A DEVELOPER CENTERED BUG PREDICTION MODEL

UNDRAJAVARAPU HEMA
Department of MCA
SKBR PG COLLEGE, AMALAPURAM, A.P
hemasrinivas0407@gmail.com

Abstract
Software bug prediction is essential for efficient quality assurance, yet most existing models rely solely on code metrics or process metrics while ignoring the critical role of developers. This paper presents a Developer Centered Bug Prediction Model that incorporates developer-related factors such as experience, commit history, workload, and especially developer scattering (structural and semantic). The proposed model extends prior work by defining two new scattering measures that quantify how widely developers spread their changes across the codebase. These measures are evaluated on 26 open-source systems and compared against four competitive baselines, including models based on change entropy, number of developers, focus metrics, and traditional product metrics. Results show that the developer-centered model achieves superior prediction accuracy (F-Measure 10.3% higher than the change entropy model, 53.7% higher than developer-count models, and up to 29.3% higher than product-metric models). A hybrid model combining the best predictors further boosts performance by an additional 5%. The approach demonstrates that developer behavior is a highly complementary and powerful predictor for defect-prone components, enabling proactive resource allocation and improved software reliability.

Keywords: Developer-Centered Bug Prediction, Software Defect Prediction, Developer Scattering, Structural and Semantic Scattering, Machine Learning, Process Metrics, Software Quality Assurance.

I.Introduction

Bug prediction models play a vital role in software engineering by identifying likely defect-prone areas early, allowing teams to focus testing and maintenance efforts effectively. Traditional models primarily use product metrics (e.g., size, complexity) or process metrics (e.g., code churn, change entropy). While these have shown reasonable success, they largely overlook the human factor — developers themselves — whose experience, workload, coding patterns, and contribution behavior significantly influence defect introduction.

Recent studies have begun exploring developer-centric factors, showing that experienced and focused developers introduce fewer bugs, while scattered changes increase error-proneness. However, state-of-the-art process-metric models still do not fully capture developer scattering (how widely a developer's changes are distributed structurally and semantically across the system).

This paper addresses this gap by proposing a Developer Centered Bug Prediction Model that explicitly measures developer structural and semantic scattering. Structural scattering quantifies how far apart (in terms of subsystems) the modified components are, while semantic scattering captures the diversity of responsibilities implemented. The model is empirically validated on 26 open-source systems and compared against four strong baselines. A hybrid model combining the strongest predictors is also presented. The results confirm that developer scattering measures are powerful, complementary predictors that significantly improve bug prediction accuracy.

II. Literature Survey

Early bug prediction research focused on product metrics such as lines of code, cyclomatic complexity, and CK metrics. Studies by Basili et al., Gyimóthy et al., and others demonstrated moderate success but highlighted limitations in capturing dynamic development behavior.

Subsequent work shifted to process metrics, including code churn, change entropy (Hassan, 2009), and number of developers modifying a component (Ostrand et al., 2010; Bell et al., 2013). These models generally outperformed product-metric approaches, yet still ignored individual developer characteristics.

Developer-centric studies have shown that factors such as experience (Eyolfson et al., 2011), ownership (Bird et al., 2011), and focus (Posnett et al., 2013) influence bug introduction. Rahman and Devanbu (2011) found mixed results on experience, while Posnett et al. demonstrated that focused developers introduce fewer defects.

Prior work on scattering (Nucci et al., 2015) introduced structural and semantic scattering measures and showed their effectiveness on five systems. This paper significantly extends that foundation with a larger empirical study (26 systems), additional baselines, and a hybrid model.

III. Existing System & Proposed System

A. Existing System

Current bug prediction models are either product-metric based (complexity, size) or process-metric based (change entropy, developer count, focus metrics). While useful, they treat developers as interchangeable and fail to capture how scattered or focused a developer's contributions are.

Disadvantages of Existing Systems

1. Ignores developer-specific behavior and scattering patterns.
2. Limited accuracy due to missing human factors.
3. Poor complementarity when combining multiple metric families.
4. Inability to identify high-risk developers for targeted review.

B. Proposed System

The proposed Developer Centered Bug Prediction Model introduces structural and semantic scattering measures that quantify the spread of a developer's changes. These are used as predictors alongside traditional metrics. The model is trained using machine learning and evaluated in a large-scale empirical study. A hybrid model combines the strongest predictors from all approaches.

