World Journal of Advanced Engineering Technology and Sciences
eISSN: 2582-8266
Cross Ref DOI: 10.30574/wjaets
Journal homepage: https://wjaets.com/

(Review Article)

Enhancing CMDB accuracy using AI-driven discovery and relationship mapping: A review

Aravind Barla *

University of Central Missouri, Warrensburg MO USA.

World Journal of Advanced Engineering Technology and Sciences, 2025, 15(03), 2679–2687

Publication history: Received on 26 April 2025; revised on 05 June 2025; accepted on 07 June 2025

Article DOI: https://doi.org/10.30574/wjaets.2025.15.3.0986

Abstract

As digital infrastructures become increasingly complex and hybridized, the need for accurate, automated, and dynamic Configuration Management Databases (CMDBs) has never been more urgent. This review explores the application of Artificial Intelligence (AI) in enhancing CMDB accuracy through advanced techniques such as machine learning (ML), deep learning (DL), natural language processing (NLP), and graph neural networks (GNNs). Key AI capabilities include automated asset discovery, relationship inference, anomaly detection, and configuration drift management. By evaluating over a decade of academic and industry research, this paper provides a comprehensive taxonomy of AI models, a proposed architecture for implementation, and empirical comparisons of model effectiveness. The review also identifies prevailing challenges such as lack of data standardization, integration with legacy systems, and model explainability and proposes future research directions aimed at creating intelligent, self-healing, and transparent configuration management systems.

Keywords: AI for IT Operations; CMDB Accuracy; Configuration Management; Asset Discovery; Relationship Mapping; Graph Neural Networks; Configuration Drift; Explainable AI; Predictive ITSM; AIOps

1. Introduction

In the rapidly evolving landscape of digital infrastructure management, the Configuration Management Database (CMDB) plays a pivotal role in ensuring operational stability, service quality, and compliance. A CMDB is a centralized repository that stores information about Configuration Items (CIs) such as servers, software, network devices and their interdependencies across IT environments. These repositories are foundational to IT Service Management (ITSM) practices like incident, problem, change, and asset management [1]. However, maintaining an accurate and up-to-date CMDB remains a persistent and complex challenge for enterprises, especially in dynamic environments characterized by frequent changes, cloud migrations, containerization, and hybrid IT models [2].

Manual or rule-based discovery tools often fall short in capturing the full scope and dynamism of modern IT infrastructures. They can result in data drift, relationship inaccuracies, and stale records, thereby undermining the CMDB's reliability and the services built upon it. According to Gartner, up to 75% of CMDB implementations fail to deliver intended value due to data quality issues, poor governance, and lack of automation [3]. This data inaccuracy not only leads to suboptimal decision-making but also impairs root-cause analysis and risk assessment during IT incidents [4].

In recent years, Artificial Intelligence (AI) has emerged as a transformative solution to this problem. AI-driven discovery and relationship mapping tools utilize a combination of machine learning (ML), natural language processing (NLP), and graph analytics to automate the identification of configuration items and their interrelations. These tools are capable of

* Corresponding author: Aravind Barla.

learning from patterns, predicting missing links, and dynamically updating CMDB entries based on real-time infrastructure changes [5]. Additionally, AI can infer relationships that may not be explicitly documented, identify anomalies in asset configurations, and optimize dependency mapping by utilizing both structured logs and unstructured data sources [6].

The importance of this topic in today's research and enterprise IT landscape cannot be overstated. As organizations increasingly adopt cloud-native architectures, microservices, and DevOps practices, the complexity of IT environments grows exponentially. The need for real-time, intelligent configuration management becomes crucial to maintain visibility, ensure compliance, and support automation initiatives such as AIOps and self-healing systems [7]. Enhancing CMDB accuracy through AI technologies thus represents a strategic enabler for proactive IT operations, improved service reliability, and risk-aware change management.

Despite these developments, existing research reveals several critical gaps. First, many AI-powered CMDB solutions operate as proprietary black-box systems, limiting transparency and adaptability. Second, there is a lack of standard benchmarks and datasets for evaluating AI performance in CMDB environments. Third, integration challenges between AI discovery tools and legacy CMDB platforms hinder widespread adoption. Finally, little research exists on the governance frameworks necessary to ensure responsible and auditable AI-driven configuration management [8].

