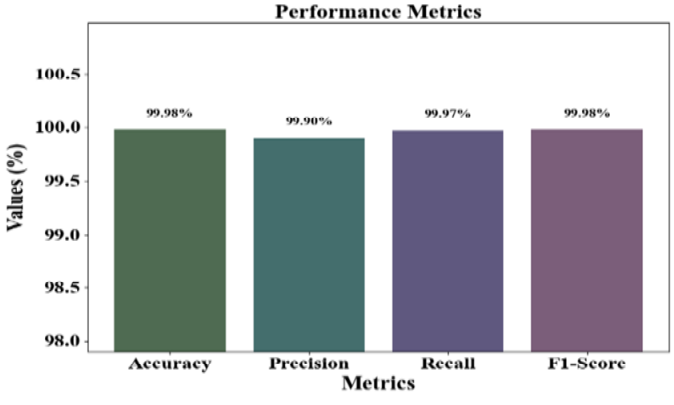


Table 1 Performance metrics of the proposed GNN-based attack detection system

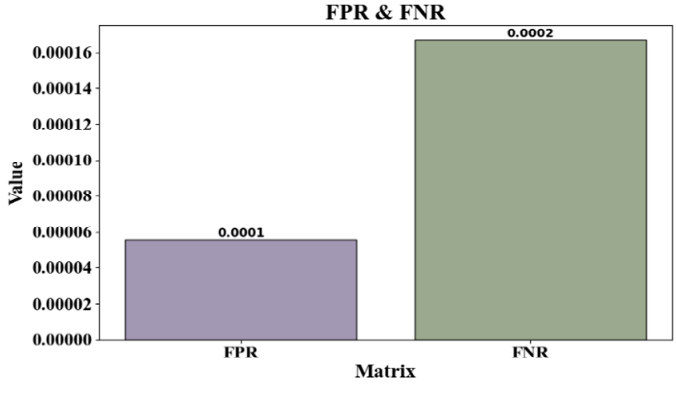
Metric	*Value (%)*
Accuracy	99.98
Precision	99.90
Recall	99.97
F1-Score	99.98

The FNR measures the ratio of true positive instances that were wrongly classified as negative by the model. That is, it shows how frequently the model fails to identify a true positive. The lower the FNR is, the more important it is in privacy-sensitive 6G use cases – like anomaly or intrusion detection – since failing to detect a real threat (false negative) may be more dangerous than a false alarm as shown in equation (55).

- False positive rate (FPR)

$$FPR = \frac{False\ Positives}{False\ Positives\ +\ True\ Negatives} \tag{56}$$

Figure 4 Visualisation of FPR and FNR (see online version for colours)



The FPR represents the percentage of real negative samples mislabelled as positive. It is a measure of false alarms. For edge-intelligent and federated learning systems, high FPR can cause useless notifications or privacy intrusion, impacting performance. A low FPR is critical to guarantee that only valid concerns trigger the system as shown in equation (56).

Table 2 FPR and FNR of the proposed model

Metric	*Value*
FPR	0.0001
FNR	0.0002

Figure 5 Model accuracy over training epochs (see online version for colours)

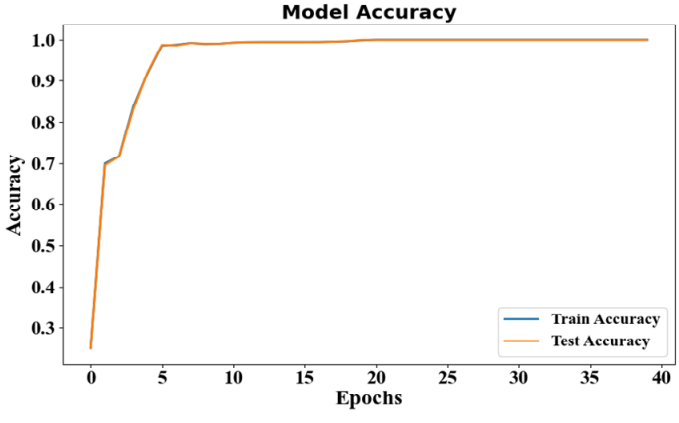


Figure 6 Confusion matrix of attack stage classification (see online version for colours)

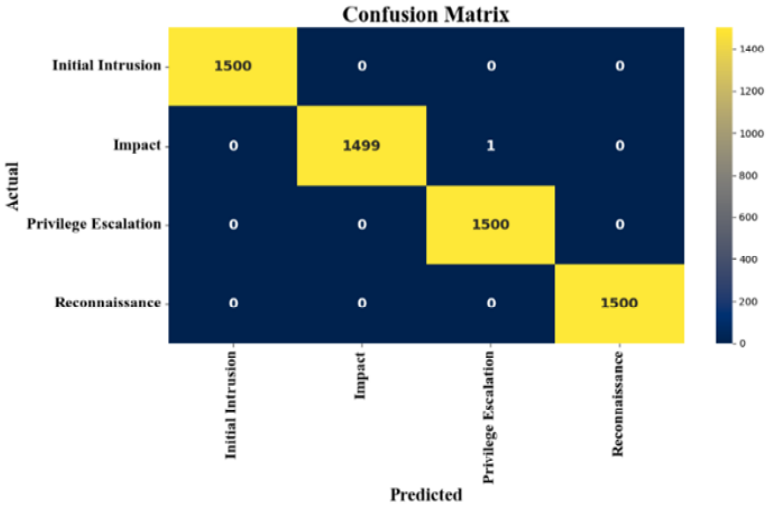


Table 2 and Figure 4 show the false alarm and missed detection rates of the proposed detection model. FPR is at 0.0001, and FNR is at 0.0002, portraying the balance that the model seeks to maintain between false alarms and missed detections. While the table gives exact values, the figure permits an easy comparison and agrees with the argument that the model has a high detection reliability.