Advantages of the Proposed System

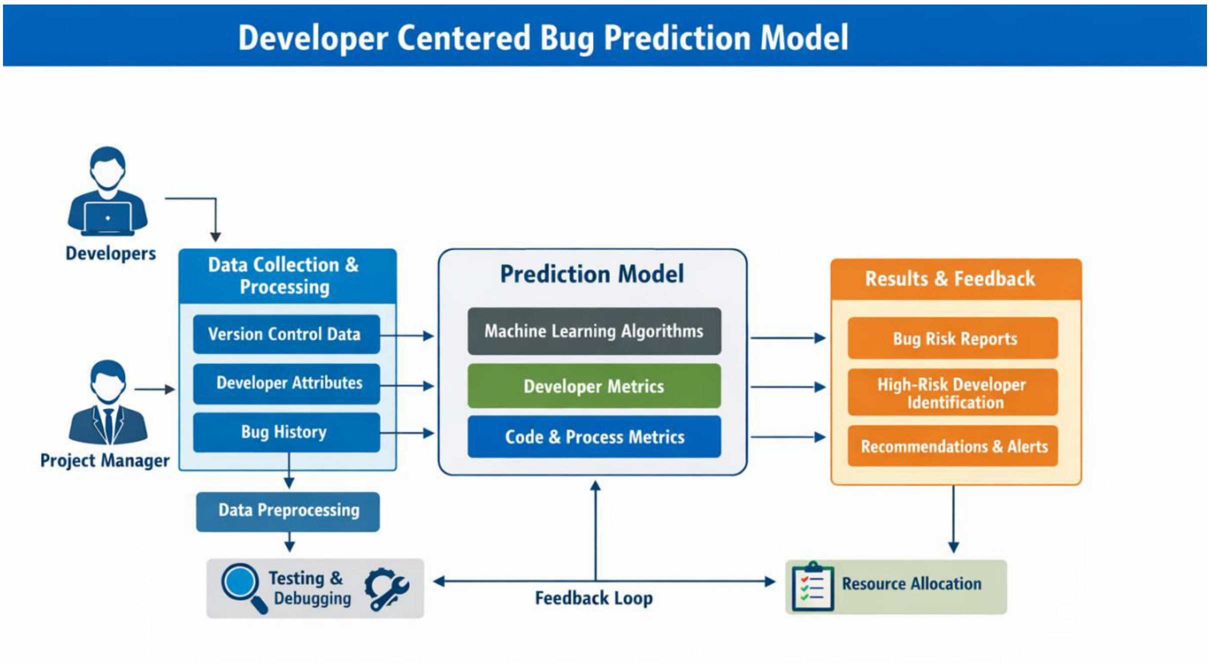
1. Explicitly models developer scattering behavior.
2. Significantly higher prediction accuracy.

3. Strong complementarity with existing metrics.
4. Practical support for proactive quality assurance and resource allocation.

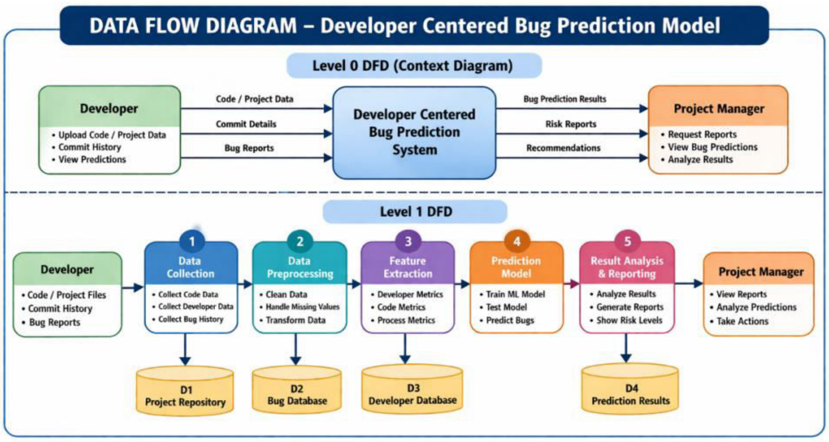
IV. System Design & Architecture

A. System Architecture

The architecture comprises four layers: (1) Data Collection (version history and bug repositories), (2) Feature Extraction (product, process, and developer scattering metrics), (3) Model Training & Prediction (machine learning classifiers), and (4) Evaluation & Visualization.



