



AI-DRIVEN SELF-HEALING COMPUTER NETWORKS: A REVIEW OF GRAPH NEURAL NETWORKS, TEMPORAL TRANSFORMERS, REINFORCEMENT LEARNING, EXPLAINABLE AI AND NETWORK DIGITAL TWINS

¹Linginedi Ushasree, ²Dr. S.G. Shanthi, ³Dr. Haritha Donavalli

¹Research Scholar, ²Associate Professor, ³Professor

^{1,2,3}Department of Computer Science and Engineering,

^{1,2}Annamalai University, Tamil Nadu, India; ³KLEF, Vaddeswaram, India

¹ushalinginedi1413@gmail.com, ²sgsau2009@gmail.com, ³haritha_donavalli@kluniversity.in

Abstract : Operators now run networks that are software-defined, virtualised, spread across cloud and edge sites, and expected to keep services alive with little manual intervention. That flexibility comes at a price. Faults travel along dependencies that are hard to see, alarms arrive in bursts, and the rule-based self-healing logic inherited from earlier generations usually reacts only after users have already noticed the problem. This paper reviews work published between 2020 and 2026 on applying artificial intelligence to the detection, prediction, diagnosis and repair of network faults. Five lines of research are examined in detail: graph neural networks (GNNs), which encode topology and dependency structure; Transformer models, which capture long-range temporal behaviour in telemetry; reinforcement learning (RL), which learns recovery and rerouting policies; explainable AI (XAI), which makes automated decisions auditable; and network digital twins (NDTs), which allow a candidate action to be tested before it touches live infrastructure. Related work on zero-touch network and service management (ZSM), alarm root-cause analysis and resilience-aware routing is also covered. Our reading of the literature is that each family of methods fixes a specific weakness of conventional operations, yet almost every published system addresses only one or two stages of the healing cycle. Very few connect perception, forecasting, causal diagnosis, action selection, validation and explanation in one loop. Building on this observation, we outline NeuroHeal-Net, a reference architecture that chains these functions together under explicit policy constraints, and we propose an evaluation protocol that measures service impact and safety rather than detection accuracy alone. We argue that the next generation of self-healing networks should be predictive, explainable, risk-aware and continuously validated.

IndexTerms - self-healing networks; autonomous network management; graph neural networks; Transformers; deep reinforcement learning; explainable AI; network digital twin; root-cause analysis; zero-touch management; network resilience.

1. INTRODUCTION

Not long ago, a campus or carrier network could be described as a fairly stable set of routers, switches and links. That description no longer holds. A present-day network mixes physical links with virtual network functions, programmable controllers, cloud tenants, edge nodes and radio access segments, and each service rides on a chain of dependencies that may change several times a day. In this setting a small local fault rarely stays local. A degraded optical interface, for example, can show up minutes later as congestion on a distant path, a flood of correlated alarms, and a drop in application-level quality, none of which points directly back to the interface itself.

For network operations this shifts the question. Knowing that *something* is wrong is no longer enough. The operator, or the automation acting on the operator's behalf, has to build a picture of the current state, anticipate which elements are about to fail, separate the originating fault from its symptoms, choose a corrective action that will not make things worse, and then confirm that service has actually been restored. Each of these steps is hard in its own right; doing all of them quickly and without human help is harder still.

The research direction we call NeuroHeal-Net is aimed at exactly this end-to-end problem. Its intended capabilities are to detect abnormal behaviour, forecast node and link failures, trace incidents to their likely root causes, select a recovery action autonomously (rerouting traffic or reassigning resources, for instance), and justify that choice in terms an engineer can check. The proposed building blocks are GNNs, Transformers, RL, XAI and a network digital twin. What makes the direction worth pursuing is that it treats self-healing as a closed-loop control problem, not as yet another anomaly classifier.

Support for each building block can be found in recent publications, but it is scattered. Work on self-healing cellular networks has drawn attention to poor data quality, class imbalance, real-time limits and the need to fuse several data sources [1]. GNN studies have shown that modelling topology and relations explicitly pays off [2, 3]. Transformer-based work has improved the handling of long temporal dependencies [4, 5]. RL has been used to pick routing and recovery actions adaptively [6, 7], XAI research has focused on trust and accountability [8], and NDTs offer a sandbox where policies can be tried safely [9]. What is missing is a consolidated view of how these pieces fit together. This review sets out to provide one, guided by three questions:

RQ1. How did AI methods for detecting, predicting, diagnosing and recovering from network faults evolve between 2020 and 2026?

RQ2. What does each of the GNN, Transformer, RL, XAI and NDT approaches contribute, and where does it fall short, when viewed as one stage of a single self-healing loop?

RQ3. What architecture and evaluation practice would close the gaps that the recent literature leaves open?

Fig. 1 places the five technology streams on the self-healing lifecycle that frames the rest of the paper. We treat them as complementary stages rather than rival model families.

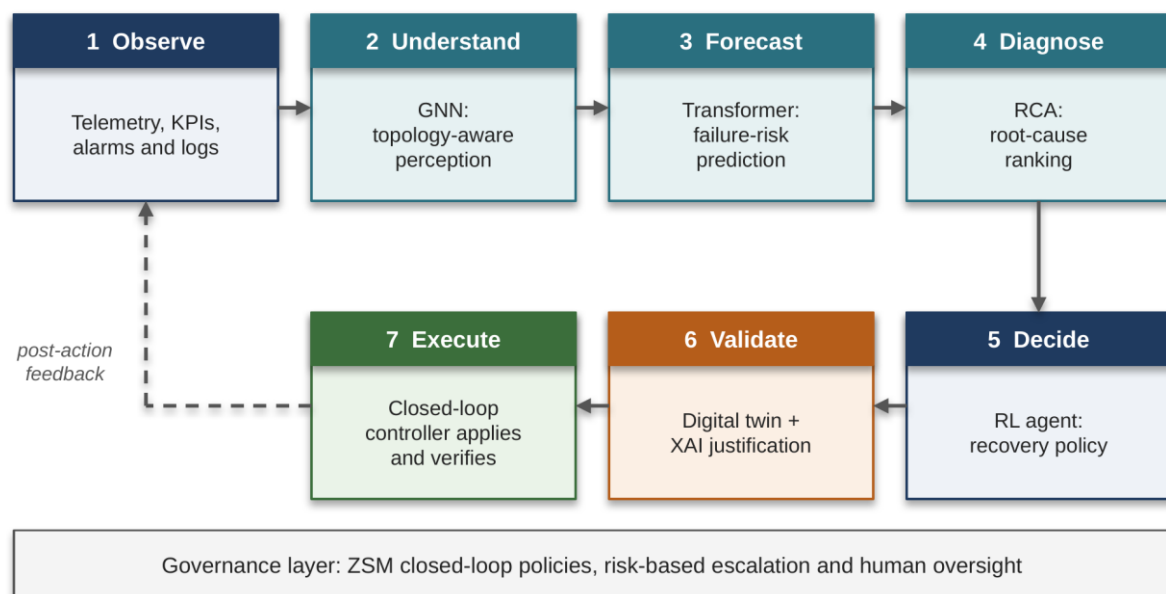


Fig. 1. The self-healing lifecycle used as the organising frame of this review, with the AI technique that primarily serves each stage. The dashed arrow denotes post-action feedback.

