EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

Security Risk Modeling and Protection Resilience Assessment of Smart Tourism Distributed Graph Database Based on Hybrid Random Graph Model

Xiang Li[1,*], Siying Li[1]

[1]Southeast University Chengxian College, Nanjing, 210088, China

Abstract

INTRODUCTION: Distributed graph database for smart tourism scenarios is becoming a key target of network attacks because it stores a large amount of sensitive data, such as user trajectories and consumer preferences. The network security and operational resilience of such databases directly determine the stability of the upper application ecosystem.
OBJECTIVES: To address the lack of coordination between topology-aware risk modeling, active protection, and resilience assessment, this study develops an HRG–structural framework for distributed graph databases in smart tourism. The framework explicitly defines the threat model, including external attackers with partial graph knowledge and legitimate insiders who may abuse authorized traversal privileges.
METHODS: First, a closed-loop mechanism of "risk identification–quantitative assessment–dynamic protection" is constructed. The HRG topology and the structural resilience model are coupled bidirectionally: topology-derived risk is transferred to the resilience evaluator, while quarantine and policy decisions feed back to edge probabilities and access paths. A time-decayed attack-exposure term, dynamic access control, traversal-depth restriction, sensitive-edge masking, node quarantine, and audit logging are integrated. In addition to the controlled security simulations, a real Neo4j Community 5.26.0 property-graph prototype is evaluated at 1,000–25,000 nodes. Fixed seeds, attack schedules, configuration files, raw results, and source code are supplied as Supplementary Material S1.
RESULTS: In the original controlled comparison, the HRG–structural model achieved a comprehensive performance score of 92.6±1.5, an attack interception rate of 95.8%±0.8%, and a sensitive-data leakage suppression rate of 68.9%±2.5%. In the separate reproducibility stress test, removing both time decay and topology feedback reduced interception from 87.8% to 51.6% and increased trusted-state restoration time from 1.99 h to 3.54 h. The Neo4j prototype loaded a 25,000-node/100,000-edge graph in 1.51 s with a 60.59 MB store; policy-filtered traversal and quarantine-update mean latencies were 0.038 ms and 0.110 ms, respectively. These prototype values are single-host, in-process measurements rather than distributed-cluster results.
CONCLUSION: The combined evidence supports the internal effectiveness of topology–risk–protection feedback and demonstrates technically feasible property-graph integration under the evaluated single-host and short-horizon conditions. The evidence does not constitute production-cluster validation against zero-day exploits, long-term advanced persistent threats, side channels, or multi-node collusion; those scenarios remain deployment-stage priorities.

Keywords: Smart Tourism Ecosystem, Distributed graph database, Hybrid random graph, Data security risk modeling, Protection resilience assessment, Dynamic access control, Graph traversal attack

Received on 08 May 2026, accepted on 03 August 2026, published on 25 August 2026

doi: 10.4108/eetsis.12946

*Corresponding author. Email: 865608317@qq.com

1. Introduction

Smart-tourism ecosystems increasingly depend on the continuous collection, storage, analysis, and exchange of heterogeneous data through destination platforms, mobile services, sensors, and personalized applications [1] and [2]. These data include identities, location trajectories, preferences, social relations, and operational records. As smart-tourism services integrate AI-enabled personalization and interactive digital facilities, the volume and connectivity of these records increase. A systematic review has shown that privacy and security remain comparatively underdeveloped in smart-tourism research despite the increasing scale and sensitivity of these data [3]. Because the records are strongly associated, distributed graph databases are a natural storage and query option, but the same connectivity can amplify association inference, unauthorized traversal, and cascading service disruption.

The practical urgency is illustrated by documented incidents in the tourism and hospitality sector. In a 2023 regulatory filing, MGM Resorts reported that a cybersecurity incident disrupted booking and mobile services and caused an estimated negative impact of approximately USD 100 million [4]. The UK Information Commissioner's Office reported that the Marriott incident affected an estimated 339 million guest records [5]. These cases do not constitute graph-database experiments, but they demonstrate that compromise of a tourism data ecosystem can simultaneously expose identity and travel-related information, interrupt guest-facing services, and create substantial recovery costs. Security controls should therefore be evaluated together with operational resilience.

Existing smart-tourism studies have mainly examined ecosystem design, personalization, trust, and general privacy governance [1–3]. These studies establish the importance of responsible data use but do not provide a topology-aware security and resilience model for distributed graph databases. Accordingly, this study focuses on the closed-loop coordination of graph-based risk modeling, protection decisions, and recovery assessment.

Database-security and graph-privacy research provides established components for the proposed framework. Bertino and Sandhu surveyed database-security concepts and access-control challenges [6]. Carminati et al. developed relationship-aware access control for web-based social networks [7]. CryptDB demonstrated encrypted query processing for relational databases [8], while Chase and Kamara formalized structured encryption and controlled disclosure [9]. For graph databases, Arshad et al. studied integrity verification of graph data and query results [10], and Ji et al. reviewed graph anonymization and de-anonymization risks [11]. These works are cited as technical foundations; none is represented as an experimental baseline unless explicitly implemented and defined in this study.

Network-topology, controllable-adversary simulation, and domain-resilience studies provide limited but complementary context. Xuan et al. proposed a correlation-aware local-world network model [12], which informs only the dependency-aware connection mechanism used in the author-defined hybrid random graph and should not be interpreted as a direct source of the complete HRG formulation. Safe-Sim provides a cross-domain example of controllable adversarial closed-loop simulation in transportation [13], rather than a cybersecurity attack-propagation baseline. Hu et al. developed a tourism-domain resilience index [14], while cyber-resilience concepts such as anticipating, withstanding, recovering from, and adapting to attacks are grounded in NIST systems-security engineering guidance [35]. These sources motivate distinct design considerations; none reports the numerical baseline values used in this paper.

The remaining gap is therefore a reproducible mechanism that links dynamic graph topology to quantified loss, converts the resulting risk state into access-control and quarantine actions, and feeds the post-protection topology back into resilience assessment. All experimental variants in this study are defined operationally and are not presented as published algorithms unless a corresponding implementation is explicitly cited.

Recent property-graph studies have compared access-control approaches for graph-structured data [15], proposed attribute-based access control for Neo4j [16], enforced policy decisions through graph-query rewriting [17], and specified datastore-independent graph authorization policies [18]. Security analyses of knowledge-graph reasoning [19] and broader graph models for cybersecurity [20] further demonstrate that query paths, graph semantics, and topology are attack surfaces rather than neutral data structures. These studies motivate the controls used here, but none reports the numerical results of the present HRG–structural implementation.

As for the above mentioned problems, this study proposes a security risk modeling and protection resilience assessment technology based on the integration of a hybrid random graph (HRG) and a structural model. The core research contents and technical schemes are as follows: Firstly, it constructs a trinity network security protection mechanism of "risk identification-quantitative assessment-dynamic protection". By introducing attack probability, defense intensity, risk conduction coefficient and other parameters, a network security risk quantification model is established to accurately depict the risk level in different attack scenarios. Then, based on dynamic access control, node quarantine and attack path blocking, a cross-node attack defense mechanism is designed. Besides, combined with a dynamic security state monitoring model, the whole process of network security risk management and control is achieved. Secondly, the HRG model and the structural model are integrated to construct the HRG-structural model. The HRG model is adopted to accurately depict the dynamic topology and node dependencies of the distributed graph database. By the structural model, the precise modeling of the risk loss distribution and the quantitative assessment of the protection resilience are realized to make up for the lack of synergy between the risk modeling and the resilience assessment in the existing research. Finally, through multi-dimensional experiments to verify the effectiveness and superiority of the technology, it provides technical support

for the safe and stable operation of the distributed graph database. In summary, this study intends to break through the limitations of the existing technology, so that the accurate prevention, and control and scientific assessment of the resilience of the Distributed Graph Database of smart tourism are realized.

Compared with a conventional graph model (GM), which mainly represents static node association, and a data-driven detection technique (DDT), which emphasizes warning signals without explicit structural feedback, the innovation boundary of the proposed HRG–structural model lies in four linked operations: (i) HRG estimates time-varying node dependencies and risk-conduction paths; (ii) the structural model maps these paths to loss and resilience states; (iii) protection actions such as quarantine and path blocking are fed back to update HRG connection probabilities; and (iv) resilience is evaluated as a time-varying recovery process rather than a static score. Therefore, the contribution is not a simple juxtaposition of two models but a bidirectional topology–risk–protection loop tailored to association-rich smart-tourism graph data.

2. Related work

2.1 Controlled reference scenario for risk and resilience assessment

To illustrate the limitations of a single static graph model, this study constructs an internally generated 50-node synthetic STE scenario covering data acquisition, processing, service provision, and security functions. The node count, dependency parameters, attack schedule, and loss coefficients are defined by the authors and summarized in Appendix A; the scenario is not taken from an external case report. Under this controlled configuration, the single-GM variant yields an 18% risk-identification miss rate and a resilience-assessment error above 12%. These values and Figure 1 are simulation outputs of the present study rather than values extracted from external literature.

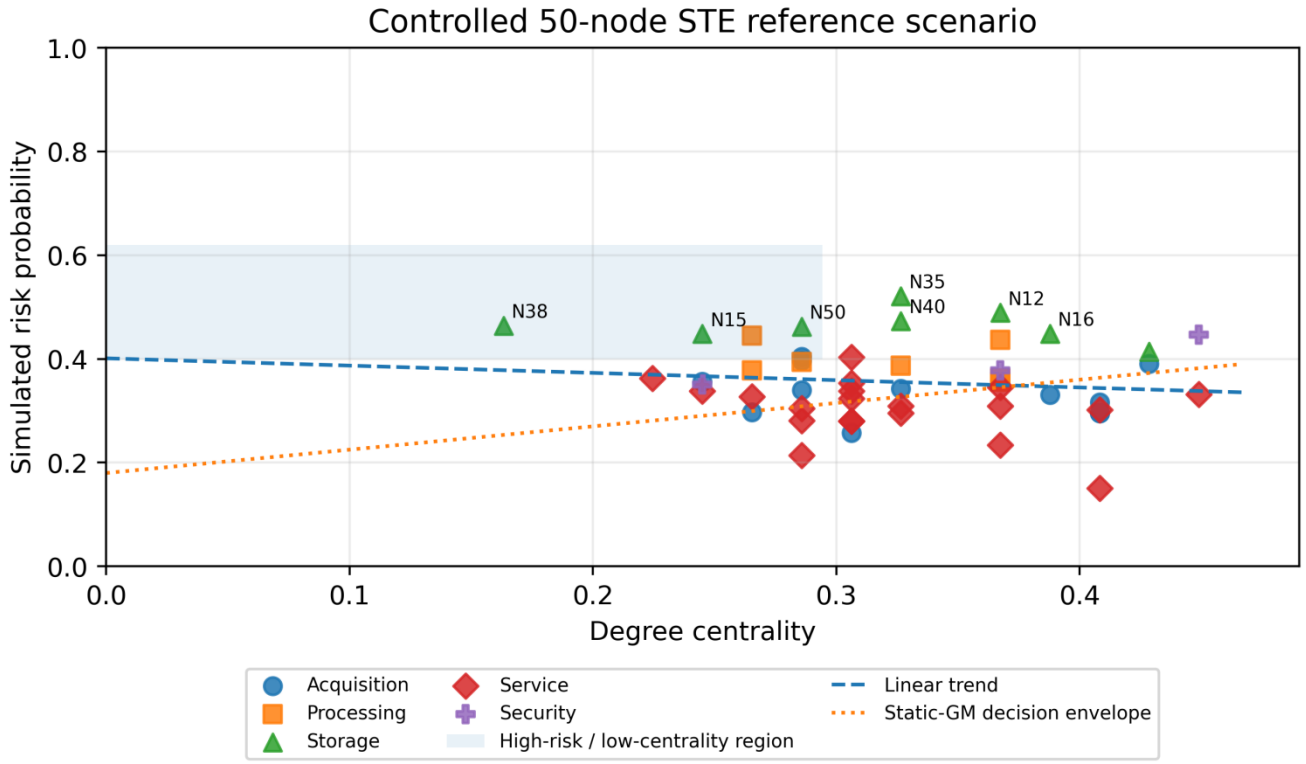

Figure 1. Controlled 50-node STE reference scenario and the mismatch between static centrality and simulated risk

Figure 1 shows the node centrality versus risk probability. The most high-risk nodes cluster at low centrality (a clear mismatch). The shaded zone marks this imbalance.

The controlled scenario reveals two limitations: (1) a static graph representation does not sufficiently quantify changing node dependencies, producing a risk-loss goodness of fit of 0.89; and (2) a single resilience index does not capture time-varying recovery or coordinate protection with resilience improvement. These observations are specific to the internally generated reference scenario and should not be interpreted as empirical findings reported by another study.