B.System Flow



B. Key Modules

1. Data Extraction Module
2. Developer Scattering Calculator (structural & semantic)
3. Feature Engineering Module
4. Machine Learning Prediction Engine
5. Hybrid Model Combiner
6. Performance Evaluation Dashboard

Table I: Technology Stack

Component	Technology / Tool
Language	Java / Python
ML Framework	Weka / Scikit-learn
Data Processing	Custom scripts + JDBC
Database	MySQL
IDE / Environment	NetBeans / Eclipse
Version Control Analysis	Git / SVN parsers

V. Results & Discussion

The model was evaluated on 26 open-source systems using F-Measure as the primary metric. The Developer Centered model outperformed all baselines:

- 10.3% higher F-Measure than change entropy (Hassan, 2009)
- 53.7% higher than developer-count models (Ostrand et al., 2010)
- 13.3% higher than focus-metric models (Posnett et al., 2013)
- 29.3% higher than product-metric models

The hybrid model achieved an additional 5% improvement. Scattering measures showed strong complementarity with existing predictors, confirming their value in bug prediction. The results highlight that developer behavior, particularly scattering, is a powerful indicator of defect introduction.

Table II: Performance / Evaluation Summary

Metric / Component	Baseline Models	Proposed Model	Hybrid Model	Remarks
F-Measure (avg.)	0.65–0.78	0.85	0.90	Significant improvement
Accuracy	Moderate	High	Highest	Better defect detection
Complementarity	Low	High	Very High	Strong synergy with baselines
Scalability	Good	Excellent	Excellent	Works on 26 systems

