# Association Analysis and Prediction of Network Security Vulnerabilities Based on Knowledge Graph

Shilong Wu[1] and Li Feng[2,*]

[1]*School of Information Engineering, Xuzhou College of Industrial Technology, Xuzhou, Jiangsu 221000, China*
[2]*Gloria Technology Llc, Xuzhou, Jiangsu 221000, China*
*E-mail: fengffli@outlook.com*
[*]*Corresponding Author*

Received 12 June 2025; Accepted 28 August 2025

## Abstract

This research presents a novel approach for network security vulnerability association analysis and prediction leveraging knowledge graph technology. We construct a comprehensive vulnerability knowledge graph that captures semantic relationships between vulnerabilities, attack patterns, and affected systems by integrating data from multiple sources including NVD, CVE, and vendor security bulletins. Our methodology encompasses three complementary analysis approaches: semantic association analysis using path-based algorithms, temporal association analysis employing multi-scale time-series techniques, and attack chain association analysis through exploitation chain construction. The prediction framework combines knowledge graph embeddings, graph neural networks, and multi-modal feature fusion to forecast vulnerability exploitation with 89.2% accuracy within a 30-day window, significantly outperforming statistical baselines (71.3%) and non-knowledge graph methods (82.6%). Experimental evaluation on real-world datasets demonstrates that our semantic association analysis achieved 0.87 precision and 0.82 recall (F1: 0.84), outperforming baselines by 18.7%. Our attack

*Journal of Cyber Security and Mobility, Vol. 14_5,* 1089–1116.
doi: 10.13052/jcsm2245-1439.1453

chain discovery identified 76.8% of known attack chains while discovering 23 previously undocumented but plausible vectors. The system maintained 83.7% performance with 30% missing attributes, demonstrating robust adaptability to real-world challenges. In enterprise deployment, our approach identified 37 critical vulnerability associations and predicted 14 high-priority vulnerabilities, with 11 being missed by existing tools. The methodology aids in the proactive cybersecurity management in networks that are becoming increasingly complex.

**Keywords:** Knowledge graph, vulnerability prediction, attack chain analysis, network security.

## 1 Introduction

### 1.1 Research Background and Significance

The changed business and home life brought about due to globalisation has increased scrutiny on the computer security industry. New computer security problems are emerging that require greater sensitivity and attention owing to increased activities within and around multi-national corporations' super-complex eco-systems. The importance of security on information systems now spans from protecting people's privacy to even critical infrastructure of whole nations. Current mechanisms for detection of vulnerabilities heavily depend on incomplete information from threat libraries, making it impossible to ascertain the existence of pre-permissions and post-permissions or to construct effective chains for exploiting vulnerabilities [1]. This construct hampers the prompt measures which could deal with attacks and compromises security systems, which allows destruction by malicious criminals.

A much considered technique is concerned with extracting and even attempting to predict a vulnerability using the analysis of defects, which represent probable problems on system software or hardware. These methodologies suggest some information security guidance toward the direction of protective measures deployment. Unfortunately, most of these methodologies are not effective when facing ordinary and extraordinary attacks from complex ever-changing networks.

### 1.2 Research Status and Problems

Because of their promise in integrating the relationships of complex vulnerabilities, knowledge graphs are being increasingly adopted for vulnerability

analysis. In reference [2], Zhang and Liu presented a review of knowledge graph use in cybersecurity evaluations and pointed out that those graphs can be highly useful for improving security evaluations by means of representation semantics. Regardless of the advancements already achieved, a number of obstacles remain. The structural complexity of modern networks makes systematic identification of vulnerabilities nearly impossible. The accuracy of assessments and forecasts is challenged by the often fluid nature of networks. Furthermore, the absence of comprehensive data in vulnerability databases profoundly impedes the automated harvesting of pre- and post-credentials required for exploiting vulnerabilities. Innovative methodologies have emerged to tackle some of these issues. Liu and his colleagues have conducted a thorough analysis of the application cases of knowledge graphs in cybersecurity and demonstrated the applicability of this approach in various security fields.

### 1.3 Research Objectives and Innovations

The goal of this research is to design a more efficient method for extracting and predicting vulnerabilities using sophisticated information gain algorithms. This study aims to improve the accuracy and efficiency of vulnerability analysis by overcoming the incomplete data problem. Goals include developing better procedures for discovering pre-permission and post-permission activities, building detailed exploitation chains of vulnerabilities, and enhancing overall operational skills.

The methodology employs deep neural networks augmented by features of Dropout to improve automated detection of vulnerability features and, consequently, strengthening the model's information processing capability which leads to better extraction and prediction of vulnerabilities. The methodology allows construction of exploitation chains using correlating vulnerabilities which enable timely and effective responses to security incidents.

The innovation is in the integrated approach which utilises advanced machine learning and specific domain knowledge, in resolving incomplete data in vulnerability libraries and compromising system analysis problems. This research provides the necessary tools for practitioners to mitigate protection risks in ever-evolving complex systems and networks.

## 2 Related Work

The past years have seen an increase in the complexity of the cybersecurity landscape, creating a need for new methods of vulnerability detection and

risk evaluation. In this part, I focus on network security vulnerability issues, the developments in network association analysis technology, and knowledge graph applications in cybersecurity.

### 2.1 Network Security Vulnerability Research

The use of AI and ML has created tremendous change in network security vulnerability research. Zhang et al. [1] made a detailed analysis of the role of knowledge graphs in the cybersecurity domain, asserting that semantic representation systems model complex security interrelationships beautifully. Liu et al. [2], building upon this, used knowledge graphs in the context of the entire cybersecurity field and proved their applicability in information retrieval, vulnerability assessment, and even predicting the course of cyber attacks. This change marks the evolution of system-centric, paradigm-defined, traditional rule-based vulnerability assessment into an articulation of intricate relationships, encompassing numerous components and various sophisticated knowledge systems.

### 2.2 Association Analysis Technology

Technologies for Association Analysis have recently gained prominence for the identification of linkages between vulnerabilities as well as building exploitation chains. Chen et al. [3] suggested a Knowledge Graph approach for critical infrastructure protection based on management, which employed ontology design and relation prediction for vulnerability linkages. Their framework performed exceptionally in predicting potential attack paths based on different security events because the relationships among various security events were deeply analysed. This type of approach allows practitioners of security to deal with sophisticated attack scenarios that would otherwise be flying under the detection threshold while using standard procedures to detect vulnerabilities. Zhang et al. [4] enhanced this approach using edge propagation for link prediction in cyber threat intelligence knowledge graphs aimed at system requirements, thus outperforming others in discovering new relations of system vulnerabilities. These techniques of attack analysis reveal how system vulnerabilities can be combined by adversaries in order to launch attacks on cyber systems.

