International Journal of Latest Engineering and Management Research (IJLEMR)
ISSN: 2455-4847
www.ijlemr.com || Volume 10 – Issue 12 || December 2025 || PP. 21-24

Predictive Cyber Resilience Modeling Based on AI-Enhanced Software Quality Assurance Metrics

Bohdan Savchuk
Anbosoft LLC, USA

Abstract: Cyber resilience increasingly depends on the internal quality characteristics of software systems rather than solely on external security controls. Traditional Software Quality Assurance (SQA) approaches primarily focus on defect prevention and compliance, leaving their predictive value for cyber resilience underutilized. This paper proposes an extended predictive cyber resilience model based on AI-enhanced SQA metrics, software dependency topology, and vulnerability propagation analysis. The approach integrates classical quality indicators with AI-oriented static analysis and graph-based influence coefficients to quantify systemic resilience. Experimental validation demonstrates that proactive, AI-driven SQA significantly improves vulnerability containment, reduces recovery time, and increases overall system robustness. The proposed framework enables early-stage cyber resilience assessment and supports data-driven security engineering decisions within DevSecOps pipelines.

Keywords: AI-oriented static analysis, cyber resilience, predictive modeling, software quality assurance, vulnerability propagation

1. Introduction

The exponential growth of software-intensive systems across critical infrastructure, industrial automation, and cloud-based services has significantly expanded the cyber threat landscape. Modern cyber incidents frequently exploit latent software vulnerabilities that originate during early development stages and propagate through complex dependency structures.

Conventional cybersecurity strategies emphasize perimeter protection, intrusion detection, and incident response. However, these measures are often reactive and fail to address the root causes of cyber fragility embedded in software quality deficiencies.

Software Quality Assurance (SQA) represents a systematic set of processes aimed at ensuring conformance to functional and non-functional requirements. Despite its proven impact on reliability and maintainability, SQA is rarely leveraged as a predictive instrument for cyber resilience assessment.

This research addresses this gap by proposing a predictive cyber resilience model that integrates AI-enhanced SQA metrics, information network topology, and vulnerability propagation analysis. The study aims to formalize the relationship between software quality characteristics and the system's ability to anticipate, withstand, and recover from cyberattacks.

2. Theoretical background and related works

Recent research confirms that a substantial proportion of cyber incidents are rooted in software design flaws, insufficient testing depth, and inadequate code review practices. Static analysis techniques have long been used to identify syntactic defects, but their effectiveness is limited when addressing complex semantic vulnerabilities.

AI-oriented static analysis introduces adaptive learning mechanisms capable of detecting vulnerability patterns, prioritizing risks based on exploitability, and correlating defects across codebases. These capabilities significantly enhance the preventive potential of SQA.

The concept of information network topology provides a mathematical foundation for analyzing dependency relationships within complex systems. The theory of suggestive influence allows modeling how defects or vulnerabilities in one component may affect others, enabling the identification of critical nodes whose compromise disproportionately impacts system resilience.

Existing cyber resilience frameworks primarily focus on operational response and recovery capabilities. However, they lack mechanisms for quantifying how development-stage quality controls influence long-term system resilience.

3. Methodology

3.1 Conceptual Overview

The proposed methodology aims to construct a predictive model of cyber resilience grounded in software quality assurance (SQA) data enriched with artificial intelligence (AI)–based static analysis and structural

influence modeling. Unlike traditional security assessments that rely on post-incident indicators, the presented approach evaluates resilience proactively during the software development lifecycle.

The central assumption of the model is that cyber resilience is not solely determined by external security controls but is significantly influenced by internal software quality characteristics and the structural role of individual components within the system. To formalize this relationship, the methodology integrates three analytical layers: quality metrics, AI-derived vulnerability indicators, and dependency-based influence coefficients [1].

3.2 Software System Representation

The software system under evaluation is modeled as a directed graph

$$G = (V , E) \tag{1}$$

Where

$$V = \{v_1, v_2, \ldots, v_n\} \tag{2}$$

Represents the set of software components (modules, services, or packages), and

$$E \subseteq V \times V \tag{3}$$

Represents dependency relationships between components.
An edge

$$(v_i, v_j) \in E \tag{4}$$

Indicates that component v_i depends on or interacts with component v_j. This representation allows the analysis of vulnerability [2] propagation paths and identification of components with disproportionate influence on system stability.

3.3 Selection of Quality and Security Metrics

Each software component v_i is associated with a vector of quality-related and security-related metrics:

$$M_i = \{m_{i1}, m_{i2}, \ldots, m_{ik}\} \tag{5}$$

The metric set includes:

- Structural quality metrics: cyclomatic complexity, code churn, dependency depth
- Process-related metrics: test coverage, code review frequency, defect density
- AI-derived security metrics: vulnerability severity score, exploit likelihood, semantic anomaly index

AI-oriented static analysis is used to extract higher-order security indicators that capture non-trivial vulnerability patterns beyond syntactic defects.

3.4 Metric Normalization

To ensure comparability between heterogeneous metrics, all metric values are normalized to the interval [0,1] using min–max scaling:

$$M_{ij}^{norm} = \frac{m_{ij} - m_j^{min}}{m_j^{max} - m_j^{min}} \tag{6}$$

For metrics with inverse impact on resilience (e.g., defect density, recovery time), inverse normalization is applied:

$$M_{ij}^{inv} = 1 - M_{ij}^{norm} \tag{7}$$

This normalization process ensures that higher normalized values consistently correspond to higher resilience contributions.