The remainder of the paper is organised as follows. Section 2 explains how the literature was selected. Section 3 traces the evolution of the field year by year, and Section 4 sets out a formal view of self-healing as constrained closed-loop control. Sections 5 to 10 review GNNs, Transformers, RL, root-cause analysis with XAI, digital twins and zero-touch management in turn. Sections 11 and 12 present the NeuroHeal-Net architecture and its formulation, Sections 13 and 14 synthesise the findings and list open gaps, Section 15 proposes an evaluation protocol, and Sections 16 to 18 discuss limitations, expected contributions and conclusions.

2. REVIEW METHOD

We carried out a structured narrative review of work published from January 2020 to mid-2026. Search strings combined the following terms: *self-healing network*, *autonomous network management*, *zero-touch network*, *network anomaly detection*, *graph neural network*, *fault localisation*, *root-cause analysis*, *Transformer*, *reinforcement learning*, *network resilience*, *explainable AI* and *network digital twin*. The searches were run against IEEE Xplore, ScienceDirect, SpringerLink, MDPI, the ACM Digital Library and arXiv, and bibliographic details were checked against publisher pages. ETSI and TM Forum documents were consulted for the standardisation side.

The selected studies were sorted into six groups: (i) classical and data-driven self-healing and fault detection; (ii) topology-aware GNN methods; (iii) Transformer-based temporal methods; (iv) RL and deep RL (DRL) for recovery and resilient routing; (v) explainability and trust; and (vi) digital twins and zero-touch closed-loop automation. We did not attempt a PRISMA-style meta-analysis. The studies cover very different network types, datasets, metrics and test environments, so pooling their numerical results would be misleading. The review is instead a technology-focused synthesis whose purpose is to locate a well-defined research gap. Table 1 summarises the selection rules.

Table 1. Inclusion and exclusion criteria applied during study selection.

Criterion	Included	Excluded	Rationale
Period	2020–2026	Pre-2020 work, except foundational concepts	Capture the recent AI shift in network operations
Domain	Computer, telecom, SDN, 5G/6G, edge and cloud networks	Studies with no network setting	Relevance to network self-healing
Role of AI	Detection, prediction, diagnosis, recovery, explanation or validation	Generic AI with no network-health function	Direct contribution to at least one loop stage
Source type	Peer-reviewed journals, IEEE/ACM conferences, standards bodies, widely cited preprints	Blogs, vendor marketing, unverifiable claims	Bibliographic reliability

3. HOW THE FIELD DEVELOPED, 2020–2026

Looking across the seven years, the overall movement is from predictive and rule-assisted fault management toward closed-loop autonomy (Fig. 2). In 2020 and 2021 most effort went into predictive self-healing, machine-learning-based fault localisation and alarm processing built on key performance indicators (KPIs) [10–12]. Around the same time, dynamic graph methods started to appear for anomaly detection [2]. By 2022, authors were openly discussing the practical obstacles: imbalanced and sparsely labelled data, the need to fuse heterogeneous sources, and tight latency budgets [1]. ZSM surveys published that year mapped machine learning onto the emerging zero-touch architecture [13, 14], and GNNs were applied to flow-level anomaly detection [15].

The years 2023 and 2024 brought two notable shifts. First, graph learning was increasingly paired with temporal models, as in GNN–Transformer pipelines for root-cause analysis in 5G radio access networks [16]. Second, diagnosis began to be linked with decisions, for example RL-driven self-healing placement of virtual network functions [6]. Systematic reviews of NDTs appeared in the same period [17, 18]. From 2025 onward, GNN-enabled DRL for resilience [7], GraphTransformer-based routing [19], intent-based RL self-healing [20] and ETSI work on agents and autonomy [21] brought the field noticeably closer to the integrated design that NeuroHeal-Net envisages.

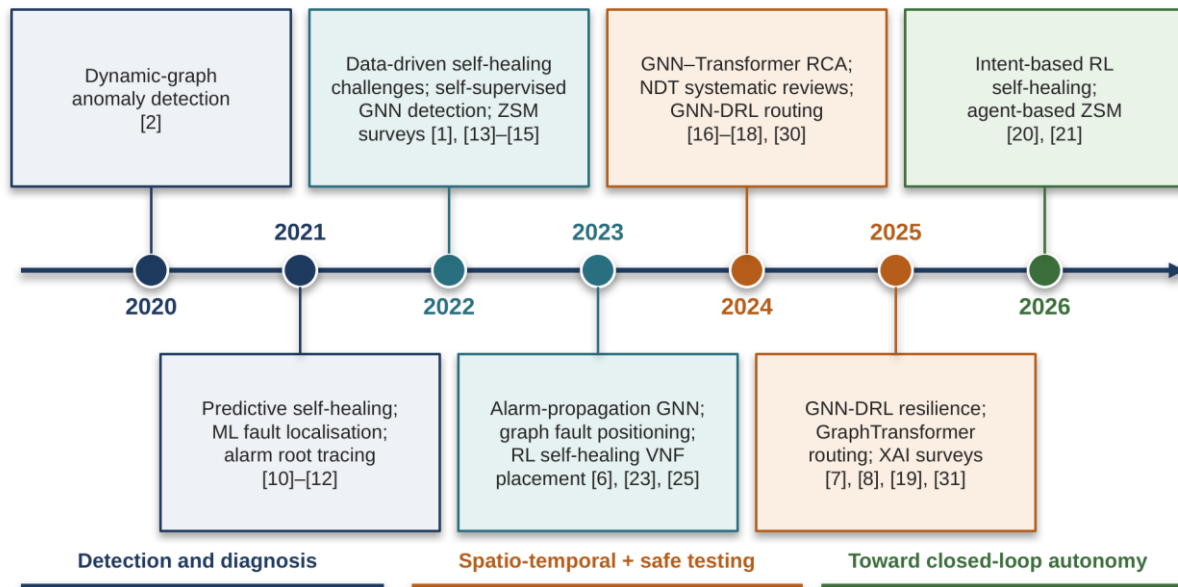


Fig. 2. Timeline of the dominant research directions in AI-enabled self-healing networks, 2020–2026, with representative works. Numbers in brackets refer to the reference list.

4. SELF-HEALING AS CONSTRAINED CLOSED-LOOP CONTROL

It helps to state the problem formally before reviewing individual techniques. Let the state of the network at time t be

$$\mathbf{x}_t = \{\mathcal{G}_t, \mathbf{K}_t, \mathbf{T}_t, \mathcal{A}_t, \mathbf{S}_t\} \quad (1)$$

where $\mathcal{G}_t$ is the network graph, $\mathbf{K}_t$ holds node and link health indicators, $\mathbf{T}_t$ contains traffic and performance measurements, $\mathcal{A}_t$ is the set of active alarms and events, and $\mathbf{S}_t$ encodes service-level objectives. A monitoring model observes a window of length L and estimates the probability that a failure F will occur within a horizon Δ :

$$P(F_{t+\Delta} = 1 \mid \mathbf{X}_{t-L+1:t}) = f_{\theta}(\mathbf{X}_{t-L+1:t}) \quad (2)$$

This horizon is what separates reactive from predictive healing. A reactive scheme waits for a threshold to be crossed; a predictive one acts while the failure is still a probability. Accurate prediction is not sufficient on its own, however, because the action that follows must also respect quality-of-service (QoS), safety, policy and resource limits. The decision step can therefore be written as

$$a_t^* = \operatorname{argmax}_{a \in \mathcal{A}_{\text{act}}} U(a \mid \mathbf{x}_t) \quad \text{s.t.} \quad C(a, \mathbf{x}_t) \leq b \quad (3)$$

where $\mathcal{A}_{\text{act}}$ is the admissible action set, U is a utility function, C is a cost or risk function and b is the allowed budget. Equations (1)–(3) map naturally onto the three main model families reviewed below. GNNs are suited to encoding the spatial structure $\mathcal{G}_t$, Transformers to modelling the temporal evolution of $\mathbf{X}$, and RL to learning a policy that chooses a_t as conditions change.