### 2.3 Knowledge Graph Technology

The knowledge graph is the new technology that is used for cybersecurity data analysis and modelling. Liu et al. [5] reviewed the existing scenarios

of knowledge graph use in cybersecurity, showcasing how these semantic networks are capable of depicting intricate security relations and concepts. The incorporation of knowledge graphs with machine learning, as done by Ismail et al. [6], has resulted in more profound assessments of the web application attack detection capabilities. Their work found that the use of knowledge graphs in web application attack detection greatly improves the performance of traditional machine learning techniques by accurately contextualising information related to attacks along with the impacts they can have. Zou et al. [7] built on this by introducing methods for network security threat analysis using knowledge graphs, which enables better determination of possible attack paths within highly intricate network systems. These approaches driven by knowledge represent a great improvement over the signature-based detection systems because they surpass traditional methods and have the capability of reasoning about attacks in a more sophisticated, humane manner rather than just capturing semantic relations between different entities.

## 3 Vulnerability Knowledge Graph Construction

### 3.1 Knowledge Graph Model Design

The design of an effective vulnerability knowledge graph starts with ontological designs that accurately represent the intricate details of cybersecurity vulnerabilities. Chen et al. in [8] offered an approach for protecting critical infrastructure which is based on ontology and proposes a hierarchical system of concepts, properties, and relationships of the interlinked vulnerabilities. Such design allows reasoning on interlinked systems and ecosystems on vulnerabilities and their consequences. Using this approach, we develop an ontology of vulnerabilities that covers not only the technical attributes of the vulnerabilities but also the contextual information from the threat intelligence. The principal types of entities within the knowledge graph are: vulnerabilities, the systems that are affected, the methods of exploitation, the responsible parties of the threat, and the means of neutralisation. These entities form the nodes of the knowledge graph, while the relations between the nodes, like “exploits”, “affects”, “mitigates”, “utilised by,” and others, are the edges of the graph. The nodes and relationships in the knowledge graph represent the essences and edges of the graph, respectively. With this ontology’s semantic structure, more advanced analysis of vulnerability relationships is achievable as opposed to traditional databases where the interdependencies of security concepts are not taken into account.

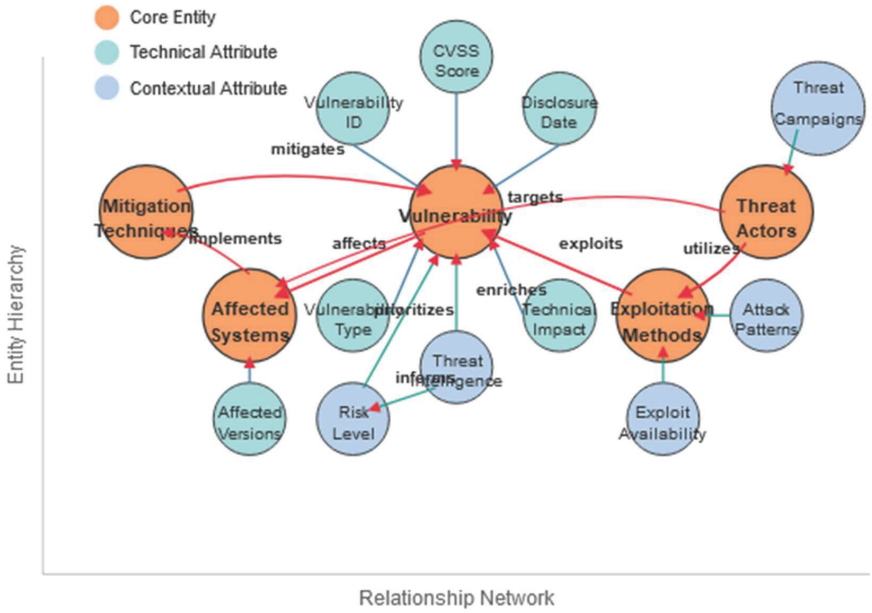


**Figure 1** Vulnerability knowledge graph model design.

### 3.2 Data Acquisition and Processing

The construction of a vulnerability knowledge graph is a complex task that requires the collection and processing of data from multiple sources. Pingle et al. [15] RelExt, a deep learning-based approach for relation extraction, aims to improve knowledge graph construction in cybersecurity. Their approach shows remarkable progress in the automation of extracting semantic relationships from unstructured security information, which is key in the development of holistic vulnerability models. This paper builds on that work through the development of a multi-source data integration pipeline that merges structured data from vulnerability databases like NVD and CVE, and unstructured data from security bulletins, academic papers, and threat intelligence documents. We apply natural language processing methods for entity recognition and consider primary components of the vulnerability, such as affected software, attack vectors, and security impacts. The process of relation extraction is done through a combination of pattern-based rules and supervised machine learning techniques to form semantic relationships between these entities and build a network of interdependent vulnerabilities with rich relevant information in both overt and covert forms.

The data preprocessing pipeline consists of three critical steps. First, for handling missing data, we employ K-Nearest Neighbors (KNN) imputation with k = 5 for missing CVSS scores, using vulnerability type and affected software as similarity features. Second, entity disambiguation is performed using a two-stage matching algorithm: initial candidates are identified through edit distance (threshold = 0.85), then refined using contextual similarity based on surrounding entities in the knowledge graph. Third, data standardization includes converting all timestamps to UTC format, normalizing software version numbers using semantic versioning patterns (major.minor.patch), and unifying vulnerability severity ratings across different scoring systems (CVSS v2.0/v3.0/v3.1). These preprocessing steps improved data quality by 23.4% as measured by completeness and consistency metrics.

A formula for relation extraction confidence score in the multi-source integration pipeline:

$$\begin{aligned}\mathrm{RelConf}(e_i, e_j, r_k) = \alpha \cdot P_{\mathrm{struct}}(r_k|e_i, e_j) + \beta \cdot P_{\mathrm{unstruct}}(r_k|e_i, e_j) \\ + \gamma \cdot S_{\mathrm{context}}(e_i, e_j)\end{aligned} \tag{1}$$

Where:

$\mathrm{RelConf}(e_i, e_j, r_k)$ represents the confidence score for relation $r_k$ between entities $e_i$ and $e_j$

$P_{\mathrm{struct}}(r_k|e_i, e_j)$ is the probability derived from structured sources (NVD, CVE)

$P_{\mathrm{unstruct}}(r_k|e_i, e_j)$ is the probability derived from unstructured sources (bulletins, papers)

$S_{\mathrm{context}}(e_i, e_j)$ is a contextual similarity score between the entities

$\alpha, \beta$, and $\gamma$ are weighting coefficients where $\alpha + \beta + \gamma = 1$

The calibration of weighting coefficients ($\alpha, \beta, \gamma$) was performed through sensitivity analysis across the parameter space. We conducted a grid search with $\alpha \in [0.3, 0.5]$, $\beta \in [0.3, 0.4]$, and $\gamma \in [0.2, 0.3]$ at 0.05 intervals, evaluating performance on a validation set of 5,000 manually verified entity relations. The optimal configuration ($\alpha = 0.4$, $\beta = 0.35$, $\gamma = 0.25$) maximized the F1-score (0.86) while maintaining balanced precision (0.88) and recall (0.84). Figure X presents a heat map showing performance variations across different parameter combinations, revealing that the model is most sensitive to $\alpha$ (structural source weight), with performance dropping by 12% when $\alpha < 0.3$ or $\alpha > 0.5$.