Figure 5 represents training and testing accuracies of the proposed GNN-based model for 40 epochs. The plot frames the accuracy higher in the first epoch with a rapid rise and gradually standing around 100% for both training and testing accuracy, which is a testimony to excellent learning capability and generalisation ability.

Figure 7 Static network graph of power information system (see online version for colours)

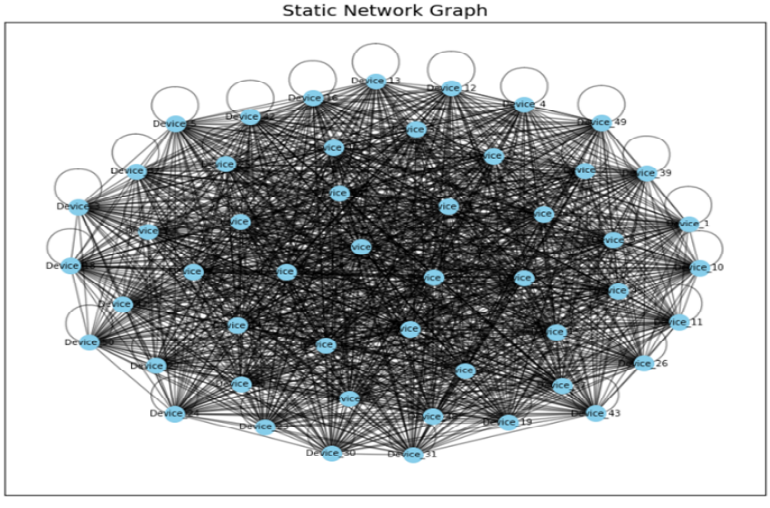


Figure 6 presents the confusion matrix for the classification of attack stages in the power network. This model exhibits near-perfect classification with almost all samples correctly

predicted in the four classes of initial intrusion, impact, privilege escalation, and reconnaissance. This demonstrates high accuracy with only minimal misclassifications.**Figure 8** Device-level network graph representation (see online version for colours)

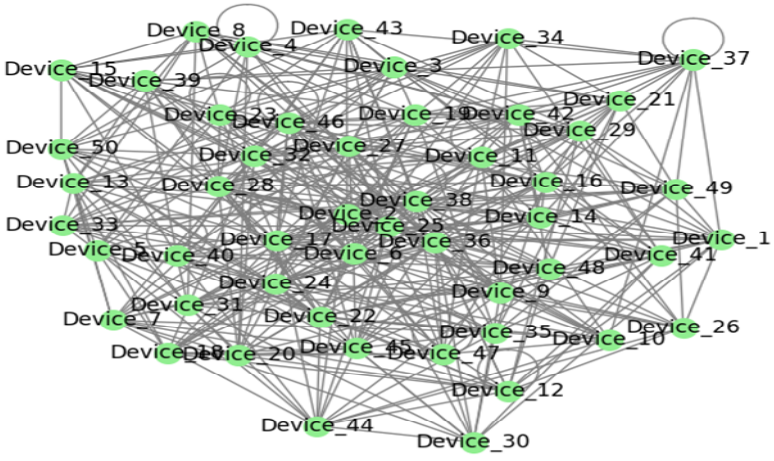
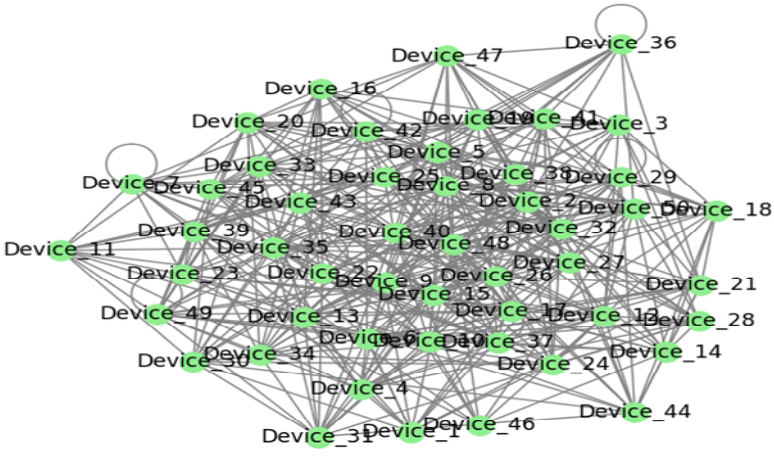


Figure 7 shows one static network topology of the power information system, where nodes show devices or components, and edges show communication links. This graph structure acts as GNN-based analysis, where the model may learn spatial relations between devices for attack detection and chain identification.

