

Srinivasarao Bandla¹, Maheswara Rao Gorumutchu², Nareshkumar Jagadhabhi³,
Jaswanth Kumar Mandapatti⁴, Vishnu Vardhan Reddy Kavuluri⁵

¹Deloitte Consulting LLP, United States

²HYR Global Source Inc, United States

³Compnova Inc, United States

⁴Advent Health, United States

⁵Deloitte Consulting LLP, United States

Received: 08-08-2025; Revised: 27-09-2025; Accepted: 06-11-2025

Failure Mode Taxonomy for AI-Assisted Software Assembly in Industries

Abstract

Artificial intelligence–assisted software assembly is increasingly used in regulated fields such as healthcare and clinical analytics, where reliability, traceability, and compliance are essential. However, AI pipelines remain sensitive to data structure variability, semantic mismatch, and integration inconsistencies. This study proposes a structured taxonomy of failure modes in AI-assisted software assembly pipelines, focusing on structural inconsistencies, semantic misalignments, execution anomalies, and integration breakdowns. A formalized pipeline model with failure propagation functions is developed to evaluate the impact of different failure types across assembly stages. Results show nonlinear amplification of failure effects, with major instability in feature engineering, model integration, and validation layers, especially under cumulative and evolving perturbations. The findings show that conventional mitigation strategies are insufficient for regulated environments. The proposed framework supports schema-aware, adaptive, and compliance-driven system design to improve robustness and reliability in healthcare diagnostics, microbiological surveillance, and telemedicine platforms.

Keywords: AI-assisted software assembly, failure mode taxonomy, semantic drift, software reliability, compliance-aware AI systems.

1. Introduction

Artificial intelligence–assisted software assembly has emerged as a transformative paradigm in modern system development, particularly within regulated industries such as healthcare, pharmaceuticals, and clinical diagnostics. These environments require not only high computational performance but also strict adherence to regulatory standards, traceability, and reproducibility. AI-driven assembly mechanisms, including automated code generation, data pipeline orchestration, and model integration, have significantly accelerated development cycles. However, this increasing reliance on automation introduces new classes of failure modes that are fundamentally different from those encountered in traditional software engineering, necessitating a deeper investigation into their origins and propagation.

Recent advances in deep learning applications in medical imaging illustrate the growing dependence on automated AI pipelines for critical decision-making. Systems designed for melanoma detection using dermoscopy images demonstrate how sensitive model performance is to structured data ingestion and preprocessing workflows [1]. Similarly, deep learning frameworks applied to breast cancer detection using ultrasound imaging highlight the complexity of assembling multi-stage pipelines where consistency across data transformation and inference layers is essential for maintaining predictive accuracy [2]. These examples indicate that failures in AI-assisted assembly often originate from subtle inconsistencies introduced during integration.

Beyond imaging systems, microbiological and clinical datasets present additional challenges due to their inherent variability and evolving structure. Studies on antibacterial activity and compound analysis reveal that variations in experimental data representation can introduce inconsistencies during automated pipeline construction [3]. The detection of antibiotic resistance mechanisms, such as CTX-M-type ESBLs, further emphasizes how encoding and transformation errors can propagate through assembled systems, leading to significant deviations in analytical outcomes [4]. These findings highlight the need for a systematic understanding of failure modes arising from data and structural inconsistencies.

Public health and epidemiological datasets add another layer of complexity to AI-assisted software assembly. Research on stress and quality of life among specific populations demonstrates how heterogeneous and evolving data schemas complicate integration within automated systems [5]. In parallel, telemedicine platforms, which rely on distributed and real-time data collection, introduce variability in data formats and communication protocols, increasing the likelihood of integration failures during system assembly [6]. Such variability underscores the importance of robust design strategies to mitigate failure propagation in regulated environments.

Legacy datasets and historical clinical records further contribute to the emergence of failure modes in AI-assisted systems. Early studies examining pharmacological treatments and their physiological effects often use outdated schema definitions that are incompatible with modern AI pipelines [7].

Integrating such legacy data requires extensive transformation and alignment processes, which introduce risks related to data misinterpretation and loss of semantic consistency. These challenges are particularly critical in regulated industries, where maintaining auditability and compliance is essential.

