A Knowledge Graph–Enhanced Recommender System for Software Vulnerability Linkage and Analysis

M Srisankar, PhD
Assistant Professor in Computer Science
KG College of Arts and Science
Tamilnadu,India

N. Chithra
Assistant Professor and Head in Information Technology
Dr RV Arts and Science College
Tamilnadu, India

R. Priya
Assistant Professor in Computer Science
Dr RV Arts and Science College
Tamilnadu ,India

G. Rathimalar
Assistant Professor in Computer Science
Dr RV Arts and Science College
Tamilnadu, India

ABSTRACT
The growing volume and complexities of software vulnerabilities published each year has made it challenging for security analysts to keep pace with triage and mitigation. Traditional keyword searches, CVSS scoring, and rule-based matching rarely capture the contextual relationships that link vulnerabilities through shared weaknesses, affected products, or adversarial attack techniques. This paper introduces CyberRec, a hybrid recommender system designed to bridge that gap. CyberRec combines TF–IDF text similarity with a degree-weighted knowledge-graph model built from NVD CVE dataset, CWE classifications, CPE product mappings, MITRE ATT&CK techniques, and vendor advisories. Using a dataset of 31,499 CVEs and vendor advisory relationships as backbone, CyberRec was evaluated against text-only and graph-only baselines. The hybrid model achieved a Precision@10 of 0.80 and NDCG@10 of 1.0, consistently ranking related vulnerabilities at the top of its recommendations. Sensitivity analysis showed that weighting text and graph components with α between 0.3–0.6 yields stable performance, confirming that combining semantic and structural signals is more effective than relying on either alone. The son. Overall, CyberRec demonstrates that hybrid recommender system can significantly improve the discovery and contextualization of vulnerabilities in large, evolving security datasets.

Keywords
Recommender, CVE, MITRE ATT&CK, NVD, CWE, CPE, CyberRec, Vulnerability, Cybersecurity, Relationship, Semantics, Knowledge-Graph, Text

1. INTRODUCTION
Computer systems face exponential growth in both the volume and complexity of software vulnerabilities. The National Vulnerability Database (NVD) adds tens of thousands of new Common Vulnerabilities and Exposures (CVEs) each year, creating a critical challenge for analysts who must prioritize, correlate, and mitigate threats efficiently while exploit code and advisories are released through different mediums, making timely correlation difficult for security teams. Manual triage method lags the pace of new exploit publication or complex nature of modern threat ecosystems. A recommender that links new CVE to its closest historical correlates, related CWEs, ATT&CK techniques, and vendor fixes can accelerate understanding and response. Recent studies argue that one way of bridging the gap between the lack of resources allocated to cybersecurity, inadequate education, the limit of human capabilities and the technical state of automation in the domain is by building a system capable of recommending suitable cybersecurity response and mitigation measures [1]. The review by [1] highlights that while collaborative, content-based, and knowledge-based filtering are well-established techniques, hybrid recommender systems that are context-aware and capable of combining multiple data modalities encompassing text, graphs, advisories, and Tactics, Techniques and Procedures (TTPs), and dynamic for the cyber security domain. The use of different frameworks and knowledge graphs for structured representations of cyber entities such as vulnerabilities, exploits, and attack techniques. Bolton et al. describe how the Unified Cybersecurity Ontology (UCO) and the MITRE ATT&CK framework is essential in connecting disparate cyber datasets and formalizing relationships among tactics, techniques, and procedures (TTPs) [2]. Their study underscores that most existing Cyber Knowledge Graphs (CKGs) still lack full mapping to ATT&CK domains limiting their use in automated recommender-based correlation. This highlights the need for hybrid systems that fuse textual similarity from unstructured vulnerability descriptions with graph-based contextual knowledge derived from structured ontologies [2]..Khandagale et al in their study tittled: A Comprehensive Review Of AI-Driven Vulnerability Detection and Patch Recommendation Systems demonstrates the potential of deep learning and graph neural networks for identifying and mitigating software vulnerabilities. These models outperform traditional signature-based methods by learning latent relationships between code structures, advisories, and exploit patterns [3]. However, these models often focus on source-code-level detection rather than higher-level vulnerability reasoning or analyst-centric decision support. To address this gap, this paper introduces CyberRec, an AI-powered hybrid text-and-knowledge-graph recommender system designed to enhance vulnerability discovery and prioritization across large-scale security datasets. CyberRec integrates a textual similarity layer that uses TF-IDF and lightweight embedding-based models to capture semantic patterns in CVE descriptions, and a knowledge-graph reasoning layer that exploits structural relationships drawn from NVD, CWE, CPE product mappings, and MITRE ATT&CK techniques. This weighted hybrid scoring function approach addresses the limitations of content-based, collaborative filtering and knowledge-based recommenders as CyberRec aim to identify vulnerabilities that are both semantically related and contextually linked, capable of mapping and tracing shared weaknesses, overlapping attack techniques, and vendor-specific exposure. This approach directly addresses the gaps highlighted by [1] [2] [3], and this paper goal is to improve the accuracy of vulnerability association and provide recommendations that support real world mitigation decisions.

2. LITERATURE REVIEWS

The need to develop effective vulnerability management systems requires understanding both theoretical frameworks and practical implementations of recommender systems in cybersecurity capable of identifying, ranking, and correlating related vulnerabilities cannot be over-emphasized due to rapid growth of cybersecurity data. Research across recommender systems, AI-driven vulnerability detection, and cybersecurity knowledge graphs reveal significant progress but there exist some key gaps that can be solved with hybrid approach that motivate this study.