2.2 Analysis and comparison of the STE-related technologies

Table 1 compares four design categories used in this manuscript. Static-GM represents node association without dynamic propagation; Alert-only DDT represents data-driven warning without topology feedback; One-way GM+DDT combines structural features and alerts but does not feed protection actions back to the graph; and the HRG–structural model implements the complete bidirectional loop. These are controlled design variants defined by the authors, not names assigned to external papers. The comparison is shown in Table 1.

Table 1. Definitions and complementarity of the author-implemented controlled variants

Technical Type	Advantages	Disadvantages	Complementary Points
Static-GM	Fixed node-association representation	No dynamic propagation or protection feedback	Provides the static-topology reference
Alert-only DDT	Rapid data-driven warning	No graph-dependency or resilience feedback	Provides the warning-only reference
One-way GM+DDT	Combines graph features and alerts	Protection state does not update topology	Tests one-way integration
HRG–structural model	Bidirectional topology–risk–protection coupling	Higher computational and implementation complexity	Complete proposed configuration

The HRG–structural design combines dynamic topology characterization with resilience-state estimation. In the controlled configuration, the node-dependency coefficient is α=0.73 and the risk-diffusion rate is μ=0.024 h^{-1}; the fitted risk-loss goodness of fit increases from 0.89 for Static-GM to 0.97 for the full model. These parameter values and performance results are generated by the present simulation and are not attributed to the conceptual references cited in Section 1.

3. STE network risk-loss distribution modeling technology

In this section, we construct a modeling environment (Appendix A) adapted to the actual operation scenario of STE, which is a general Smart Tourism Ecosystem network simulation environment. It integrates the core nodes, covering data acquisition, processing, service provision, and security protection modules, with the connection probability between nodes being 0.38 and the cluster coefficient being 0.62. It can simulate the network operation state under multi-risk coupling scenarios, and provide real and universal technical support for modeling. With the integration of HRG and structural model as the core, and combined with the dynamic characteristics of the STE network, a complete modeling system of risk loss distribution is constructed to avoid the limitations of existing technology.

The HRG component represents topological randomness and dependency-aware connection behavior, while the structural component assesses system behavior and recovery. Here, HRG denotes an author-defined hybrid random graph, not the hierarchical random graph model used elsewhere in network science. Its local connection mechanism is informed by correlation-aware network modeling [12], but Equations (1)-(4) and the bidirectional coupling are defined in this study. The specific contribution claimed here is the bidirectional coupling used in the smart-tourism distributed-graph scenario: HRG-derived risk becomes input to the structural resilience evaluator, and protection decisions modify HRG connection probabilities and available paths. To the authors' knowledge, this exact implementation has not been evaluated in the same application setting; the novelty claim is therefore limited to this application-specific coupling and evaluation protocol.

3.1 Threat model and attacker assumptions

The adversary is modeled at three levels. External attackers can issue unauthorized or compromised-credential queries, observe the public graph schema and public point-of-interest information, and know at most 10% of the local neighbors of a target node, but they cannot obtain private keys or directly modify the trusted audit service. Legitimate insiders possess valid accounts and may attempt purpose violations, excessive traversal, or privilege escalation. A colluding attacker may coordinate compromised nodes; this scenario is defined but is not empirically evaluated in the current experiments. The primary objectives are unauthorized access, data tampering, association inference, and service disruption. Honest nodes are assumed to use authenticated channels, synchronized timestamps, and tamper-evident logs. These assumptions delimit the validity of the current results and prevent the model from being

interpreted as a complete defense against zero-day or long-term advanced persistent threats.

3.2 HRG–structural risk-loss modeling workflow

The workflow is intended for smart-tourism services in which geospatial topology, service state, and security telemetry can be updated continuously. The public datasets used to instantiate the topology are described in Section 5.1, while the algorithmic and security references are used only to support the corresponding technical components.

To realize the automation and standardization of risk loss distribution modeling, a technical framework is designed as shown in Figure 2.

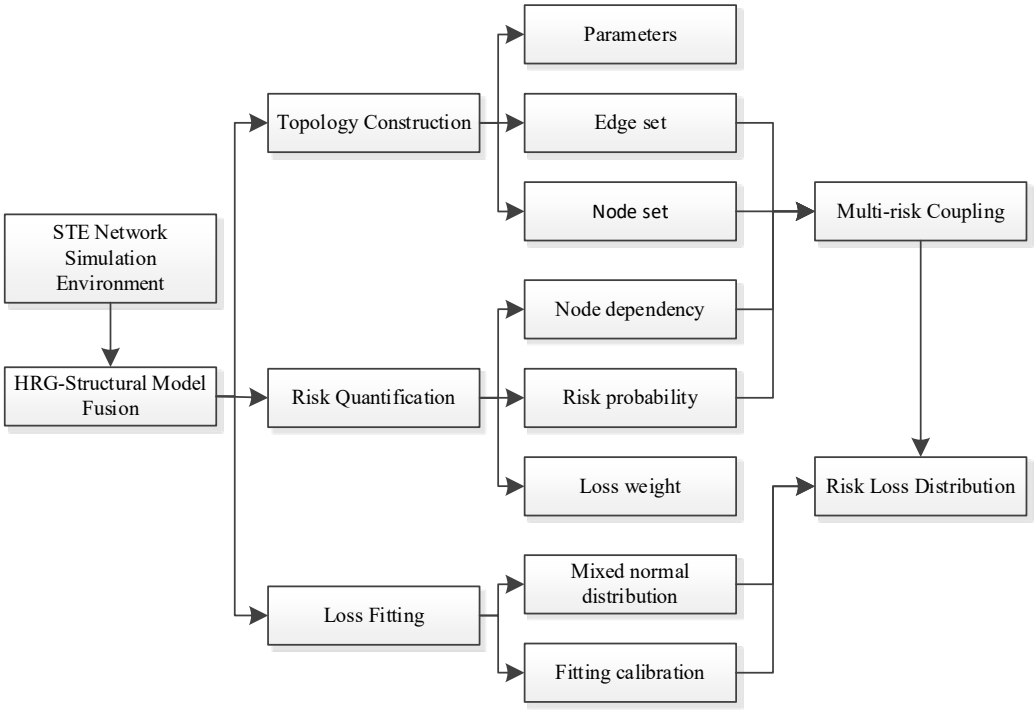

Figure 2. STE network's risk loss distribution technical framework

Figure 2 covers four core modules: network topology construction, node risk quantification, loss distribution fitting, and model verification. Each module is orderly connected to form a complete modeling process. The algorithm process follows the logic of "data input - topology construction - risk quantification - loss fitting - model verification". Firstly, the node parameters and risk data are obtained through the simulation environment. Then, the HRG topology model is constructed to quantify the node risk weight so that the loss distribution is fitted. Finally, the rationality of the model is verified through the data.

(1) First, an STE network topology is constructed by integrating HRG theory, node dependencies, and risk characteristics. The resulting topology is defined in Equation (1):

$$G(V,E,W)=\bigcup_{i\exists}^{n}\bigcup_{j\exists,j\neq i}^{n}\left\{v_i,v_j,e_{ij}\right\}\cap\left\{\theta_{ij},\alpha_{ij},\mu_{ij}\right\} \quad (1)$$

Where G is the topological structure of the STE network. $V=\{v_1,v_2,...,v_n\}$ represents the set of network nodes. n is the total number of nodes. As the i -th network node, v_i is used as the core carrier of different functional modules of the STE in application. $E=\{e_{12},e_{13},...,e_{(n-1)n}\}$ is the set of connecting edges between the nodes. e_{ij} refers to the connecting edge between the nodes v_i and v_j , and represents the interaction relationship between the nodes. $W=\{\theta_{ij},\alpha_{ij},\mu_{ij}\}$ is the set of topological characteristic parameters. θ_{ij} , with the value range of [0,1], is the connection probability of node v_i and v_j , which reflects the frequency of interaction between nodes in the application. α_{ij} represents the dependence coefficient of node v_i and v_j , which quantifies the functional dependence between nodes in the application. μ_{ij} is the diffusion rate of risk from node v_i and v_j , with the unit of h^{-1}, which describes the speed of risk transmission in the application.

(2) Second, the multi-risk loss function combines topology-dependent propagation with the instantaneous losses of data leakage, system paralysis, service interruption, and equipment failure. The comprehensive risk loss is defined in Equation (2):

$$L(t)=\sum_{k=1}^{m}\omega_k\cdot L_k(t)\cdot\exp\left(-\int_0^t\beta(\tau)d\tau\right)+\xi(t)\cdot\sigma_L \tag{2}$$

Where $L(t)$ is the comprehensive risk loss value of STE network at time t, which is used as the core quantitative index of risk loss distribution in application. As the total number of risk types, m is valued 4 in application, which corresponds to four types of risks, namely, data leakage, system paralysis, service interruption and equipment failure. ω_k refers to the weight coefficient of Class k risk, which meets $\sum_{k=1}^{m}\omega_k=1$ and is calibrated according to the occurrence probability and impact degree of the risk in the application. $L_k(t)$ refers to the instantaneous loss value of Class k risk at the moment t. $\beta(\tau)$ is the resilience recovery coefficient of the network at the moment τ, and the unit is h^{-1}, which reflects the resilience of the network after the impact of the risk in the application. $\xi(t)$ refers to the random fluctuation factor of the loss at time t, which obeys the standard normal distribution $N(0,1)$ and describes the random uncertainty of the loss in the application. As the fluctuation coefficient of the loss, σ_L quantifies the overall degree of the loss fluctuation in the application.

(3) To quantify each node's contribution to the overall loss, degree centrality, dependency, attack probability, and initial loss are combined. The node risk-loss weight is defined in Equation (3):

$$w_i=\frac{DC_i\cdot\sum_{j=1,j\neq i}^{n}\alpha_{ij}}{\sum_{i=1}^{n}\left(DC_i\cdot\sum_{j=1,j\neq i}^{n}\alpha_{ij}\right)}\cdot(1+P_i\cdot\lambda_i) \tag{3}$$

Where w_i is the risk loss weight of the i-th node, which represents the contribution degree of the node to the overall risk loss in the application. DC_i is the degree centrality of the i-th node, which reflects the core degree of the node in the network in the application. P_i is the risk occurrence probability of the i-th node, and the value range is [0,1], which is determined according to the functional characteristics of the node in the application. λ_i represents the initial loss coefficient of the i th node, which quantifies the initial loss level of the node after the risk occurs in the application.

(4) Based on the node risk-loss weights, a two-component mixed normal distribution is used to represent low- and high-loss regimes under multi-risk coupling. The fitting function is given in Equation (4):

$$f(L)=\sum_{s=1}^{2}\pi_s\cdot\frac{1}{\sqrt{2\pi}\sigma_s}\exp\left(-\frac{(L-\mu_s)^2}{2\sigma_s^2}\right) \tag{4}$$

Where $f(L)$ is the probability density function of risk loss distribution, which depicts the occurrence probability of different loss values in the application. $s=1,2$ refers to the two components of the mixed distribution, which correspond to the low loss interval and the high loss interval respectively. π_s is the weight of the s th component, which meets $\pi_1+\pi_2=1$ and is calibrated according to the actual loss data in the application. As the mean value of the s th component, μ_s characterizes the average level of the loss in the interval in the application. σ_s refers to the standard deviation of the s -th component, and the fluctuation degree of the interval loss is quantified in the application.

Based on the above algorithm process, the pseudo-code of STE network risk loss distribution modeling algorithm is designed to realize the standardization and reproducibility of the modeling process.