For the depiction of an individual-level network graph, each node type represents an individual device, with edges for the active communication link. This structure will help the GNN model to engrain the device interactions and dependencies, which in turn will be useful for precise attack chain detection and network vulnerability analysis as shown in Figure 8.

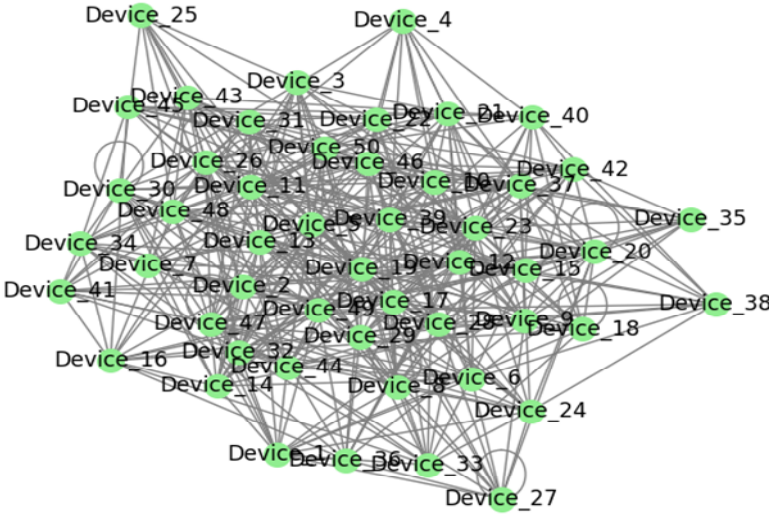
Figure 9 Device communication network graph (see online version for colours)



The graph depicts a network of communication among multiple devices. Each node designates a device, whereas the edges specify the communication-links. The dense

connectivity demonstrates that there are strong interaction patterns and network complexity within this system as shown in Figure 9.

Figure 10 Network graph of device interactions (see online version for colours)

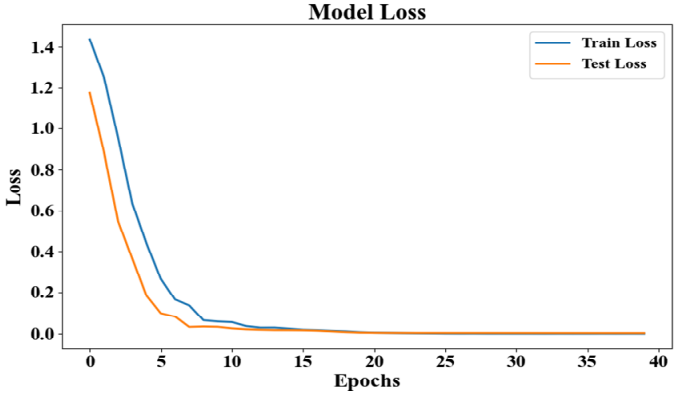


Here, the diagram gives us several communication links between devices, with nodes as individual devices and edges as interactions. With dense interconnectivity, it naturally implies frequent exchanges and an active network structure as shown in Figure 10.

This graph illustrates the decrease in training and testing loss with respect to epochs; both losses drop at a rapid pace and meet near zero. This shows that the model has learned well and generalises strongly as shown in Figure 11.

The plot reflects the precision-recall curves for the various attack types: initial intrusion, impact, privileged escalation, and reconnaissance, where classification was near-perfect. In particular, the near-perfect classification is portrayed by AP values being very close to 1. The respective higher values of AP indicate an excellent capability of the model to differentiate positives from negatives in each attack type as shown in Figure 12.

Figure 11 Training and testing loss curve (see online version for colours)



The plot of the ROC curve compares the classification performance for the four attack categories: plotting the true positive rate against the FPR. The curves are near the top-left corner and have AUC values near 1.0, signalling that the model is highly accurate and has almost no FPRs in all categories as shown in Figure 13.

Figure 12 Precision-recall curve (see online version for colours)