5. GRAPH NEURAL NETWORKS FOR TOPOLOGY-AWARE DETECTION AND DIAGNOSIS

A communication network is a graph whether or not the model treats it as one. Routers, switches, base stations, virtual functions and service instances are nodes; physical links, logical paths and service dependencies are edges. When telemetry is flattened into a feature vector, this structure is thrown away. GNNs keep it. Early dynamic-graph anomaly detection work showed that structure, node and edge attributes and their change over time can be learned jointly [2], and that idea underpins most of the network-specific methods that followed.

Two studies from 2021–2022 are especially relevant here. Deng and Hooi learned the dependency graph among sensor variables directly from multivariate time series and used graph attention to forecast each variable; the attention weights also gave operators a starting point for explaining which relationships had broken down [3]. Anomal-E applied self-supervised, edge-aware graph learning to network flows, which reduced the need for fully labelled traffic [15]. Label efficiency matters a great deal for self-healing, since genuine failures are rare, varied and costly to annotate. Tang et al. approached the problem from the theory side and showed that anomalies tend to shift graph spectral energy toward higher frequencies, a finding that motivates band-pass filter designs for anomaly-sensitive GNNs [22].

Turning to fault localisation, Wang, Fu and Li represented Internet communication networks as graphs and used a two-stage graph deep-learning pipeline first to detect abnormal nodes and then to position the fault [23]. Ko et al. used a GNN to examine correlations among traffic features and identify which features drive DDoS detection on the CIDDs and KDDCup benchmarks [24]. APGNN went a step further for alarm handling. It builds an alarm propagation graph from a live radio access network and trains a GNN to separate true faults and root alarms from derived ones; on operational data it improved the average F1 score over the strongest baseline the authors tested [25]. GNNs have also been used on the control side, for example to drive flowlet-level multipath routing in OpenFlow networks [26]. Table 2 collects the most relevant studies.

Table 2. Representative GNN-based studies relevant to network self-healing.

Year	Study	Main task	Key strength	Main limitation
2020	Guo et al. [2]	Anomaly detection in dynamic graphs	Learns topology, attributes and temporal change together	Detection only; no recovery stage
2021	Deng & Hooi [3]	Multivariate time-series anomaly detection	Learns inter-variable graph; attention supports explanation	Generic sensor setting; no action stage
2022	Anomal-E [15]	Flow-based intrusion and anomaly detection	Self-supervised; uses edge features	Security focus; no repair
2022	Tang et al. [22]	Theory of graph anomaly detection	Spectral explanation of anomalies	Not network-specific
2022	Yan et al. [26]	GNN-driven multipath routing in SDN	Topology-aware traffic control	Optimises routing, not fault handling
2023	Wang et al. [23]	Fault detection and positioning	Graph-level view of the communication network	Two separate stages; no autonomous repair
2023	APGNN [25]	Alarm root-cause analysis and fault detection	Propagation graph built from real RAN data	Relies on historical labels
2024	Ko et al. [24]	Feature-correlation GNN for DDoS detection	Shows which traffic features matter	Attack detection, not fault recovery
2025	Li et al. [7]	Resilience via GNN-enabled DRL	Couples graph perception with adaptive policy	Security and generalisation still open

Most of these methods build on the graph convolution introduced by Kipf and Welling [27], whose layer-wise propagation rule is

$$\mathbf{H}^{(l+1)} = \sigma \left(\hat{\mathbf{D}}^{-\frac{1}{2}} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-\frac{1}{2}} \mathbf{H}^{(l)} \mathbf{W}^{(l)} \right) \quad (4)$$

Here $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix with self-loops, $\hat{\mathbf{D}}$ is its diagonal degree matrix, $\mathbf{H}^{(l)}$ holds node representations at layer l , $\mathbf{W}^{(l)}$ is a trainable weight matrix and $\sigma(\cdot)$ is a nonlinear activation. For NeuroHeal-Net we argue that the graph should contain both the physical topology and the logical and service dependencies layered on top of it. With both present, a fault in one element can be propagated along dependency edges and read in terms of the services it actually affects, which is what an operator ultimately cares about.

6. TRANSFORMERS FOR TEMPORAL PREDICTION AND EARLY WARNING

Network faults almost always have a time signature. Link utilisation, delay, packet loss, queue occupancy, CPU and memory load, radio measurements and alarm sequences all show periodic patterns, bursts and delayed effects, sometimes spread over long intervals. Self-attention suits this kind of data because it can relate any two points in a window without the vanishing-gradient problems of recurrent models. Transformer variants have been used for cellular traffic forecasting [4] and multi-scale intrusion detection [5], and spatio-temporal hybrids have been applied to traffic anomaly detection on road networks, a problem with the same graph-plus-time structure [28]. The Simba model combines a GNN and a Transformer to perform anomaly detection and root-cause analysis in 5G radio access networks [16].

For self-healing, the main gain is not a few points of forecasting accuracy. It is the chance to move the control loop earlier. If the model predicts that a link will degrade within the next Δ interval, traffic can be moved before the service-level objective is breached, which turns failure response into failure avoidance. The scaled dot-product attention at the core of the Transformer [29] is

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax} \left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}} \right) \mathbf{V} \quad (5)$$

where $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ are learned projections of the input sequence and d_k is the key dimension. With several heads, different temporal patterns (short bursts and slow drifts, say) can be captured in parallel. Rather than feeding raw counters into the Transformer, a NeuroHeal-Net implementation would more sensibly apply it to the node and edge embeddings produced by the GNN:

$$\mathbf{z}_t = \text{Transformer}(\mathbf{h}_{t-L+1}, \dots, \mathbf{h}_t) \quad (6)$$

where $\mathbf{h}_t$ is the topology-aware embedding at time t and L is the window length. The reasoning is simple: a 70% utilisation reading means one thing on an access link with spare paths and quite another on a backbone link that carries several critical services. A purely temporal model cannot tell the two apart, whereas a spatio-temporal one can.

7. REINFORCEMENT LEARNING FOR AUTONOMOUS RECOVERY AND RESILIENT ROUTING

Detecting or predicting a fault does not tell the system what to do about it. There are usually several plausible responses: reroute a flow, adjust link weights, migrate a virtual function, rebalance load, restart a service, bring standby capacity online, retune radio parameters or isolate a suspect component. Recovery is also sequential, because each action changes the state in which the next decision is made. Both properties point toward RL.