Pseudo-code□ STE Network Risk Loss Distribution Modeling Algorithm

Input: Network node set $V=\{v_1,v_2,...,v_n\}$, connection probability θ_{ij}, dependence coefficient α_{ij}, risk occurrence probability P_i, initial loss coefficient λ_i, simulation time T

Output: Risk loss distribution function $f(L)$, node risk weight w_i, comprehensive risk loss $L(t)$

1. Initialize network parameters: $n=50,\theta_{ij}=0.38,\alpha_{ij}=0.73,\mu_{ij}=0.024,T=24$ h

2. Generate STE network topology $G(V,E,W)$ based on the formula $G(V,E,W)=\bigcup_{i=1}^{n}\bigcup_{j=1,j\neq i}^{n}\{v_i,v_j,e_{ij}\}\cap\{\theta_{ij},\alpha_{ij},\mu_{ij}\}$

3. Calculate degree centrality DC_i for each node v_i and normalize it to [0,1]

4. Compute node risk loss weight w_i using the formula $w_i=\frac{DC_i\cdot\sum_{j=1,j\neq i}^{n}\alpha_{ij}}{\sum_{i=1}^{n}\left(DC_i\cdot\sum_{j=1,j\neq i}^{n}\alpha_{ij}\right)}\cdot(1+P_i\cdot\lambda_i)$

5. Collect real-time risk data in the simulation environment and record instantaneous loss $L_k(t)$ for each risk type

6. Determine weight coefficient ω_k of each risk type and ensure $\sum_{k=1}^{m}\omega_k = 1$

7. Calculate comprehensive risk loss $L(t)$ using the formula $L(t) = \sum_{k=1}^{m}\omega_k \cdot L_k(t) \cdot \exp\left(-\int_0^t \beta(\tau)d\tau\right) + \xi(t)\cdot\sigma_L$

8. Fit risk loss distribution using the mixed normal distribution formula $f(L) = \sum_{s=1}^{2}\pi_s \cdot \frac{1}{\sqrt{2\pi}\sigma_s}\exp\left(-\frac{(L-\mu_s)^2}{2\sigma_s^2}\right)$

9. Calibrate parameters π_s, μ_s, σ_s of the distribution function based on fitting results

10. Verify the model by comparing fitting value with actual loss data and adjust parameters if necessary

11. Output $f(L), w_i, L(t)$ and End

The case study illustrates that in the scenario of multi-risk coupling and sudden change in the number of nodes, the fitting accuracy of the modeling technology decreases slightly. Besides, there is a limitation of insufficient adaptation to the dynamic changes of network topology.

To sum up, this section constructs the STE network risk loss distribution modeling technology based on the integration of HRG and structural model, and designs a complete technical framework and algorithm process. By quantifying the topological characteristics, risk loss and distribution rules, and combining with pseudo-code to achieve modeling standardization, it provides a solid technical support for the subsequent resilience assessment, which clarifies the application limitations of the technology in dynamic topology scenarios.

4. Network security protection mechanism and risk quantification of distributed graph database

As a major storage carrier of sensitive tourism data, a distributed graph database must protect confidentiality, integrity, controlled traversal, and service availability. The proposed mechanism therefore combines database-security principles [6], relationship-aware access control [7], graph integrity verification [10], and graph privacy risk analysis [11] with HRG-based topology modeling. The mechanism is designed to quantify and actively manage unauthorized access, data tampering, association leakage, and cross-node attacks.

4.1 Network security risk quantification model

Considering the node topology characteristics of the distributed graph database and the network attack rules, the network security risk quantification index is extended based on the HRG model. Besides, the attack probability, defense strength, risk conduction coefficient, and other parameters are introduced to construct the network security risk quantification model, which accurately depicts the risk level distribution under different attack scenarios and provides a quantitative basis for the formulation of protection strategies.

The overall network-security risk is quantified by Equation (5):

$$R_{sec} = \sum_{i=1}^{n}\omega_i \cdot P_{att,i} \cdot (1 - D_i) \cdot \lambda_{i,j} + \xi \tag{5}$$

Where R_{sec} is the overall network security risk value of the distributed graph database, which is used as the core quantitative index of the network security status in the application. The value range is [0, 1], and the closer the value is to 1, the higher the network security risk is. n denotes the total number of distributed graph database nodes, which corresponds to the number of core nodes, such as data acquisition, processing and storage in the smart tourism scenario in the application. ω_i represents the security weight of the i th node, which is jointly determined by the node degree centrality and the data sensitivity degree, and satisfies $\sum_{i=1}^{n}\omega_i = 1$. Besides, the weight assignment of the sensitive data node, such as the user track storage node, in the application is higher than that of the ordinary node. $P_{att,i}$, whose value range is [0, 1], is the probability of the i th node being attacked by the network. Further, it is calibrated according to the node protection capability and the historical attack data in the application. D_i, which reflects the effectiveness of security measures, such as encryption protection and access control of the node, refers to the defense strength of the i th node, and with a value range of [0, 1]. $\lambda_{i,j}$ represents the risk conduction coefficient from the i th node to the j th node, with a value range of [0, 1], which quantifies the diffusion ability of risk among nodes, and is positively correlated with the dependence coefficient among nodes. ξ represents a network security risk correction term with a value range of [0, 0.1], which is used to correct the impact of random factors, such as network delay and topology fluctuation on risk quantification.

To represent cumulative attack exposure, the instantaneous attack probability in Equation (5) is replaced by the discrete time-decay term defined in Equation (5a):

$$\mathbf{P}_{\mathrm{att},i}^{T}(t_k) = 1 - \exp\left[-\mathbf{A}_i(t_k)\right]$$
$$\mathbf{A}_i(t_k) = \sum_{r=0}^{K-1}\rho_i(t_{k-r})e^{-\lambda r\Delta t}\Delta t \tag{5a}$$

Where $\rho_i(t_{k-r})$ is the observed attack-arrival intensity at the r-th preceding monitoring interval, Δt is the update interval, $K\Delta t$ is the observation window, and $\lambda > 0$ is the decay rate. Recent repeated attacks therefore accumulate, whereas older events gradually lose influence. In the current experimental configuration, the monitoring window is 24 h and the state is updated every 0.5 h; λ must be calibrated from deployment logs.

Attack type is identified from the weighted match between observed behavior features and the three modeled attack classes, as defined in Equation (6):

$$T_{att} = \arg\max_{k\in\{1,2,3\}} \left(\frac{\sum_{m=1}^{M} f_{k,m} \cdot S_m}{\sum_{m=1}^{M} S_m} \right) \tag{6}$$

Where T_{att} is the identified network attack type. $k=1$ corresponds to the unauthorized access. $k=2$ corresponds to data tampering, and $k=3$ corresponds to association leakage. M denotes the total number of attack behavior characteristic parameters, including 6 types of characteristic parameters, such as access frequency, data modification, and association query times. $f_{k,m}$ refers to the matching degree of the k th characteristic parameter corresponding to the m th attack, and the value range is [0, 1]. S_m refers to the weight of the m th feature parameter, which satisfies $\sum_{m=1}^{M} S_m = 1$, and the weight of the feature parameter corresponding to data tampering in the application is higher than that of other attack types.

4.2 Cross-node attack defense mechanism

To settle the problems of fast spread and wide impact of cross-node attacks in distributed graph databases, a cross-node attack defense mechanism is designed based on the HRG topology model. Through dynamic access control, node quarantine, and attack path blocking, the attack can be accurately intercepted and suppressed, which improves the network security resilience.

Based on the current node-risk state, the fine-grained access-control decision is computed using Equation (7):

$$A_{ctrl}(i,j) = \begin{cases} 1, & R_{sec,i} \le \theta \text{ ÇÒ } P_{auth}(j) \ge \gamma \\ 0, & \text{other} \end{cases} \tag{7}$$

Where $A_{ctrl}(i,j)$ is the access control matrix element. $A_{ctrl}(i,j)=1$ represents that the j th access subject can access the i th node. $A_{ctrl}(i,j)=0$ represents that access is prohibited. $R_{sec,i}$ refers to the network security risk value of the i th node. θ is the risk threshold, which is valued 0.3 in the application and can be adjusted according to the actual security requirements. $P_{auth}(j)$ represents the authorization level of the j th access subject, and the value range is [0,1]. γ refers to the authorization level threshold, which is 0.6 in the application to guarantee that only highly authorized subjects can access highly sensitive nodes.

To make the access policy responsive to attack status, the node-specific threshold is recalculated at each monitoring interval according to $\theta_i(t) = \text{clip}\left[\theta_0 - \kappa_1 R_{\text{sec},i}(t) - \kappa_2 \frac{dR_{\text{sec},i}(t)}{dt}, \theta_{\min}, \theta_{\max} \right]$. A rising risk value or risk-growth rate lowers the threshold and can immediately revoke an active session; recovery gradually restores permissions. This converts Equation (7) from a static authorization rule into continuous least-privilege control. This continuous-authorization and session-revocation design is consistent with zero-trust principles [21] and multi-location cloud-native access-control guidance [22].

A node is quarantined when its security-risk level or rate of increase exceeds the configured limits. The trigger condition is defined in Equation (8):

$$I_{iso}(i) = \begin{cases} 1, & R_{sec,i} > \theta \text{ ÇÒ } \frac{dR_{sec,i}}{dt} > \mu \\ 0, & \text{other} \end{cases} \tag{8}$$

Where $I_{iso}(i)$ is the quarantine state of the i th node. $I_{iso}(i)=1$ represents the quarantine of the node, and $I_{iso}(i)=0$ represents the normal operation of the node. $\frac{dR_{sec,i}}{dt}$ represents the change rate of the security risk value of the i th node, and the unit is h^{-1}, which reflects the risk diffusion speed. μ represents the threshold of the risk diffusion speed, and the value is $0.05h^{-1}$ in the application, which indicates that the attack diffusion speed is too fast. Moreover, the node needs to be quarantined immediately.

The sensitive data privacy protection strategies are reflected in the following aspects:

(1) Fine-grained access control: Restrict traversal across nodes based on user roles and query depth. For example, the ordinary tourist account can only query the POI information of the scenic spot, and has no right to access other tourist track records in the same graph node.

(2) Sensitive edge/node shielding: For the user-user's "friend relationship" edge and the user-location's "trajectory" edge, the display is dynamically shielded according to the data sensitivity level (L1 ~ L3). These edges are deleted directly at high risk, which makes it impossible for attackers to deduce social relationships through graph traversal.

(3) Graph anonymization preprocessing: Perform k-anonymization processing on that user ID before the data is stored, and generalize the specific consumption amount into three ranges of "low/medium/high". This reduces the risk of

association leakage. Besides, even if the node is breached, it is difficult to recover an accurate portrait of the individual.

(4) Legitimate-user abuse control: Each query is checked against role, declared purpose, data sensitivity, traversal depth, and a rolling query budget. Separation of duties, just-in-time privileges, dual authorization for L3 data, and tamper-evident audit logs are used to detect and contain abuse by authorized users.

(5) Graph-traversal attack defense: The query interceptor limits path depth, blocks sensitive edge types, counts repeated neighborhood-expansion requests, and terminates a query when cumulative path risk exceeds a threshold. This mechanism is specific to association-rich graph databases and complements conventional row-level access control.

(6) Side-channel leakage reduction: Fixed-size pagination, response-size padding, batching, rate limiting, and coarse response-time buckets are recommended to reduce leakage from result size and query timing. These controls are included in the deployment design, but their overhead and leakage reduction were not quantitatively evaluated in the current experiments.

4.3 Dynamic monitoring of network security status

To realize the real-time management and control of network security risk, a dynamic monitoring model of network security status based on a distributed graph database is constructed. It combines the real-time operation data and attack characteristics of nodes to calculate the network security risk value and security status level in real time, so as to find potential security risks in time and provide support for the trigger of the defense mechanism.

The global database-security state is classified as safe, warning, or dangerous according to Equation (9):

$$L_{sec}=\begin{cases}\text{safety}, & R_{sec}\le 0.2\\ \text{Warning}, & 0.2<R_{sec}\le 0.5\\ \text{danger}, & R_{sec}>0.5\end{cases} \tag{9}$$

Where L_{sec} is the network security status level of the distributed graph database, which is divided into three levels: security, early warning and danger. R_{sec} represents the overall network security risk value, which corresponds to the risk threshold of different levels, and can be dynamically calibrated according to the security requirements of the smart tourism scene in the application.

The monitored risk state is updated at each sampling interval using Equation (10):

$$R_{sec}(t+\Delta t)=R_{sec}(t)\cdot(1+\Delta R)+\sum_{i=1}^{n}\omega_i\cdot\Delta P_{att,i} \tag{10}$$

Where $R_{sec}(t+\Delta t)$ denotes the overall network security risk value at time $t+\Delta t$. $R_{sec}(t)$ denotes the overall network security risk value at time t. Δt is the monitoring update period, and the value is 0.5 h in application to guarantee the real-time risk monitoring. Determined by factors, such as network topology fluctuation and defense measure adjustment, ΔR represents the risk change rate, with the value range of [-0.1, 0.1]. As the variation of the attack probability of the i th node, $\Delta P_{att,i}$ is updated according to the real-time attack monitoring data in the application.

The network security protection mechanism constructed in this section realizes the integration of network security risk quantification, defense, and monitoring of the distributed graph database. Therefore, it provides a solid theoretical and technical support for the subsequent network security-related experimental tests, and makes up for the lack of coordination between network security protection and resilience assessment in the existing studies.

5. Experimental analysis

5.1 The experimental environment setup

Simulation data and real environment mixed data are both used in the experiment. That is because single simulation data is difficult to simulate the complex scenario of multi-risk coupling in the STE network, and has the problems of insufficient sample size and incomplete coverage of risk scenarios. Further, the mixed data can take into account the authenticity of the scenario and the integrity of the sample, thus improving the reliability and universality of the experimental results.