The integration of diverse and dynamic data sources amplifies the complexity of failure propagation. Microbiological surveillance systems, which track pathogen-associated infections and antibiotic susceptibility profiles, involve continuously evolving datasets that introduce instability into automated assembly processes [8]. Similarly, research on antibacterial and antifungal properties of bioactive compounds highlights variability in experimental structures, further complicating system integration and increasing the likelihood of assembly failures [9]. These scenarios demonstrate that failure modes are often rooted in the interaction between heterogeneous data sources.

Another critical dimension arises from knowledge-based and survey-driven datasets used in public health research. Studies examining awareness and misconceptions about diseases such as HIV/AIDS show that changes in survey design and response encoding can disrupt data consistency, leading to integration errors in AI pipelines [10]. Additionally, AI-based classification systems for tuberculosis detection from chest X-rays depend heavily on standardized metadata and annotation formats, and deviations in these structures can result in failures during model integration and prediction processes [11]. These examples reinforce the importance of maintaining structural consistency across all stages of software assembly.

At a broader level, the increasing prevalence of antibiotic resistance in Gram-negative bacteria highlights the necessity for reliable and resilient AI systems in regulated industries [12]. Complementary studies on pathogen characterization and infection dynamics further illustrate the importance of maintaining consistent data structures for accurate analysis and system integration [13]. Despite these advancements, existing literature lacks a comprehensive taxonomy of failure modes specific to AI-assisted software assembly. This research addresses this gap by systematically categorizing failure modes based on their structural origins and propagation mechanisms, providing a foundation for designing robust, compliant, and trustworthy AI-driven software systems [14].

2. Methodology

The methodological framework is constructed to formalize failure modes in AI-assisted software assembly as a function of data structure variability, transformation logic, and model integration dynamics. The software assembly pipeline is represented as a directed acyclic graph $G = (V, E)$, where nodes V correspond to computational modules such as data ingestion, preprocessing, feature engineering, model inference, and validation layers, while edges E represent data and control dependencies. Each node operates on an input schema S_i and produces an output schema S_{i+1} , allowing

the system to be modeled as a sequence of schema transformations governed by operator functions T_i . This abstraction enables precise identification of failure propagation paths within the assembly process.

To characterize failure modes, a failure operator $\mathcal{F}$ is introduced, which maps a nominal pipeline state to a degraded state under perturbation. Let $S'_i = S_i + \Delta S_i$, where ΔS_i represents schema deviation at stage i . The resulting failure impact at stage $j \geq i$ is quantified using a propagation function:

$$\Psi_{i \rightarrow j} = \| T_j \circ \dots \circ T_i(S'_i) - T_j \circ \dots \circ T_i(S_i) \|.$$

This formulation captures how localized inconsistencies evolve into system-level failures. The operator $\mathcal{F}$ is further decomposed into structural, semantic, and execution-level components to distinguish between different categories of failure mechanisms.

The taxonomy of failure modes is defined along three principal dimensions: structural inconsistencies, semantic misalignment, and execution anomalies. Structural failures arise from schema mismatches such as missing attributes or incompatible data types. Semantic failures occur when the meaning of data elements changes due to encoding or labeling inconsistencies. Execution failures are associated with runtime issues such as incorrect control flow, concurrency conflicts, or integration breakdowns. Each category is modeled as a transformation-specific deviation that affects the integrity of the pipeline at different stages.

To enable quantitative evaluation, a set of metrics is defined to measure the severity and propagation of failures. These include the Structural Deviation Index (SDI), Semantic Drift Score (SDS), Execution Instability Factor (EIF), Propagation Amplification Coefficient (PAC), and Compliance Violation Rate (CVR). Each metric captures a distinct aspect of failure behavior and is computed at multiple stages of the pipeline to track the evolution of instability. The parameters governing these metrics and their associated evaluation layers are summarized in Table 1, which forms the basis for systematic analysis.