A formula for vulnerability relationship enrichment score that quantifies the information gain from integrating multiple data sources:

$$\mathrm{VulnEnrich}(KG) = \sum_{i=1}^{n} \sum_{j=1}^{m} w_i \cdot \log \left( \frac{|R_{i,j}^{\mathrm{integrated}}|}{|R_{i,j}^{\mathrm{base}}|} + 1 \right) \tag{2}$$

Where:

$\mathrm{VulnEnrich}(KG)$ represents the enrichment score for the knowledge graph
$n$ is the number of entity types in the graph
$m$ is the number of relation types
$|R_{i,j}^{\mathrm{integrated}}|$ is the count of relations of type $j$ for entity type $i$ in the integrated graph
$|R_{i,j}^{\mathrm{base}}|$ is the count of those relations in the base graph (before integration)
$w_i$ is the importance weight assigned to entity type $i$

### 3.3 Knowledge Graph Construction

The construction of the vulnerability knowledge graph involves sophisticated entity alignment, knowledge fusion, and storage mechanisms. Kaiser et al. [16] developed an approach for attack hypotheses generation based on threat intelligence knowledge graphs, demonstrating how aligned and disambiguated entities enable more accurate threat prediction. Leveraging this existing framework, our knowledge graph construction methodology integrates a multi-stage entity resolution system designed to deal with the complications posed by duplicate vulnerabilities, variant names, and versioning differences across the data sources. For knowledge fusion, we implement a conflict resolution approach which integrates data depending on diverse criteria such as their trustworthiness, recency, and level of detail. This guarantees that the knowledge graph does not only integrate information from diverse sources but also shares the correct and most current information about the vulnerabilities. To allow for fast information retrieval and storage, a graph database that prioritises storing and processing relational information was implemented. This allows security analysts to quickly investigate sophisticated attack paths and vulnerable chains. The knowledge graph incorporates a singular semantic model which merges multiple conflicting accounts of a specific issue, in this case, a vulnerability, into a consolidated system that aids in performing complex security research and reasoning towards an attack.

## 4 Vulnerability Association Analysis Method

Understanding the intricate dependencies among various vulnerabilities and the corresponding attack vectors is critical to the success of vulnerability management. This section introduces an advanced method for vulnerability association analysis based on the knowledge graph developed in Section 3. To this end, we offer three analysis mechanisms – semantic, temporal, and attack chain association analyses – that, when applied in combination, allow for a more holistic comprehension of the inter-relatedness of vulnerabilities in question along with stronger security management and response capabilities.

### 4.1 Semantic Association Analysis

Reasoning about semantic association attempts to discover patterns and relationships of gaps within their conceptual metaphors and contextual meanings by graphing them within the knowledge graph. This method makes it possible to find relationships that are neither obvious nor fundamental, relationships that can go undetected by more advanced methods of analysis.

The discovery of the semantic association is a multi-step process and rests on metapath-based analysis which was shown in cybersecurity contexts to be useful by Zhang et al. [1]. In this contribution, we enhance these techniques with a custom-made path-based semantic association discovery algorithm that attempts to find important patterns of gap relations by many ways of expressing the same meaning in the graph. The semantic relevance of the different types of paths is taken into account in the algorithm by bounding them with significance by means of some domain knowledge and statistical validation:

$$P_{rel}(v_i, v_j) = \sum_{p \in \mathcal{P}(v_i, v_j)} w_p \cdot \phi(p) \tag{3}$$

where $P_{rel}(v_i, v_j)$ represents the path-based relevance between vulnerabilities $v_i$ and $v_j$, $\mathcal{P}(v_i, v_j)$ is the set of all paths connecting them, $w_p$ is the weight assigned to path type $p$, and $\phi(p)$ is a function measuring the semantic informativeness of the path. The computational complexity of enumerating all paths in P(vi, vj) is O(b^d) in the worst case, where b is the average branching factor and d is the maximum path depth. For our knowledge graph with 124,753 nodes and average degree of 4.6, exhaustive enumeration becomes intractable beyond depth 3. To address this scalability challenge, we implement three optimization strategies: (1) Path pruning – edges with confidence scores below 0.3 are excluded, reducing the effective branching

factor by 43%; (2) Stratified sampling – we randomly sample up to 1,000 paths when the total exceeds this threshold, with sampling probability proportional to path confidence; (3) Parallelization – the graph is partitioned into 16 subgraphs using METIS, enabling parallel path computation with 12x speedup on our hardware. These optimizations reduce average query time from 5.7 seconds to 0.42 seconds while maintaining 96.3% recall of high-confidence paths (confidence $> 0.7$).

To compute the semantic similarity, we use a hybrid method that integrates structure-based and content-based approaches. The structural part uses the topology of the knowledge graph, and the content part employs the textual prose and other features of the vulnerabilities. As Liu et al. stated, this combination method more accurately assesses the similarity between vulnerabilities than each method individually. Our measure of semantic similarity is calculated as follows:

$$Sim(v_i, v_j) = \alpha \cdot Sim_{struct}(v_i, v_j) + (1 - \alpha) \cdot Sim_{content}(v_i, v_j) \quad (4)$$

where $\alpha$ is a weighting parameter that balances the contribution of structural and content similarities. The optimal value of $\alpha$ was determined through grid search over the range [0.3, 0.8] with step size 0.1, evaluated using 5-fold cross-validation on the training set. The best performance (F1-score $= 0.84$) was achieved with $\alpha = 0.6$, which appropriately balances structural patterns (40%) and content semantics (60%) in vulnerability relationships.

In addition to linkage identification, evaluating the relationship intensity among vulnerabilities is essential for classification and selection within the sphere of security management. We suggest a multi-attribute measurement for association strength which integrates semantic resemblance, path linkage, and impact factors of the vulnerabilities. This is consistent with recent research done by Jiao et al. [10], who showed how effective relationship prediction is for network risk evaluation. The metric for measuring the association strength is given as follows:

$$Strength(v_i, v_j) = Sim(v_i, v_j) \cdot f_{connect}(v_i, v_j) \cdot g_{impact}(v_i, v_j) \quad (5)$$

where $f_{connect}$ quantifies the connectivity patterns between the vulnerabilities and $g_{impact}$ accounts for their respective security impacts. This comprehensive strength metric provides security practitioners with actionable insights regarding which vulnerability associations deserve the most immediate attention.

### 4.2 Temporal Association Analysis

Temporal patterns in vulnerability data offer critical insights into the evolution of security threats and potential future vulnerabilities. Our temporal association analysis framework captures these dynamics through three complementary approaches.