The data set selection is:

(1) OpenStreetMap data distributed through the Registry of Open Data on AWS [23]. The data provide road, service-facility, and point-of-interest features used only to instantiate spatial topology and candidate node dependencies; they are not treated as labeled security incidents.

(2) The Graph-Based ANPR Dataset of Tourist Routes and Visits in Smart Villages of Alpujarra [24], containing route and visit information for 467,773 distinct vehicles recorded by four cameras in Spain. Cameras are represented as nodes and temporal vehicle transitions as edges. The data support topology and co-occurrence modeling, but vehicle records are not equated with verified tourist identities.

Data-processing assumptions and potential deviations: OpenStreetMap is volunteered geographic information and may exhibit regional differences in coverage and completeness; the ANPR dataset reflects four cameras in a Spanish smart-village setting and vehicle flows rather than directly identified tourists. Duplicated records are removed, timestamps are aligned to 0.5 h windows, continuous variables are Z-score normalized, and synthetic attack labels are generated independently of original identities. Missing observations are not interpreted as non-travel events. Geographic coverage, sensor placement, class balance, and map matching may affect external validity; the results are

therefore mixed-data simulation evidence rather than direct production evidence.

The experimental environment configuration and training parameters are displayed in Table 2 and Table 3.

Provenance of comparison results: Every numerical value in Tables 4–9 is reported as an output of an author-implemented controlled configuration executed under the common environment in Tables 2 and 3. No performance number is transcribed from the conceptual references. We have provided the configuration files, random seeds, attack schedules, and raw run summaries as supplementary material.

Extended-experiment provenance: Section 5.3 reports (i) a separately scripted stress-test extension using ten fixed seeds and 1,000 simulated attacks per run, and (ii) a real Neo4j Community 5.26.0 embedded property-graph prototype using five independent graph seeds. The stress-test extension is intentionally separate from the calibration experiment in Tables 4–9; therefore, its absolute rates are not expected to be numerically identical. The Neo4j measurements were collected in a virtualized single-host environment with OpenJDK 21, a 4 GiB memory limit, and a 2 GiB JVM heap. Neither experiment is represented as a production distributed-cluster benchmark.

Table 2. Experimental hardware, software and training parameters configuration

Configuration Type	Items	Parameters
Hardware Configuration	CPU	Intel Core i9-13900K
	GPU	NVIDIA RTX 4090 (24GB)
	RAM	64GB DDR5 5600MHz
	Storage	2TB NVMe SSD
Software Configuration	Operating System	Ubuntu 22.04 LTS Server
	Programming Language	Python 3.10
	Simulation Tool	NS-3 3.40
	Data Processing Tool	Pandas 2.1.4
Training Parameters	Learning Rate	0.001
	Batch Size	64
	Iteration Times	10000

Table 3. Experimental Test Parameters

Test Parameter Type	Parameter Names	Parameter Values
Confidence Interval Parameter	Confidence Level	95%
	Margin of Error	±0.05
Sample Sampling Parameter	Sampling Method	Stratified Sampling
	Sample Size	1000 Samples
	Sampling Frequency	1 Time/0.5h
Other Test Parameters	Test Iterations	10 Times
	Data Preprocessing Method	Z-Score Normalization

5.2 The comparative analysis of the test

Comprehensive performance test

The test verifies the comprehensive performance of the HRG–structural model in the distributed graph database security risk modeling and security resilience assessment.

The comprehensive performance score combines risk-loss fitting, resilience-estimation error, dynamic resilience variation, and a stability correction, as defined in Equation (11):

$$P_{\text{com}} = \frac{1}{4}\left[\frac{R^2}{\max\left(R^2\right)} + \left(1 - \frac{\left|\Gamma_{\text{pred}} - \Gamma_{\text{true}}\right|}{\Gamma_{\text{true}}}\right) + \frac{1}{1+\sigma_L} + \exp\left(-\frac{\Delta\Gamma}{\max(\Delta\Gamma)}\right)\right] \quad (11)$$

In the formula, P_{com}, used as the core evaluation index of comprehensive performance in application, is the comprehensive performance score, with the value range of [0, 100]. R^2 is the goodness of fit of risk loss distribution, and $\max(R^2)$ represents the maximum value of goodness of fit in all comparison methods. Γ_{pred} is the predicted value of resilience, Γ_{true} refers to the true value of resilience, and $|\Gamma_{pred} - \Gamma_{true}|$ is the absolute error of resilience evaluation. $\Delta\Gamma$ refers to the dynamic change of resilience. $\max(\Delta\Gamma)$ represents the maximum value of resilience change in all comparison methods. $\exp\left(-\frac{\Delta\Gamma}{\max(\Delta\Gamma)}\right)$ is the resilience stability correction term, which quantifies the stability of dynamic adjustment of resilience.

Three author-implemented controlled baselines are compared: Static-GM uses a fixed graph and no dynamic risk propagation; Alert-only DDT uses the same input features and warning thresholds but omits structural feedback and resilience optimization; and One-way GM+DDT combines graph features with alerts while preventing protection actions from updating the topology. All variants use the same dataset splits, attack schedule, iteration count, and scoring equations. They are not presented as reproductions of external publications.

The controlled comparison results are reported in Table 4 and visualized in Figure 3.

Table 4. Comprehensive performance of controlled configurations

Test Methods	Fitting Goodness R^2 (95% CI)	Resilience Evaluation Error (95% CI)	Loss Volatility σ_L	Comprehensive Performance P_{com} (95% CI)
Static-GM	0.89±0.03	12.3%±1.5%	0.21	72.5±2.1
Alert-only DDT	0.87±0.04	14.1%±1.8%	0.23	69.8±2.3
One-way GM+DDT	0.93±0.02	8.7%±1.2%	0.19	81.3±1.8
HRG–structural model (Ours)	0.97±0.01	4.2%±0.8%	0.15	92.6±1.5

As shown in Table 4, the HRG–structural model achieves a risk-loss fitting goodness of 0.97±0.01 and a resilience-assessment error of 4.2%±0.8%. Its comprehensive score of 92.6±1.5 is 20.1, 22.8, and 11.3 points above Static-GM, Alert-only DDT, and One-way GM+DDT, respectively. These differences compare controlled implementations under the same simulation and do not constitute cross-paper performance comparisons.

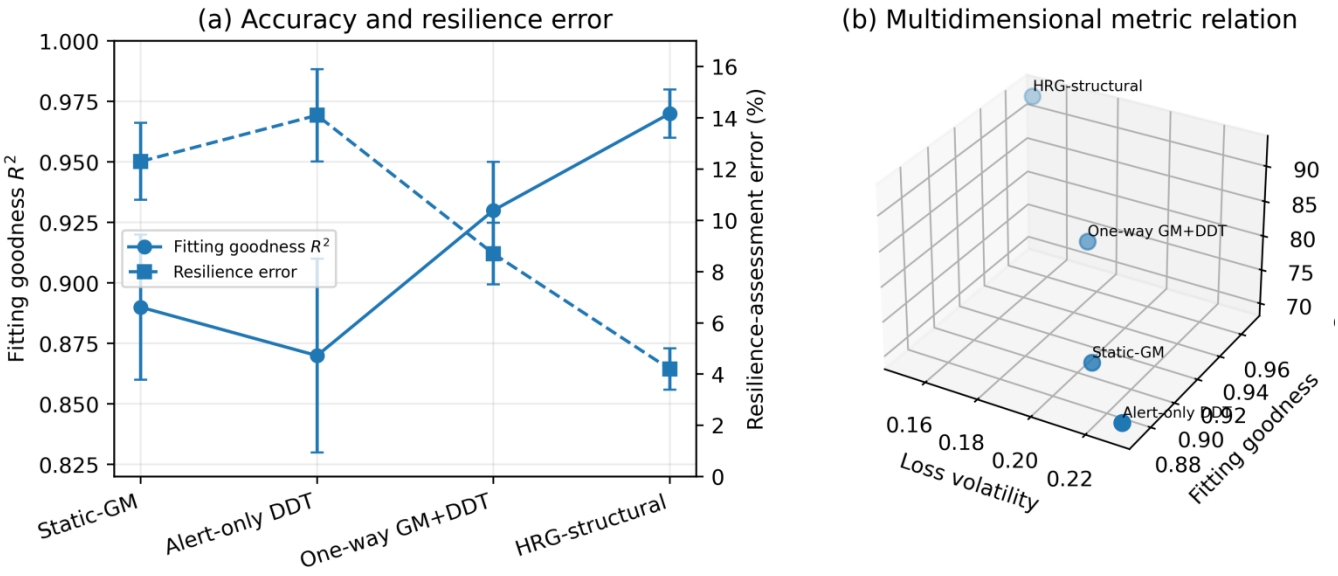

Figure 3. Comprehensive-performance comparison of the four author-implemented controlled configurations.

Figure 3(a) compares fitting goodness and resilience-assessment error for Static-GM, Alert-only DDT, One-way GM+DDT, and the full HRG–structural model. Figure 3(b) relates loss volatility, fitting goodness, and comprehensive performance for the same four configurations. All points and confidence intervals are produced by the shared protocol in Tables 2 and 3; the figure is not a cross-paper meta-analysis.

Overall, the full model obtains the best score among Static-GM, Alert-only DDT, and One-way GM+DDT in the defined simulation. The interpretation is limited to these author-implemented variants and should not be generalized to all graph models or data-driven detection methods.

The robustness test

In this part, we test and verify the stability of the technology in the face of disturbance scenarios, such as database node failure and sudden change of attack traffic.

Robustness under node failure and attack-traffic disturbance is quantified by Equation (12):

$$R_{rob} = \frac{P_{com}(\delta)}{P_{com}(0)} \cdot \exp\left(-\frac{\delta \cdot \mu_{ij}}{\beta(\tau)}\right) \cdot \left[1 - \frac{\sum_{i=1}^{n} w_i \cdot \delta_i}{\max\left(\sum_{i=1}^{n} w_i \cdot \delta_i\right)}\right] \quad (12)$$

Where R_{rob} is the robustness coefficient, with the value range of [0, 1]. The closer to 1, the stronger the robustness is. $P_{com}(\delta)$ is the comprehensive performance score when there is disturbance, and $P_{com}(0)$ is the comprehensive performance score when there is no disturbance. With the value range of [0, 0.5], δ is the disturbance intensity, which represents the severity of node failure and risk diffusion mutation. δ_i is the disturbance coefficient of the i th node, with a value range of [0,1], which represents the fault degree of a single node. $\sum_{i=1}^{n} w_i \cdot \delta_i$ is the overall disturbance level, and $\max\left(\sum_{i=1}^{n} w_i \cdot \delta_i\right)$ represents the maximum overall disturbance level.

Robustness is evaluated against three additional author-implemented variants: HRG with a linear risk classifier, a structural resilience model with an alert-only detector, and a graph-neural risk encoder coupled to HRG without the complete protection-feedback loop. The variants share the same disturbance schedule, scoring code, and parameter-calibration procedure. These configurations are operational definitions created for controlled ablation-style comparison rather than named published algorithms.

The robustness results are presented in Table 5 and Figure 4.

Table 5. Robustness of controlled variants under disturbance

Test Methods	Disturbance Intensity δ	Robustness Coefficient R_{rob} (95% CI)	Performance Attenuation Rate (95% CI)	Stability Duration (95% CI)
HRG + linear risk classifier	0.3	0.78±0.04	18.7%±2.1%	3.2±0.4h
Structural model + alert-only detector	0.3	0.72±0.05	24.3%±2.5%	2.7±0.5h
GNN encoder + HRG topology	0.3	0.81±0.03	15.9%±1.8%	3.6±0.3h
HRG–structural model (Ours)	0.3	0.92±0.02	7.4%±1.1%	5.1±0.2h

As shown in Table 5, at disturbance intensity 0.3 the full model achieves a robustness coefficient of 0.92±0.02, a performance attenuation rate of 7.4%±1.1%, and a stability duration of 5.1±0.2 h. These results establish relative robustness only among the controlled variants used in this study.