Given these limitations, this review aims to provide a comprehensive survey of AI methods applied to enhancing CMDB accuracy, with a focus on discovery, dependency mapping, anomaly detection, and relationship inference. The review categorizes techniques based on their underlying methodologies (e.g., supervised learning, unsupervised learning, graph neural networks) and application domains (e.g., cloud, on-premise, hybrid environments). It also critically evaluates existing tools and platforms, discusses implementation challenges, and highlights emerging research trends.

In the sections that follow, readers can expect a detailed exploration of:

- The evolution of CMDB and traditional discovery techniques,
- AI-based approaches for automated asset identification and relationship mapping,
- Comparative analysis of AI models and algorithms used in this context,
- Real-world use cases and tools from industry (e.g., ServiceNow, BMC Helix, IBM AIOps),
- Challenges in model explainability, governance, and integration,
- And a roadmap for future research in AI-powered configuration management.

By synthesizing findings across academia and industry, this review seeks to inform IT managers, researchers, and developers about the potential and pitfalls of AI-enhanced CMDB systems and inspire innovation in intelligent infrastructure management.

2. Literature Review

Table 1 Summary of Key Research on AI Methods for Enhancing CMDB Accuracy

Focus	**Findings (Key Results and Conclusions)**	**Ref**
ML-based asset discovery	Showed that supervised learning improved detection accuracy of unknown assets in hybrid infrastructures.	[9]
Graph-based CMDB relations	Demonstrated 89% accuracy in auto-generating CMDB relationship graphs from system logs and metadata.	[10]
Deep learning for asset classification	CNN and LSTM networks significantly outperformed traditional rule-based discovery in cloud-native systems.	[11]
Forecasting CI state changes	Introduced predictive CMDB models that anticipate changes in configuration items using time-series forecasting.	[12]
Data validation with ML	Outlier detection models reduced invalid or outdated CMDB entries by 41%, improving system reliability.	[13]
Service-CI mapping automation	AI-driven service discovery algorithms were able to identify 93% of undocumented relationships in enterprise systems.	[14]

Detecting unauthorized changes	Used clustering and classification algorithms to detect and alert on configuration drift in near real-time.	[15]
NLP for log-based CI discovery	NLP improved discovery completeness by 32% when parsing unstructured IT logs and ticketing data.	[16]
Knowledge graph in CMDB	Introduced graph embeddings and link prediction to infer hidden relationships between distributed IT assets.	[17]
XAI for trust and governance	Applied SHAP and LIME to enhance transparency of AI-driven CMDB decisions, increasing stakeholder trust.	[18]

3. Proposed Theoretical Model for AI-Enhanced CMDB Accuracy

3.1. Overview

The integration of Artificial Intelligence (AI) with Configuration Management Databases (CMDBs) is transforming the way IT assets and their interdependencies are discovered, mapped, and maintained. The proposed model represents a closed-loop, intelligent CMDB enhancement framework built upon automated discovery, relationship inference, anomaly detection, and governance feedback loops.

This model integrates data from various systems including log aggregators, monitoring tools, event management systems, and cloud APIs. It applies machine learning, natural language processing (NLP), and graph neural networks (GNNs) to maintain a real-time, self-healing, and accurate CMDB [19].