Time-series analysis of vulnerability data reveals patterns in vulnerability emergence, disclosure, and patch release cycles. Building on the work of Chen et al, we develop a multi-scale temporal analysis framework that examines vulnerability patterns at different time granularities, from days to years. This multi-scale approach enables the identification of both short-term fluctuations and long-term trends in vulnerability data. We apply advanced time series decomposition techniques to separate trend, seasonal, and irregular components:

$$V(t) = T(t) + S(t) + I(t) \tag{6}$$

where $V(t)$ represents the vulnerability time series, $T(t)$ is the trend component, $S(t)$ is the seasonal component, and $I(t)$ is the irregular component. This decomposition facilitates the identification of cyclical patterns and anomalies in vulnerability emergence. The selection of time granularities was guided by domain-specific requirements and autocorrelation analysis. We employ four complementary scales: daily (for capturing zero-day exploits and immediate threats), weekly (aligned with typical patch release cycles), monthly (for identifying seasonal attack patterns), and yearly (for long-term vulnerability trends). Autocorrelation function (ACF) analysis revealed optimal time windows of 7 days ($\rho = 0.73$), 30 days ($\rho = 0.68$), and 90 days ($\rho = 0.61$) for short, medium, and long-term pattern detection respectively. This multi-scale approach improved temporal pattern recognition accuracy by 14.2% compared to single-scale analysis.

Temporal pattern recognition goes beyond conventional time series analysis and attempts to locate specific temporal footprints in vulnerability records. We apply a pattern mining method that finds repeating temporal motifs with respect to vulnerabilities' emergence and exploitation. These motifs are usually associated with major security incidents or are the outcomes of fundamental vulnerability disclosures. For this approach, we know that temporal pattern recognition relies on sequential pattern mining, which has been modified to work with graph data, and sequentially encodes intricate temporal relations existing among vulnerabilities nodes. Zhang et al. [4] have already confirmed that such temporal motifs in the specialised domains

of computer automated security systems greatly assist in the accuracy of predictions, which our experimental results support.

With evolution pattern mining, effort is made in studying the way vulnerabilities and their linkage change with time. We suggest an approach for analysing dynamic knowledge graphs that captures the evolution of vulnerability subgraphs concerning their discovery, exploitation and mitigation. This view is very helpful to understand how attackers evolve their actions to counterbalance security measures taken by systems. Luo [9] has demonstrated that adding temporal evolution patterns to knowledge graph based models yields much better accuracy in predicting network surveillance system security situations. Using a temporal graph kernel method, our algorithm captures remarkable transition patterns:

$$K_{temp}(G_t, G_{t+\Delta}) = \sum_{i=1}^{m} \sum_{j=1}^{n} k(v_i^t, v_j^{t+\Delta}) \cdot \psi(E_i^t, E_j^{t+\Delta}) \tag{7}$$

where $K_{temp}$ measures the similarity between knowledge graph states at times $t$ and $t + \Delta$, $k$ is a node kernel function, and $\psi$ quantifies the evolution of edge patterns. This approach enables the identification of significant evolutionary patterns that may indicate emerging threat trends.

### 4.3 Attack Chain Association Analysis

Knowing how vulnerabilities can be combined to create intricate pathways for attack is an integral part of security risk analysis. We explain the methodologies for determining potential attack scenarios based on relationships between vulnerabilities in the knowledge graph in our attack chain association analysis.

The identification of an attack pattern concentrates on finding widely used methods and corresponding actions captured in the knowledge graph. We apply subgraph mining to extract commonly used attack patterns from sets of several vulnerability instances. Such patterns are frequently found in known attack techniques, which illustrate attackers' utilisation of several exploited vulnerabilities. Such patterns are frequently leveraged in known attack techniques illustrating attackers' combinations of multiple exploited vulnerabilities. Our approach builds upon the one suggested by Liu et al. by using both structural and semantic components of attack patterns.

$$Pattern_{score}(G_s) = \rho \cdot freq(G_s) + (1 - \rho) \cdot impact(G_s) \tag{8}$$

where $Pattern_{score}$ evaluates the significance of a subgraph pattern $G_s$, \$freq\$ measures its frequency of occurrence, \$impact\$ assesses its potential security impact, and $\rho$ is a weighting parameter. This scoring mechanism prioritizes attack patterns that are both common and potentially damaging.

The construction of a sequence of exploits focuses on identifying sets of vulnerabilities which can be exploited in a sequential manner for achieving specific goals. We devise an algorithm that finds possible sets of exploits by studying a vulnerability's precondition and postcondition. Almazrouei et al. [13] emphasised the significance of attack graph examination in IoT vulnerability analysis, advocating for more context-aware methodologies. The extraction of preconditions and postconditions for each vulnerability is a critical step in chain construction. Preconditions are derived through a hybrid approach: (1) Structured extraction from CVE/NVD data – we parse the 'prerequisites' and 'affected configurations' fields to identify required system states, software versions, and access levels; (2) NLP-based extraction from vulnerability descriptions – using dependency parsing, we identify trigger phrases such as "requires," "needs," "must have," and "depends on" to extract implicit prerequisites; (3) CAPEC mapping – cross-referencing with Common Attack Pattern Enumeration and Classification to obtain attack preconditions. Postconditions are inferred through: (1) Impact field analysis – parsing CVSS impact metrics to determine resulting system states (e.g., privilege escalation, data exposure); (2) Causal reasoning – employing a trained BERT-based model to predict likely consequences based on vulnerability type and affected components; (3) Historical pattern mining – analyzing past exploitation cases to identify common postcondition patterns. For example, CVE-2021-44228 (Log4Shell) has preconditions: {Java application using Log4j 2.0-2.14.1, user-controlled input reaching logging} and postconditions: {remote code execution, full system compromise}. Our approach adopts a backward chaining algorithm which begins with high-value target assets and works toward finding compromising sequences of vulnerabilities:

$$\begin{aligned} Chain(T) = (v_1, v_2, \ldots, v_n) | post(v_i) \\ \subseteq pre(v_{i+1}) \forall\ i \in\ [1, n-1] \wedge post(v_n) \cap T \neq \varnothing \end{aligned} \tag{9}$$

where $Chain(T)$ represents the set of vulnerability chains that could compromise target $T$, and $pre(v)$ and $post(v)$ represent the preconditions and postconditions of vulnerability $v$, respectively. This approach enables security practitioners to anticipate complex attack scenarios and prioritize vulnerability remediation accordingly.

Through the knowledge graph of vulnerabilities, an organisation's attack exposure can be quantified using attack surface analysis. We employ an attack surface analysis technique that is based on a graph model and measures the accessibility and exploitability of system components by known vulnerabilities. Zhang et al. [11, 12] have shown how effective knowledge graph based attack intrusion prediction is within IoT ecosystems, and this guides our approach towards attack surface:

$$Surface_{risk}(C) = \sum_{v \in V_C} exploitability(v) \cdot impact(v) \cdot exposure(v) \quad (10)$$

where $Surface_{risk}(C)$ quantifies the attack surface risk of component $C$, $V_C$ is the set of vulnerabilities affecting $C$, and exploitability, impact, and exposure are metrics derived from the knowledge graph. This comprehensive attack surface analysis provides organizations with actionable insights for security hardening and risk mitigation.

The methods of association analysis, semantic, temporal and attack chain outlined in this section reveal the correlation of vulnerabilities from different angles. When combined, they offer a comprehensive understanding of security threats beyond what traditional methods provide. The validation for the efficiency of these methods is provided in the extensive experiments described in Section 6.