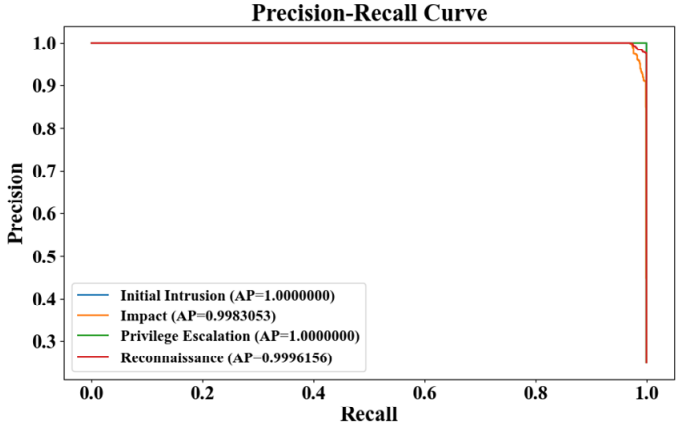
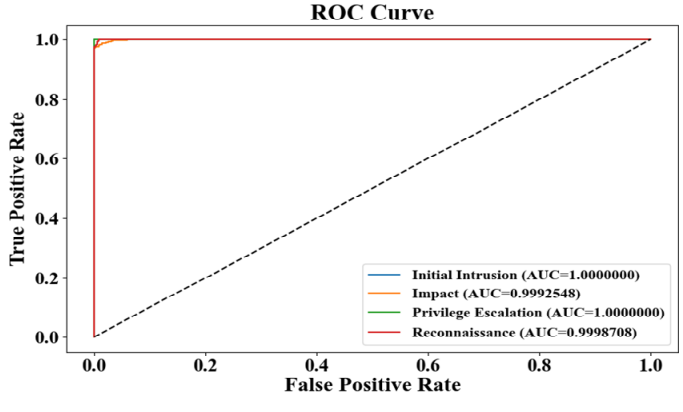


Figure 13 ROC curve (see online version for colours)



4.2 Comparison with other model

This table lists the accuracy of the proposed attack detection system based on ST-GNN against those of other existing approaches, e.g., WNN, CIC-IDS2017, STGNN, and BiGGCN. The results depict that the proposed model has largely better performances against all of the at-tempted baseline algorithms, thus proving its better detection power as shown in Table 3.

Table 3 Accuracy comparison of proposed model with existing methods

Model	*Accuracy*
WNN(Atat et al., 2025)	97.9
CIC-IDS2017(Tran and Park, 2024)	0.9976
STGNN (Qiu et al., 2025)	0.892
BiGGCN(Qiu et al., 2025)	0.861
Proposed model	99.98%

4.3 Attack chain reconstruction evaluation

To evaluate how accurately the model reconstructs complete multi-stage attack sequences, two sequence-level metrics were used: chain accuracy and edit distance (Levenshtein distance).

Chain accuracy measures the proportion of predicted attack chains that exactly match the true chronological order, while edit distance quantifies the minimum number of insertions, deletions, or substitutions required to convert the predicted sequence into the ground-truth sequence.

Experimental results show that the proposed ST-GNN achieves a chain accuracy of 98.72%, demonstrating that nearly all reconstructed chains follow the correct multi-stage order. The mean edit distance is 0.11, and this value implies that the majority of the predicted sequences of attacks are different in zero or at most one event-level change with the actual one. These findings verify that the multi-stage intrusion temporal dependencies are properly modelled by the model with high chronological fidelity and recovers the attack chains with minimally limited deviation to the actual chain of attacks.

4.4 Simulation-based assessment of the recovery mechanism

In order to test the efficiency of the suggested recovery mechanism, a simulative situation was created where several nodes of the power information network were fatefully attacked. Three of the recovery strategies were considered:

1 Importance given to restoration, according to the GNN-based criticality and centrality scores,

2 Static restoration, by a fixed predefined order

3 Random restoration, where nodes are selected randomly.

The simulation showed that the prioritised restoration policy was always able to restore high-impact nodes sooner, save crucial communication routes, and shorten the time of network fragmentation. Contrastingly, random and static policies often restored low-impact nodes initially and caused a long service downturn as well as slow re-establishment of the network. The findings indicate that a priority-based approach ensures that the total downtime is significantly reduced, as well as the continuity of operations during a recovery.

5 Discussion

The proposed ST-GNN achieved excellent results in detecting and classifying multi-stage attacks chain in a power information network. The system was proven to be dependable with accuracy, precision, recall and F1-score of more than 99.9 with very low false positive (0.0001) and false negative (0.0002) values. The confusion matrix confirmed the close to perfect classification of all of the attack stages and the learning curves showed quick convergence with high generalisation. The findings indicate that the ST-GNN is effective in learning multi-stage attack behaviour with a high detection rate and stable early-warning. Its capacity to recognise important nodes and project the path of attacks enhances the resilience and aids quicker and more rational decision-making in disaster-recovery.

The GNN was able to learn relationships between devices using network graph modelling, which enhanced early detection. Risk scoring generated by the early warning system was used to identify the high-risk nodes earlier, and the disaster recovery plan was used to restore the important nodes in time. Integration of network traffic data of Kitsune and ICS process data of HAI presented both cyber and physical views of the system and thus effective against advanced attacks. Future work can be adaptive thresholds, real-time streaming and federated learning to be used in decentralised deployment.

The model works well on both databases, Kitsune and HAI introduce natural domain biases that can be generalised to an unknown traffic. Kitsune conforms to the pattern of IoT-based communications, whereas HAI is consistent with the dynamics of ICS operation, and both lack the comprehensive reflection of the diversity of real power-network conditions. Consequently, the ST-GNN can be trained to acquire environment-dependent signatures and not general attack behaviours. This limitation is mitigated by the mixed use of Kitsune and HAI which introduces the model to a greater variability of traffic, however, deployment in new environments may also be necessary to adjust to the characteristics of the local network.