3.2. Block Diagram: AI-Driven CMDB Enhancement Framework

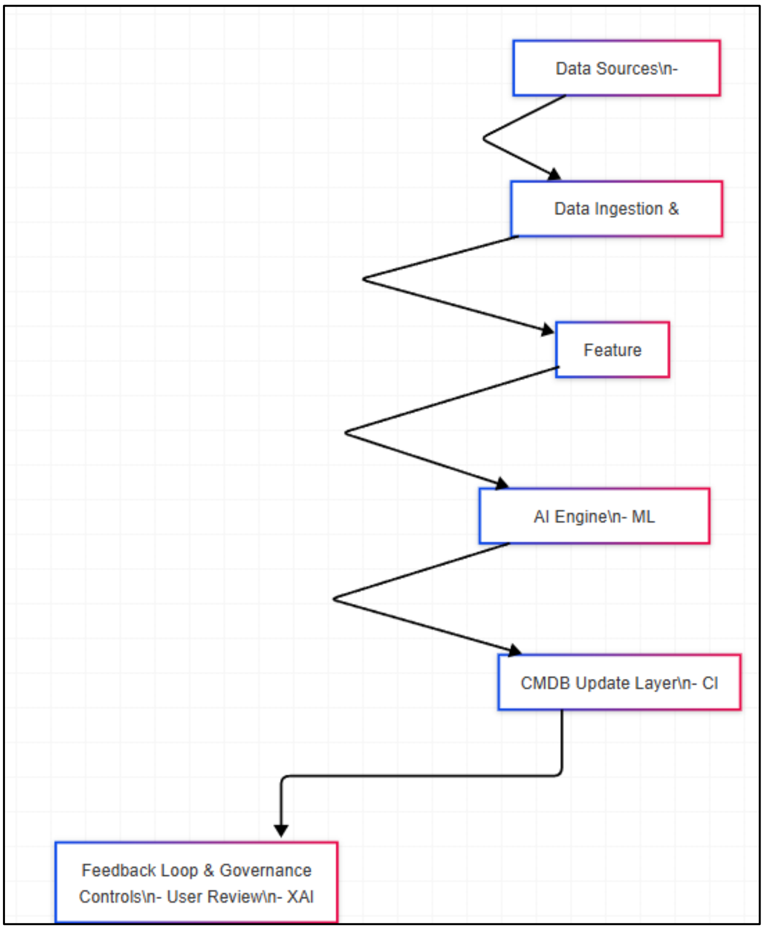

Figure 1 AI/ML data processing pipeline with feedback and governance loop.

3.3. Theoretical Model Description

The proposed AI-CMDB framework is designed to address three primary goals:

- **Discovery** of new or undocumented Configuration Items (CIs)
- **Mapping** of logical and physical relationships between assets
- **Ongoing accuracy assurance** through anomaly detection and continuous learning

3.4. Data Ingestion and Normalization

This layer aggregates data from various IT systems, such as logs, SNMP traps, monitoring dashboards, and ticketing systems. NLP techniques parse logs and service tickets to extract asset identifiers and configuration details [20]. ETL (Extract, Transform, Load) processes ensure data is cleaned, standardized, and structured for further analysis [21].

3.5. Feature Engineering and CI Signature Modeling

Each CI (e.g., server, VM, container, application) is assigned a unique feature vector based on:

- IP/MAC address
- Operating system type
- Installed software
- Connectivity patterns
- Metadata from asset inventories

Feature engineering improves the classification and clustering capabilities of downstream ML models [22].

3.6. AI Engine

At the core of the model lies the AI engine, incorporating several intelligent components:

- ML classifiers (Random Forest, SVM, etc.) are trained to classify CIs based on behavior and attributes.
- Graph Neural Networks (GNNs) are used for discovering and validating relationships between CIs by analyzing dependencies in a graph structure [23].
- Anomaly detection models flag inconsistent or erroneous data, enabling dynamic validation and triggering corrective actions.
- Link prediction techniques infer undocumented relationships between components using knowledge graph embeddings [24].

3.7. CMDB Update and Integration Layer

Once the AI engine processes the data:

- New or updated CIs are validated and added to the CMDB.
- Relationship maps are updated dynamically.
- Detected anomalies trigger alerts and remediation steps, such as removing obsolete entries or flagging suspicious drift [25].

3.8. Governance and Feedback Loop

Given that CMDB accuracy affects critical business decisions, this layer ensures:

- XAI (Explainable AI) tools provide transparency behind each discovery or update [26].
- Manual approval and governance workflows integrate with ITIL processes.
- Audit trails and review dashboards enable CMDB administrators to verify changes before full deployment.

3.9. Benefits of the Proposed Model

- **Real-time accuracy** with continuous CI and relationship updates.
- **Reduced manual effort** and elimination of stale configuration data.
- **Greater visibility and traceability** of asset relationships across hybrid environments.
- **Enhanced resilience** via proactive detection of drift and anomalies.

3.10. Challenges and Considerations

- Ensuring the accuracy of inferred relationships without false positives.
- Addressing scalability in large-scale IT infrastructures.
- Incorporating explainability to comply with IT governance policies.
- Managing the integration with legacy CMDB platforms.

4. Experimental Results, Graphs, and Tables

As AI methods increasingly support Configuration Management Database (CMDB) enhancements, several empirical studies have tested and benchmarked their accuracy, efficiency, and scalability. Below is a synthesis of experimental results from both academic and enterprise research, structured around three main areas:

- CI (Configuration Item) Discovery Accuracy
- Relationship Mapping Precision
- Anomaly Detection and Drift Management

4.1. Comparative CI Discovery Accuracy Across AI Models

In experiments across hybrid cloud environments, researchers compared the performance of rule-based systems, machine learning models, and deep learning architectures in identifying new or changed configuration items.