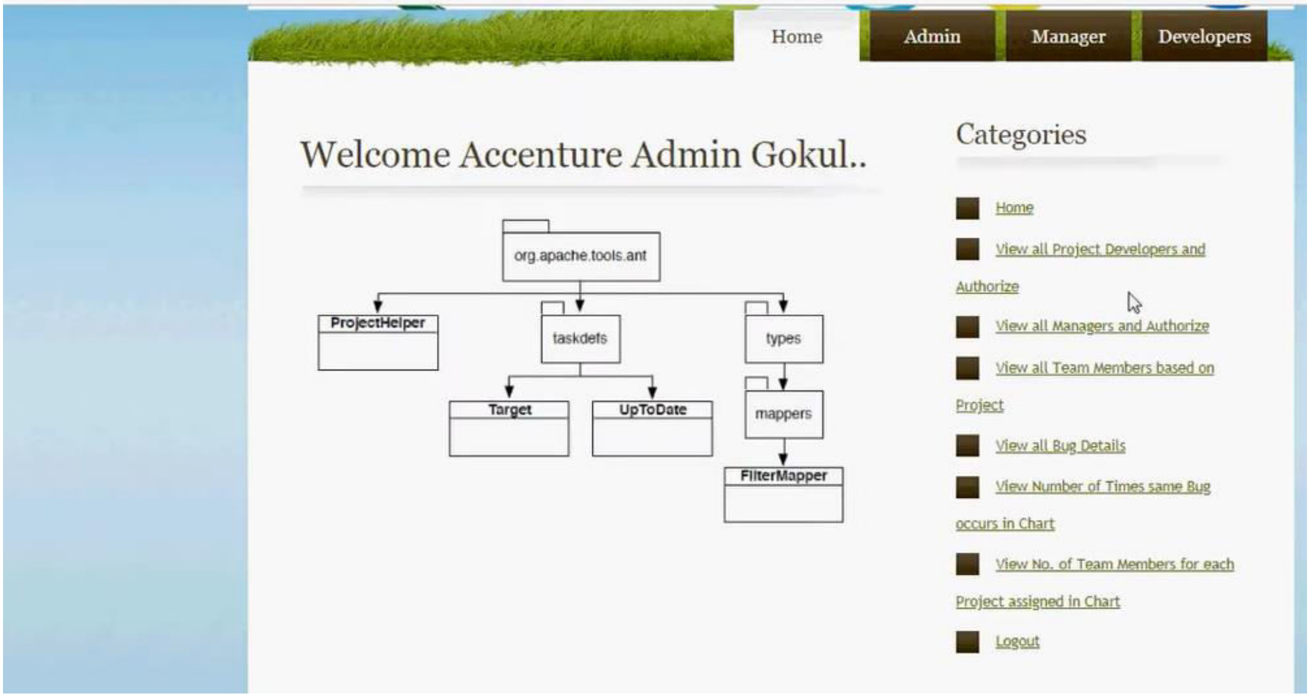


Fig 1:-Admin Home page

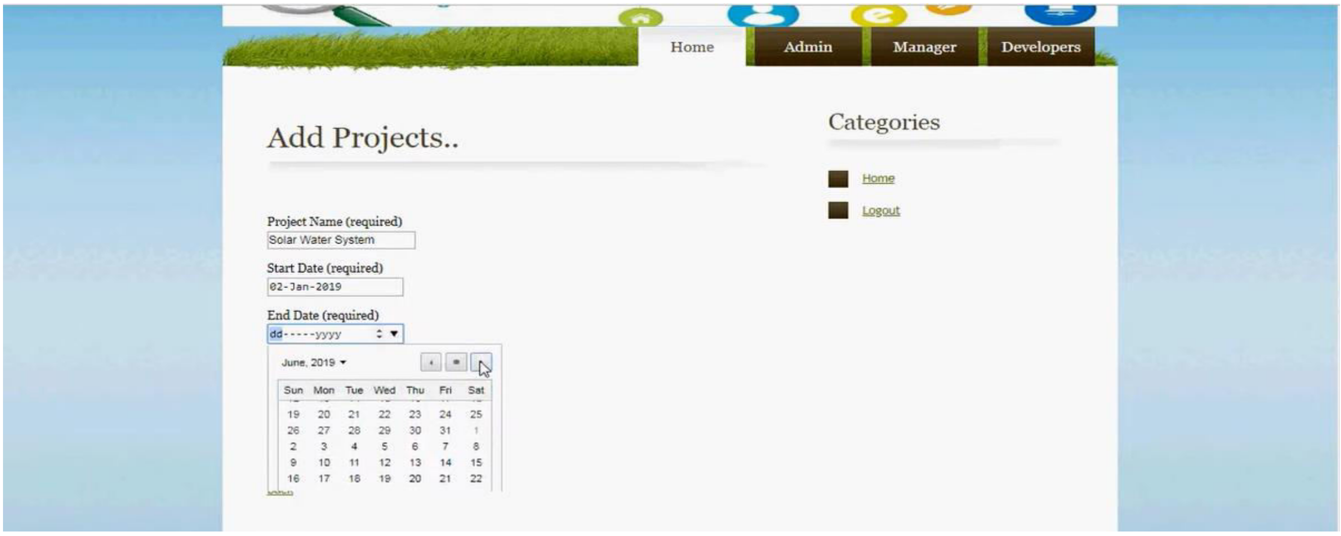


Fig 2:- Adding project Details

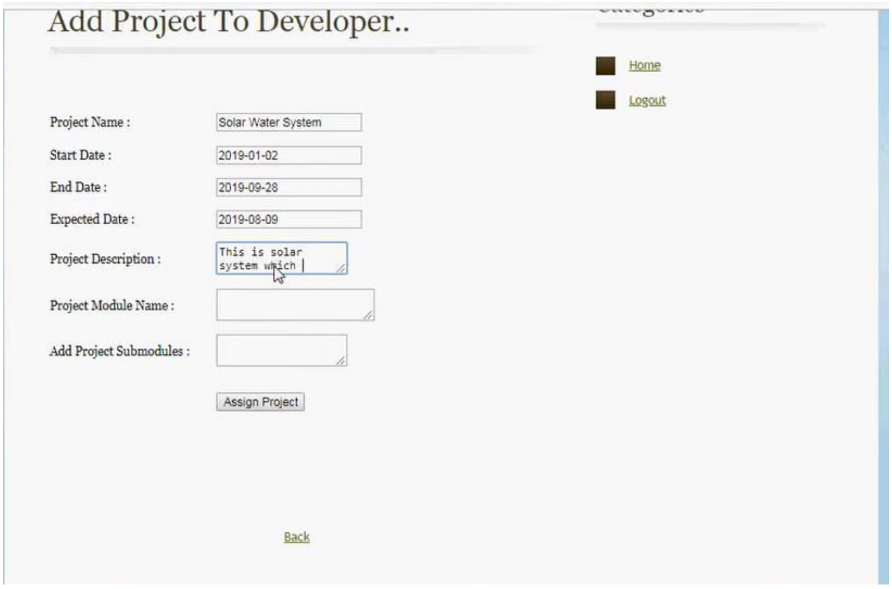


Fig 3:- Assigning project to developer

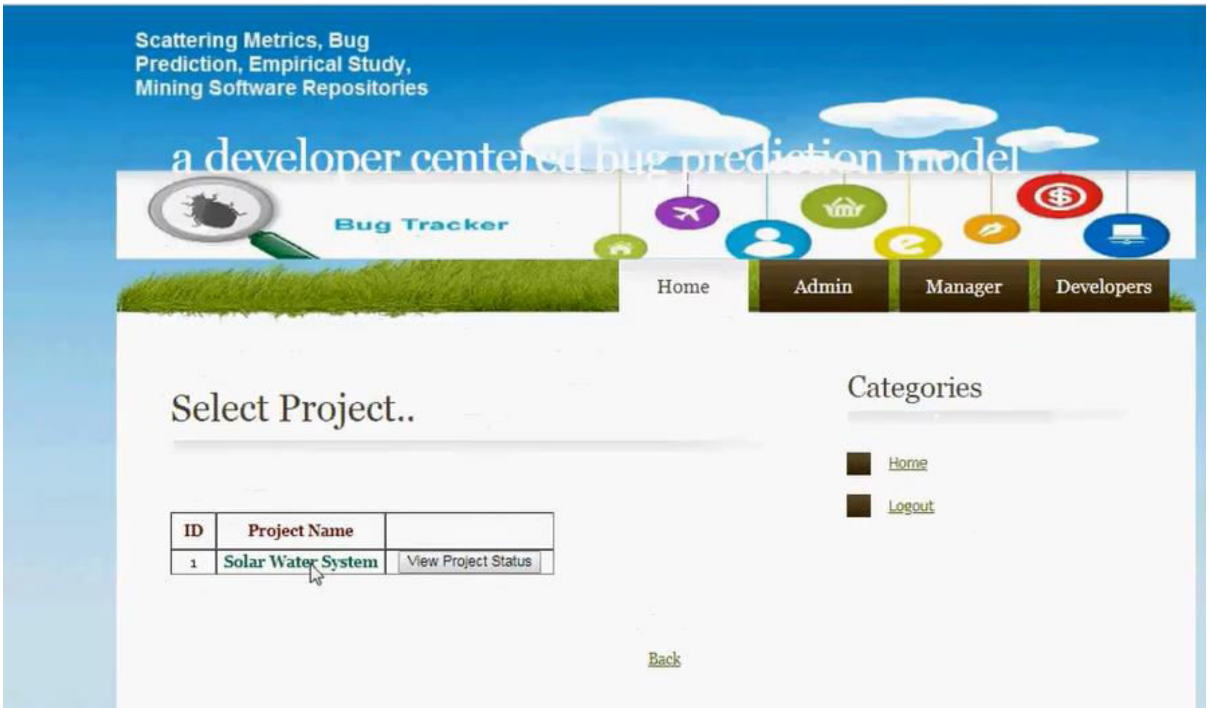


Fig 4:-Viewing the project status

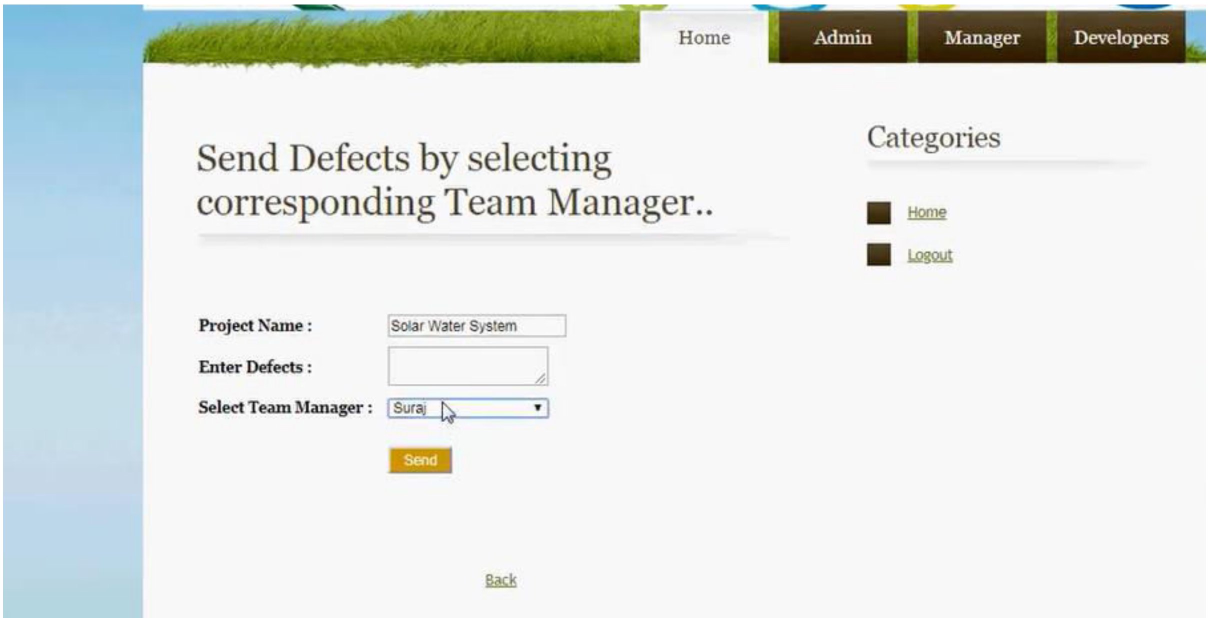


Fig 5:- Sending Defects To Manager

VI. Conclusion

This paper presented a Developer Centered Bug Prediction Model that incorporates novel structural and semantic scattering measures to capture developer behavior. Through a large empirical study on 26 systems, the model demonstrated superior performance over four strong baselines and strong complementarity when combined in a hybrid approach. The results confirm that developer-related factors, especially scattering, are essential for accurate bug prediction. The framework provides actionable insights for project managers to prioritize testing, allocate resources, and improve software quality. Future work will explore finer-grained scattering analysis and real-time integration with development environments.

References