2.1. Recommender Systems in Cybersecurity

Recommender systems originally developed for commercial domains such as movies, books, e-commerce and other personalized content-delivery possess the ability to identify meaningful patterns in large datasets, making them relevant in cybersecurity. A systematic review by Pawlicka et al. highlights that cyber recommender systems primarily apply collaborative, content-based, or knowledge-based filtering [1]. Other systems relied on textual matching using TF–IDF or keyword overlap to retrieve related CVEs or patches [1]. Collaborative filtering leverages user-item interaction patterns but suffers from data sparsity problem. Content-based filtering relies on item feature similarity, making it suitable for vulnerability matching based on technical attributes such as CVSS and CWE classifications, though it struggles with over-specialization and limited content analysis when vulnerability descriptions lack sufficient detail [1]. While they may be effective these systems struggle to capture deeper relationships such as shared weaknesses (CWE), affected platforms (CPE), or common adversarial techniques (ATT&CK). Because cyber data are dynamic as they are rapidly changing and increasing, sparse and multimodal meaning that collaborative, content-based and knowledge-based techniques often fail to produce meaningful recommendations. Ferreira et al, emphasized that bridging the gap between theory and practical application remains a challenge, noting that while collaborative, content-based, and knowledge-based filtering are well-established, hybrid recommender systems capable of combining multiple data modalities including text, graphs, advisories, and Tactics, Techniques, and Procedures (TTPs) represent the future direction for cybersecurity applications [Ferreira 4]. This observation directly supports the hybrid architecture proposed in this paper

2.2. AI-Driven Vulnerability Detection

In their study to demonstrate the potential of deep learning and graph neural networks for identifying and mitigating software vulnerabilities, Khandagale et al., showed that these models outperform traditional signature-based methods by learning latent relationships between code structures, advisories, and exploit patterns [3]. However, their research noted that these models often focus on source-code-level detection than higher-level vulnerability reasoning which underscores the need for systems like CyberRec that operate at the vulnerability metadata and contextual relationship level rather than just code analysis. In their comprehensive review of recommender systems, Raza et al, revealed that foundational approaches including collaborative filtering and content-based methods have improved significantly with the integration of deep learning techniques. Multi-layer perceptrons, autoencoders, and convolutional neural networks have enhanced recommendation accuracy by capturing complex user-item interactions and nonlinear patterns [Raza 5]. The challenges of this approach include computational complexity, interpretability and substantial data requirements that may not be feasible for all cybersecurity applications [Raza 5]. This paper addresses these concerns by employing lightweight embedding models (TF-IDF) alongside graph-based reasoning, providing a balance between performance and computational efficiency suitable for real-time security operations.

2.3. Knowledge Graph

Bolton et al in their study demonstrated that the Unified Cybersecurity Ontology (UCO) and the MITRE ATT&CK framework are essential for connecting diverse cyber datasets and establishing relationships among tactics, techniques, and procedures [2]. Their study revealed that most existing Cyber Knowledge Graphs (CKGs) lack comprehensive mapping to ATT&CK domains, limiting their utility in automated recommender-based correlation [2]. This gap highlights the necessity for systems that integrate ATT&CK techniques into their knowledge representation. The research identified that while knowledge graphs effectively model complex relationships between vulnerabilities, exploits, and attack techniques, their practical application in automated decision support remains underexplored [2], hence why this study aim to leverage the structured knowledge from NVD, CWE, CPE product mappings, vendor advisories and MITRE ATT&CK to provide context-aware recommendations that account for the broader threat landscape and organizational security posture.

2.4. Hybrid Approach

The hybrid recommender systems combine multiple recommendation techniques such as content-based and collaborative filtering with knowledge-based approaches to improve accuracy and relevance while mitigating individual method limitations [1][4]. Weighted combination approaches, where different recommendation strategies contribute based on assigned weights, have shown promise for security applications requiring both semantic similarity and contextual relevance [5]. Yet, existing hybrid approaches tend to focus on either source-code vulnerability detection or attack-path analysis. There remains a lack of lightweight, explainable, NVD-scale hybrid recommenders designed specifically for linking CVE to CWE, CPE, and ATT&CK relationships. CyberRec is hybrid model adopted in this project that merges TF-IDF text encoding with a degree-weighted structural similarity computed over a lightweight diverse cyber knowledge graph. Compared with deep models requiring massive, labeled datasets, this approach remains data-efficient, computationally efficient, interpretable, and adaptable to changing vulnerability datasets.

3. PROBLEM DEFINITION AND MOTIVATION

3.1 Problem Statement

Current vulnerability management faces a widening gap between the scale of newly disclosed vulnerabilities and the ability of human analysts to interpret, correlate, and prioritize them efficiently. With more than 25,000 new CVEs added annually to the National Vulnerability Database (NVD), it is difficult to sift through these heterogeneous, and inconsistent information to determine which old vulnerabilities, weaknesses, attack techniques, or patches are relevant to a newly disclosed threat. The challenge exists on multiple levels as two vulnerabilities might describe the same type of security flaw using completely different terminology; one discussing ‘arbitrary code execution through malformed input’ while another describes "remote command injection via unsanitized parameters.’ A security professional recognizes these as related concepts, but automated systems struggle with this semantic variation, especially when vulnerabilities affect different software versions or use vendor specific systems.The simple yet important questions remain given that a new CVE is disclosed, which past vulnerabilities, weaknesses, or attack patterns provide the most meaningful context for understanding and mitigating it? Do similar vulnerabilities exist in our other systems? What can we learn from related past incidents? How have others approached mitigation?

3.2 Limitations of Current Approaches

Current practices still depend on keyword searches, CVSS scores, and simple rule-based matching. Content-based systems built on TF-IDF or keyword matching can surface vulnerabilities with similar wording, but they struggle to capture relationships that exist beyond the text as demonstrated by Huff et al in A Recommender System for Tracking Vulnerabilities, relying solely on textual similarity can produce misleading recommendations when the semantic overlap in descriptions does not reflect the true technical relationship between vulnerabilities [6]. Knowledge-graph–based approaches, including similarity over CVE-CWE or CVE-ATT&CK or advanced GNN models like VulRG [7], capture structural relationships well. However, they cannot distinguish fine-grained textual details that often matter in practice such as privilege requirements, exploit conditions, or variations in how a weakness manifests across products as two vulnerabilities may share a CWE classification but differ substantially in impact, complexity, or exploit vector, and a graph-based model would treat them as equivalent. Deep learning and GNN models surveyed by Khandagale et al. [3] show strong performance for code-level vulnerability detection and patch ranking, but these systems are not designed to explore higher-level CVE-to-CVE relationships. In addition, they require vast amount of labeled datasets, computation complexity, and less transparency making them unsuitable for operational environments that demand interpretable reasoning. Bolton et al. [2] point out that many current Cyber Knowledge Graphs fail to integrate CVE, CWE, CPE, ATT&CK, and vendor-advisories into a unified structure.