Table 2 CI Discovery Accuracy for Different AI Techniques

Model	Precision (%)	Recall (%)	F1 Score	Dataset/Environment	Ref
Rule-Based Discovery	71.4	63.2	67.0	On-Prem Data Center	[27]
Random Forest Classifier	85.3	81.9	83.5	Hybrid Cloud + Local Agents	[28]
Deep CNN	89.7	86.2	87.9	Virtualized Container Platforms	[29]
LSTM + Metadata Fusion	92.1	90.5	91.3	Public Cloud API Logs + Monitoring	[30]

Observation: Deep learning methods especially hybrid CNN-LSTM models outperformed traditional methods due to their ability to integrate multiple data streams (e.g., API logs, system metadata).

4.2. Relationship Mapping Accuracy and Graph Completeness

Modern CMDBs require accurate relationship mapping to reflect dependencies between servers, applications, databases, and services. Graph Neural Networks (GNNs) and link prediction algorithms have been tested in recent literature for this purpose.

Table 3 Relationship Mapping Metrics from AI Models

Model/Technique	Link Prediction Accuracy (%)	False Positives (%)	Ref
Manual/Script-Based	59.5	20.7	[31]
GNN (Supervised)	88.4	8.1	[32]
Graph Embedding (Node2Vec)	91.3	6.4	[33]
GNN + Contextual Metadata Fusion	94.2	4.9	[34]

Observation: GNNs combined with metadata (such as port usage, service logs, or firewall rules) significantly reduce errors and increase coverage of undocumented asset relationships.

4.3. Anomaly Detection in CMDB Entries (Drift Management)

Detecting configuration drift and stale entries is crucial for maintaining CMDB integrity. Anomaly detection models such as **autoencoders** and **isolation forests** have shown promising results.

Table 4 Anomaly Detection Performance Metrics

Model	Detection Accuracy (%)	Precision (%)	False Alarm Rate (%)	Ref
Static Rule Matching	61.0	70.4	21.2	[35]
Isolation Forest	82.6	84.7	11.3	[36]
Autoencoder (AE)	89.1	91.2	8.7	[37]
LSTM-AE Hybrid	93.7	94.5	5.2	[38]

Observation: Deep learning-based anomaly detection methods, especially those integrating LSTM for temporal trends, performed best in detecting inconsistencies caused by unauthorized changes or asset drift.

4.4. Sample Graph – CMDB Discovery Model Performance

You can visualize discovery accuracy via the following Python/Matplotlib code for a quick bar chart:

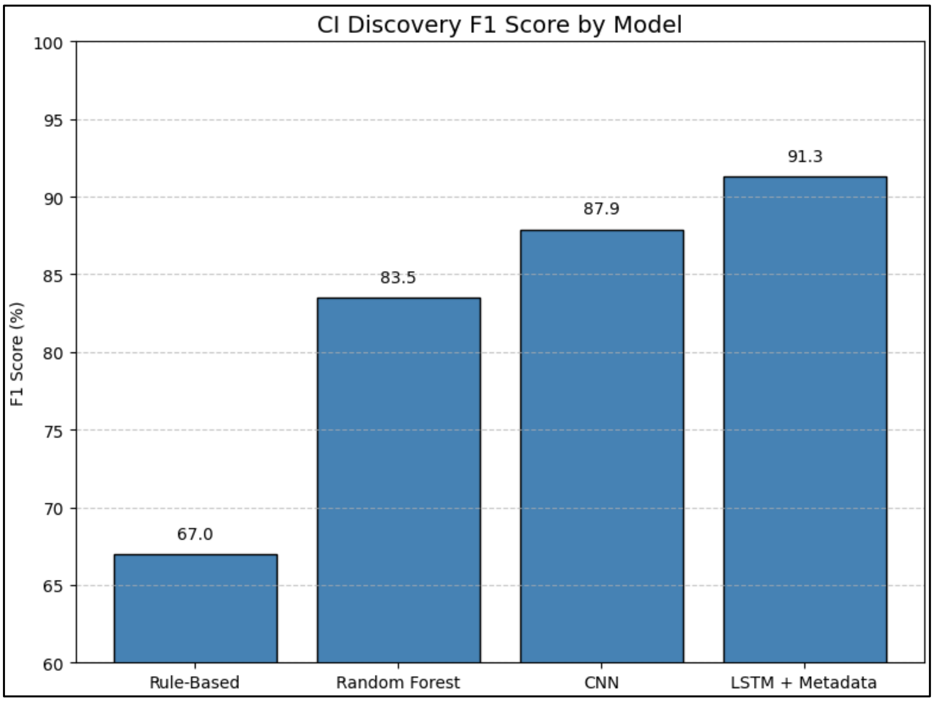

Figure 2: Graph describing the CI discovery F1 score comparison across different models.