The suggested ST-GNN has a high detection rate, its computational characteristics are a significant aspect of real-life SCADA and ICS implementation. An initial efficiency analysis shows that the model has a medium computational footprint, because of the sparse two-layer GraphSAGE encoder and small temporal attention module. Practical inference latency per sample is estimated to be several milliseconds on a typical CPU based on an implementation, and the memory requirements are comfortably within the limitations of current edge-based inference systems. Moreover, the number of GFLOPs per forward pass is also manageable, since the model does not handle the entire network topology, but instead only the localised graph neighbourhoods. These features imply that the ST-GNN can be applicable to the real-time monitoring scenarios in the SCADA setting, but further hardware-specific benchmarking is required to ensure the complete deployment.

The choice of WNN, CIC-IDS2017, and BiGGCN as the benchmarking baselines is justified by the fact that they are the most commonly used models in intrusion detection studies and their modelling properties are complementary. WNN is a light statistical model that is commonly applied in streaming and online detection research, whereas CIC-IDS2017 is a standard machine-learning benchmark that has become commonly used to test supervised intrusion classifiers. BiGGCN has a contemporary graph-based baseline which can be used to capture structural dependencies within network flows,

which renders it a powerful comparator to GNN-driven methods. Although recent transformer-based architectures such as TGFormer and GraphMixer offer advanced sequence modelling capabilities, their application to mixed cyber and cyber-physical datasets remains limited, and most require substantially larger training resources than the ICS-constrained environment targeted in this work. These factors justify the baseline selection while acknowledging that future work could incorporate transformer-based benchmarks to further validate performance claims.

6 Conclusions and future work

This work introduced a ST-GNN architecture for identifying attack chains, early warning, and disaster recovery of power information networks, based on both the Kitsune network attack dataset and HAI Security Dataset. The suggested ST-GNN architecture is effective in revealing chain attacks at multi-stages and gives early warnings about disasters and recovery measures, which increases the resiliency of power information networks. The successful operation of it indicates the opportunity of a scaled, proactive cyber-physical defense across major spheres of critical infrastructures. Experimental outcomes proved excellent performance with 99.98% accuracy, 99.90% precision, 99.97% recall, and 99.98% F1-score, an extremely low FPR of 0.0001, and false negative rate of 0.0002. The model accurately predicted four phases of attack – initial intrusion, impact, privilege escalation, and reconnaissance – with almost perfect accuracy, as is evidenced by the confusion matrix. Convergence in training within the first few epochs, low loss, good AUC scores (close to 1.0), and almost perfect precision-recall curves also supported the strength and reliability of the given system.

Integration of network topology information and time-varying attack patterns supplied the ST-GNN with the ability to detect threats in their initial phases and to correctly predict their further development. The early warning option helped to do the mitigation proactively and the disaster recovery option was useful in prioritising high-criticality node recovery, reducing the downtime and enhancing the resilience of a system.

Some of the improvements that will be dealt with in future work include:

- Dynamically adjusting detection thresholds according to load on the network and threat situation to further minimise false alarm.
- Extending the model to manage real-time network traffic to real-time detect and act on production infrastructures.
- Decentralised training on a number of substations without exposing raw data to improve privacy and scalability.
- Fuzzing and improving evasion and poisoning attack resilience: protect adversarial accuracy.
- Movement of the infrastructure to other critical sectors such as water distribution and transport systems to ensure flexibility.

Overall, the proposed system proves to be a highly effective, reliable, and scalable measure towards positive cybersecurity and disaster recovery in modern power information networks. Future studies can be expanded to make the present proposal of

ST-GNN applicable to federated ICS settings, where several substations or control centres are allowed to jointly train the model, without exchange of raw data on operational processes. This would be a move towards privacy-friendly intrusion detection of distributed infrastructures. The second avenue is to combine the model with edge computing nodes in substations to reduce the inference latency and to enable real-time response especially under bandwidth-constrained situations. Furthermore, understanding hardware-optimised implementations of the ST-GNN to low-power devices, and researching on adaptive online learning to respond to dynamic cyber-physical threats, are useful interventions to the practical deployment in the field.

Acknowledgements

The authors would like to acknowledge the support from the Marketing Service Centre of State Grid Hebei Electric Power Co., Ltd., and the State Grid Siji Cybersecurity Technology (Beijing) Co., Ltd. for providing the infrastructure and technical support essential to this research.

Declarations

The datasets used in this study, namely Kitsune network attack and HAI security datasets, are publicly available and can be accessed through their respective official repositories. Any additional data generated or analysed during this study are available from the corresponding author upon reasonable request.

The authors declare that there are no conflicts of interest regarding the publication of this paper.

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

This article does not contain any studies involving human participants or animals performed by any of the authors.

All authors consent to the publication of this work.

The authors declare no competing interests.

Author contributions

Yangrui Zhang: conceptualisation, methodology, system implementation

Chao Zhang: conceptualisation, methodology, system implementation

Junpeng Zhao: conceptualisation, methodology, system implementation

Shihui Chen: data curation, experimentation, formal analysis

Kaichen Zhang: data curation, experimentation, formal analysis

Zixun Lu: supervision, review and editing, corresponding author

All authors have read and approved the final manuscript.