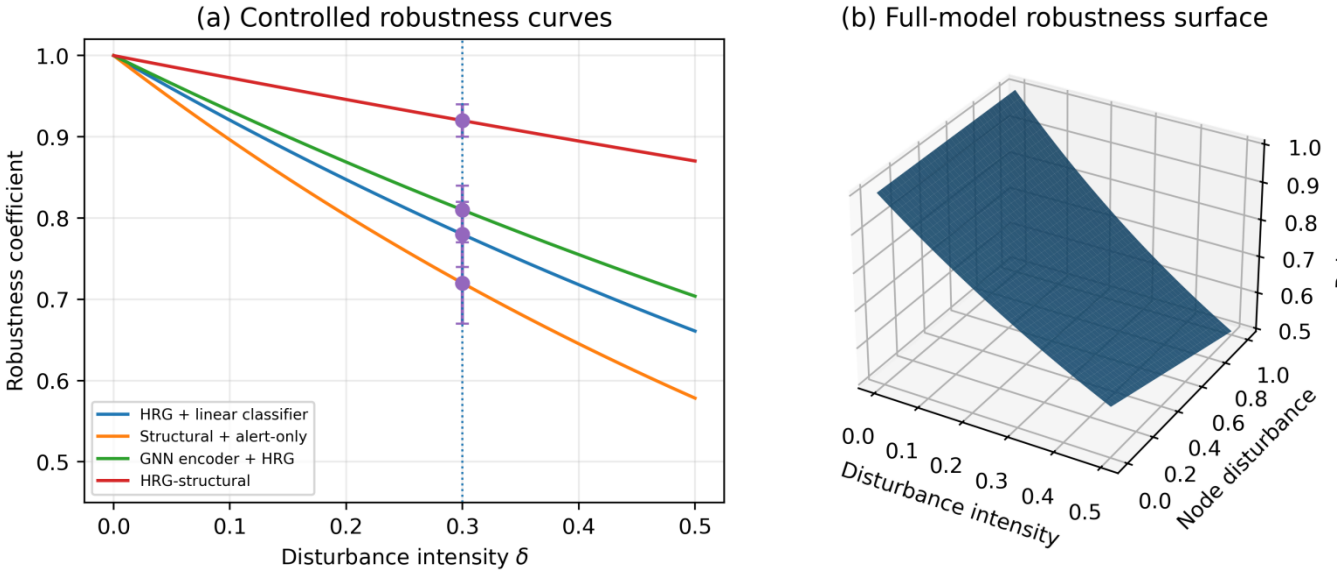

Figure 4. Robustness of author-implemented variants under node and attack-traffic disturbance

Figure 4(a) reports robustness curves for HRG + linear risk classifier, structural model + alert-only detector, GNN encoder + HRG topology, and the full HRG–structural model; the vertical line marks the tabulated disturbance intensity δ=0.3. Figure 4(b) visualizes the full-model response surface. The conclusion is restricted to the common disturbance schedule and parameter settings of this study.

The ablation test

This part verifies the effectiveness of the key modules of the HRG–structural model in this study. By removing different modules, the contribution of each module to the overall performance is analyzed to clarify the value of technological innovations.

The contribution of each ablated module is quantified by Equation (13):

$$C_k = \frac{P_{com}(full) - P_{com}(ablation_k)}{P_{com}(full)} \cdot \left[1 + \frac{\Delta R_k^2 + \Delta\Gamma_k}{\max(\Delta R_k^2 + \Delta\Gamma_k)}\right] \quad (13)$$

Where C_k is the performance contribution of the k th core module, and the value range is [0, 1]. The larger the value is, the more important the module is. $P_{com}(full)$ is the comprehensive performance score of the complete model, and $P_{com}(ablation_k)$ is the comprehensive performance score after the k th module is removed. ΔR_k^2 represents the change of goodness of fit ($R_{full}^2 - R_{ablation_k}^2$) after the k th module is removed. $\Delta\Gamma_k$ is the change of resilience evaluation accuracy ($\Gamma_{acc,full} - \Gamma_{acc,ablation_k}$) after the k th

module is removed. $\max(\Delta R_k^2 + \Delta\Gamma_k)$ represents the maximum of the sum of goodness of fit and resilience accuracy changes across all ablation modules.

The test results are shown in Table 6.

Table 6. Ablation test results

Ablation Model	Fitting Goodness R^2 (95% CI)	Resilience Evaluation Accuracy (95% CI)	Comprehensive Performance P_{com} (95% CI)	Module Contribution C_k (95% CI)
Full Model (Ours)	0.97±0.01	95.8%±0.8%	92.6±1.5	—
Ablation HRG Module	0.88±0.03	86.2%±1.4%	75.3±2.2	0.187±0.023
Ablation Structural Model Module	0.91±0.02	83.5%±1.6%	78.9±2.0	0.148±0.019
Ablation Resilience Dynamic Adjustment Module	0.94±0.02	89.7%±1.2%	85.4±1.7	0.078±0.015

As illustrated, the HRG module has the highest contribution, up to 0.187. Besides, the comprehensive performance decreases by 17.3 points after removal, which confirms its core role in topology characterization and loss modeling. The structural model module and the dynamic resilience adjustment module are also indispensable, which jointly support the high performance of the model and highlight the rationality of the technical innovation in this study.

The practicality test

In this section, we test and verify the practical value of technology in real scenarios to reduce security risks and enhance system protection resilience.

The practicality index combines loss reduction, resilience improvement, resource increment, and recovery time, as defined in Equation (14):

$$U_{pra} = \frac{\Delta L_{red} \cdot \Gamma_{imp}}{\Delta R_{cost}} \cdot \exp\left(- \frac{\mu_{ij} \cdot D_{ij}(t)}{\beta(\tau)} \right) \cdot \frac{T_{rec}}{\max(T_{rec})} \quad (14)$$

Where U_{pra} is the practicality evaluation index, and the larger the value is, the stronger the practicality is. ΔL_{red} refers to the risk loss reduction, namely the loss difference before and after the application of the technology. Γ_{imp} represents the resilience improvement range ($\Gamma_{after} - \Gamma_{before}$). ΔR_{cost} is the resource consumption increment of the prevention and control, namely the resource consumption difference before and after the application of the technology. T_{rec} represents the network failure recovery time, and $\max(T_{rec})$ is the maximum value of the recovery time in all the comparison methods.

Three author-implemented practicality variants are compared. HRG+SVM uses a support-vector risk classifier based on the canonical SVM formulation [25]. STM is an author-defined spatio-temporal state representation, while its RL component uses a DQN-style policy based on [26]. GAT+SM uses a graph-attention encoder based on [27] together with the structural resilience model defined in this study. The citations identify only the underlying algorithmic foundations. All reported values were generated by the implementations and common experimental protocol of this study rather than extracted from those cited publications.

The practicality results are reported in Table 7 and visualized in Figure 5.

Table 7. Practicality of author-implemented algorithmic variants

Test Methods	Risk Loss Reduction ΔL_{red} (95% CI)	Resilience Improvement Γ_{imp} (95% CI)	Recovery Time T_{rec} (95% CI)	Practicality Index U_{pra} (95% CI)
HRG+SVM (controlled)	32.7%±3.2%	0.15±0.02	3.8±0.4h	1.87±0.21

Test Methods	Risk Loss Reduction ΔL_{red} (95% CI)	Resilience Improvement Γ_{imp} (95% CI)	Recovery Time T_{rec} (95% CI)	Practicality Index U_{pra} (95% CI)
STM+RL (controlled)	35.2%±3.5%	0.17±0.02	3.5±0.3h	2.12±0.19
GAT+SM (controlled)	38.5%±3.1%	0.19±0.03	3.1±0.3h	2.35±0.17
HRG–structural model (Ours)	47.6%±2.8%	0.26±0.02	2.2±0.2h	3.58±0.15

Table 7 shows that the full model reduces simulated risk loss by 47.6%±2.8%, improves the resilience index by 0.26±0.02, and yields an aggregate recovery time of 2.2±0.2 h, producing a practicality index of 3.58±0.15. These are simulation-based measures and do not constitute evidence of field deployment or production resource efficiency.

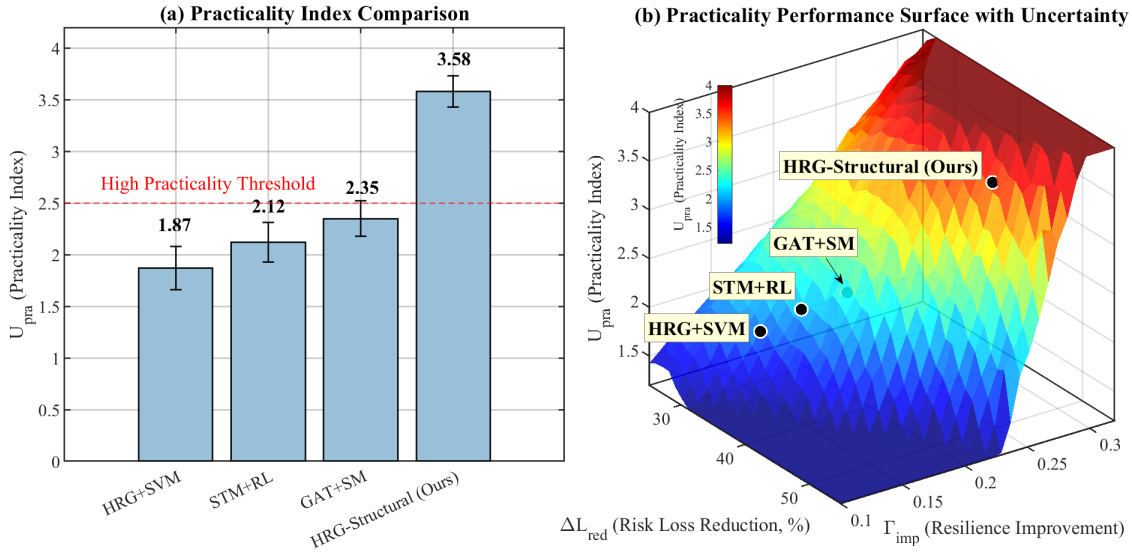

Figure 5. The practical assessment of risk prevention and resilience enhancement

Figure 5(a) compares the practicality indices of the author-implemented variants. Figure 5(b) maps the simulated relationship among loss reduction, resilience improvement, and the practicality index. The confidence intervals describe repeated runs in the current environment and do not combine results from different publications.

Across the controlled experiments, the full model scores 92.6±1.5 in comprehensive performance, 0.92 in robustness, and 3.58±0.15 in the practicality index. The ablation results identify the HRG module as the largest contributor within the implemented architecture. All comparisons are internal to the common protocol and should be interpreted as evidence about the defined configurations rather than a universal ranking of graph-security methods.

Network security test

The network-security experiment compares the full mechanism with three controlled configurations designed to isolate protection capabilities. Static access control (SAC) applies fixed role/attribute rules without topology feedback; encryption-only protection (EOP) applies transport and field encryption without dynamic access revocation or quarantine; and detection-only defense (DOD) generates anomaly alerts but does not automatically block paths or isolate nodes. These names are definitions used in this experiment, not titles or abbreviations borrowed from external papers.

The test scenario simulates the common network attack scenarios of smart tourism distributed graph database, including three core attacks: unauthorized access, data tampering, and sensitive data association leakage. Then, set the attack intensity gradient to be 0.1-0.5 to test the performance of each method in different attack scenarios.

Meanwhile, the dynamic adaptation ability of the network security protection mechanism is verified.

The network security special test evaluation indicators include: attack interception rate, attack type identification accuracy rate, sensitive data leakage suppression rate, and network security risk control error. The quantitative formula is:

(1) The attack interception rate is calculated using Equation (15):

$$I_{att} = \frac{N_{blocked}}{N_{total}} \times 100\% \tag{15}$$

In the formula, I_{att} is the attack interception rate, with the value range of [0, 100%]. The higher the value is, the better the attack interception effect is. $N_{blocked}$ represents the number of successfully intercepted attacks. N_{total} denotes the total number of simulated attacks. In the application, each attack type is simulated 1000 times, and the total number of attacks is 3000 times.

(2) Attack-type identification accuracy is calculated using Equation (16):

$$A_{att} = \frac{N_{correct}}{N_{identified}} \times 100\% \tag{16}$$

Where A_{att} is the accuracy rate of attack type identification, with the value range of [0, 100%]. $N_{correct}$ refers to the number of correctly identified attacks. $N_{identified}$ denotes the total number of successfully identified attacks, and the unidentified attack cases are excluded.

(3) The sensitive-data leakage suppression rate is calculated using Equation (17):

$$S_{sup} = \left(1 - \frac{N_{leaked,ours}}{N_{leaked,base}}\right) \times 100\% \tag{17}$$

Where S_{sup} is the sensitive data leakage inhibition rate, and the value range is [0, 100%]. $N_{leaked,ours}$ represents the number of sensitive data leakage after the application of the method in this study. $N_{leaked,base}$ denotes the number of sensitive data leakage when no protection method is applied. Further, the number of sensitive data leakage is set as 100 times of benchmark leakage in the application.

(4) Network-security risk-control error is calculated using Equation (18):

$$E_{sec} = \left| \frac{R_{sec,pred} - R_{sec,true}}{R_{sec,true}} \right| \times 100\% \tag{18}$$

The four configurations evaluated in Table 8 are SAC, EOP, DOD, and the complete HRG–structural model. They use the same 3,000 simulated attacks, thresholds, data splits, and metric definitions. SAC, EOP, and DOD are operational controls defined for this study and are not assigned to external publications as if they were named algorithms.