The literature shows a clear path from RL-based placement to graph-enhanced resilient routing. Avgeris, Leivadreas and Lambadaris used RL to re-place virtual network functions after failures, one of the first explicit RL formulations of self-healing in this setting [6]. GROM coupled GNN state encoding with DRL so that a routing policy trained on one topology could generalise to

topologies it had not seen [30]. Li et al. framed GNN-enabled DRL directly as a mechanism for network resilience [7], while Song et al. used GraphTransformer representations to make DRL-driven routing more robust [19]. Most recently, Makiyah et al. proposed an RL framework for self-healing fault recovery in intent-based SDNs [20]. In standard notation, the agent follows a stochastic policy

$$\pi_{\theta}(a | s) = P(A_t = a | S_t = s) \quad (7)$$

and learns its parameters by maximising expected discounted return over an episode of length T :

$$J(\theta) = \mathbb{E}_{\pi_{\theta}} \left[\sum_{t=0}^T \gamma^t r_t \right] \quad (8)$$

A reward built only on throughput is a poor fit for self-healing. It will happily trade a short outage on a minor service for a small gain elsewhere. We suggest a multi-objective reward that weighs service restoration against the main costs:

$$r_t = w_1 Av_t - w_2 PL_t - w_3 DL_t - w_4 Co_t - w_5 Rk_t \quad (9)$$

where Av_t measures availability or service restoration, PL_t is packet loss, DL_t is delay, Co_t is the cost of the action and Rk_t is a penalty for risk or policy violation, with non-negative weights $w_1, \dots, w_5$. The weights make the operator's priorities explicit and can be tuned per service class. Table 3 compares the main families of recovery intelligence.

Table 3. Comparison of candidate recovery-intelligence paradigms.

Method	State representation	Adaptation	Strength	Limitation for self-healing
Rules / thresholds	Hand-picked KPIs	None	Simple and predictable	Reactive; brittle on unseen fault combinations
Supervised ML	Feature vector	Offline	Can reach high classification accuracy	Needs labels; does not learn actions
GNN	Graph with attributes	Offline or online	Topology-aware perception	Usually stops at detection or localisation
Transformer	Temporal sequence	Offline or online	Long-range temporal modelling	Compute-heavy; does not choose actions
RL / DRL	State and action history	Online, by interaction	Learns sequential recovery policies	Safe exploration and training cost
GNN + DRL	Graph state with policy	Adaptive	Topology-aware routing and recovery	Scalability and robustness
GNN + Transformer + DRL	Spatio-temporal graph with policy	Adaptive	Potential end-to-end perception-to-action	Complex to train and validate

8. ROOT-CAUSE ANALYSIS AND EXPLAINABLE AI

A self-healing system has to tell causes from symptoms. A congested link, rising packet loss and a dozen alarms may all trace back to one failed card. Liu et al. tackled this with a real-time root-tracing framework that orders alarms in time and applies Bayesian inference to rank likely origins [12]. APGNN later recast the same problem as learning over an alarm propagation graph [25], and Simba carried it into the spatio-temporal setting [16]. The common lesson is that root-cause analysis (RCA) works better when it models how faults propagate instead of classifying each alarm in isolation.

Once a system is allowed to change the network on its own, explainability stops being optional. An engineer reviewing an automated action should be able to see which nodes, links, metrics or alarm relations drove the decision, what alternatives were considered, and why the chosen action met the relevant constraints. Surveys of XAI for 6G O-RAN [8] and for traffic classification, prediction and intrusion detection [31] identify transparency, trust, robustness and the human-machine interface as open problems, echoing the broader challenges catalogued in a meta-survey of XAI [32]. We therefore propose that the explanation attached to each action a_t should bundle four kinds of evidence:

$$\mathcal{E}(a_t) = \{\mathcal{E}_{\text{topo}}, \mathcal{E}_{\text{temp}}, \mathcal{E}_{\text{causal}}, \mathcal{E}_{\text{policy}}\} \quad (10)$$

At the perception level, GNN attention or feature attribution ($\mathcal{E}_{\text{topo}}$) highlights the nodes and edges that mattered most. At the temporal level, the Transformer can point to the time segments that drove its forecast ($\mathcal{E}_{\text{temp}}$). The RCA module contributes its ranked causal chain ($\mathcal{E}_{\text{causal}}$), and the decision layer reports the expected utility of the alternatives together with the constraints and service objectives that were respected ($\mathcal{E}_{\text{policy}}$). An explanation of this kind can be audited after the fact, which a single heat map cannot.

9. NETWORK DIGITAL TWINS AS A SAFETY LAYER

A network digital twin is a virtual counterpart of an operational network in which configurations and policies can be tried before deployment. Research on NDTs has grown quickly, and systematic reviews list optimisation, offloading, resource allocation and monitoring among the main uses [17, 18]. Almasan et al. describe the NDT as a data-driven model able to run in real time, and show how it can support ML-based performance evaluation and routing optimisation [9]. Si-Mohammed et al. have since shown how a discrete-event network simulator can be embedded in an NDT for online optimisation of IoT networks [33].

In a self-healing pipeline, the twin is most useful as a gate rather than a dashboard. Before the RL agent touches the physical network, its proposed action is replayed in the twin. If the predicted outcome would breach latency, capacity, availability or security limits, the action is rejected or replaced with a more conservative one:

$$a^* = \underset{a \in \mathcal{A}_{\text{act}}}{\operatorname{argmax}} U_{\text{twin}}(a) \quad \text{s.t.} \quad \text{QoS}_{\text{twin}}(a) \geq \text{QoS}_{\text{min}} \quad (11)$$

The twin also answers counterfactual questions. What happens to latency on the remaining paths if this link is drained? What if the virtual function is moved to the neighbouring site, or the failing node is isolated? Answers of this kind feed directly into the explanation in (10), so that operators see predicted consequences and not just after-the-fact feature importance.

10. ZERO-TOUCH AND CLOSED-LOOP NETWORK MANAGEMENT

Self-healing should not be bolted on as a standalone AI module. ETSI's Zero-touch network and Service Management (ZSM) framework defines a wider architecture for automated, end-to-end management of networks and services [34]. Its closed-loop automation specifications cover governance, policy, coordination, delegation and escalation [35], and later reports address intent-driven closed loops [36] and the role of agents in the move from automation to autonomy [21]. Machine-learning surveys of ZSM reach a similar conclusion from the research side [13, 14].

This governance layer matters for NeuroHeal-Net because it bounds what the AI components are allowed to do. No model should hold unrestricted authority over a production network. Policy has to state which actions may run automatically, which must pass digital-twin validation first, and which carry enough risk that a human must approve them. Table 4 maps each NeuroHeal-Net component onto this closed-loop view.

Table 4. Mapping of NeuroHeal-Net components to closed-loop management functions.

Component	Input	Output	Operational role
GNN perception	Topology with node and edge telemetry	Graph embeddings and anomaly scores	Detect and localise abnormal elements
Transformer predictor	Sequence of graph embeddings	Failure and service-risk probability	Early warning
RCA engine	Alarms, graph evidence, history	Ranked list of root causes	Separate causes from symptoms
RL decision agent	State, predicted risk, policy constraints	Recovery action	Adaptive repair
Digital twin	Candidate action and current state	Predicted outcome	Safety validation
XAI layer	Model states and attributions	Human-readable rationale	Trust, audit and governance
Closed-loop controller	Validated action	Configuration change	Execute and monitor recovery

11. THE NEUROHEAL-NET REFERENCE ARCHITECTURE

None of the model families above is enough by itself. NeuroHeal-Net is therefore organised as a layered pipeline in which each technique handles the part of the problem it is best at (Fig. 3).