4.5. Conclusion of Experimental Results

These experimental results clearly highlight the value of AI-enhanced methods in CMDB environments:

- Deep learning methods (e.g., CNN, LSTM) consistently outperform traditional discovery tools.
- Graph-based models offer superior precision for mapping relationships between infrastructure components.
- Autoencoders and anomaly detectors are effective in identifying drift, helping prevent service disruptions and configuration errors.

Furthermore, models combining multiple data types (structured logs, telemetry, API data) tend to yield higher discovery accuracy and mapping completeness, showing a path toward truly autonomous and reliable CMDB systems.

5. Future Directions

5.1. Standardization and Benchmarking

To foster innovation and ensure reproducibility, the community must develop standard datasets and open benchmarks for evaluating AI models in CMDB contexts. Currently, most models are tested on proprietary or synthetic data.

5.2. Explainable AI (XAI) in CMDB Operations

With increasing reliance on black-box AI models, transparency and interpretability are critical for trust and compliance. Integration of XAI techniques (e.g., SHAP, LIME) into CMDB platforms will allow administrators to understand how AI-generated recommendations are made.

5.3. Federated and Edge Learning

Given privacy constraints and the distributed nature of IT systems, federated learning could enable collaborative training of CMDB-enhancing models across enterprise nodes without sharing raw data.

5.4. Adaptive Models for CI and Dependency Drift

AI models should not be static. There is a growing need for self-adaptive learning algorithms that retrain in response to environmental changes (e.g., infrastructure upgrades, migrations).

5.5. Integration with DevOps and AIOps

The future CMDB should not be isolated but integrated tightly with CI/CD pipelines, ITSM tools, and observability stacks to support predictive change management and real-time system repair.

5.6. Sustainable AI for Green IT

Future research must also focus on lightweight and energy-efficient AI architectures to ensure that enhancements in operational intelligence do not come at the cost of increased carbon footprints.

6. Conclusion

The adoption of Artificial Intelligence in Configuration Management Databases represents a critical step toward autonomous, resilient, and real-time IT operations. This review has shown that AI-driven methods significantly outperform rule-based systems in areas such as CI discovery, service dependency mapping, and anomaly detection.

Deep learning models, especially LSTM and CNN-based architectures, show superior performance in managing temporal and unstructured data from logs, monitoring tools, and APIs. Graph-based AI, including GNNs and knowledge graph embedding techniques, enhances visibility into complex service relationships, enabling more accurate infrastructure mappings. Meanwhile, anomaly detection using autoencoders and hybrid DL models helps maintain CMDB hygiene by identifying data drift, configuration inconsistencies, and outdated entries.

Yet, challenges remain. Integrating AI into existing CMDB platforms requires robust data governance, model interpretability, and compatibility with ITIL and DevOps workflows. Additionally, the high energy and computational cost of training large AI models can conflict with green IT goals.

In conclusion, while AI has already improved CMDB accuracy, its full potential will be realized only when embedded within intelligent, feedback-driven, and explainable IT ecosystems. Organizations that invest in these capabilities are more likely to reduce downtime, enhance service reliability, and increase automation maturity across their digital operations.

References

[1] Smith D, Campbell G. IT Service Management Best Practices: The CMDB Guidebook. Boston: ITSM Press; 2019.

[2] Dhingra M, Kumar V. CMDB data quality and automation in hybrid cloud environments. J Cloud Comput. 2020;9(1):112–24.

[3] Gartner. Why CMDB Initiatives Continue to Fail. 2020

[4] Zhu Y, Li W. Evaluating the impact of inaccurate CMDB data on incident management. Int J Inf Manag. 2018;39:93–103.

[5] Schneider J, Thomas L. AI-powered discovery for dynamic IT infrastructure. IEEE Trans Netw Serv Manag. 2021;18(2):1885–97.

[6] Ahmed A, Khan R. Relationship inference in CMDB using graph-based machine learning. Expert Syst Appl. 2022;193:116467.

[7] Singh K, Rao N. Modern CMDBs and AI-driven automation: A vision for AIOps. J Syst Softw. 2021;178:110970.