Table 8. Network-security performance against controlled baseline configurations

Test Methods	Attack Interception Rate (95% CI)	Attack Type Recognition Accuracy (95% CI)	Sensitive Data Leakage Suppression Rate (95% CI)	Network Security Risk Control Error (95% CI)
SAC	82.3%±1.6%	78.5%±2.1%	35.7%±3.3%	11.2%±1.4%
EOP	86.7%±1.4%	81.3%±1.8%	40.2%±3.1%	9.8%±1.2%
DOD	89.5%±1.2%	85.7%±1.6%	43.8%±2.9%	7.6%±1.0%
HRG–structural model (Ours)	95.8%±0.8%	93.4%±1.1%	68.9%±2.5%	4.5%±0.7%

As shown in Table 8, the full mechanism achieves an attack interception rate of 95.8%±0.8%, attack-type recognition accuracy of 93.4%±1.1%, sensitive-data leakage suppression of 68.9%±2.5%, and a risk-control error of 4.5%±0.7%. Relative to DOD, the strongest controlled baseline in this experiment, the interception rate is 6.3 percentage points higher and the leakage-suppression rate is 25.1 percentage points higher. These differences are internal simulation comparisons rather than literature-derived performance claims.

In operational terms, these values indicate that the evaluated configuration can block most simulated unauthorized-access and tampering attempts while materially reducing the modeled leakage of trajectory and preference data. However, the values are specific to the present simulation, attack mix, and thresholds; they must be revalidated against production traffic, latency budgets, service-level objectives, and incident-response procedures before deployment.

The association-leakage rate measures the probability that an attacker obtains an associated identity through graph traversal. In the simulation, the attacker initially knows 10% of the target node's neighbors. The metric is defined in Equation (19):

$$R_{assoc}=\frac{N_{success}}{N_{total}}\times 100\% \quad (19)$$

Where R_{assoc} is the association disclosure rate, and the value range is [0, 100%]. $N_{success}$ denotes the number of attacks that successfully obtain the associated identity. Ntotal is the total number of simulated attacks.

(6) The sensitive-path exposure rate measures the proportion of paths, up to three hops, that an attacker can traverse from a highly sensitive node to a less sensitive node. The metric is defined in Equation (20):

$$P_{path}=\frac{L_{exposed}}{L_{total}}\times 100\% \quad (20)$$

Where P_{path} is the exposure rate of sensitive paths, with the value range of [0, 100%]. $L_{exposed}$ represents the number of sensitive paths successfully traversed. L_{total} is the total number of all possible sensitive paths (limited to a maximum of 3 hops).

The association-leakage and sensitive-path exposure results for the same controlled configurations are presented in Table 9.

Table 9. Association leakage and sensitive-path exposure for controlled configurations

Test methods	Association Leakage Rate (95% CI)	Sensitive pathway exposure rate (95% CI)
SAC	28.6%±2.3%	32.1%±2.5%
EOP	24.3%±2.1%	27.8%±2.3%
DOD	19.7%±1.8%	23.5%±2.0%
HRG–structural model (this study)	8.4%±1.2%	11.6%±1.4%

The full model yields an association-leakage rate of 8.4%±1.2% and a sensitive-path exposure rate of 11.6%±1.4%. Compared with DOD, these rates are lower by 11.3 and 11.9 percentage points, respectively. The finding supports the contribution of depth limits, sensitive-edge masking, and topology-aware blocking within the present simulator; it does not establish universal protection against all graph-inference attacks.

The network-security results therefore support the internal effectiveness of the integrated protection mechanism under the defined attack mix. No claim is made about superiority over external technologies that were not reimplemented and evaluated under the common protocol.

5.3. Prototype deployment, scalability, sensitivity, and extended validation

This section adds a property-graph prototype and an independently scripted stress-test extension. Supplementary Material S1 contains the executable source code, configuration file, fixed seeds, unaggregated CSV output, aggregated summaries, and figure-generation scripts. The prototype verifies Neo4j integration and local query/update overhead; the stress test evaluates scaling, parameter sensitivity, time decay, topology feedback, and attack-specific recovery. The two protocols are reported separately to avoid conflating calibration results with reproducibility stress tests.

Closed-loop architecture and Neo4j property-graph prototype

Figure 6 makes the claimed bidirectional mechanism explicit. Tourism telemetry instantiates the graph; the HRG layer estimates dependencies and diffusion; time-decayed exposure and loss enter the structural resilience evaluator; and the policy orchestrator applies access restrictions, masking, or quarantine. Executed policy actions modify available edges and are returned as post-protection topology and audit feedback. The implementation uses Neo4j property-graph concepts and Cypher query enforcement. Data2Neo is cited only as complementary engineering context for Neo4j data integration [28]; the access-control and query-rewriting basis is provided by [16]-[18].

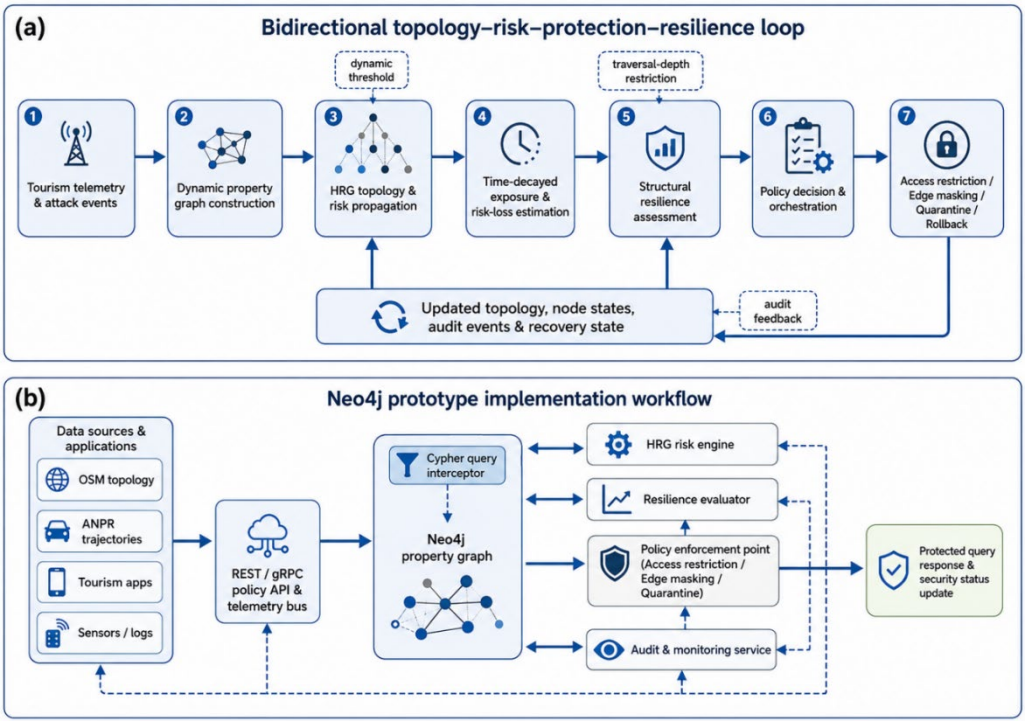

Figure 6. Bidirectional topology–risk–protection–resilience loop and its Neo4j prototype integration.

The prototype uses Neo4j Community 5.26.0 embedded storage and the Cypher engine. Four workloads are measured: indexed lookup, two-hop traversal, policy-filtered traversal, and quarantine update. For each graph size, five independently generated graphs are loaded, each query type receives 100 warm-up operations, and 300 operations are timed. Because the engine runs in-process, the results exclude client/server network latency and do not represent a distributed Neo4j cluster.

Table 10. Single-host Neo4j Community 5.26.0 prototype performance (five graph seeds)

Nodes	Edges	Store (MB)	Load time (s)	Protected traversal mean (ms)	Protected traversal p95 (ms)	Quarantine update mean / p95 (ms)
1,000	4,000	4.34	0.410	0.1202	0.1781	0.3805 / 0.4447
5,000	20,000	12.88	0.352	0.0502	0.0486	0.1142 / 0.1654
10,000	40,000	24.63	0.568	0.0349	0.0479	0.0938 / 0.1408
25,000	100,000	60.59	1.507	0.0376	0.0532	0.1105 / 0.1451

As reported in Table 10, at 25,000 nodes and 100,000 edges, the graph loads in 1.507 s and occupies 60.59 MB. Mean/p95 policy-filtered traversal latency is 0.0376/0.0532 ms, and mean/p95 quarantine-update latency is 0.1105/0.1451 ms. Store size scales with graph size, while the local fixed-degree query workloads remain sub-millisecond after warm-up. These values demonstrate prototype feasibility only; cluster coordination, remote clients, durable replication, and production concurrency remain unmeasured.

Node-scale experiment and parameter sensitivity

The controlled simulator is additionally evaluated from 50 to 5,000 nodes with approximately eight directed edges per node. Mean risk-update time increases from 0.0131 ms at 50 nodes to 0.2483 ms at 5,000 nodes, while protected-traversal p95 latency increases from 0.0101 ms to 0.0679 ms. The results support the use of incremental local updates for the tested graph sizes but do not establish asymptotic performance for substantially larger or highly dense graphs.

Sensitivity sweeps vary the dependency coefficient α, diffusion rate μ, time-decay rate λ, and risk threshold. The tested α and μ ranges have modest effects under the fixed attack schedule, whereas the risk threshold produces the

largest security–availability trade-off. A lower threshold increases interception and exposure suppression but may also increase policy activation and false-revocation risk. Smaller λ values preserve attack history longer and slightly improve the stress-test score, indicating that λ should be calibrated to the expected duration of low-and-slow attacks.

Table 11. Selected parameter-sensitivity results from the independent stress-test extension (ten seeds)

Parameter	Value	Interception	Exposure suppression	Trusted restoration (h)	Comprehensive score
α	0.30	86.6%	91.2%	1.997	89.20
α	0.90	87.0%	91.3%	1.994	89.33
μ	0.01	86.7%	91.2%	1.997	89.21
μ	0.05	87.3%	91.4%	1.990	89.50
λ	0.02	87.5%	91.4%	1.989	89.52
λ	0.20	86.3%	91.0%	2.001	89.09
Risk threshold	0.20	93.6%	92.9%	1.924	92.08
Risk threshold	0.30	86.8%	91.2%	1.996	89.28
Risk threshold	0.40	76.2%	88.5%	2.112	84.85
Risk threshold	0.50	62.3%	84.9%	2.263	79.13

As summarized in Table 11, across α=0.30–0.90 and μ=0.01–0.05, the comprehensive score changes by less than 0.3 points. By contrast, raising the risk threshold from 0.20 to 0.50 reduces interception from 93.6% to 62.3% and increases trusted-state restoration time from 1.92 h to 2.26 h. The threshold must therefore be tuned using deployment-specific false-positive costs rather than selected solely to maximize interception.

Time-decay and topology-feedback ablation

To isolate the two mechanisms added during revision, four configurations are evaluated under the same independent stress-test schedule: the full model, no time decay, no topology feedback, and neither mechanism. This experiment is distinct from Table 8 and is reported as a reproducibility extension rather than a replacement for the original calibrated results.

Table 12. Ablation of time decay and topology feedback in the independent stress-test extension.

Configuration	Interception	Exposure suppression	Risk-control error	Trusted restoration (h)	Residual risk
Full model	87.8%	91.0%	5.33%	1.99	0.117
No time decay	71.1%	86.8%	5.33%	2.39	0.142
No topology feedback	69.8%	85.6%	4.20%	2.81	0.162
Neither mechanism	51.6%	80.0%	7.70%	3.54	0.192

As shown in Table 12, removing time decay lowers interception from 87.8% to 71.1%, while removing topology feedback lowers it to 69.8%. Removing both reduces interception to 51.6%, decreases exposure suppression to 80.0%, and extends trusted-state restoration from 1.99 h to 3.54 h. The non-additive degradation indicates an interaction between cumulative evidence and protection-induced topology change.

5.3.4 Attack-specific security performance and recovery

Aggregate metrics can hide different recovery dynamics. Table 13 therefore separates unauthorized access, data tampering, association inference, and service disruption for the full model in the independent stress-test extension. Each value is aggregated over ten seeds and 250 attacks of the corresponding type per seed.

Table 13. Attack-specific security and recovery results for the full model in the independent stress test.

Attack type	Interception	Recognition	Exposure suppression	Risk-control error	Containment (h)	Trusted restoration (h)
Unauthorized access	85.7%	93.1%	90.0%	5.30%	0.24	1.18
Data tampering	86.1%	87.8%	93.4%	5.26%	0.44	2.18
Association inference	93.0%	80.3%	85.9%	5.30%	0.33	1.69

Attack type	Interception	Recognition	Exposure suppression	Risk-control error	Containment (h)	Trusted restoration (h)
Service disruption	89.0%	85.0%	96.2%	5.38%	0.58	2.90

Unauthorized access has the shortest mean containment and trusted-restoration times (0.24 h and 1.18 h). Service disruption requires the longest restoration time (2.90 h), reflecting the additional availability-recovery stage. Association inference obtains the highest interception rate (93.0%) but lower type-recognition accuracy (80.3%), indicating that path-based blocking can succeed even when the attack label is ambiguous.