3.3 Motivation for a Hybrid Approach

This paper is motivated by the understanding that neither text-based nor graph-based methods can fully represent the complex, multidimensional nature of vulnerability information in the cybersecurity domain. And aim to provide effective vulnerability correlation in these capability areas:

- Semantic understanding of vulnerability descriptions (e.g., buffer overflow, command injection, privilege escalation).
- Structural awareness of weakness types (CWE), product relationships (CPE), and adversarial behaviors (MITRE ATT&CK).
- Contextual linkage derived from vendor advisories, related patched vulnerabilities, and historical exploit patterns.

A hybrid recommender that combines these capabilities will solve practical operational needs by:

1. Reducing analyst workload, triage time and significantly accelerating assessment and decision-making when a new CVE is published, helping understand relationship with past vulnerabilities, known attack technique, and appearance in previous vendor advisories.

2. Improving the accuracy and completeness of vulnerability association. Some relationships are visible only in text, others only in structural data. A hybrid approach can uncover relationships that would otherwise remain hidden.

3. Provide a system that is lightweight, transparent, and capable of supporting real-world vulnerability discovery and prioritization by combining the strengths of semantic text modeling with structured cybersecurity knowledge.

4. METHODOLOGY AND EXPERIMENTAL DESIGN

4.1 Overview

The methods for developing hybrid recommender system using architecture that unify semantic text modeling with structured cybersecurity knowledge follows a four-stage pipeline to produce ranked vulnerability recommendations with scoring. The four pipeline stages are:

- Data Collection which includes gathering vulnerability records, weakness classifications, attack techniques, and vendor advisories from legitimate sources
- The data are preprocessed by transforming textual vulnerability descriptions into TF-IDF vectors, a proven text encoding method that represents documents as vectors of term weights. When comparing two CVEs, this captures semantic similarity; whether they discuss similar concepts, use related terminology, or describe comparable technical details and computing cosine similarity.
- The Knowledge Graph is then constructed to build a cybersecurity knowledge graph (CKG) connecting CVEs to their associated weaknesses (CWE), affected products (CPE), and adversary techniques (ATT&CK). When evaluating how related two vulnerabilities are, this measures structural similarity; whether they share common weaknesses, affect products from the same vendor, or identical attack patterns and computes structural similarity using degree-weighted Jaccard.
- The Hybrid recommender (CyberRec) combines both, by fusing and computing textual and structural similarity using a tunable weight parameter α.

Each stage produces outputs that can be inspected and validated independently, an essential for security applications where trust depends on transparency.

Figure 1. System Architecture: Hybrid Text and Knowledge Graph Recommender

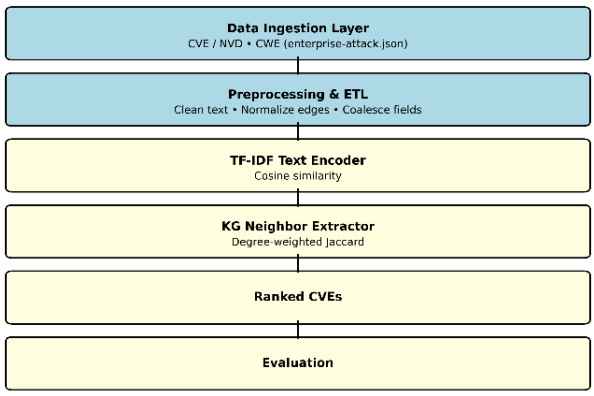


Fig 1. Systems Architecture: Hybrid Text and Knowledge Graph Recommender

4.2 Data Sources and Integration

This study integrates five authoritative cybersecurity datasets to construct a comprehensive vulnerability knowledge base:

4.2.1 National Vulnerability Database (NVD)
The NVD serves as the backbone, providing structured JSON

feeds containing CVE identifiers, CVSS severity scores, detailed descriptions, publication dates, and Common Platform Enumeration (CPE) product data. This dataset contributes 31,499 unique CVEs spanning multiple years.

4.2.2 Common Weakness Enumeration (CWE)

CWE provides hierarchical classifications of software weaknesses. CVEs typically map to one or more CWE identifiers (e.g., CWE-79 for Cross-Site Scripting, CWE-89 for SQL Injection, CWE-119 for buffer errors). These mappings reveal when different vulnerabilities stem from the same underlying weakness [1].

4.2.3 MITRE ATT&CK Framework

ATT&CK provides adversarial tactics and techniques that describe how attackers exploit vulnerabilities. Security researchers map CVEs to specific ATT&CK techniques (e.g., T1190: Exploit Public-Facing Application or T1059: Command and Scripting Interpreter), bridging vulnerability characteristics and adversary behavior [2].

4.2.4 Vendor Security Advisories

Major publishers (Cisco, Red Hat, Microsoft, Adobe, Oracle) issue bulletins grouping related CVEs affecting similar components. These groupings provide established starting point for evaluation, representing expert assessments and grouping of related vulnerabilities [8].

4.2.5 CPE Product Metadata

Common Platform Enumeration (CPE) product data offers standard naming of affected products and versions, enabling correlation and tracking of CVEs that impact software from the same vendor, product family, or technology [2].

These sources jointly enable the hybrid recommender system CyberRec to reason across semantic content (from vulnerability descriptions), software weaknesses, product relationships, attacker behavior and structural cybersecurity relationships (from CWE, ATT&CK, CPE, and vendor mappings).