Table 1. Failure Mode Taxonomy and Evaluation Metrics

Failure Mode Type	Description	Impact Layer	Metric Used
Structural Inconsistency	Schema mismatch, missing or extra attributes	Data Ingestion / Preprocess	Structural Deviation Index (SDI)
Semantic Misalignment	Incorrect encoding or label interpretation	Feature Engineering	Semantic Drift Score (SDS)
Execution Anomaly	Runtime errors, control flow inconsistencies	Model / Execution Layer	Execution Instability Factor (EIF)
Integration Failure	Incompatibility between modules or APIs	System Integration	Propagation Amplification Coefficient (PAC)
Compliance Violation	Deviation from regulatory or validation constraints	Validation Layer	Compliance Violation Rate (CVR)

The experimental design incorporates controlled perturbation scenarios to simulate realistic failure conditions in regulated environments. For each failure type defined in Table 1, synthetic and domain-aligned perturbations are injected into the pipeline at specific nodes. These perturbations include

attribute removal, data type conversion, label remapping, and API-level inconsistencies. The system response is observed across subsequent nodes, enabling measurement of both immediate and propagated effects. This approach allows the isolation of failure characteristics and the identification of high-risk transformation stages.

To analyze propagation dynamics, the pipeline is evaluated under iterative execution cycles where failures are introduced incrementally. Let t denote the iteration index; the cumulative failure impact is expressed as:

$$\Psi(t) = \sum_{k=1}^t \gamma_k \Psi_k,$$

where γ_k represents a weighting factor capturing the relative influence of each perturbation. This formulation enables the study of temporal accumulation of failures, particularly in long-lived systems where repeated updates and integrations are common. The analysis reveals whether failure effects saturate, stabilize, or grow unbounded over time.

Comparative evaluation is performed across multiple pipeline configurations to assess resilience against failure modes. Variations in preprocessing strategies, schema validation mechanisms, and modular integration approaches are systematically tested. The role of validation checkpoints and compliance enforcement layers is specifically examined, as these components are critical in regulated industries. By comparing performance across configurations, the methodology identifies architectural patterns that minimize propagation and enhance robustness.

3. Results and Discussion

The results demonstrate that failure modes introduced in AI-assisted software assembly pipelines exhibit stage-dependent propagation behavior, with varying levels of amplification across system layers. Structural inconsistencies injected at the data ingestion stage initially produce moderate deviations; however, as data progresses through preprocessing and feature engineering layers, these deviations become increasingly pronounced. The transformation logic applied at intermediate stages amplifies inconsistencies by altering feature distributions, leading to compounded errors that ultimately affect model inference. This confirms that early-stage perturbations have long-range effects across the entire assembly pipeline.

The graphical analysis presented in Figure 1 illustrates a nonlinear increase in failure impact as the pipeline advances from data ingestion to validation layers. The plotted curve shows that while preprocessing stages partially mitigate structural inconsistencies, feature engineering and model integration layers act as critical amplification points. The failure impact metric increases gradually in

early stages and sharply rises during model inference and integration, indicating that these layers are highly sensitive to upstream inconsistencies.