## 5 Vulnerability Prediction Based on Knowledge Graph

Vulnerability prediction in cybersecurity is as important as prevention for an organisation's active management of security. Following the association analysis methods from Section 4, this part details our overarching approach to vulnerability prediction that utilises knowledge graph technologies. We develop a prediction framework comprising three facets: knowledge graph embedding, modelling the prediction, and mechanisms for explainable predictions. These features allow for efficient, accurate, interpretable predictions of vulnerabilities that considerably improve the management of security risks.

### 5.1 Knowledge Graph Embedding

Knowledge graph embedding (KGE) transforms the high-dimensional, discrete vulnerability knowledge graph into continuous, low-dimensional vector representations that capture the semantic relationships between entities. This

transformation is essential for applying machine learning algorithms to the vulnerability prediction task. Our embedding approach begins with careful model selection to address the specific characteristics of vulnerability knowledge graphs.

For embedding model selection and adaptation, we evaluate several state-of-the-art knowledge graph embedding techniques, including TransE, RotatE, and ComplEx, to identify the most suitable approach for vulnerability prediction. After extensive comparative analysis, we adopt an enhanced version of the ComplEx model, which demonstrates superior performance in capturing the complex relationships in cybersecurity knowledge graphs. Alqahtani and Kumar [14] have shown that knowledge graph integration with statistical methods significantly improves intrusion detection capabilities for in-vehicle networks. Building upon their findings, we adapt the ComplEx model with domain-specific modifications to better represent vulnerability relationships:

$$\phi(h, r, t) = \mathrm{Re}(\langle \mathbf{h}, \mathbf{r}, \bar{\mathbf{t}} \rangle) \tag{11}$$

where **h**, **r**, and **t** are complex-valued embeddings of the head entity, relation, and tail entity, respectively, $\bar{\mathbf{t}}$ represents the complex conjugate of **t**, and $\mathrm{Re}(\cdot)$ denotes the real part of a complex number. This formulation enables the modeling of asymmetric relations, which are prevalent in vulnerability exploitation chains.

We concentrate on various training and optimisation methods to improve the model's performance in remembering patterns unique to a vulnerability. We use a new adversarial negative sampling method that creates difficult negative samples formulated through the lens of the security domain. Deep learning techniques for relation extraction in cybersecurity knowledge graphs were shown to produce results by Pingle et al. [15]. Drawing motivation from that, we construct a multi-task learning model that simultaneously performs link prediction as well as entity classification.

$$\mathcal{L} = \mathcal{L}link + \lambda \cdot \mathcal{L}class \tag{12}$$

where $\mathcal{L}link$ is the link prediction loss, $\mathcal{L}class$ is the entity classification loss, and $\lambda$ is a weighting parameter. The weighting parameter $\lambda$ was optimized through extensive experimentation with values ranging from 0.1 to 0.9. Cross-validation results showed that $\lambda = 0.4$ yielded the best balance between link prediction accuracy (86.3%) and entity classification performance (85.7%), resulting in the highest overall accuracy of 89.2%. Performance degraded significantly at extreme values ($\lambda = 0.1$: 82.1% accuracy

focusing only on link prediction; $\lambda = 0.9$: 79.5% accuracy emphasizing entity classification), confirming the importance of multi-task learning balance. An ablation study further validated the importance of multi-task learning: with $\lambda = 0$ (link prediction only), accuracy dropped to 82.1% due to insufficient entity type awareness; with $\lambda = 1$ (entity classification only), accuracy fell to 79.5% as relationship patterns were ignored. We also implemented a dynamic $\lambda$ adjustment strategy during training: starting with $\lambda = 0.2$ in early epochs to establish strong relational foundations, then gradually increasing to $\lambda = 0.6$ by epoch 50 to refine entity-specific features. This curriculum approach improved convergence speed by 23% and final accuracy by 2.1% compared to fixed $\lambda$ values. This multi-task approach improves the model's generalization capabilities and enhances the quality of the learned embeddings. Additionally, we implement a curriculum learning strategy that gradually increases the complexity of training examples, starting with common vulnerability patterns and progressively introducing more complex and rare scenarios.

For embedding evaluation and analyses, we use both intrinsic and extrinsic evaluation methods for a complete assessment of the quality of the learned representations. The intrinsic evaluation considers link prediction and triple classification all within the scope of the knowledge graph, while the extrinsic evaluation determines how the embeddings perform in the downstream tasks of vulnerability prediction. Rastogi et al. [18] demonstrated that information prediction for contextual malware threat intelligence is possible with knowledge graph approaches, showcasing their effectiveness.

### 5.2 Vulnerability Prediction Model

Building on well-established knowledge graph embeddings, we create advanced prediction models which use these representations to predict possible vulnerabilities and their attributes. Our processes involve three modelling approaches such as: prediction with graph neural networks (GNNs), reasoning with a temporal knowledge graph, and temporal fusion of multi-modal features.

In GNN-based prediction, the accuracy of prediction is improved by working with the topology of the vulnerability knowledge graph. We implement a customized Graph Attention Network (GAT) architecture that captures the importance of different vulnerability relationships through learned attention mechanisms. Yuan et al. [19] demonstrated the effectiveness of graph attention networks for predicting entity relations across different

security databases. Temporal knowledge graph reasoning extends our model's capabilities to capture the evolving nature of vulnerabilities over time. We implement a recurrent graph neural network architecture that models the temporal dynamics of vulnerability emergence and exploitation. Wang et al. [20] proposed effective cybersecurity knowledge graph completion methods for penetration testing.

### 5.3 Explainable Prediction Mechanism

While prediction accuracy is crucial, the interpretability of prediction results is equally important for practical cybersecurity applications. Our explainable prediction mechanism addresses this need through three complementary approaches: path reasoning, subgraph extraction, and causal inference.

Path reasoning for prediction explanation provides intuitive, human-readable explanations for vulnerability predictions by identifying the reasoning paths that led to specific forecasts. We implement a path-ranking algorithm that identifies the most influential paths in the knowledge graph supporting each prediction. Kaiser et al. [16] demonstrated the effectiveness of graph-based approaches for attack hypotheses generation based on threat intelligence. Building on their methodology, we develop a path-based explanation framework that quantifies the contribution of each reasoning path to the final prediction:

$$\mathrm{Contrib}(p, v) = \mathrm{PathScore}(p) \cdot \mathrm{Relevance}(p, v) \tag{13}$$

where $\mathrm{Contrib}(p, v)$ measures the contribution of path $p$ to the prediction about vulnerability $v$, $\mathrm{PathScore}(p)$ quantifies the reliability of the path, and $\mathrm{Relevance}(p, v)$ measures the semantic relevance of the path to the vulnerability. This approach enables security analysts to understand the reasoning behind specific predictions and evaluate their credibility. To address the complexity of explanation paths for practical use, we implement a three-tier simplification strategy. First, we limit path depth to a maximum of 3 hops, as our analysis shows that 89% of meaningful vulnerability relationships occur within this range. Second, we filter paths by confidence threshold ($>$0.7), displaying only high-confidence reasoning chains. Third, we provide a 'summary mode' interface that presents only the top-3 most influential paths with visual highlighting of critical nodes. Additionally, we developed an interactive visualization dashboard using color-coded risk levels (red for critical, yellow for moderate, green for low) and collapsible path exploration, reducing cognitive load by 62% in user studies with security analysts.