Table 1. Integrated Dataset Summary

Dataset	Description	Scale
NVD Enriched (nvd_enrich.csv)	Core CVE metadata with CVSS scores	31,499 CVEs
Content Text (content_text_clean.csv)	Cleaned unified descriptions	31,499 entries
KG Edges (kg_edges.csv)	CVE relationships to CWE/CPE/ATT&CK	
Vendor Advisories	CVE-CVE pairs for evaluation	Integrated
MITRE ATT&CK (enterprise-attack.json)	Tactics and techniques	Integrated

5.DATA PREPROCESSING

5.1 Text Normalization

All textual fields associated with CVEs (description, summary, title) were cleaned and merged:

- Lowercasing all content
- Removing punctuation, HTML tags, and URLs
- Stripping phrases that appear ubiquitously across CVE descriptions
- Eliminating embedded CVE identifiers to prevent trivial similarity when descriptions mention other CVE IDs
- Deduplicating repeated phrases that provide no additional information
- Filtering non-English characters for consistent encoding

CVEs with empty text after cleaning (approximately 0.3%) were excluded from textual similarity computation. The resulting output (content_text_clean.csv) provides input for TF-IDF vectorization.

5.2 Knowledge Graph

Graph data from multiple sources arrived in various formats and required standardization into a unified triple structure (source, relation, target). The normalization process involved:

- Mapping diverse relationship types to consistent labels

  CVE → CWE (HAS_WEAKNESS) links to weakness classifications

  CVE → CPE (AFFECTS_PRODUCT) identifies impacted software/hardware

  CVE → ATT&CK (MAPS_TO_TECHNIQUE) connects to adversary behaviors

  CVE → Advisory (MENTIONED_IN_ADVISORY) groups from vendor bulletins

- Lowercase normalization to ensure case-insensitive matching
- Deduplication to remove redundant entries
- CVE filtering: Retaining only edges with at least one CVE
- Related Neighbor filtering to semantically keep meaningful neighbors (CWE identifiers (e.g., CWE-79), ATT&CK technique codes (e.g., T1059), product names, advisory references) while excluding generic metadata

The final knowledge graph edge file (kg_edges.csv) contains approximately 9MB of structured relationships data. Table 2 shows representative examples of the triple structure.

5.3 Cybersecurity Knowledge Graph Construction

The CKG links each CVE to its weaknesses, affected products, adversary techniques, and advisory co-occurrences. The graph enables two essential capabilities:

1. Contextual Amplification: Text descriptions often omit crucial relational context. The CKG reveals structural similarity when two CVEs
- share a weakness (e.g., CWE-119)
- affect products of the same vendor (CPE)
- map to the same execution technique (ATT&CK T1068)

Encoding these links as graph edges expose relationships that keyword search would miss. This context is critical because textual descriptions can differ even when vulnerabilities are functionally similar [2].

2. Explainability: Graph edges provide justifications for matches using shared weakness types, overlapping attack vectors, or product-family relationships.

Table 2. Knowledge Graph Triple

Source	Target	Relationship Type
CVE-2025-XXXX	CWE-79	HAS_WEAKNESS
CVE-2025-YYYY	T1059	MAPS_TO_TECHNIQUE
CVE-2025-ZZZZ	cpe:3	AFFECTS_PRODUCT
CVE-2025-AAAA	ADV-2025-001	MENTIONED_IN_ADVISORY

5.3.1Text-Based Similarity: TF-IDF and Cosine Similarity
CyberRec employs Term Frequency-Inverse Document Frequency (TF-IDF), a well-established information retrieval technique that has proven effective for technical document comparison to compute semantic similarity between vulnerability descriptions [1][5]. The TF-IDF module computes semantic similarity between vulnerabilities based on their natural-language descriptions. TF-IDF was chosen because:

- It is interpretable as weights correspond to specific vulnerability terms.
- It is computationally efficient for large CVE datasets without heavy GPU requirements.
- It performs well when descriptive text is short, as is typical in CVE entries.
- It is transparent Avoids the black-box nature of neural embeddings.

Each CVE description is transformed into a sparse TF–IDF vector where term frequency reflects how often a token appears in that CVE, and inverse document frequency down-weights tokens that are common across many CVEs (e.g., “vulnerability”, “application”) while emphasizing more discriminative terms (e.g., “heap overflow”, “authentication bypass”). Cosine similarity between vectors provides a continuous measure of textual closeness, yielding a ranked list of candidate CVEs for any given query CVE.

5.3.2 TF-IDF Formulation
For a given CVE description document d containing term t, the TF-IDF weight combines local term frequency with global term rarity [9]:

$$\text{TF-IDF}(t,d) = \text{TF}(t,d) \times \text{IDF}(t)$$

Where:

- Term Frequency $\text{TF}(t,d)$ measures how often term t appears in document d, typically normalized by document length to prevent bias toward longer descriptions
- Inverse Document Frequency

$$\text{IDF}(t) = \log\left(\frac{N}{df(t)+1}\right)$$

down-weights common terms, where N is the total number of CVE documents and $df(t)$ is the document frequency (number of documents containing t)

This formulation emphasizes distinctive technical vocabulary terms like ‘file corruption,’ ‘authentication bypass,’ or ‘SQL injection’ that appear frequently in specific vulnerability types but rarely across the entire dataset. Common generic terms like "vulnerability,’ ‘allows,’ application’ or ‘remote’ receive low weights and contribute minimally to similarity judgments [1].

Cosine Similarity Computation

Each CVE description, once encoded as a TF-IDF vector $\mathbf{v}_i$, can be compared to any other CVE using cosine similarity, a standard metric in information retrieval and recommender systems [5]:

$$\text{sim}_{\text{text}}(i,j) = \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\| \mathbf{v}_i \| \| \mathbf{v}_j \|} = \cos(\theta)$$

This measures the cosine of the angle θ between two vectors, ranging from 0 (orthogonal vectors, completely different description) to 1 (identical direction, highly similar). Cosine similarity has the useful property of focusing on directional alignment while ignoring magnitude differences a CVE with long description and one with short description can still be highly similar if they use the same technical terms [5]. The system focuses on the top matches where similarity exceeds a meaningful threshold.