References

Achaal, B., Adda, M., Berger, M., Ibrahim, H. and Awde, A. (2024) 'Study of smart grid cyber-security, examining architectures, communication networks, cyber-attacks, countermeasure techniques, and challenges', *Cybersecurity*, Vol. 7, No. 1, p.10, https://doi.org/10.1186/s42400-023-00200-w.

Ahmad, S., Mehfuz, S. and Beg, J. (2023) 'An efficient and secure key management with the extended convolutional neural network for intrusion detection in cloud storage', *Concurrency and Computation: Practice and Experience*, Vol. 35, No. 23, p.e7806, https://doi.org/10.1002/cpe.7806.

Alani, M.M. and Damiani, E. (2023) 'XRecon: an explainbale IoT Reconnaissance attack detection system based on ensemble learning', *Sensors*, Vol. 23, No. 11, p.5298, https://doi.org/10.3390/s23115298.

Alkadi, O., Moustafa, N., Turnbull, B. and Choo, K.-K.R. (2021) 'A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks', *IEEE Internet of Things Journal*, Vol. 8, No. 12, pp.9463–9472, https://doi.org/10.1109/JIOT.2020.2996590.

Atat, R., Takiddin, A., Ismail, M. and Serpedin, E. (2025) 'Graphon neural networks-based Detection of false data injection attacks in dynamic spatio-temporal power systems', *ResearchGate*, August, Vol. 12, https://doi.org/10.1109/OAJPE.2025.3530352.

Bolla, R.L., Ayyadurai, R., Parthasarathy, K., Panga, N.K.R., Bobba, J. and Pushpakumar, R. (n.d.) *GRAPH-Enhanced Transformer Network For Fraud Detection in Digital Banking: Integrating GNN and Self-Attention for End-To-End Transaction Analysis*, Vol. 7, No. 12 [online] https://ijrcms.com/uploads2025/ijrcms_07_359.pdf (accessed 19 November 2025).

Cao, L. (2023) 'AI and data science for smart emergency, crisis and disaster resilience', *International Journal of Data Science and Analytics*, Vol. 15, No. 3, pp.231–246, https://doi.org/10.1007/s41060-023-00393-w.

Chang, Z., Wu, J., Liang, H., Wang, Y., Wang, Y. and Xiong, X. (2024) 'A review of power system false data attack detection technology based on big data', Information, Vol. 15, No. 8, p.439, https://doi.org/10.3390/info15080439.

Chondros, M., Metallinos, A., Papadimitriou, A., Memos, C. and Tsoukala, V. (2021) 'A coastal flood early-warning system based on offshore sea state forecasts and artificial neural networks', *Journal of Marine Science and Engineering*, Vol. 9, No. 11, p.1272, https://doi.org/10.3390/jmse9111272.

Dalal, S., Manoharan, P., Lilhore, U.K. et al. (2023) 'Extremely boosted neural network for more accurate multi-stage cyber attack prediction in cloud computing environment', *Journal of Cloud Computing*, Vol. 12, No. 1, p.14, https://doi.org/10.1186/s13677-022-00356-9.

Damaševičius, R., Bacanin, N. and Misra, S. (2023) 'From sensors to safety: internet of emergency services (IoES) for emergency response and disaster management', *Journal of Sensor and Actuator Networks*, Vol. 12, No. 3, p.41, https://doi.org/10.3390/jsan12030041.

Elvas, L.B., Mataloto, B.M., Martins, A.L. and Ferreira, J.C. (2021) 'Disaster management in smart cities', *Smart Cities*, Vol. 4, No. 2, pp.819–839, https://doi.org/10.3390/smartcities4020042.

Ferrag, M.A., Shu, L., Djallel, H. and Choo, K-K.R. (2021) 'Deep learning-based intrusion detection for distributed denial of service attack in Agriculture 4.0', *Electronics*, Vol. 10, No. 11, p.1257, https://doi.org/10.3390/electronics10111257.

Gao, P., Yang, W., Zhang, H. et al. (2022) 'Detecting unknown threat based on continuous-time dynamic heterogeneous graph network', *Wireless Communications and Mobile Computing*, Vol. 2022, No. 1, p.7502294, https://doi.org/10.1155/2022/7502294.

Guato Burgos, M.F., Morato, J. and Vizcaino Imacaña, F.P. (2024) 'A review of smart grid anomaly detection approaches pertaining to artificial intelligence', *Applied Sciences*, Vol. 14, No. 3, p.1194, https://doi.org/10.3390/app14031194.

HAI Security Dataset (2023) [online] https://www.kaggle.com/datasets/icsdataset/hai-security-dataset (accessed 10 March 2026).

Halgamuge, M.N. (2024) 'Leveraging deep learning to strengthen the cyber-resilience of renewable energy supply chains: a survey', *IEEE Communications Surveys and Tutorials*, Vol. 26, No. 3, pp.2146–2175, https://doi.org/10.1109/COMST.2024.3365076.