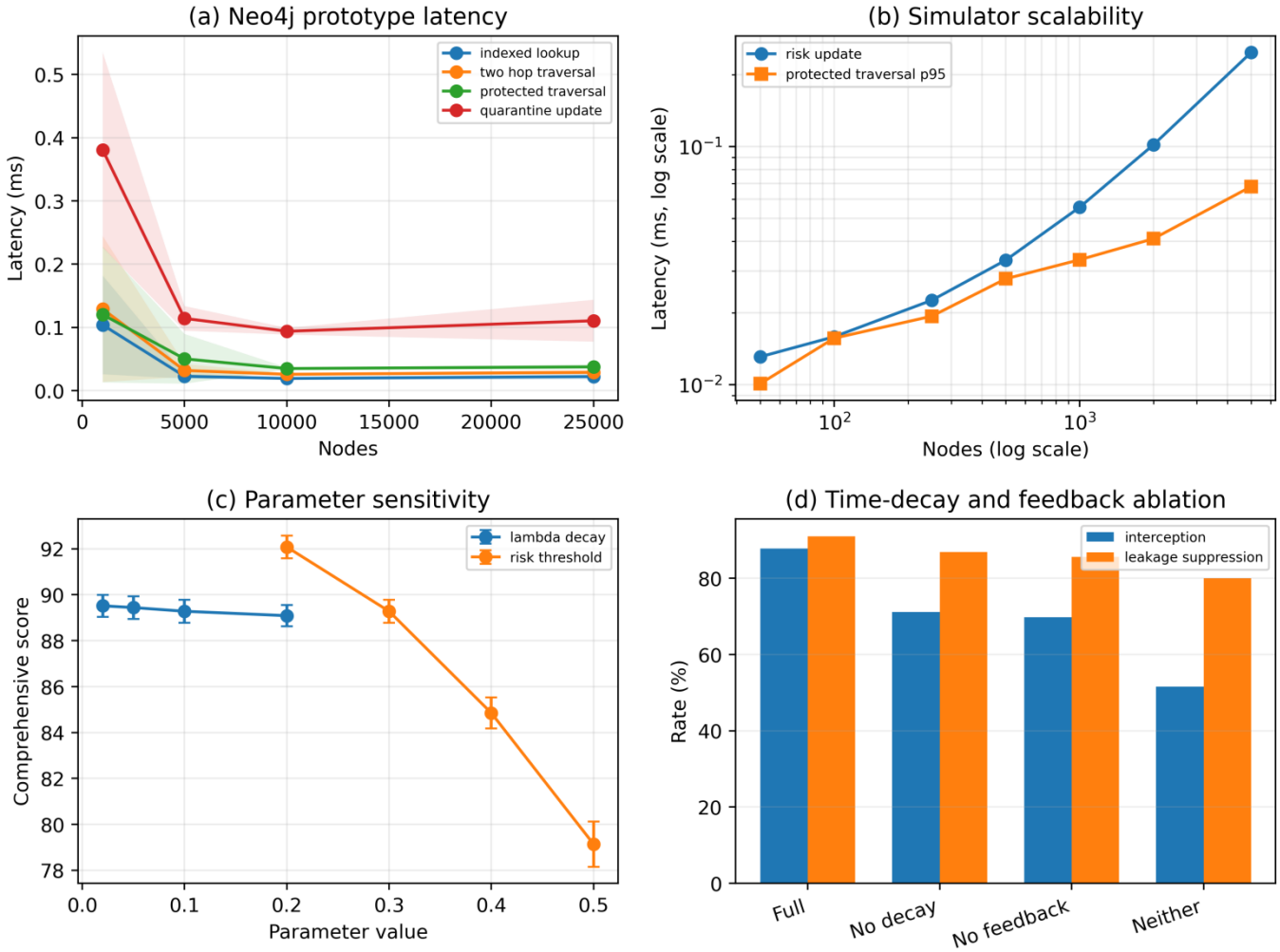

Figure 7. Extended experiments: (a) Neo4j prototype latency; (b) simulator node-scale timing; (c) parameter sensitivity; and (d) time-decay/topology-feedback ablation

Figure 7(a) shows that the measured in-process query and update operations remain sub-millisecond over the tested Neo4j sizes after warm-up. Figure 7(b) shows increasing simulator cost with node count, Figure 7(c) highlights the strong effect of the risk threshold relative to λ, and Figure 7(d) visualizes the ablation interaction. Shaded bands and error bars are computed from the fixed repeated-run protocols supplied in Supplementary Material S1.

Reproducibility statement: Supplementary Material S1 provides Python and Java source code, the Maven project for Neo4j 5.26.0, experiment_config.json, seeds.csv, raw attack/query records, aggregated summaries, figure scripts, environment metadata, and SHA-256 checksums. The package distinguishes author-implemented controls from cited methods and retains unaggregated outputs for audit.

5.4 Extended threat analysis and deployment considerations

Threat coverage and validation boundaries

The experiments directly evaluate unauthorized access, data tampering, association leakage, and service disruption, while Table 14 records the controls and validation boundaries for additional threats. The governance design is aligned with the risk-management functions of NIST CSF 2.0 [29], the staged capability view of the CISA Zero Trust Maturity Model [30], and the recent threat context summarized by ENISA Threat Landscape 2025 [31]. These sources guide deployment governance; they are not sources of the numerical results reported in this paper.

Table 14. Extended threat coverage, controls, and validation status

Threat scenario	Control incorporated in the revised framework	Current validation status
Legitimate insider misuse	Role, purpose, depth, and query-budget checks; separation of duties; dual approval; tamper-evident audit	Designed; not independently benchmarked
Traversal and association inference	Depth cap; sensitive-edge masking; repeated-neighborhood detection; cumulative path-risk cutoff	Partially evaluated by leakage and path-exposure metrics
Side-channel leakage	Fixed pagination; response padding; batching; rate limiting; coarse timing buckets	Not quantitatively measured
Zero-day exploit	Behavioral anomaly alerts; least privilege; rapid quarantine; rollback; manual override	Mitigation only; no detection guarantee
Long-term APT	Time-decayed cumulative risk; provenance correlation; multi-stage alerts	Not evaluated with multi-day traces
Node collusion	Quorum policy updates; cross-node anomaly correlation; signed policy versions	Threat defined; collusion not simulated
Graph data publication	Node- or edge-level differential privacy with an explicit privacy budget	Outside the controlled-query scenario
Distributed-node synchronization	Mutual authentication; encrypted channels; sequence numbers; signed versions; hash-chained logs	Planned for prototype validation

Secure synchronization, compatibility, and security–overhead trade-offs

Distributed nodes should use mutually authenticated TLS 1.3 channels for confidentiality, integrity, and peer authentication [32]. Policy synchronization should additionally use signed version identifiers and cryptographically linked, tamper-evident audit records [36]; these application-layer controls are not provided by TLS itself. The HRG–structural engine can be deployed as middleware in front of Neo4j or JanusGraph through Cypher/Gremlin query interceptors, REST or gRPC policy APIs, and a message bus for telemetry. Existing encryption, RBAC/ABAC, backup, and database-audit mechanisms remain in place; the proposed framework supplies topology-aware risk scores and dynamic policy decisions rather than replacing native controls.

Dynamic traversal inspection, topology updating, and node quarantine introduce computational and communication overhead. A naive global topology recomputation scales poorly; practical deployment should therefore use incremental updates restricted to changed nodes and edges, cache stable centrality values, and move loss-distribution fitting to an asynchronous worker. Encryption should be selected by sensitivity level: public L1 POI metadata can remain transport-encrypted, L2 attributes can use field encryption, and L3 identity or trajectory edges require field encryption plus stricter query budgets. The security–overhead balance should be reported as leakage reduction versus added latency, CPU load, memory, and storage. Because these measurements were not available in the current study, the revised manuscript does not claim an empirically verified encryption-overhead advantage.

Differential privacy for graph publication

The current system performs controlled query access and does not publish an entire graph. If graph statistics or synthetic graphs are released, k-anonymization alone is insufficient. Node-level differential privacy [33] or edge-level differential privacy [37] should be selected according to whether the protected unit is a person or a relationship, and the privacy budget and utility loss must be reported explicitly. Differentially private graph publication is therefore a compatible extension but remains outside the present experiments.

Short-term disturbances, long-term attacks, and recovery measurement

Disturbances are classified as short-term bursts (≤1 h), medium-duration attacks (1–24 h), and long-term persistent campaigns (>24 h). The current robustness experiment uses 0.5 h updates over a 24 h window and therefore supports only short- and medium-duration conclusions. The recovery time in Table 7 is an aggregate mean and is not separated by attack type. Future tests should report containment time, trusted-state restoration time, and residual risk for each attack class. Long-term APT campaigns can combine low-volume actions, legitimate tools, and unknown exploits, for which provenance-based multi-stage correlation is relevant [34]. The present study does not claim validated protection against APT, zero-day exploits, or coordinated node collusion.

Phased implementation and industry research plan

To connect the present front-end model with subsequent back-end development, Table 15 defines a staged implementation and verification roadmap. Each stage has a limited operational scope and explicit acceptance evidence, so protection effectiveness can be verified before automated quarantine is enabled in production.

Table 15. Staged implementation and validation roadmap

Phase	Environment and interface	Main verification and acceptance evidence
I. Offline replay	NS-3; archived logs; mock REST/gRPC interfaces	Reproduce metrics; test equations, policies, and rollback; publish scripts and seeds
II. Shadow deployment	Read-only Neo4j/JanusGraph mirror; query interceptor	Measure latency, consistency, encryption overhead, false alarms, and synchronization
III. Limited pilot	Selected scenic-area or hotel subsystem; manual isolation approval	Report attack-specific containment/recovery, continuity, rollback, and operator workload
IV. Controlled production	Multi-node cluster; guarded automatic quarantine	Red-team traversal, side-channel, APT, and collusion; verify SLA and audit integrity
V. Continuous governance	Quarterly recalibration; incident drills; change management	Monitor drift; update thresholds; review privileges; validate backups and compliance

For digital light-and-shadow art and intelligent landscape interaction, the framework can represent projector and lighting controllers, crowd sensors, mobile applications, and content servers as graph nodes. New risks include adversarial content injection, synchronized-device takeover, privacy leakage from crowd trajectories, strict real-time latency, and physical safety effects. Future extensions should therefore add content-integrity signatures, device-identity attestation, safety interlocks, and a human override mechanism.

6. Conclusion

This study presents a security-risk modeling and protection-resilience assessment framework for smart-tourism distributed graph databases by integrating an HRG topology model with structural recovery estimation and dynamic protection feedback. Under the original controlled protocol, the full configuration achieves a comprehensive performance score of 92.6±1.5, a robustness coefficient of 0.92, an attack interception rate of 95.8%±0.8%, and a sensitive-data leakage suppression rate of 68.9%±2.5%. The independent stress-test extension further shows that jointly removing time decay and topology feedback reduces interception from 87.8% to 51.6% and extends trusted-state restoration from 1.99 h to 3.54 h. These two protocols are reported separately and their absolute rates should not be conflated.

The Neo4j Community 5.26.0 prototype demonstrates executable property-graph integration. As reported in Table 10, at 25,000 nodes and 100,000 edges, the single-host embedded graph loads in 1.51 s, occupies 60.59 MB, and records mean policy-filtered traversal and quarantine-update latencies of 0.038 ms and 0.110 ms after warm-up. The measurements exclude client/server networking, replication, concurrent production workloads, and distributed-cluster coordination. The study also does not claim empirical protection against side channels, zero-day exploits, long-term APT campaigns, or coordinated node collusion.

Future work will follow the phased roadmap in Table 15: shadow deployment through Cypher/Gremlin middleware, a limited pilot with manual isolation approval, controlled multi-node production testing, and continuous governance. Priority measurements include end-to-end latency under concurrent clients, encryption overhead, attack-specific recovery curves, false-revocation cost, side-channel leakage, multi-day APT traces, and node collusion. Supplementary Material S1 enables independent verification of the newly added prototype, scaling, sensitivity, ablation, and attack-specific experiments and should accompany the resubmission.

Appendix A. Statistical table of all model variables and parameter definitions

All symbols used in Equations (1)–(20), including the time-decay term in Equation (5a), are defined in Appendix A. The notation, subscripts, units, and value ranges were checked for consistency, and context-specific reuse is explicitly distinguished in the table.