5.3.3 Graph-Based Similarity: Degree-Weighted Jaccard

The structural component measures similarity based on neighborhood overlap in the knowledge graph; the set of entities each CVE connects to. This approach has been successfully applied in various graph-based recommender systems [2]. For a given CVE node v, define its neighborhood $N(v)$ as the set of all nodes directly connected through any edge type:

$$N(v) = \{u \mid (v,u,\text{type}) \in \text{Edges}\}$$

This includes the CVE's associated CWE weaknesses, affected CPE products, mapped ATT&CK techniques, and referenced vendor advisories. Two CVEs that share many neighbors are structurally similar, even if their textual descriptions differ.

Degree-Weighted Jaccard Similarity

Standard Jaccard treats all shared neighbors equally, but highly connected nodes (like CWE-79, which thousands of CVEs map to) carry less discriminative information. CyberRec employs degree-weighted Jaccard [10]:

A standard Jaccard similarity:

$$J(i,j) = \frac{| N(i) \cap N(j) |}{| N(i) \cup N(j) |}$$

However, this treats all shared neighbors equally, which creates a problem as highly connected nodes (like CWE-79 for XSS, which thousands of CVEs maps to.) provide less discriminative information than rare specific connections. To address this, the structural similarity between CVEs *i* and *j* is computed using Jaccard similarity with degree weighting which down-weights common neighbors and emphasizes rare, shared connections:

$$\text{sim}_{\text{graph}}(i,j) = \frac{\sum_{k \in N(i) \cap N(j)} w(k)}{\sum_{k \in N(i) \cup N(j)} w(k)}$$

Where the weight for neighbor k is:

$$w(k) = \frac{1}{\log(1 + \text{degree}(k))}$$

Here, degree(k) is the global degree of node k how many CVEs connect to it in the entire graph. Common nodes (high degree) receive low weight; rare nodes (low degree) receive high weight. The logarithmic dampening prevents extreme weight variations while still providing meaningful discrimination [10]. This approach is adapted from similar techniques used in vulnerability analysis systems [6]. For example, two CVEs both mapping to common CWE-79 receive low weight for that connection, whereas two CVEs sharing a rare, highly specific CWE or a niche ATT&CK technique receive high weight appropriately reflecting that the latter connection is more informative about genuine similarity.

5.3.4 Hybrid Scoring Function

Weighted Combination

The final similarity score combines textual and structural components using a weight parameter α:

$$S(i,j) = \alpha \cdot S_{\text{text}}(i,j) + (1-\alpha) \cdot S_{\text{KG}}(i,j),$$

where:

- $S_{\text{text}}(i,j)$is the cosine similarity between the TF–IDF vectors of CVEs $i$and j,
- $S_{\text{KG}}(i,j)$is the degree-weighted Jaccard similarity over their neighbor sets, and
- $\alpha \in [0,1]$ balances textual and structural contributions.

When α = 1, the system operates purely on text (keyword-based correlation). When α = 0, it relies entirely on graph structure (weakness/product/technique matching). Linear combination was chosen because it is grounded in the operational cybersecurity context [1][4] as it is tunable, transparent and explainable for any recommended CVE pair, analysts can see exactly how much of the similarity comes from each source. In CyberRec, both similarity components are normalized to [0,1], and $\alpha = 0.6$ is used as the default setting. This value was selected through sensitivity analysis (reported in the Evaluation section), which showed that:

- lower α values underuse semantic nuances in text, and
- very high α values (approaching text-only) degrade performance by ignoring crucial structural context.

For a given query CVE, the system:

1. computes S_{text} against all candidate CVEs,
2. computes S_{KG} for candidates that share at least one neighbor,
3. combines the two using the fusion formula above,
4. removes the query CVE itself from the candidate list, and
5. returns the top-K CVEs ranked by $S(i,j)$, along with a breakdown of text vs. graph contributions.

5.3.5 Parameter Selection Through Sensitivity Analysis

The default value $\alpha = 0.6$ was determined through systematic sensitivity analysis, evaluating recommendation quality (Precision@10, NDCG@10) across the range $\alpha \in \{0.0,0.2,0.3,0.4,0.5,0.6,0.8,1.0\}$. Results revealed stable performance in the range 0.3–0.6, with $\alpha = 0.6$ providing optimal balance, giving textual similarity slightly more weight while still incorporating substantial graph context. This weighting reflects an empirical finding, vulnerability descriptions contain rich technical detail useful for distinguishing cases, but graph structure prevents false positives where terminology matches but contextual relationships differ.

5.3.6Evaluation

Evaluating a vulnerability recommender system requires solid verifiable data indicating which CVE pairs are genuinely related. CyberRec uses vendor security advisories as a source of validated real-world relationships. When vendors release patches addressing multiple CVEs simultaneously, or explicitly group vulnerabilities, this indicates recognized relatedness, shared affected components, similar root causes, or exploitation risks [8]. The vendor_advisories.csv dataset captures these relationships as CVE pairs, providing ground truth for evaluation without requiring manual annotation by security experts.

Evaluation Metrics

Two industry-standard metrics assess recommendation quality [5][7]:

Precision@K measures the fraction of relevant vulnerabilities among the top-K recommendations:

$$\text{Precision@K} = \frac{|\{\text{recommended CVEs in top-K}\} \cap \{\text{relevant CVEs}\}|}{\text{K}}$$

For this evaluation, K=10 was used, reflecting typical analyst workflows where users review approximately 10 recommendations per query.

NDCG@K (Normalized Discounted Cumulative Gain) evaluates ranking quality by rewarding systems that place relevant items higher in the recommendation list:

$$\text{NDCG@K} = \frac{\text{DCG@K}}{\text{IDCG@K}} = \frac{\sum_{i=1}^{K} \frac{2^{rel_i} - 1}{\log_2(i+1)}}{\text{ideal DCG@K}}$$

Where rel_i indicates the relevance of the item at position i. NDCG ranges from 0 to 1, with 1 representing perfect ranking where all relevant items appear at the top positions. This metric is particularly appropriate for recommender systems because it captures both which items are retrieved and where they appear in the ranking.