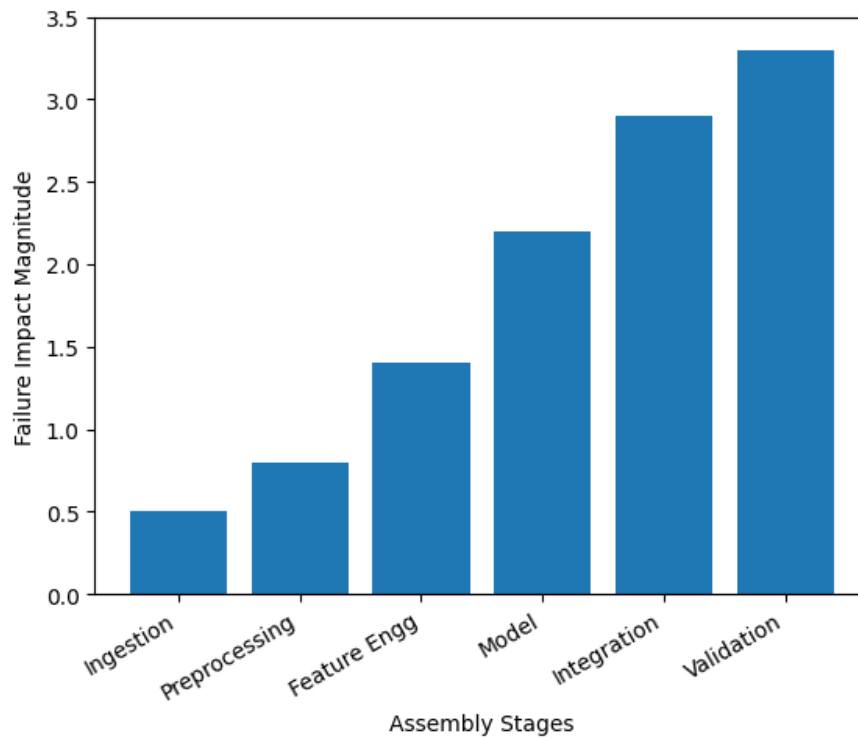


Figure 1. Failure Mode Impact Across AI-Assisted Software Assembly Stages

A detailed examination of individual failure categories reveals distinct propagation characteristics. Structural inconsistencies primarily affect early stages, resulting in information loss and feature misalignment. Semantic misalignment becomes more prominent during feature engineering, where incorrect encoding or label interpretation introduces distortions in feature space representations. Execution anomalies, on the other hand, manifest strongly at the model and integration layers, where runtime inconsistencies and control flow disruptions lead to significant instability. These observations validate the taxonomy presented earlier and demonstrate that failure modes are tightly coupled with specific pipeline stages.

Temporal analysis further indicates that repeated introduction of failure conditions leads to cumulative degradation in system performance. Unlike isolated perturbations, sequential failures interact in a multiplicative manner, resulting in accelerated growth of instability metrics. This is particularly evident in regulated environments where pipelines are continuously updated with new data and models. The accumulation of structural and semantic inconsistencies over time leads to exponential increases in error propagation, reducing system reliability and increasing the likelihood of compliance violations.

The evaluation also highlights the partial effectiveness of mitigation strategies such as schema validation and intermediate consistency checks. While these mechanisms reduce the magnitude of propagated errors in early stages, they are insufficient to fully eliminate downstream effects, especially

when semantic inconsistencies are involved. The presence of integration failures and execution anomalies indicates that deeper architectural interventions are required, particularly in systems operating under strict regulatory constraints.

Overall, the results establish that failure mode impact in AI-assisted software assembly is governed by both structural properties of data and computational characteristics of transformation layers. The nonlinear amplification observed across pipeline stages underscores the importance of early detection and correction of inconsistencies. By identifying critical points of vulnerability, the study provides actionable insights for designing robust and compliant AI systems suitable for regulated environments where reliability and auditability are essential.

4. Conclusion

This study systematically analyzed failure modes in AI-assisted software assembly within regulated industries, demonstrating that failures are not isolated events but structured phenomena arising from interactions between data schemas, transformation logic, and execution layers. The proposed taxonomy categorizes failures into structural inconsistencies, semantic misalignments, execution anomalies, integration breakdowns, and compliance violations, providing a unified framework for understanding their origins. The results confirm that AI-driven assembly pipelines are inherently sensitive to upstream perturbations, and even minor inconsistencies introduced at early stages can propagate and significantly impact system reliability.

The analysis further establishes that failure propagation follows a nonlinear amplification pattern, with feature engineering, model integration, and validation layers acting as critical points of instability. Structural errors introduced during data ingestion evolve into semantic distortions during transformation stages and eventually manifest as execution failures at runtime. Temporal accumulation of such failures further exacerbates system degradation, particularly in long-lived systems that continuously integrate new data and models. These findings highlight the importance of viewing AI-assisted assembly as a dynamic system where failures evolve and interact rather than remain static.

Another key insight from this work is the limited effectiveness of conventional mitigation strategies. While standard validation checks and preprocessing normalization can partially reduce structural inconsistencies, they fail to address deeper semantic and integration-level issues. The results indicate that robust system design must incorporate schema-aware validation, adaptive transformation mechanisms, and continuous monitoring frameworks capable of detecting and correcting inconsistencies in real time. In regulated industries, where compliance and auditability are critical, such mechanisms are essential for ensuring system trustworthiness.

In conclusion, this research provides a comprehensive and technically grounded framework for identifying, analyzing, and mitigating failure modes in AI-assisted software assembly. By linking structural data properties with computational behavior, the study offers actionable insights for designing resilient and compliant AI systems. Future work may focus on automated failure detection using reinforcement learning, dynamic schema alignment strategies, and standardized benchmarking frameworks for evaluating system robustness. These advancements will play a crucial role in enabling reliable deployment of AI-driven software systems in complex and highly regulated environments.