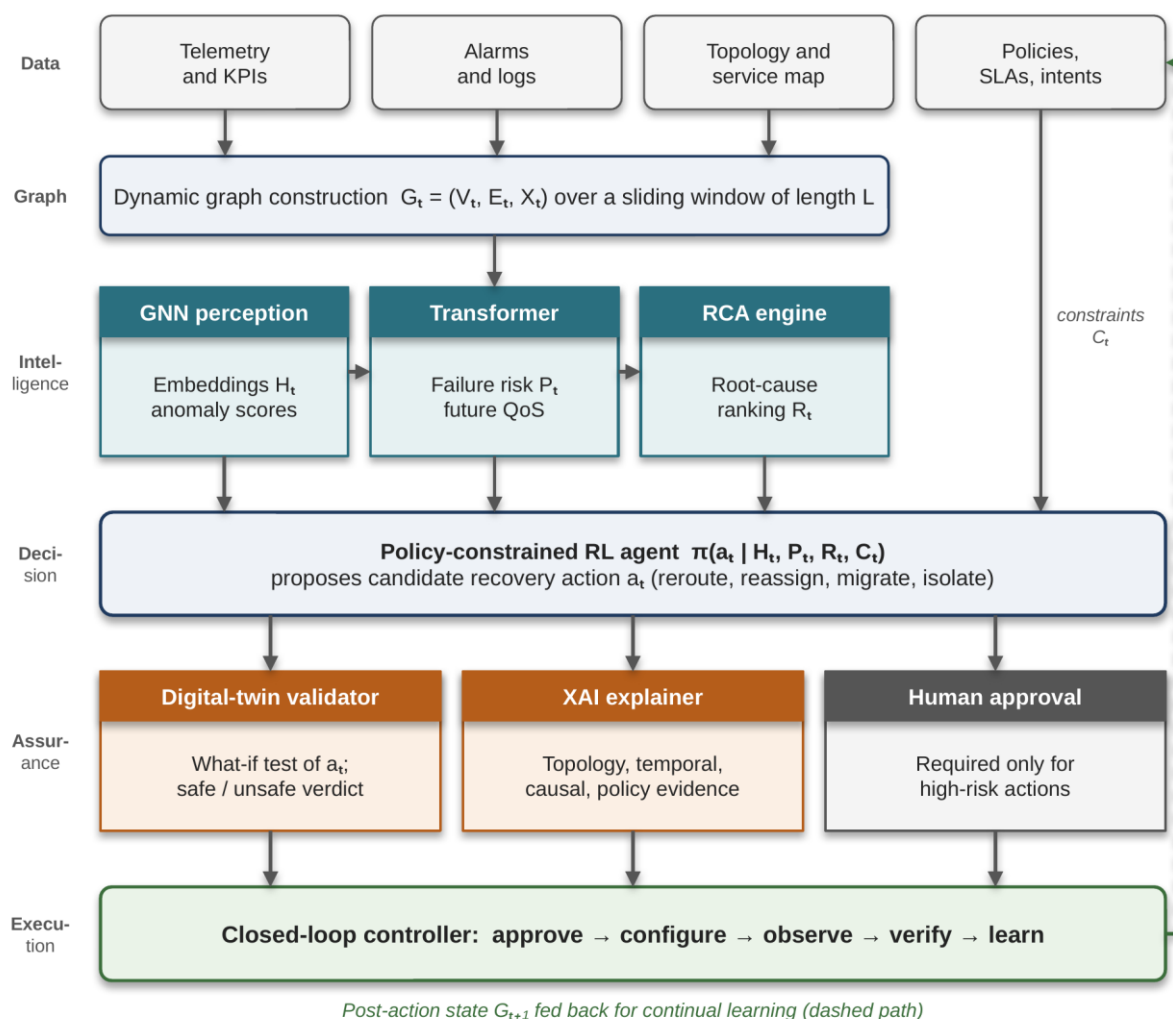


Fig. 3. NeuroHeal-Net reference architecture. Telemetry is turned into a dynamic graph, processed by perception, forecasting and root-cause modules, and passed to a policy-constrained RL agent. Every candidate action is checked in the digital twin and explained by the XAI layer before execution; the post-action state is fed back for continual learning.

The working hypothesis is that this combination narrows the distance between spotting an anomaly and repairing it safely. The GNN provides spatial context. The Transformer estimates how risk will evolve. The RCA module names the most likely originating

element. The RL agent proposes a recovery policy, the twin tests it, and the XAI layer records why it was chosen. The controller executes only validated actions and returns the resulting state to the learning loop, so the system improves with each incident it handles.

12. UNIFIED FORMULATION

Let the physical network at time t be described by the attributed graph

$$\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \mathbf{X}_t) \quad (12)$$

where $\mathcal{V}_t$ is the node set, $\mathcal{E}_t$ the set of links and dependencies, and $\mathbf{X}_t$ their attributes. Let $\mathbf{Y}_t$ be the service-level measurements, $\mathcal{A}_t$ the active alarms and $\mathcal{C}_t$ the policy constraints. The pipeline then proceeds as follows:

$$\mathbf{H}_t = \text{GNN}_{\theta}(\mathcal{G}_t) \quad (13)$$

$$\mathbf{P}_t = \text{TF}_{\phi}(\mathbf{H}_{t-L+1:t}, \mathbf{Y}_{t-L+1:t}, \mathcal{A}_{t-L+1:t}) \quad (14)$$

$$\mathbf{R}_t = \text{RCA}_{\psi}(\mathcal{G}_t, \mathcal{A}_t, \mathbf{H}_t, \mathbf{P}_t) \quad (15)$$

$$\mathbf{a}_t = \pi_{\omega}(\mathbf{H}_t, \mathbf{P}_t, \mathbf{R}_t, \mathcal{C}_t) \quad (16)$$

$$\mathbf{a}'_t = \text{Twin}(\mathbf{a}_t, \mathcal{G}_t, \mathcal{C}_t) \quad (17)$$

$$\mathcal{G}_{t+1} = \text{Execute}(\mathcal{G}_t, \mathbf{a}'_t) \quad (18)$$

Here $\mathbf{H}_t$ are the graph embeddings, $\mathbf{P}_t$ the predicted risks, $\mathbf{R}_t$ the root-cause ranking, $\mathbf{a}_t$ the proposed action and $\mathbf{a}'_t$ the action after twin filtering, which may be the original action, a safer substitute or a request for human approval. The overall goal is to minimise service disruption while respecting operational limits:

$$\min J = \alpha T_{\text{down}} + \beta \text{PL} + \gamma D + \delta C_{\text{rec}} + \varepsilon R_{\text{risk}} \quad (19)$$

$$\text{s.t. } A_v \geq A_{\min}, \quad D \leq D_{\max}, \quad \rho \leq \rho_{\max}, \quad R_{\text{sec}} \leq R_{\max} \quad (20)$$

In (19), T_{down} is downtime, PL packet loss, D latency, C_{rec} recovery cost and R_{risk} operational risk, weighted by $\alpha, \dots, \varepsilon \geq 0$. In (20), A_v is availability, ρ link utilisation and R_{sec} security risk. We have kept the formulation deliberately general so that different network types can plug in their own state variables, actions and limits. Its other purpose is to make the evaluation targets explicit, which is a precondition for comparing future implementations fairly.