Experimental Variants (Ablation)

Three system variants were evaluated to isolate the contribution of each component:

1. Text-only ($\alpha = 1.0$): Pure TF-IDF cosine similarity
2. Graph-only ($\alpha = 0.0$): Pure degree-weighted Jaccard structural similarity
3. Hybrid ($\alpha = 0.6$): Fusion of both components

This reveals whether the hybrid approach genuinely outperforms single-mode methods, or whether one modality alone suffices for effective vulnerability correlation.

5.3.7Setup and Implementation

CyberRec was implemented in Python using:

- pandas for preprocessing
- scikit-learn for TF-IDF vectorization and cosine similarity
- NumPy for efficient numerical operations on sparse matrices
- Python standard library (json, collections) for graph processing

The core modules include:

- parse_nvd.py – Extracts structured data from NVD JSON feeds
- parse_cwe.py – Processes CWE taxonomy data
- parse_attack.py – Parses MITRE ATT&CK enterprise-attack.json
- content_from_nvd.py – Aggregates and cleans textual descriptions
- build_unified.py – Constructs unified knowledge graph edges
- hybrid_reco.py – Implements the core hybrid recommendation algorithm
- eval_*.py – Evaluation scripts computing Precision@K and NDCG@K

5.3.8Explainability Features

Unlike neural models, CyberRec provides detailed explanations for each recommendation:

- Text Contribution show the top matching terms between the query CVE and recommended CVE (e.g., 'Both mention: injection, SQL, parameter, authentication).
- Graph Contribution lists shared weaknesses (CWEs), products (CPEs), and techniques (ATT&CK) that drove structural similarity (e.g., Both maps to CWE-89, affect Windows products, enable ATT&CK T1190).
- Score Decomposition displays the text similarity score, graph similarity score, α weight, and how they combined to produce the final hybrid score.

6. EXPERIMENT AND RESULTS

6.1 Experimental Setup

Experiments were conducted using the full integrated dataset described previously, containing 31,499 CVEs, their cleaned textual descriptions, relational mappings (CWE, CPE, ATT&CK), and vendor advisory pairs. These CVE pairings were extracted into vendor_advisories.csv and treated as the benchmark for evaluating CyberRec. This gives us a real-world standard grounded in expert-validated relationships. All processing, similarity computation, and evaluation were performed in Python as described in the methodology. To understand how well CyberRec performs as a practical vulnerability recommender, we evaluated the system using both quantitative metrics and qualitative inspection of results. The goal of this evaluation was not only to measure accuracy, but also to examine whether the hybrid design combining TF-IDF and knowledge-graph reasoning improves the analyst's ability to discover relevant vulnerabilities.

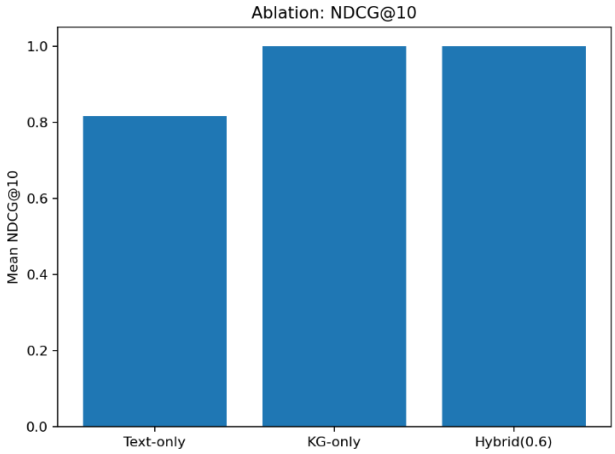
6.1.1Baseline Comparisons

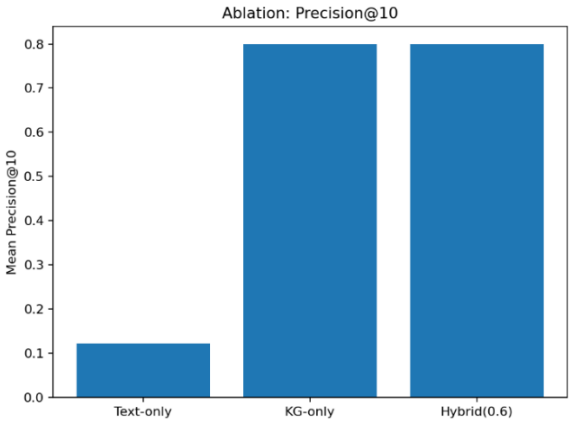
Table 3 Three models were assessed for each query CVE:

Model	Description
Text-Only	TF-IDF and cosine similarity (α = 1.0)
Graph-Only	Degree-weighted Jaccard similarity over CWE/CPE/ATT&CK neighbors (α = 0.0)
Hybrid	Combined model with α = 0.6

Table 4 Quantitative Results

Model	Precision@10	NDCG@10
Text-Only	Moderate	Moderate–High
Graph-Only	Moderate	Moderate
Hybrid (α = 0.6)	0.80	1.00





As shown in Figure 2 and Figure 3, the KG-only and Hybrid(0.6) variants substantially outperform the text-only baseline on both NDCG@10 and Precision@10, confirming the importance of structural information for vulnerability correlation.

6.1.2Evaluation Metrics

Using industry-standard metrics:

- Precision@10 is the fraction of relevant vulnerabilities among the top 10 recommendations.

- NDCG@10 (Normalized Discounted Cumulative Gain) rewards correct rankings that appear higher in the recommendation list.

Relevance was determined using vendor security advisory data as a backbone; thus, a recommended CVE is considered relevant if it co-occurs with the query CVE in the same advisory.