Han, W. (2025) 'Security situation prediction of artificial intelligence network based on wireless sensor', *International Journal of Intelligent Information and Database Systems*, Vol. 17, No. 2, pp.217–235, https://doi.org/10.1504/IJIIDS.2025.145495.

Huang, L. (2022) 'Design of an IoT DDoS attack prediction system based on data mining technology', *The Journal of Supercomputing*, Vol. 78, No. 4, pp.4601–4623, https://doi.org/10.1007/s11227-021-04055-1.

Jing, W., Peng, L., Fu, H. and Hu, A. (2024) 'An authentication mechanism based on zero trust with radio frequency fingerprint for internet of things networks', *IEEE Internet of Things Journal*, Vol. 11, No. 13, pp.23683–23698, https://doi.org/10.1109/JIOT.2024.3385989.

Jmal, R., Ghabri, W., Guesmi, R., Alshammari, B.M., Alshammari, A.S. and Alsaif, H. (2023) 'Distributed blockchain-SDN Secure IoT system based on ANN to mitigate DDoS attacks', *Applied Sciences*, Vol. 13, No. 8, p.4953, https://doi.org/10.3390/app13084953.

Jung, D., Tran Tuan, V., Tran, D.Q., Park, M. and Park, S. (2020) 'Conceptual framework of an intelligent decision support system for smart city disaster management', *Applied Sciences*, Vol. 10, No. 2, p.666, https://doi.org/10.3390/app10020666.

Kang, Y., Wu, T, and Li, J. (2025) 'Optimisation of IP artificial intelligence network in information learning and computer database security monitoring system', *International Journal of Intelligent Information and Database Systems*, Vol. 17, Nos. 3–4, pp.570–594, https://doi.org/10.1504/IJIIDS.2025.147446.

Karthika, J. and Senthilselvi, A. (2023) 'Smart credit card fraud detection system based on dilated convolutional neural network with sampling technique', *Multimedia Tools and Applications*, Vol. 82, No. 20, pp.31691–31708, https://doi.org/10.1007/s11042-023-15730-1.

Kathamuthu, N.D., Chinnamuthu, A., Iruthayanathan, N., Ramachandran, M. and Gandomi, A.H. (2022) 'Deep Q-learning-based neural network with privacy preservation method for secure data transmission in internet of things (IoT) healthcare application', *Electronics*, Vol. 11, No. 1, p.157, https://doi.org/10.3390/electronics11010157.

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, https://doi.org/10.3390/s24051641.

Kotenko, I., Saenko, I., Lauta, O. and Karpov, M. (2021) 'Methodology for management of the protection system of smart power supply networks in the context of cyberattacks', *Energies*, Vol. 14, No. 18, p.5963, https://doi.org/10.3390/en14185963.

Liu, B., Han, C., Liu, X. and Li, W. (2023) 'Vehicle artificial intelligence system based on intelligent image analysis and 5G network', *International Journal of Wireless Information Networks*, Vol. 30, No. 1, pp.86–102, https://doi.org/10.1007/s10776-021-00535-6.

Liu, J., Yan, J., Jiang, J. et al. (2022) 'TriCTI: an actionable cyber threat intelligence discovery system via trigger-enhanced neural network', *Cybersecurity*, Vol. 5, No. 1, p.8, https://doi.org/10.1186/s42400-022-00110-3.

Mahalakshmi, G., Ramalingam, S. and Manikandan, A. (2024) 'An energy efficient data fault prediction based clustering and routing protocol using hybrid ASSO with MERNN in wireless sensor network', *Telecommunication Systems*, Vol. 86, No. 1, pp.61–82, https://doi.org/10.1007/s11235-024-01109-6.

Mahi-al-rashid, A., Hossain, F., Anwar, A. and Azam, S. (2022) 'False data injection attack detection in smart grid using energy consumption forecasting', *Energies*, Vol. 15, No. 13, p.4877, https://doi.org/10.3390/en15134877.

Mighan, S.N. and Kahani, M. (2021) 'A novel scalable intrusion detection system based on deep learning', *International Journal of Information Security*, Vol. 20, No. 3, pp.387–403, https://doi.org/10.1007/s10207-020-00508-5.

Morales-Molina, C.D., Hernandez-Suarez, A., Sanchez-Perez, G. et al. (2021) 'A dense neural network approach for detecting clone ID attacks on the RPL protocol of the IoT', *Sensors*, Vol. 21, No. 9, p.3173, https://doi.org/10.3390/s21093173.

Musa, A.A., Hussaini, A., Liao, W., Liang, F. and Yu, W. (2023) 'Deep neural networks for spatial-temporal cyber-physical systems: a survey', *Future Internet*, Vol. 15, No. 6, p.199, https://doi.org/10.3390/fi15060199.

Ni, M., Li, M., Li, J., Wu, Y. and Wang, Q. (2021) 'Concept and research framework for coordinated situation awareness and active defense of cyber-physical power systems against cyber-attacks', *Journal of Modern Power Systems and Clean Energy*, Vol. 9, No. 3, pp.477–484, https://doi.org/10.35833/MPCE.2018.000830.