13. COMPARATIVE SYNTHESIS

No single algorithm dominates the 2020–2026 literature. The strengths are spread across families. Classical ML is easier to deploy and interpret but leans on hand-engineered features. GNNs are good at relational and topological reasoning. Transformers capture long-range temporal structure. RL offers a natural way to make sequential recovery decisions, though it brings its own training and safety problems. XAI raises trust but only if it is designed into the decision pipeline from the start. Digital twins provide a safety net whose value depends on how faithfully, and how promptly, they track the real network. Table 5 sets these observations against the timeline.

Table 5. Year-by-year synthesis of the literature and the remaining gaps.

Year	Main capability	Representative works	What improved	Persistent gap	NeuroHeal-Net response
2020	Graph-based anomaly detection	[2]	Graph context for anomalies	Detection only	Joint graph and temporal prediction
2021	Predictive self-healing; fault localisation; alarm RCA	[10–12]	Earlier detection; automated root-cause inference	Little autonomous repair	RCA output feeds the RL agent
2022	Data-driven self-healing; GNN anomaly detection; ZSM surveys	[1, 13–15]	Data challenges and multi-source fusion made explicit	Labels, imbalance, latency	Self-supervised graph fusion
2023	Graph RCA; RL-based self-healing placement	[6, 23, 25]	Topology-aware diagnosis; learned actions	Stages remain disconnected	One closed loop
2024	GNN–Transformer fusion; NDT reviews	[16–18, 33]	Spatio-temporal reasoning; safe testing	Validation and deployment	Twin and XAI gates
2025	GNN–DRL resilience; robust routing; XAI surveys	[7, 8, 19, 31]	Robust, generalisable routing; trust frameworks	Scalability, security, verification	Policy-constrained DRL with twin
2026	Intent-based RL; agent-based ZSM	[20, 21]	Closer to autonomous operation	Governance and safety	Human-governed autonomy

14. OPEN RESEARCH GAPS

Reading the studies side by side brings out ten gaps that recur across network types and research groups.

Gap 1: Fragmented intelligence. Most work optimises one stage, whether detection, diagnosis or routing. An autonomous network needs these stages to work together, and the hand-offs between them are where many practical failures occur.

Gap 2: Static or partial topology. Many ML systems still treat telemetry as a flat feature vector. Real networks call for dynamic graphs, since topology, dependencies and service chains keep changing.

Gap 3: Healing that starts too late. When detection follows a KPI violation, little time is left for a careful response. Failure-probability forecasts need to feed the decision loop directly.

Gap 4: Weak causal reasoning. Correlation-based anomaly scores can flag symptoms without pinning down the initiating fault. Alarm propagation, dependency graphs and causal evidence should be used together.

Gap 5: Unsafe exploration. RL can learn good policies, but trial-and-error on a live network is not acceptable. Training in a twin or sandbox, together with hard policy constraints, is a prerequisite.

Gap 6: Explanations that stop short of the action. Feature importance does not say why one recovery action was preferred to another. Useful explanations link the evidence, the predicted outcome, the active constraints and the action chosen.

Gap 7: Scarce and imbalanced data. Real failures are infrequent and labels are expensive. Self-supervised learning, synthetic fault injection, digital twins and transfer learning all have a role.

Gap 8: Poor transfer to new topologies. A model trained on one network may fail on another. Graph and GraphTransformer representations look promising [19, 30], but systematic cross-topology testing is still rare.

Gap 9: No end-to-end evaluation. Detection F1 tells only part of the story. A self-healing system should also be judged on recovery time, availability, packet loss, false-recovery rate, safety violations and operator trust.

Gap 10: Governance and human oversight. Autonomy has to scale with risk. High-impact actions need policy controls, escalation paths and auditable explanations, in line with the ZSM closed-loop model [35, 36].

15. PROPOSED EVALUATION PROTOCOL

An experimental study of NeuroHeal-Net, or any comparable system, should evaluate the whole loop rather than report classification accuracy alone. Table 6 lists the metrics we recommend for each layer.

Table 6. Recommended evaluation metrics for integrated self-healing systems.

Layer	Metric	Definition	What it shows
Detection	Precision, recall, F1	Standard classification measures	Quality of fault detection
Prediction	MAE, RMSE, Brier score; lead time	Forecast error, calibration and warning horizon	Quality of early warning
Localisation	Top-k RCA accuracy	Share of incidents with the true root cause in the top k	Diagnostic usefulness
Recovery	MTTR	Mean time from fault onset to restoration	Speed of healing
Service	Availability	Uptime divided by observation period	Service continuity
QoS	Packet loss, latency, jitter	Measured service metrics during and after recovery	Impact on users
Safety	Constraint-violation rate	Violations divided by actions taken	Operational safety
Autonomy	Automation success rate	Successful autonomous recoveries divided by incidents	Degree of zero-touch operation
Robustness	Unseen-topology performance	Metrics on held-out graphs	Generalisation
Explainability	Fidelity, stability, operator usefulness	XAI quality measures and user studies	Trust and auditability
Twin fidelity	Prediction-to-reality error	Mean absolute gap between twin and real outcomes	Reliability of the twin

One metric deserves particular mention because it captures resilience in a single number. We define the service-impact integral over an observation period T as

$$SI = \int_0^T \left(1 - \frac{Q(t)}{Q_{\text{target}}} \right) dt \quad (21)$$

where $Q(t)$ is normalised service quality and Q_{target} is the agreed target. A lower value means less cumulative degradation. The integral rewards what users actually experience. It separates a system that notices a fault quickly but takes a long time to fix it from one that notices slightly later but restores service almost at once, a distinction that detection metrics miss entirely. Fig. 4 shows the validation workflow we recommend, with the metrics collected at each stage.

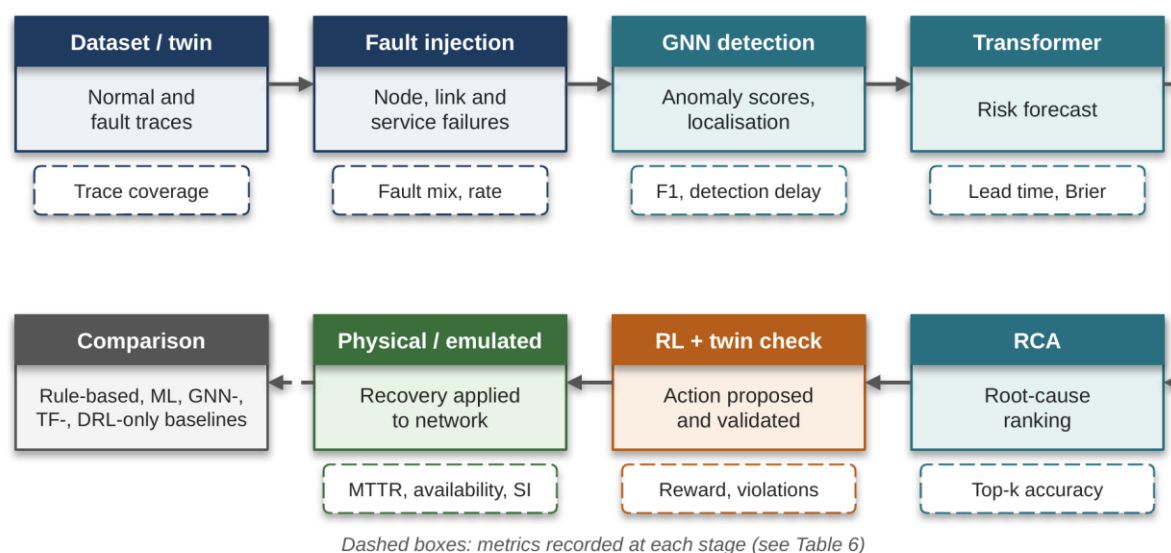


Fig. 4. Recommended experimental validation pipeline for an integrated self-healing network, with the metrics recorded at each stage.