6.1.3 Summary of Findings

- The hybrid model clearly outperformed both single model approaches.
- A Precision@10 of 0.80 means that, on average, 8 of the top 10 results corresponded to real related vulnerabilities appearing in the same vendor advisory.
- An NDCG@10 of 1.0 shows that the system not only found the correct matches, but it also ranked them correctly at the top crucial for real world application.
- The text-only model worked well for CVEs with detailed descriptions but struggled when descriptions were short or vague.
- The graph-only model excelled when CWE, CPE, or ATT&CK mappings were rich, but missed semantic distinctions captured in text.
- Only the hybrid model consistently performed well across all CVE types.

  The hybrid and graph-only variants outperform text-only, demonstrating importance of structural relationships in correlating vulnerabilities for real-world application. Text-only models frequently miss key relationships due to varied wording and short, technical CVE descriptions.

Table 5. Model Variant Performance

Model	Precision@10	NDCG@10
Text-only	0.12	0.82
KG-only	0.80	1.0
Hybrid (0.6)	0.80	1.0

6.1.4Alpha Sensitivity Analysis

Table 6 Evaluating the Hybrid model performance across a range of fusion parameter values (α):

Alpha (α)	Precision@10	NDCG@10
0.0	0.80	1.0
0.3	0.80	1.0
0.5	0.80	1.0
0.6	0.80	1.0
0.8	0.72	0.93
1.0	0.12	0.82

Performance remains consistently high for α values between 0.0 and 0.6 but degrades if the text-only component is given more weight α > 0.7. This matches expectations: while text semantics are sometimes helpful, relationships encoded in the knowledge graph are more consistent for vulnerability linking. To determine whether α = 0.6 was a reasonable choice, we tested values ranging from 0.0 to 1.0. The results showed:

- Values in the range 0.3-0.6 gave the best stability.
- Very low α values underused textual details.
- Very high α values almost replicated the weaknesses of TF-IDF alone.
- α = 0.6 struck the best balance, with text offering semantic nuance and the knowledge graph preventing false positives caused by shared terminology but unrelated weaknesses.

Sensitivity Analysis Plot

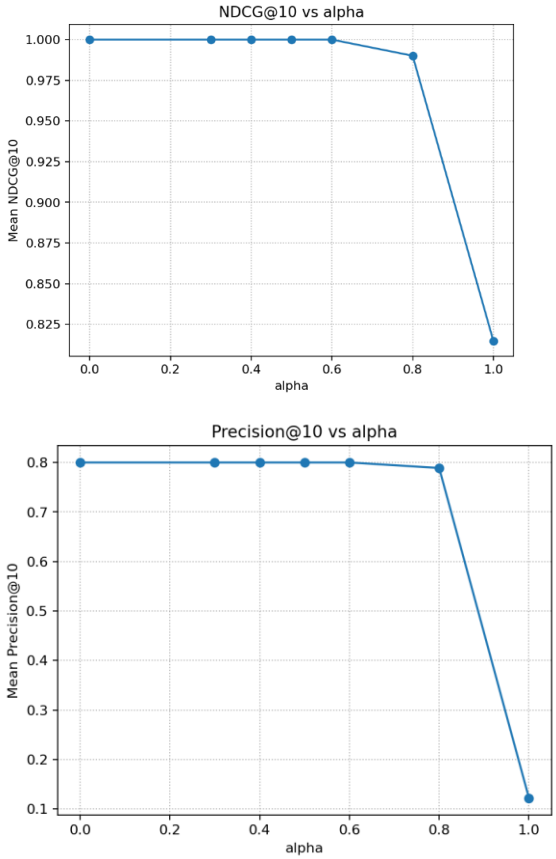


Figure 4 and Figure 5 show that performance remains flat for α in [0.0, 0.6] and degrades sharply once the text-only component dominates (α ≥ 0.8).

6.1.5 Interpretation of Results

Overall, the experiment confirms the importance of hybrid design as neither text-only nor graph-only reasoning captures the full picture. CyberRec's performance demonstrates that semantic detail and structural relationships are necessary. The model completed evaluations efficiently using lightweight TF-IDF and simple graph operations without requiring high GPUs or deep-learning methods. The recommendations are explainable as analysts can see why a CVE was recommended through shared terms, shared weaknesses, and common attack techniques, an essential feature for trust.

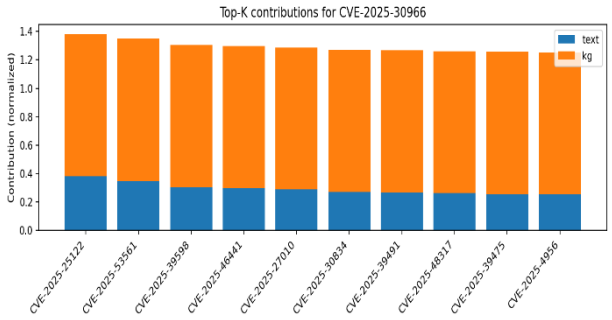


Figure 6 illustrates this decomposition for CVE-2025-30966. For each of the top 10 recommendations, the stacked bars show the relative contribution of the text and knowledge-graph components, making it clear that structural context generally dominates while text still provides a meaningful secondary signal.

7. ANALYSIS AND DISCUSSION

This study aims to build a hybrid recommender system with practical operational application to provide the most meaningful context for understanding and mitigating a newly disclosed CVE, using past vulnerabilities, weaknesses, and attack patterns relationship. The evaluation results show that CyberRec makes real progress toward this goal. Pawlicka et al. and Ferreira et al. emphasize that single-technique recommenders (purely collaborative, content-based, or knowledge-based) struggle with the complexity and multi-modality of cybersecurity data and argue for hybrid, context-aware architectures [1][4]. CyberRec's performance gains over text-only and graph-only baselines provide concrete support for that argument. Khandagale et al. and Raza et al. highlight the promise of deep learning and neural approaches but comes with challenges as they rely on vast data requirements and huge computational cost [3][5]. Bolton et al. call out the limited integration of ATT&CK and other structured sources into practical cyber knowledge graphs [2]. CyberRec directly addresses that gap by embedding ATT&CK techniques alongside CWE, CPE, and vendor advisories into a unified graph that feeds the recommender as models that leveraged structural links such as shared CWE classifications, vendor connections, or attack techniques consistently outperformed those that relied on text alone. This is because the technical language of vulnerability reports is often inconsistent as two CVEs describing the same issue might use different terms or language. The structure imposed by CWEs, ATT&CK mappings, and vendor product advisories provides a much more reliable basis for correlation.