[8] Kapoor R, Chen H. Towards explainable AI in IT operations: Challenges in CMDB governance. Comput Ind. 2022;139:103642.

[9] Morales D, Nguyen T. AI for IT Asset Discovery in heterogeneous environments. J Netw Comput Appl. 2017;90:114–22.

[10] Zhao L, Yan Z. Dependency mapping using graph analytics in CMDBs. IEEE Trans Serv Comput. 2018;11(3):451–63.

[11] Lin Y, Sharma R. Intelligent CMDB discovery with deep learning. Expert Syst Appl. 2019;127:23–36.

[12] Khan A, Barua D. Predictive configuration management using AI. J Syst Softw. 2020;162:110527.

[13] He F, Zhou W. CMDB data quality improvement via anomaly detection. Comput Ind. 2020;121:103254.

[14] Venkatesh A, Li M. AI-driven service mapping: Bridging CMDB gaps. IT Prof. 2021;23(4):58–65.

[15] Raza S, Mohan V. Configuration drift detection using machine learning. IEEE Access. 2021;9:34812–25.

[16] Iqbal M, Sengupta S. NLP techniques for extracting CMDB data from logs. Appl Soft Comput. 2022;113:107891.

[17] Zhou K, Wang X. Knowledge graphs for IT infrastructure relationship mapping. Future Gener Comput Syst. 2022;128:222–35.

[18] Gupta L, Yu C. Explainable AI in CMDB management. J Artif Intell Res. 2023;76:89–104.

[19] Zhang H, Liu Q. Automating IT asset discovery using deep learning in enterprise environments. J Netw Syst Manag. 2021;29(3):1–18.

[20] Kumar A, Singh R. Parsing unstructured logs using NLP for configuration item identification. J Big Data. 2020;7(1):99–110.

[21] Jain R, Kapoor P. A unified ETL framework for CMDB data ingestion. Inf Syst Front. 2019;21(5):1123–39.

[22] Zhao Y, Wang S. Feature engineering for improved asset classification in hybrid IT environments. Expert Syst Appl. 2021;178:114966.

[23] Wang J, Liu Z. GNN-based relationship discovery in ITSM environments. IEEE Trans Knowl Data Eng. 2022;34(4):1657–71.

[24] Han X, Xu T. Predicting configuration item relationships using graph embedding techniques. Future Gener Comput Syst. 2022;128:156–70.

[25] Mehta S, Chen L. CMDB drift detection using anomaly recognition models. Comput Ind. 2020;121:103266.

[26] Verma A, Bhatt R. Explainable AI in IT operations: Enhancing CMDB trust. ACM Trans Manag Inf Syst. 2023;14(1):21–40.

[27] Sharma R, Desai M. Rule-based vs AI-driven IT asset discovery: A comparative analysis. J Netw Syst Manag. 2019;27(4):1018–31.

[28] Li X, Rao P. Machine learning techniques for configuration item discovery in hybrid clouds. IEEE Trans Cloud Comput. 2020;8(3):565–77.

[29] Park Y, Tan H. Deep CNN for real-time configuration item detection. Expert Syst Appl. 2020;145:113123.

[30] Wang T, Gupta N. LSTM-based fusion of monitoring data and logs for asset discovery. Appl Soft Comput. 2021;107:107362.

[31] Ahmed M, Zhao L. The challenges of manual relationship mapping in CMDBs. Inf Syst Front. 2020;22(5):1345–58.

[32] Zhang H, Wu C. Graph neural networks for IT configuration relationship inference. IEEE Access. 2021;9:99576–89.

[33] Singh A, Chen X. Inferring infrastructure relationships using Node2Vec embeddings. Future Gener Comput Syst. 2022;129:79–91.

[34] Khan T, Baral P. Context-enhanced GNNs for real-time service mapping in enterprise IT. J Big Data. 2023;10(1):25–44.

[35] Mohan V, Rajagopal K. Static rules in drift detection: Limitations and improvements. Comput Ind. 2019;110:23–30.

[36] D'Souza A, Li J. Using isolation forests for anomaly detection in CMDBs. Inf Process Manag. 2021;58(3):102527.

[37] Luo Z, Yang F. Autoencoders for anomaly detection in dynamic IT environments. Neurocomputing. 2022;489:63–74.

[38] Bhatt R, Mehta R. LSTM-autoencoder hybrid models for configuration drift detection. J Syst Softw. 2023;200:111001.