16. DISCUSSION

The evidence reviewed here favours a shift from isolated AI functions to what might be called compositional network intelligence. Each family has a natural role and a natural blind spot. GNNs handle the non-Euclidean structure of networks well but do not, by themselves, reason over long sequences or choose actions. Transformers fill the temporal gap but are expensive and know nothing about topology unless it is supplied. RL brings sequential decision-making but depends on careful reward design and safe exploration.

XAI is needed for trust, yet explanations bolted on afterwards rarely help an engineer decide whether an action was sound. Digital twins offer a safe test bed, but only as long as they stay synchronised with the real network and can reproduce rare failure conditions.

These observations point to a hierarchical design. At the bottom, lightweight detectors trigger fast, low-risk responses. In the middle, GNN and Transformer models build a richer estimate of state and risk. At the top, an RL agent proposes recovery actions, the twin acts as a validation barrier, and the XAI layer produces an explanation and an audit trail. The controller executes only what passes policy and safety checks.

Graceful degradation should be a design requirement, not an afterthought. If the Transformer's forecast is uncertain, the RL action should not be executed blindly. If the twin is unavailable, the system should fall back to a conservative policy or ask a human. If the GNN encounters a topology unlike anything in its training data, the controller should reduce its own level of autonomy rather than extrapolate. Autonomy that adjusts to confidence in this way is more realistic than assuming that every component always works.

Limitations of this review. The study has several limitations. Its scope was restricted to English-language sources indexed in the databases listed in Section 2, so relevant work published elsewhere may have been missed. Because the included studies use different datasets, topologies and metrics, we compare them qualitatively rather than numerically. Several 2025–2026 items are recent and have yet to be widely replicated. Finally, NeuroHeal-Net is presented as a reference architecture drawn from the literature; its benefits remain hypotheses until they are tested experimentally.

17. EXPECTED CONTRIBUTIONS OF NEUROHEAL-NET

On the basis of this review, NeuroHeal-Net is best positioned as an integrated framework rather than one more fault classifier. Its expected contributions are:

- a dynamic, topology-aware GNN representation that models physical links, logical dependencies and service relations together;
- a temporal Transformer that forecasts node, link and service degradation before an SLA is breached;
- a graph-based root-cause module that combines alarm propagation with topological and temporal evidence;
- a policy-constrained RL agent that learns recovery strategies such as rerouting, resource reassignment and service migration;
- a digital-twin layer that checks every candidate action before it reaches production;
- an XAI layer that explains the evidence, predicted impact, constraints and reasons behind each action; and
- a unified evaluation protocol covering detection, prediction, diagnosis, recovery, service continuity, safety, explainability and generalisation.

18. CONCLUSION

This review traced how AI-enabled self-healing networks developed between 2020 and 2026. Clear progress has been made in automated fault detection, topology-aware graph learning, temporal forecasting, root-cause analysis, RL-based routing and recovery, explainable AI, digital twins and zero-touch management. The progress, however, is largely modular. Many studies report strong results for one part of the lifecycle, but few cover the full path from observation through prediction, diagnosis, action and validation to post-action checking.

Our main conclusion is that future self-healing networks should be engineered as safe, closed-loop intelligent systems. In such a system the GNN supplies the spatial foundation, the Transformer supplies forecasting, RCA supplies causal interpretation, RL supplies adaptive decisions, XAI supplies transparency, and the digital twin supplies a place to test counterfactuals safely. Taken together, these map directly onto the five aims of NeuroHeal-Net: topology-aware anomaly detection, early failure prediction, autonomous recovery, twin-based validation and quantitative measurement of resilience.

The real opportunity, then, is not a model with a slightly higher accuracy score. It is to show that a complete AI stack can operate dependably under unseen topologies, rare faults, noisy telemetry, competing QoS goals and hard safety limits. A NeuroHeal-Net implementation should be judged by how much disruption it prevents and how safely it restores service, not by the F1 score of its detector. Our follow-up work will implement the architecture, train it in a digital twin, test it under controlled fault injection, and compare it with rule-based, standalone ML, GNN-only, Transformer-only and DRL-only baselines.

DECLARATIONS

Funding. This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

Conflict of interest. The author declares no competing interests.

Data availability. No new data were generated or analysed in this review.

REFERENCES

- [1] T. Zhang, K. Zhu, and E. Hossain, "Data-driven machine learning techniques for self-healing in cellular wireless networks: Challenges and solutions," *Intelligent Computing*, vol. 2022, Art. no. 9758169, 2022, doi: 10.34133/2022/9758169.
- [2] J. Guo, R. Li, Y. Zhang, and G. Wang, "Graph neural network based anomaly detection in dynamic networks," *Journal of Software*, vol. 31, no. 3, pp. 748–762, 2020, doi: 10.13328/j.cnki.jos.005903.
- [3] A. Deng and B. Hooi, "Graph neural network-based anomaly detection in multivariate time series," in *Proc. AAAI Conf. Artificial Intelligence*, vol. 35, no. 5, pp. 4027–4035, 2021, doi: 10.1609/aaai.v35i5.16523.
- [4] J. Zhang *et al.*, "Cellular traffic forecasting based on inverted transformer for mobile perception dual-level base station sleep control," *Ad Hoc Networks*, vol. 161, Art. no. 103505, 2024, doi: 10.1016/j.adhoc.2024.103505.
- [5] C. Xi, H. Wang, and X. Wang, "A novel multi-scale network intrusion detection model with transformer," *Scientific Reports*, vol. 14, Art. no. 23239, 2024.
- [6] M. Avgeris, A. Leivadreas, and I. Lambadaris, "A reinforcement-learning self-healing approach for virtual network function placement," in *Proc. IEEE/IFIP Network Operations and Management Symp. (NOMS)*, 2023, doi: 10.1109/NOMS56928.2023.10154429.
- [7] X. Li, T. Zhang, J. Wang, Z. Han, J. Liu, J. Kang, D. Niyato, and A. Jamalipour, "Achieving network resilience through graph neural network-enabled deep reinforcement learning," *IEEE Network*, vol. 39, no. 6, pp. 279–287, 2025, doi: 10.1109/MNET.2025.3542450.
- [8] B. Brik *et al.*, "Explainable AI in 6G O-RAN: A tutorial and survey on architecture, use cases, challenges, and future research," *IEEE Communications Surveys & Tutorials*, vol. 27, no. 5, pp. 2826–2859, 2025, doi: 10.1109/COMST.2024.3510543.