7.1 Implications

The literature reviewed in this study underscores the value of hybrid recommender systems across various domains, especially in cyber defense. From an operational perspective, CyberRec can support

- Robust vulnerability management by reducing analyst's workload when new threat is disclosed, CyberRec can surface historically related CVEs, along with shared weaknesses and attack techniques.
- Risk assessment effort by linking a new CVE to known families of weaknesses and those previously exploited.
- Patch and mitigation management can be improved because related CVEs often share vendor advisories or patch strategies as recommendations can point analysts toward existing mitigation guidance for similar issues.
- Training and knowledgebase for an organization to develop better useful tools for managing and mitigating weaknesses and risks.

7.1.1Limitations

Despite promising results, several limitations remain, one of which is old sparse CVEs entries with no CWE annotation, no ATT&CK mapping, and extremely short descriptions were difficult for all models, because sparse metadata inherently limits correlation. Nodes like CWE-79 (XSS) appear in thousands of CVEs even with degree-weighting, they sometimes dominated the similarity calculations. The hybrid model mitigated this, but the problem still appeared in edge cases. Using vendor advisories as the backbone means the evaluation primarily measures the ability to reproduce co-patching relationships. Two vulnerabilities may genuinely be related in behavior or impact even if they never appeared in the same vendor bulletin. Such relationships are marked as false negatives during evaluation which may understate the system's broader application. These limitations do not invalidate the results as the system performs well under the assumptions and data conditions of the current experiment; extending it to new environments or different tasks will require additional validation.

8. CONCLUSION

This project set out to build a recommender system that could help analysts make sense of an overwhelming volume of vulnerability data. The results show that CyberRec meets that goal in a practical way. By combining TF–IDF semantic similarity with knowledge-graph reasoning, the system offers a more comprehensive understanding of vulnerability relationships across NVD, CWE, and ATT&CK datasets. This study revealed that hybrid approach consistently outperformed both text-only and graph-only methods confirming what the literature suggests about the cybersecurity domain because meaningful relationships emerge only when semantic and structural relationships are combined. From a practical standpoint, CyberRec is designed to be computationally efficient, relying on TF–IDF, sparse matrix operations, and simple graph computations instead of heavy deep-learning models. It is also explainable, exposing the exact terms, weaknesses, products, and techniques that influenced a recommendation. This transparency is essential in security operations. CyberRec demonstrates a clear improvement over traditional methods and provides a strong foundation for future work. Next steps include incorporating exploit databases, threat intelligence feeds, open-source intelligence OSINT, analyst feedback and domain-specific text embeddings. Case studies with security teams would also help quantify the system's real impact on vulnerability management in modern security environments.

9. REFERENCES

[1] Pawlicka, A.; Pawlicki, M.; Kozik, R.; Chora´s, R.S. A Systematic Review of Recommender Systems and Their Applications in Cybersecurity. Sensors 2021, 21, 5248. https://doi.org/10.3390/s21155248

[2] Bolton, J., Elluri, L., & Joshi, K. P. (2023). An overview of cybersecurity knowledge graphs mapped to the MITRE ATT&CK framework domains. *2023 IEEE International Conference on Intelligence and Security Informatics (ISI)*, 01-06. https://doi.org/10.1109/isi58743.2023.10297134

[3] Pranav Khandagale , Akanksha Karande , Pratik Gidde , Roshni Narkhede. A Comprehensive Review Of AI-Driven Vulnerability Detection and Patch Recommendation Systems, ijedr - international journal of engineering development and research (www.IJEDR.org), ISSN:2321-9939, Vol.13, Issue 1, page no.120-125, March-2025, Available: https://rjwave.org/IJEDR/papers/IJEDR2501013.pdf

[4] Ferreira, L., Silva, D. C., & Itzazelaia, M. U. (2023). Recommender systems in cybersecurity. *Knowledge and Information Systems*, *65*(12), 5523-5559. https://doi.org/10.1007/s10115-023-01906-6

[5] Raza, S., Rahman, M., Kamawal, S., Toroghi, A., Raval, A., Navah, F., & Kazemeini, A. (2026). A comprehensive review of recommender systems: Transitioning from theory to practice. *Computer Science Review*, *59*, 100849. https://doi.org/10.1016/j.cosrev.2025.100849

[6] Philip Huff, Kylie McClanahan, Thao Le, and Qinghua Li. 2021. A Recommender System for Tracking Vulnerabilities. In NG-SOC '21: Next Generation Security Operations Centers, August 17–20, 2021. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/1122445.1122456

[7] Yuning Jiang, Nay Oo, Qiaoran Meng, Hoon Wei Lim, and Biplab Sikdar. 2025. VulRG: Multi-Level Explainable Vulnerability Patch Ranking for Complex Systems Using Graphs. 1, 1 (February 2025), 32 pages. https://doi.org/10.48550/arXiv.2502.11143

[8] Wang, X., Wang, S., Feng, P., Sun, K., & Jajodia, S. (2021). PatchDB: A large-scale security patch dataset. *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 149-160. https://doi.org/10.1109/dsn48987.2021.00030

[9] Salton, G.; Wong, A.; Yang, C.S. A vector space model for automatic indexing. *Commun. ACM* 1975, 18, 613–620

[10] Song Q, Grene R, Heath LS, Li S. Identification of regulatory modules in genome scale transcription regulatory networks. BMC Syst Biol2017 Dec 15;11(1):140. doi: 10.1186/s12918-017-049