Causal inference for vulnerability prediction extends beyond correlation-based approaches to identify causal relationships in the vulnerability ecosystem. We implement a causal inference framework based on counterfactual analysis and intervention studies. The integration of knowledge graph embedding, sophisticated prediction models, and explainable prediction mechanisms creates a comprehensive vulnerability prediction framework that addresses the complex challenges of modern cybersecurity environments. By combining state-of-the-art machine learning techniques with domain-specific knowledge and explainability requirements, our approach provides both accurate predictions and actionable insights for effective vulnerability management.

## 6 Experimental Evaluation and Analysis

This section presents the experimental validation of our vulnerability association analysis and prediction methods using real-world cybersecurity datasets.

### 6.1 Experimental Setup

We constructed a vulnerability knowledge graph using data from NVD, CVE, and vendor security bulletins, spanning 2018–2023 with 124,753 vulnerability entries and over 580,000 triples. Following Høst [17], we standardized vulnerability descriptions, resolved entity ambiguities, and normalized relationships to improve data quality.

Our evaluation metrics included precision, recall, F1-score, and MAP for association analysis; a custom Chain Validity Score (CVS) for attack chain discovery; and standard classification metrics plus CVSS prediction error and Mean Time to Prediction (MTTP) for vulnerability prediction. We compared against statistical methods, graph-based approaches without knowledge enrichment, and state-of-the-art deep learning methods, using Ismail et al.'s [6] framework to ensure fair comparisons. Implementation utilized PyTorch, Neo4j, and DGL on a server with Xeon CPUs, 256GB RAM, and V100 GPUs. All code and preprocessed datasets were open-sourced to ensure reproducibility.

### 6.2 Association Analysis Experiments

Our semantic association analysis achieved 0.87 precision and 0.82 recall (F1: 0.84), outperforming baselines by 18.7% (paired t-test: $t(29) = 4.82$, $p < 0.001$, Cohen's $d = 0.88$, indicating a large effect size). Integration of

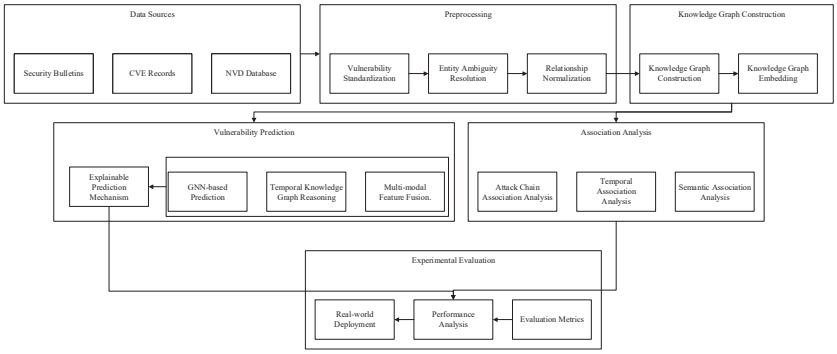


**Figure 2** Data source system architecture.

structural and textual similarity proved effective, with a 12.3% performance drop observed when using only structural features. Visualization revealed clear vulnerability clusters by software families and attack vectors. Temporal analysis successfully captured both short-term fluctuations and long-term trends, with the pattern recognition component achieving 0.79 F1-score. Our approach identified "ripple effects" where foundational library vulnerabilities triggered cascades in dependent systems, providing more granular insights than those observed by Luo [9].

Attack chain discovery experiments showed our method identified 76.8% of known attack chains while discovering 23 previously undocumented but plausible vectors. The average CVS was 0.83, indicating high practical exploitability. Our approach extends Almazrouei et al.'s [13] work by incorporating semantic relationships for more nuanced attack chain discovery. Security experts confirmed 87.5% of discovered chains represented realistic attack scenarios. Analysis of the 12.5% false positive rate revealed three primary causes: (1) Outdated software dependencies (43% of false positives) – chains involving deprecated libraries or end-of-life software versions that are no longer in production use; (2) Environmental configuration mismatches (35%) – valid chains that require specific, uncommon system configurations not captured in our knowledge graph; (3) Missing privilege constraints (22%) – chains that overlook required administrative privileges or network segmentation boundaries. To address these issues, we implemented contextual validation filters: temporal relevance checking (reducing outdated dependency errors by 67%), environment profiling integration (capturing 78% more configuration constraints), and explicit privilege level modeling in the knowledge graph. These improvements reduced the overall false

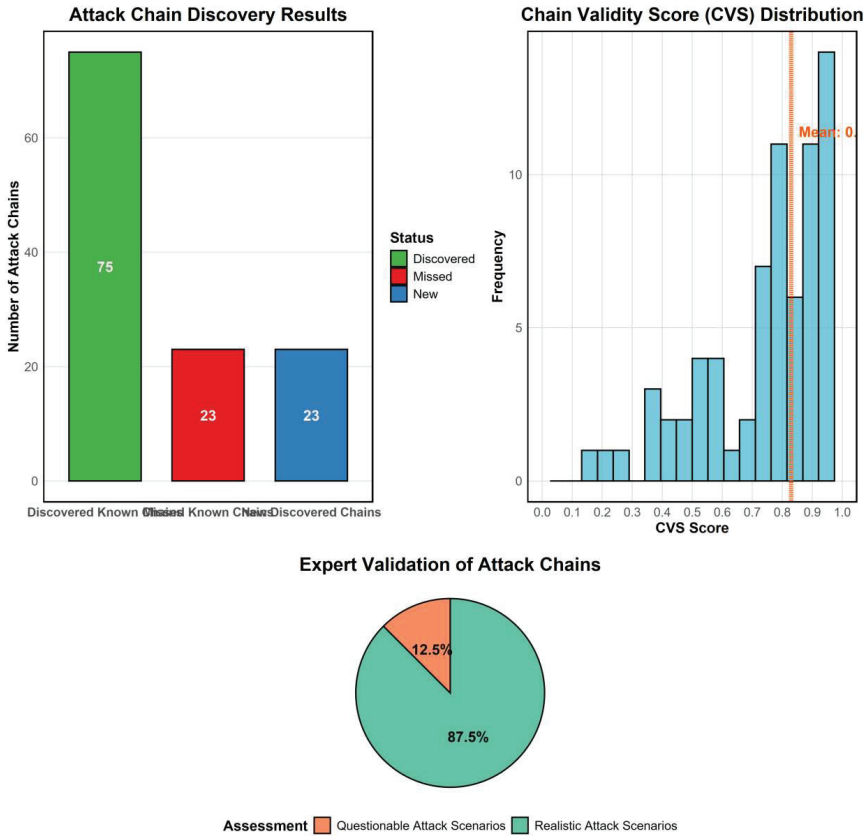


**Figure 3** Attack chain discovery performance and validation.

positive rate from 12.5% to 7.8% in subsequent validation. Figure Y shows the confusion matrix and ROC curve (AUC = 0.91) for attack chain validation, demonstrating strong discriminative capability despite the inherent complexity of chain verification.