- [9] P. Almasan *et al.*, “Network digital twin: Context, enabling technologies, and opportunities,” *IEEE Communications Magazine*, vol. 60, no. 11, pp. 22–27, 2022.
- [10] T. R. Reshmi and M. Azath, “Improved self-healing technique for 5G networks using predictive analysis,” *Peer-to-Peer Networking and Applications*, vol. 14, pp. 375–391, 2021, doi: 10.1007/s12083-020-00926-1.
- [11] A. R. Mohammed, S. A. Mohammed, D. Côté, and S. Shirmohammadi, “Machine learning-based network status detection and fault localization,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, 2021, doi: 10.1109/TIM.2021.3094223.
- [12] M. Liu, X. Qi, L. Liu, and H. Pan, “Roots-tracing of communication network alarm: A real-time processing framework,” *Computer Networks*, vol. 192, Art. no. 108037, 2021, doi: 10.1016/j.comnet.2021.108037.
- [13] J. Gallego-Madrid, R. Sanchez-Iborra, P. M. Ruiz, and A. F. Skarmeta, “Machine learning-based zero-touch network and service management: A survey,” *Digital Communications and Networks*, vol. 8, no. 2, pp. 105–123, 2022, doi: 10.1016/j.dcan.2021.09.001.
- [14] M. Liyanage, Q.-V. Pham, K. Dev, S. Bhattacharya, P. K. R. Maddikunta, T. R. Gadekallu, and G. Yenduri, “A survey on zero touch network and service management (ZSM) for 5G and beyond networks,” *Journal of Network and Computer Applications*, vol. 203, Art. no. 103362, 2022, doi: 10.1016/j.jnca.2022.103362.
- [15] E. Caville, W. W. Lo, S. Layeghy, and M. Portmann, “Anomal-E: A self-supervised network intrusion detection system based on graph neural networks,” *Knowledge-Based Systems*, vol. 258, Art. no. 110030, 2022, doi: 10.1016/j.knosys.2022.110030.
- [16] A. Hasan, C. Boeira, K. Papry, Y. Ju, Z. Zhu, and I. Haque, “Root cause analysis of anomalies in 5G RAN using graph neural network and Transformer,” arXiv preprint arXiv:2406.15638, 2024.
- [17] R. Verdecchia, E. Vicario, L. Scommegna, B. Picano, and M. Becattini, “Network digital twins: A systematic review,” *IEEE Access*, vol. 12, pp. 145400–145416, 2024, doi: 10.1109/ACCESS.2024.3453034.
- [18] A. M. V. V. Sai, C. Wang, Z. Cai, and Y. Li, “Navigating the digital twin network landscape: A survey on architecture, applications, privacy and security,” *High-Confidence Computing*, vol. 4, no. 4, Art. no. 100269, 2024, doi: 10.1016/j.hcc.2024.100269.
- [19] Y. Song, J. Zhou, X. Li, and J. Liu, “Enabling robust and generalized DRL-driven routing optimization via exploration of graph neural networks,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 11, no. 6, pp. 4281–4298, 2025, doi: 10.1109/TCCN.2025.3544276.
- [20] E. H. Makiyah, M. N. Rasool, and F. A. Rawdhan, “A reinforcement learning framework for self-healing fault recovery in intent-based SDNs,” *Discover Networks*, vol. 2, Art. no. 14, 2026, doi: 10.1007/s44354-026-00028-z.
- [21] ETSI, “Zero-touch network and Service Management (ZSM); Study on the utilization of agents in autonomous networks,” ETSI GR ZSM 020, and “Study of ZSM framework from automation to autonomy,” ETSI GR ZSM 021, 2026.
- [22] J. Tang, J. Li, Z. Gao, and J. Li, “Rethinking graph neural networks for anomaly detection,” in *Proc. 39th Int. Conf. Machine Learning (ICML)*, PMLR vol. 162, pp. 21076–21089, 2022.
- [23] X. Wang, Z. Fu, and X. Li, “A graph deep learning-based fault detection and positioning method for Internet communication networks,” *IEEE Access*, vol. 11, pp. 102261–102270, 2023, doi: 10.1109/ACCESS.2023.3313003.
- [24] H. Ko, I. Praça, and S. G. Choi, “Anomaly detection analysis based on correlation of features in graph neural network,” *Multimedia Tools and Applications*, vol. 83, pp. 25487–25501, 2024, doi: 10.1007/s11042-023-15635-z.
- [25] W. Jiang and Y. Bai, “APGNN: Alarm propagation graph neural network for fault detection and alarm root cause analysis,” *Computer Networks*, vol. 220, Art. no. 109485, 2023, doi: 10.1016/j.comnet.2022.109485.
- [26] B. Yan, Q. Liu, J. Shen, and D. Liang, “Flowlet-level multipath routing based on graph neural network in OpenFlow-based SDN,” *Future Generation Computer Systems*, vol. 134, pp. 140–153, 2022, doi: 10.1016/j.future.2022.04.006.
- [27] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *Proc. Int. Conf. Learning Representations (ICLR)*, 2017.
- [28] J. Zhang *et al.*, “Automatic traffic anomaly detection on the road network with spatial-temporal graph neural network representation learning,” *Wireless Communications and Mobile Computing*, vol. 2022, Art. no. 4222827, 2022, doi: 10.1155/2022/4222827.
- [29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.
- [30] M. Ding, Y. Guo, Z. Huang, B. Lin, and H. Luo, “GROM: A generalized routing optimization method with graph neural network and deep reinforcement learning,” *Journal of Network and Computer Applications*, vol. 229, Art. no. 103927, 2024, doi: 10.1016/j.jnca.2024.103927.
- [31] A. Nascita *et al.*, “A survey on explainable artificial intelligence for Internet traffic classification and prediction, and intrusion detection,” *IEEE Communications Surveys & Tutorials*, vol. 27, no. 5, pp. 3165–3198, 2025, doi: 10.1109/COMST.2024.3504955.
- [32] W. Saeed and C. Omlin, “Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities,” *Knowledge-Based Systems*, vol. 263, Art. no. 110273, 2023, doi: 10.1016/j.knosys.2023.110273.
- [33] S. Si-Mohammed, A. Bardou, T. Begin, I. Guérin Lassous, and P. Vicat-Blanc, “NS+NDT: Smart integration of network simulation in network digital twin, application to IoT networks,” *Future Generation Computer Systems*, vol. 157, pp. 124–144, 2024, doi: 10.1016/j.future.2024.03.038.
- [34] ETSI, “Zero-touch network and Service Management (ZSM); Reference architecture,” ETSI GS ZSM 002, European Telecommunications Standards Institute, Sophia Antipolis, France.
- [35] ETSI, “Zero-touch network and Service Management (ZSM); Closed-loop automation; Part 1: Enablers,” ETSI GS ZSM 009-1 V1.1.1, 2021.
- [36] ETSI, “Zero-touch network and Service Management (ZSM); Intent-driven closed loops,” ETSI GR ZSM 016 V1.1.1, 2024.