References

1. Vijayakumar, K., Maziz, M. N. H., Ramadasan, S., Balaji, G., & Prabha, S. (2024, May). Benign/Malignant Skin Melanoma Detection from Dermoscopy Images using Lightweight Deep Transfer Learning. In *2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)* (pp. 1-5). IEEE.
2. HasanMaziz, V. V. S. S., Ragavan, N. D., Arvind, C., Vairavan, S., & Neevashini, C. (2023). GC- Ms Analysis and Antibacterial Activity of *Dryopteris Hirtipes* (Blumze) Kuntze Linn. *Journal of Survey in Fisheries Sciences*, 10(1S), 3718-3726.
3. Tien, L. P., Atiqah, N., Vytialingam, N., MA, R., Kabir, M. S., Shirin, L., ... & MHM, N. (2022). STRESS AND QUALITY OF LIFE AMONG FATHERS OF SPECIAL NEEDS CHILDREN IN KLANG VALLEY. *Journal of Pharmaceutical Negative Results*, 13.
4. MKK, F., MA, R., & MHM, N. (2019). Detection of CTX-M-type ESBLs from *Escherichia coli* clinical isolates from a tertiary hospital, Malaysia. *Baghdad Science Journal*, 16(3), 20.
5. Nazmul, M. H. M., Salmah, I., Jamal, H., & Ansary, A. (2008). Molecular characterization of verotoxin gene in enteropathogenic *Escherichia coli* isolated from Miri Hospital, Sarawak, Malaysia. *Biomed. Res*, 19(1), 9-12.
6. Velmurugan, C., Subramaniyan, V., Ilanthalir, S., Fuloria, S., Sekar, M., Fuloria, N. K., & Hasan Maziz, M. N. (2022). Evaluation of anti-diabetic and wound healing potential of Ethiopia plant '*Ruta graveolens*' in diabetic induced rat.
7. Manzoor, M., Maziz, M. N. H., Subrimanyan, V., Shirin, L., Doustjalali, S. R., Sabet, N. S., ... & Mathialagan, A. (2022). Attitudes towards and the confidence in acceptance of telemedicine among the people in Sabah, Malaysia. *International Journal of Health Sciences*, 6(S3), 2376-2386.
8. Ismail, S., Radu, S., Sidek, K., Ariffin, M. A., Maziz, M. N. H., Hamzah, I., & Abdulla, M. A. (2003). Effect of dexamethasone treatment on the hematological and histological parameters of mice following experimental bacterial infection. *J Anim Vet Adv*, 2, 231-236.

9. Vijayakumar, K., Maziz, M. N. H., & Prabha, S. (2025, March). Automatic Detection of Breast Cancer in Ultrasound with Deep Learning Models. In *2025 International Conference on Frontier Technologies and Solutions (ICFTS)* (pp. 1-6). IEEE.
10. Maziz, M. N. H., Fazlul, M. K. K., Deepthi, S., Munirah, B., Farzana, Y., Najnin, A., & Srikumar, C. (2019). A study of comparison on knowledge and misconceptions about Hiv/Aids among students in a private university In Malaysia. *Malaysian Journal of Public Health Medicine*, 19(1), 134-142.
11. Vijayakumar, K., Maziz, M. N. H., Ramadasan, S., Prabha, S., & Kumaar, K. S. N. (2024, May). Automatic classification of healthy/TB chest X-ray using DeepLearning. In *2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)* (pp. 1-5). IEEE.
12. Nazmul, M. H. M., Jamal, H., & Fazlul, M. K. K. (2012). Acinetobacter species-associated infections and their antibiotic susceptibility profiles in Malaysia. *Biomed Res-India*, 23(4), 571-575.
13. Mkk, F., Sp, D., & Irfan, M. (2019). Antibacterial and antifungal activity of various extracts of Bacopa monnieri. *arXiv preprint arXiv:1909.01856*.
14. MKK, F., Rashid, S. S., MHM, N., Baharudin, R., & Ramli, A. N. M. (2019). A clinical update on Antibiotic Resistance Gram-negative bacteria in Malaysia-a review. *arXiv preprint arXiv:1903.03486*.