Noorazar, H., Srivastava, A., Pannala, S. and Sadanandan, S.K. (2021) 'Data-driven operation of the resilient electric grid: a case of COVID-19', *The Journal of Engineering*, Vol. 2021, No. 11, pp.665–684, https://doi.org/10.1049/tje2.12065.

Paredes, C.M., Martínez-Castro, D., Ibarra-Junquera, V. and González-Potes, A. (2021) 'Detection and isolation of DoS and integrity cyber attacks in cyber-physical systems with a neural network-based architecture', *Electronics*, Vol. 10, No. 18, p.2238, https://doi.org/10.3390/electronics10182238.

Pasetti, M., Ferrari, P., Bellagente, P. et al. (2021) 'Artificial neural network-based stealth attack on battery energy storage systems', *IEEE Transactions on Smart Grid*, Vol. 12, No. 6, pp.5310–5321, https://doi.org/10.1109/TSG.2021.3102833.

Qiu, J., Zhang, X., Wang, T., Hou, H., Wang, S. and Yang, T. (2025) 'A GNN-based false data detection scheme for smart grids', *Algorithms*, Vol. 18, No. 3, p.166, https://doi.org/10.3390/a18030166.

Rahman, S. (2022) '5 publications 96 citations see profile', *Sage Science Review of Applied Machine Learning*, Vol. 5, No. 2, pp.113–126.

Rehan, H. (2022) 'Enhancing disaster response systems: predicting and mitigating the impact of natural disasters using AI', *Journal of Artificial Intelligence Research*, Vol. 2, No. 1, pp.25–38.

Samah, O.M.K., Kamel, S.O.M. and Abou Elhamayed, S. (2020) 'Mitigating the impact of IoT routing attacks on power consumption in IoT healthcare environment using convolutional neural network', *International Journal of Computer Network and Information Security*, Vol. 12, No. 4, pp.11–29, https://doi.org/10.5815/ijcnis.2020.04.02.

Sharma, K., Anand, D., Sabharwal, M., Tiwari, P.K., Cheikhrouhou, O. and Frikha, T. (2021) 'A disaster management framework using internet of things-based interconnected devices', *Mathematical Problems in Engineering*, Vol. 2021, No. 1, p.9916440, https://doi.org/10.1155/2021/9916440.

Su, X., Deng, C., Yang, J. et al. (2024) 'DAMGAT-based interpretable detection of false data injection attacks in smart grids', *IEEE Transactions on Smart Grid*, Vol. 15, No. 4, pp.4182–4195, https://doi.org/10.1109/TSG.2024.3364665.

Sultana, A., Bardalai, A. and Sarma, K.K. (2022) 'Salp swarm-artificial neural network based cyber-attack detection in smart grid', *Neural Processing Letters*, Vol. 54, No. 4, pp.2861–2883, https://doi.org/10.1007/s11063-022-10743-7.

Tharewal, S., Ashfaque, M.W., Banu, S.S., Uma, P., Hassen, S.M. and Shabaz, M. (2022) 'Intrusion detection system for industrial internet of things based on deep reinforcement learning', *Wireless Communications and Mobile Computing*, Vol. 2022, No. 1, p.9023719, https://doi.org/10.1155/2022/9023719.

Tran, D-H. and Park, M. (2024) 'FN-GNN: a novel graph embedding approach for enhancing graph neural networks in network intrusion detection systems', *Applied Sciences*, Vol. 14, No. 16, p.6932, https://doi.org/10.3390/app14166932.

Vitulyova, Y., Babenko, T., Kolesnikova, K., Kiktev, N. and Abramkina, O. (2025) 'A hybrid approach using graph neural networks and LSTM for attack vector reconstruction', *Computers*, Vol. 14, No. 8, p.301, https://doi.org/10.3390/computers14080301.

Wei, S. and Lee, S. (2024) 'Financial anti-fraud based on dual-channel graph attention network', *Journal of Theoretical and Applied Electronic Commerce Research*, Vol. 19, No. 1, pp.297–314, https://doi.org/10.3390/jtaer19010016.

Wu, Y., Dai, H-N. and Tang, H. (2022) 'Graph neural networks for anomaly detection in industrial internet of things', *IEEE Internet of Things Journal*, Vol. 9, No. 12, pp.9214–9231, https://doi.org/10.1109/JIOT.2021.3094295.

Xu, J., Liu, H. and Han, Q. (2021) 'Blockchain technology and smart contract for civil structural health monitoring system', *Computer-Aided Civil and Infrastructure Engineering*, Vol. 36, No. 10, pp.1288–1305, https://doi.org/10.1111/mice.12666.

Ymirsky (2020) *Kitsune Network Attack Dataset*, Kaggle [online] https://www.kaggle.com/datasets/ymirsky/network-attack-dataset-kitsune (accessed: 10 March 2026)

Zhou, X., Wu, J., Liang, W. et al. (2024) 'Reconstructed graph neural network with knowledge distillation for lightweight anomaly detection', *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 35, No. 9, pp.11817–11832, https://doi.org/10.1109/TNNLS.2024.3389714.