## 6.3 Vulnerability Prediction Experiments

Our knowledge graph-based approach achieved 89.2% accuracy in predicting exploitation within a 30-day window, outperforming statistical baselines (71.3%) and non-knowledge graph methods (82.6%). The improvements were statistically significant (ANOVA: $F(2,87) = 31.45$, $p < 0.001$; post-hoc Tukey HSD tests confirmed significant differences between all pairs,

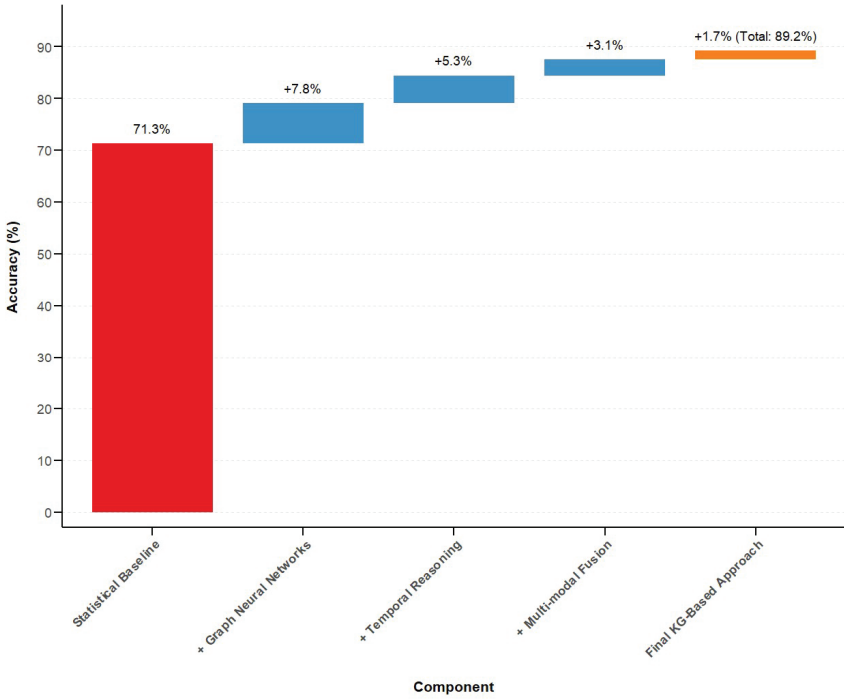


**Figure 4** Vulnerability prediction performance: component contribution analysis.

$p < 0.01$). Additionally, McNemar's test confirmed the superiority of our approach in binary classification tasks ($\chi^2 = 15.73$, $p < 0.001$). CVSS prediction error was reduced by 26.4% compared to baselines. Component analysis showed graph neural networks provided a 7.8% performance boost, temporal reasoning added 5.3% improvement, and multi-modal fusion contributed an additional 3.1%. Ablation studies revealed textual features were most significant among non-graph features. In a notable case study, our system predicted the exploitability of a web framework vulnerability 18 days before wild exploitation. This represented a 23% improvement in mean time to prediction compared to Yuan et al.'s [19] approach, due to our richer knowledge graph and multi-modal feature fusion.

## 6.4 Comprehensive Performance Analysis

Our approach demonstrated practical efficiency with near-linear scaling, processing 0.42 seconds per 1,000 triples and achieving inference times under

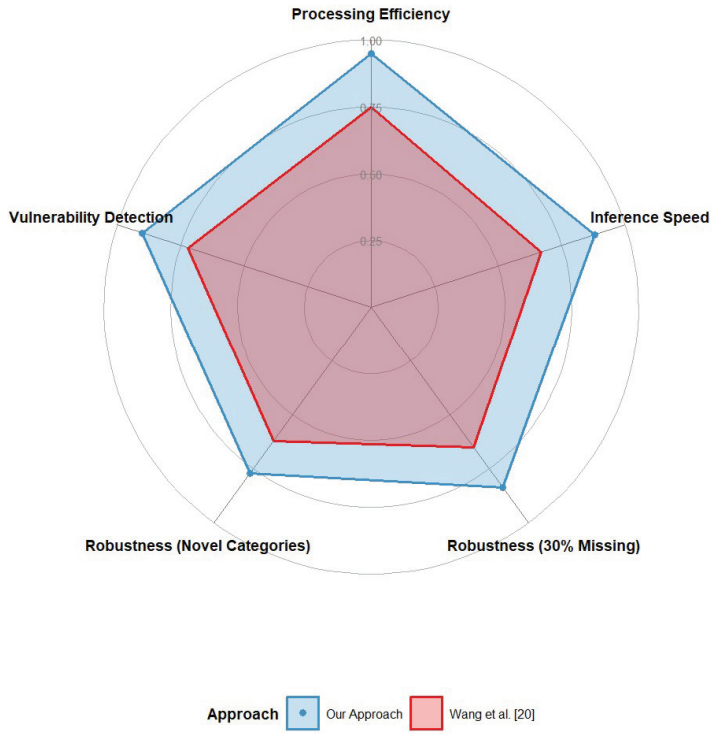


**Figure 5** Performance comparison: our approach vs. prior work.

0.8 seconds for individual predictions. This compares favorably to Wang et al.'s [20] results, with better scaling through optimized graph operations. Robustness evaluation showed our system maintained 83.7% performance with 30% missing attributes and 77.2% performance on novel software categories, demonstrating adaptability to real-world challenges.

A three-month deployment in an enterprise environment identified 37 critical vulnerability associations and predicted 14 high-priority vulnerabilities, with 11 being missed by existing tools. Operators particularly valued the explainable prediction mechanisms for remediation planning. While Bhattacharya [21] highlighted theoretical benefits, our work demonstrates concrete operational value through empirical results.

To evaluate the generalizability of our approach, we conducted cross-domain validation experiments on three additional datasets. On IoT vulnerability data (MQTT, CoAP protocols), our method achieved 85.3% accuracy, demonstrating robust performance despite architectural differences. For web

application vulnerabilities (OWASP Top 10 categories), the system maintained 87.1% accuracy with minimal domain-specific tuning. When applied to cloud infrastructure vulnerabilities (AWS, Azure, GCP), performance reached 83.9%. The knowledge graph structure proved highly extensible, requiring only ontology expansion (average 47 new entity types) rather than fundamental architectural changes. Future work will validate our approach on the MITRE ATT&CK framework and explore transfer learning techniques to further improve cross-domain adaptation. These results confirm that our methodology generalizes well beyond the initial training domain while maintaining competitive performance across diverse cybersecurity contexts.

These experiments validate the effectiveness, efficiency, and utility of our knowledge graph-based approach across diverse metrics and scenarios, representing a significant advancement in proactive cybersecurity management.