3.5 Influence Coefficient Modeling

Not all components contribute equally to cyber resilience. To capture structural importance, an influence coefficient I_i is assigned to each component based on its position in the dependency graph.

The influence coefficient is computed using normalized betweenness centrality:

$$I_i = \frac{B_i}{max(B)} \tag{8}$$

Where B_i denotes the centrality of component v_i.

Betweenness centrality was selected because it reflects the degree to which a component mediates information and control flow, making it particularly suitable for modeling vulnerability propagation. Components with high I_i values are considered critical nodes whose failure or compromise may disproportionately affect system resilience.

3.6 Predictive Cyber Resilience Score

The predictive cyber resilience score R_p is computed as a weighted aggregation of normalized metrics and influence coefficients:

$$R_p=\sum_{i=1}^{n} \quad w_i \cdot M_i \cdot I_i \tag{9}$$

Where:

- M_i is the aggregated normalized metric value for component v_i,
- w_i is the weight reflecting the relative importance of the metric group,
- I_i is the influence coefficient derived from network topology.

The weighting scheme is determined through expert evaluation and empirical observation. Moderate variations in weight values do not significantly affect comparative results; however, automated weight optimization is considered a direction for future research.

3.7 Workflow Integration

The overall workflow of the proposed methodology consists of the following stages:

- Collection of SQA and AI-derived metrics
- Construction of the software dependency graph
- Calculation of influence coefficients
- Metric normalization and weighting
- Computation of the predictive cyber resilience score

This workflow enables continuous resilience monitoring and can be integrated into DevSecOps pipelines for early detection of systemic vulnerabilities [3].

3.8 Methodological Assumptions and Limitations

The methodology assumes the availability of reliable SQA and static analysis data during development. While the model supports comparative assessment and trend analysis, absolute resilience values may vary across domains and architectures. Validation across large-scale industrial systems and heterogeneous environments remains an important direction for future research.

4. Experimental setup and data collection

The experimental study was conducted using two comparable software projects developed under controlled conditions. Both systems implemented identical functional requirements but differed in SQA maturity.

The baseline configuration relied on manual code reviews, limited test automation, and rule-based static analysis. The enhanced configuration integrated AI-oriented static analysis, automated testing pipelines, and topology-aware quality prioritization.

Data were collected over an eight-week development cycle, including metrics [4] such as test coverage, vulnerability density, mean time to recovery, and vulnerability propagation paths.

The experimental scope of this study was intentionally limited to controlled development environments to ensure reproducibility and precise metric collection. While this approach enables clear comparative analysis, it does not fully capture the diversity of real-world industrial systems. Validation across heterogeneous architectures and long-lived production systems represents an important direction for future research.

5. Experimental results and analysis

The enhanced SQA configuration demonstrated a significant reduction in vulnerability density and propagation potential. AI-based prioritization enabled early identification of high-impact defects.

The predictive cyber resilience score increased from 0.44 in the baseline system to 0.81 in the enhanced system, indicating a strong correlation between SQA maturity and cyber resilience [5].

Table 1: Comparative results of experimental configurations

Metric	Baseline	Enhanced	Improvement
Test Coverage (%)	35	85	+50
Vulnerability Density (per KLOC)	4.6	1.1	-76%
Mean Recovery Time (h)	52	9	-83%
Predictive Resilience Score	0.44	0.81	+84%

6. Discussion

The results confirm that AI-enhanced SQA provides measurable benefits beyond traditional quality improvement. The integration of influence modeling enables identification of critical components that require prioritized security controls.

The nonlinear growth of the resilience score suggests the existence of threshold effects [6], where incremental improvements in SQA maturity lead to disproportionate gains in cyber resilience.

7. Conclusion

From a practical perspective, the model may serve as a decision-support mechanism for prioritizing quality assurance activities with the highest expected impact on cyber resilience.

This paper presents an extended predictive cyber resilience model grounded in AI-enhanced software quality assurance. The proposed approach formalizes the relationship between development-stage quality controls and system-level cyber resilience.

The findings demonstrate that proactive SQA practices, supported by AI-oriented analysis and topology-aware metrics, significantly strengthen a system's ability to anticipate and withstand cyber threats.

Future work will focus on real-time resilience monitoring, automated weight optimization, and validation across large-scale industrial software systems.

Acknowledgements

The author acknowledges the support of academic peers and industry experts who contributed to this research.

References

[1]. Y. Nakonechna, B. Savchuk, Information network topology: mathematical model of suggestive influence, Theoretical and Applied Cybersecurity, 6(2), 2024, 52–65.
[2]. B. Savchuk, Development of software quality assurance performance indicators for assessing cyber resilience of systems. Measuring and computing devices in technological processes. 47-51. 10.31891/2219-9365-2025-83-6, 2025
[3]. A. Kovalova, Controlling software code vulnerabilities using AI-oriented static analysis, Measuring and Computing Devices in Technological Processes, 3, 2025, 7-11. https://doi.org/10.31891/2219-9365-2025-83-1.
[4]. M. Klima et al., Selected code-quality characteristics and metrics for IoT systems, IEEE Access, 10, 2022, 46144–46161.
[5]. P. Verma et al., towards a unified understanding of cyber resilience, IEEE Access, 2025.
[6]. M. Sinan et al., Integrating security controls in DevSecOps, Journal of Software: Evolution and Process, 37(6), 2025.