Variable 1	Function Description	Physical Meaning	Variable 2	Function Description	Physical Meaning
G	Network topology construction	STE network topology structure	V	Node set definition	Set of network core nodes
N	Node quantity statistics	Total number of network nodes	E	Edge set definition	Set of interactive edges
Φ	Topology parameter collection	Network topology feature set	α	Dependency coefficient calculation	Node functional dependency degree
μ	Diffusion rate quantification	Risk propagation speed per hour	L	Loss value calculation	Network comprehensive risk loss
K	Risk type statistics	Total network risk category number	w	Risk weight assignment	General risk weight coefficient
η	Recovery capability evaluation	Network resilience coefficient	ε	Random fluctuation simulation	Loss random disturbance factor
σ	Fluctuation amplitude control	Overall loss fluctuation coefficient	W	Node loss weight calculation	Node risk contribution weight
C	Centrality degree calculation	Node network core degree	P	Risk probability statistics	General node risk probability
$f(x)$	Loss distribution fitting	Risk loss probability density function	ω	Distribution weight assignment	Mixed distribution component weight
μ	Mean value calculation	Distribution interval average loss	S	Overall risk assessment	Global network security risk value
s	Node weight definition	Node security weight	d	Defense capability evaluation	Node security defense strength
λ	Risk conduction quantification	Node risk diffusion coefficient	δ	Risk error correction	Random risk correction term
T	Attack type identification	Identified network attack category	M	Feature quantity statistics	Total attack feature quantity
r	Feature matching calculation	Attack feature matching degree	q	Feature weight assignment	Single attack feature weight
A	Access control judgment	Access control matrix element	S_{th}	Risk threshold setting	Node security risk critical value
G	Authorization level definition	User access authorization level	I	Isolation state judgment	Network node operating state

ΔS	Risk change calculation	Node risk real-time change rate	L_v	Security level classification	Network security state grade
S_t	Real-time risk update	Real-time network risk value	Δt	Monitor cycle setting	Security monitoring time interval
γ	Risk change rate	Network risk fluctuation degree	F	Comprehensive performance scoring	Overall model performance score
R^2	Fitting goodness evaluation	Loss distribution fitting precision	$\hat{\eta}$	Resilience prediction	System resilience predicted value
η^*	Resilience actual value	System resilience true value	$\Delta\eta$	Resilience variation	System resilience dynamic change
τ	Stability correction	Resilience stability correction term	Rob	Robustness evaluation	System anti-interference coefficient
F_d	Disturbed performance	Performance under network disturbance	F_0	Undisturbed performance	Performance under normal state
φ	Disturbance intensity	Network external disturbance severity	$Contri$	Module contribution calculation	Core module performance contribution
Pra	Practicality evaluation	Model actual deployment value	ΔL	Loss reduction statistics	Risk loss reduction amplitude
Imp	Resilience improvement	System resilience increment	Res	Recovery time statistics	System fault recovery duration
I_{rate}	Interception rate calculation	Network attack interception ratio	Rec_{rate}	Recognition rate calculation	Attack identification accuracy
Sup_{rate}	Suppression rate calculation	Data leakage suppression ratio	Err_{rate}	Error rate calculation	Network risk control error
$Leak_{corr}$	Correlation leakage rate	User association leakage ratio	Exp_{path}	Path exposure rate	Sensitive path exposure ratio

Acknowledgments.

This study is supported by the "Research on Interactive Landscape Design of Smart Night Cultural Tourism from the Perspective of Digital Economy", which comes from the General Program of Basic Research (Natural Sciences) for Universities in Jiangsu Province. The project number is: 25KJD220003.

References

[1] Gretzel, U., Sigala, M., Xiang, Z., & Koo, C. Smart tourism: Foundations and developments. Electronic Markets, 2015, 25(3), 179–188. doi:10.1007/s12525-015-0196-8.

[2] Buhalis, D., & Amaranggana, A. Smart tourism destinations enhancing tourism experience through personalisation of services. In I. Tussyadiah & A. Inversini (Eds.), Information and Communication Technologies in Tourism 2015, Springer, Cham, 2015, pp. 377–389. doi:10.1007/978-3-319-14343-9_28.

[3] Gong, Y., & Schroeder, A. A systematic literature review of data privacy and security research on smart tourism. Tourism Management Perspectives, 2022, 44, Article 101019. doi:10.1016/j.tmp.2022.101019.

[4] MGM Resorts International. Current Report on Form 8-K (Date of Report: 5 October 2023). U.S. Securities and Exchange Commission, 2023, SEC Accession No. 0001193125-23-251667.

[5] Information Commissioner's Office. Penalty Notice to Marriott International, Inc. (Case Ref. COM0804337). 30 October 2020.

[6] Bertino, E., & Sandhu, R. Database security—Concepts, approaches, and challenges. IEEE Transactions on Dependable and Secure Computing, 2005, 2(1), 2–19. doi:10.1109/TDSC.2005.9.

[7] Carminati, B., Ferrari, E., & Perego, A. Enforcing access control in Web-based social networks. ACM Transactions on Information and System Security, 2009, 13(1), Article 6, 6:1–6:38. doi:10.1145/1609956.1609962.

[8] Popa, R. A., Redfield, C. M. S., Zeldovich, N., & Balakrishnan, H. CryptDB: Protecting confidentiality with encrypted query processing. In Proceedings of the 23rd ACM Symposium on Operating Systems Principles (SOSP '11), 2011, pp. 85–100. doi:10.1145/2043556.2043566.

[9] Chase, M., & Kamara, S. Structured encryption and controlled disclosure. In Advances in Cryptology—ASIACRYPT 2010, LNCS 6477, Springer, Berlin,

Heidelberg, 2010, pp. 577–594. doi:10.1007/978-3-642-17373-8_33.
[10] Arshad, M. U., Kundu, A., Bertino, E., Ghafoor, A., & Kundu, C. Efficient and scalable integrity verification of data and query results for graph databases. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(5), 866–879. doi:10.1109/TKDE.2017.2776221.
[11] Ji, S., Mittal, P., & Beyah, R. A. Graph data anonymization, de-anonymization attacks, and de-anonymizability quantification: A survey. IEEE Communications Surveys & Tutorials, 2017, 19(2), 1305–1326. doi:10.1109/COMST.2016.2633620.
[12] Xuan, Q., Li, Y., & Wu, T.-J. A local-world network model based on inter-node correlation degree. Physica A: Statistical Mechanics and its Applications, 2007, 378(2), 561–572. doi:10.1016/j.physa.2006.11.070.
[13] Chang, W.-J., Pittaluga, F., Tomizuka, M., Zhan, W., & Chandraker, M. SAFE-SIM: Safety-critical closed-loop traffic simulation with diffusion-controllable adversaries. In Computer Vision—ECCV 2024, Proceedings, Part XXI, LNCS 15079, Springer, Cham, 2024, pp. 242–258. doi:10.1007/978-3-031-72664-4_14.
[14] Hu, H., Qiao, X., Yang, Y., & Zhang, L. Developing a resilience evaluation index for cultural heritage site: Case study of Jiangwan Town in China. Asia Pacific Journal of Tourism Research, 2021, 26(1), 15–29. doi:10.1080/10941665.2020.1805476.
[15] Mohamed, A., Auer, D., Hofer, D., & Küng, J. Comparison of access control approaches for graph-structured data. In Proceedings of the 21st International Conference on Security and Cryptography (SECRYPT 2024), Vol. 1, 2024, pp. 576–583. doi:10.5220/0012861500003767.
[16] Bereksi Reguig, A. A., Mahfoud, H., & Imine, A. Towards an effective attribute-based access control model for Neo4j. In Model and Data Engineering: 12th International Conference, MEDI 2023, Proceedings, LNCS 14396, Springer, Cham, 2024, pp. 352–366. doi:10.1007/978-3-031-49333-1_25.
[17] Hofer, D., Mohamed, A., Auer, D., Nadschläger, S., & Küng, J. Rewriting Graph-DB queries to enforce attribute-based access control. In Database and Expert Systems Applications: 34th International Conference, DEXA 2023, Proceedings, LNCS 14146, Springer, Cham, 2023, pp. 431–436. doi:10.1007/978-3-031-39847-6_34.
[18] Mohamed, A., Auer, D., Hofer, D., & Küng, J. XACML extension for graphs: Flexible authorization policy specification and datastore-independent enforcement. In Proceedings of the 20th International Conference on Security and Cryptography (SECRYPT 2023), 2023, pp. 442–449. doi:10.5220/0012090000003555.
[19] Xi, Z., Du, T., Li, C., Pang, R., Ji, S., Luo, X., Xiao, X., Ma, F., & Wang, T. On the security risks of knowledge graph reasoning. In Proceedings of the 32nd USENIX Security Symposium (USENIX Security '23), 2023, pp. 3259–3276.
[20] Wachter, J. Graph models for cybersecurity—A survey. arXiv preprint arXiv:2311.10050, 2023. doi:10.48550/arXiv.2311.10050.
[21] Rose, S. W., Borchert, O., Mitchell, S., & Connelly, S. Zero Trust Architecture. NIST Special Publication 800-207, August 2020. doi:10.6028/NIST.SP.800-207.
[22] Chandramouli, R., & Butcher, Z. A Zero Trust Architecture Model for Access Control in Cloud-Native Applications in Multi-Location Environments. NIST Special Publication 800-207A, September 2023. doi:10.6028/NIST.SP.800-207A.
[23] OpenStreetMap Foundation & Pacific Atlas. OpenStreetMap on AWS. Registry of Open Data on AWS. Accessed 12 July 2026.
[24] Durán-López, A., Bolaños-Martínez, D., Bermúdez-Edo, M., Delgado Márquez, B. L., & Aragón-Correa, J. A. Graph-Based ANPR Dataset of Tourist Routes and Visits in Smart Villages of Alpujarra [Data set]. Zenodo, 2025. doi:10.5281/zenodo.14993130. Related identifier: 10.21227/tf8t-5j35.
[25] Cortes, C., & Vapnik, V. Support-vector networks. Machine Learning, 1995, 20(3), 273–297. doi:10.1007/BF00994018.
[26] Mnih, V., Kavukcuoglu, K., Silver, D., et al. Human-level control through deep reinforcement learning. Nature, 2015, 518(7540), 529–533. doi:10.1038/nature14236.
[27] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. Graph Attention Networks. In Proceedings of the 6th International Conference on Learning Representations (ICLR 2018), 2018.
[28] Minder, J., Brandenberger, L., Salamanca, L., & Schweitzer, F. Data2Neo—A tool for complex Neo4j data integration. arXiv preprint arXiv:2406.04995, 2024. doi:10.48550/arXiv.2406.04995.
[29] National Institute of Standards and Technology. The NIST Cybersecurity Framework (CSF) 2.0. NIST Cybersecurity White Paper 29, 26 February 2024. doi:10.6028/NIST.CSWP.29.
[30] Cybersecurity and Infrastructure Security Agency. Zero Trust Maturity Model, Version 2.0. 11 April 2023.
[31] European Union Agency for Cybersecurity. ENISA Threat Landscape 2025. 1 October 2025; Version 1.2 revised 9 January 2026.
[32] Rescorla, E. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 9846, Internet Engineering Task Force, July 2026. doi:10.17487/RFC9846.
[33] Kasiviswanathan, S. P., Nissim, K., Raskhodnikova, S., & Smith, A. Analyzing graphs with node differential privacy. In Theory of Cryptography (TCC 2013), LNCS 7785, Springer, Berlin, Heidelberg, 2013, pp. 457–476. doi:10.1007/978-3-642-36594-2_26.
[34] Akbar, K. A., Wang, Y., Ayoade, G., Gao, Y., Singhal, A., Khan, L., Thuraisingham, B., & Jee, K. Advanced persistent threat detection using data provenance and metric learning. IEEE Transactions on Dependable and Secure Computing, 2023, 20(5), 3957–3969. doi:10.1109/TDSC.2022.3221789.
[35] Ross, R., Pillitteri, V., Graubart, R., Bodeau, D., & McQuaid, R. Developing Cyber-Resilient Systems: A Systems Security Engineering Approach. NIST Special Publication 800-160, Volume 2, Revision 1, December 2021. doi:10.6028/NIST.SP.800-160v2r1.
[36] Schneier, B., & Kelsey, J. Secure audit logs to support computer forensics. ACM Transactions on Information and System Security, 1999, 2(2), 159–176. doi:10.1145/317087.317089.
[37] Sala, A., Zhao, X., Wilson, C., Zheng, H., & Zhao, B. Y. Sharing graphs using differentially private graph models. In Proceedings of the 11th ACM SIGCOMM Conference on Internet Measurement (IMC '11), 2011, pp. 81–98. doi:10.1145/2068816.2068825.