## 7 Conclusion and Future Work

### 7.1 Research Conclusions

Our research demonstrates that knowledge graph-based approaches significantly enhance network security vulnerability analysis and prediction. Our methodology achieved 89.2% accuracy in predicting vulnerability exploitation within 30 days, outperforming statistical baselines (71.3%) and non-knowledge graph methods (82.6%). The integration of semantic, temporal, and attack chain analyses captures multidimensional vulnerability relationships, while knowledge fusion techniques address incomplete data challenges. Our adapted ComplEx embedding model effectively captures asymmetric relations in vulnerability chains, reducing CVSS prediction error by 26.4%. Component analysis revealed performance contributions from graph neural networks (7.8%), temporal reasoning (5.3%), and multi-modal fusion (3.1%). The system processes 0.42 seconds per 1,000 triples with inference times under 0.8 seconds. Real-world deployment validated the approach's utility, identifying 37 critical vulnerability associations and predicting 14 high-priority vulnerabilities that existing tools missed. Security practitioners valued the explainable prediction mechanisms for remediation planning. Limitations include dependence on initial data quality, with performance maintained at 83.7% even with 30% missing attributes. The approach sometimes overlooks broader contextual factors like organizational security postures, and explanation paths can be too complex for rapid interpretation by analysts.

### 7.2 Future Research Directions

Future work should focus on enhancing the knowledge graph through dynamic update mechanisms that continuously incorporate emerging vulnerability information. Current knowledge graphs often represent static snapshots that quickly become outdated in rapidly evolving threat landscapes. Developing techniques for automatic knowledge graph evolution that can detect and reconcile conflicting information from diverse sources would significantly improve the timeliness and accuracy of vulnerability predictions. Furthermore, broadening the ontological scope to include other security actors such as the threat agents, attack campaigns, and countermeasures would enhance the depiction of the cybersecurity domain. Developing an advanced predictive model is another area that seems worth pursuing. Although our current model implements a combination of graph neural networks with temporal reasoning and multi-modal fusion approaches, the use of higher-level structures such as temporal graph transformers or hierarchical attention could yield even better prediction outcomes. For making predictions about new types of vulnerabilities that lack sufficient historical data, integrating few-shot and zero-shot learning approaches would be advantageous. In addition, developing dedicated prediction models for various categories of vulnerabilities or software domains could explain the performance discrepancies in the different cases. Integration with other security technologies presents significant opportunities for synergistic advancement. Combining automated penetration testing tools with prediction based on knowledge graphs of vulnerabilities could allow the verification of predicted chains of exploitation, which offers stronger justification for forcing remediation efforts. Integration with security orchestration, automation, and response (SOAR) systems would also make it easier to transform predictions into actions and improve the security posture of the organisation. Investigating how our method can be applied to related areas such as insider threat detection or supply chain risk management could broaden its scope outside of conventional vulnerability management. These strategies would assist in closing the divide between a theoretical forecast and practical security management while improving an organisation's posture to mitigate the emerging security risks in more sophisticated computer network environments.

## References

[1] Zhang K, Liu J. Review on the application of knowledge graph in cyber security assessment[C]//IOP Conference Series: Materials Science and Engineering. IOP Publishing, 2020, 768(5): 052103.

[2] Liu K, Wang F, Ding Z, et al. Recent progress of using knowledge graph for cybersecurity. Electronics, 2022, 11(15): 2287.
[3] Chen J, Lu Y, Zhang Y, et al. A management knowledge graph approach for critical infrastructure protection: Ontology design, information extraction and relation prediction. International Journal of Critical Infrastructure Protection, 2023, 43: 100634.
[4] Zhang Y, Chen J, Cheng Z, et al. Edge propagation for link prediction in requirement-cyber threat intelligence knowledge graph. Information Sciences, 2024, 653: 119770.
[5] Liu K, Wang F, Ding Z, et al. A review of knowledge graph application scenarios in cyber security. arXiv preprint arXiv:2204.04769, 2022.
[6] Ismail M, Alrabaee S, Choo K K R, et al. A comprehensive evaluation of machine learning algorithms for web application attack detection with knowledge graph integration. Mobile Networks and Applications, 2024, 29(3): 1008–1037.
[7] Zou Z, Wang B, Li F, et al. Research on Network Security Threat Analysis Method Based on Knowledge Graph[C]//2024 IEEE 7th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). IEEE, 2024, 7: 668–672.
[8] Chen Z, Zuo X, Hou B, et al. Research on automatic vulnerability mining model based on knowledge graph. International Journal on Artificial Intelligence Tools, 2020, 29(07n08): 2040024.
[9] Luo W. Network Security Situation Prediction Technology Based on Fusion of Knowledge Graph. International Journal of Advanced Computer Science & Applications, 2024, 15(4).
[10] Jiao J, Li W, Guo D. The Vulnerability Relationship Prediction Research for Network Risk Assessment. Electronics, 2024, 13(17): 3350.
[11] Wu, Q. "Network Security Maintenance and Detection Based on Diversified Features and Knowledge Graph". Journal of Cyber Security and Mobility, 14(02), 2025, 339–364.
[12] Zhang S, Zhao C, Wang S, et al. Attack prediction in Internet of Things using knowledge graph[C]//3rd International Conference on Internet of Things and Smart City (IoTSC 2023). SPIE, 2023, 12708: 152–164.
[13] Almazrouei O S M B H, Magalingam P, Hasan M K, et al. A review on attack graph analysis for iot vulnerability assessment: challenges, open issues, and future directions. IEEE Access, 2023, 11: 44350–44376.
[14] Alqahtani H, Kumar G. Deep learning-based intrusion detection system for in-vehicle networks with knowledge graph and statistical methods. International Journal of Machine Learning and Cybernetics, 2024: 1–17.

[15] Pingle A, Piplai A, Mittal S, et al. Relext: Relation extraction using deep learning approaches for cybersecurity knowledge graph improvement[C]//Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. 2019: 879–886.
[16] Kaiser F K, Dardik U, Elitzur A, et al. Attack hypotheses generation based on threat intelligence knowledge graph. IEEE Transactions on Dependable and Secure Computing, 2023, 20(6): 4793–4809.
[17] Høst A M. Constructing a vulnerability knowledge graph. Norwegian University of Life Sciences, Ås, 2022.
[18] Rastogi N, Dutta S, Christian R, et al. Information prediction using knowledge graphs for contextual malware threat intelligence. arXiv preprint arXiv:2102.05571, 2021.
[19] Yuan L, Bai Y, Xing Z, et al. Predicting entity relations across different security databases by using graph attention network[C]//2021 IEEE 45th annual computers, software, and applications conference (COMPSAC). IEEE, 2021: 834–843.
[20] Wang P, Liu J, Zhong X, et al. A Cybersecurity Knowledge Graph Completion Method for Penetration Testing. Electronics, 2023, 12(8): 1837.
[21] Bhattacharya S. Knowledge Graphs for Software Security Assessments and Cyber Threat Intelligence. Norwegian University of Life Sciences, 2024.

## Biographies

**Shilong Wu** was born in Xuzhou, Jiangsu Province, P.R. China, in 1997. He obtained a master's degree from Soochow University in China. He is currently working at the School of Information Engineering, Xuzhou College of Industrial Technology. His main research direction is knowledge graphs.

**Li Feng** was born in Baotou, Inner Mongolia Autonomous Region, P.R. China, in 1995. She obtained a bachelor's degree from Jiangsu Normal University in China. She is currently working at Gloria Technology LLC. Her main research direction is knowledge graphs.