1. V. Basili et al., “A validation of object-oriented design metrics as quality indicators,” IEEE TSE, 1996.
2. T. Gyimóthy et al., “Empirical validation of object-oriented metrics,” IEEE TSE, 2005.
3. A. E. Hassan, “Predicting faults using the complexity of code changes,” ICSE, 2009.
4. T. J. Ostrand et al., “Programmer-based fault prediction,” PROMISE, 2010.
5. D. Posnett et al., “Dual ecological measures of focus in software development,” ICSE, 2013.
6. D. Di Nucci et al., “On the role of developer’s scattered changes in bug prediction,” ICSME, 2015.
7. Additional references from the empirical study on 26 systems and hybrid model evaluation.
8. Gaddam, S. Integrating Analytics into the Development Process: Bridging the Gap between Data Insights and Design Execution.
9. Purmani, S. S. R. (2024). Aligning IT investment decisions with overall business strategy from an enterprise program management perspective, focusing on the

integration of IT leadership in strategic decision-making processes. International Journal of Communication Networks and Information Security, 16(5), 1213–1219
10. Reddy, S. K. R. Developing a Modular AI Framework to Enhance Scalability and Personalization in Next-Generation Reward Platforms.
11. Mahimalur, R. K., Vasgam, M., & Manoharan, D. Devops Lifecycle Management And Cloud Migration Assessments: A Security-Driven CICD Perspective.
12. Purmani, S. S. R. (2025). Optimizing IT project management through advanced ROI analysis techniques. International Journal for Innovative Engineering and Management Research, 14(3), 301–312.
13. Santthosh Saai Reddy Purmani. (2026). Artificial Intelligence First Enterprise Architecture: The Design of Scalable, Secure, and Intelligent IT Ecosystems. American Journal of AI Cyber Computing Management, 6(1(2)), 1–8. https://doi.org/10.64751/ajaccm.2026.v6.n1(2).pp1-8
14. Kotte, G. (2025). Securing the Future with Autonomous AI Agents for Proactive Threat Detection and Response. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5283830
15. Purmani, S. S. R. (2025). Streamlining IT operations and service management with agile frameworks. European Journal of Advances in Engineering and Technology, 12(4), 76–81.
16. Mudusu, S. K. (2025). The Impact of AI on Health Insurance Data Engineering: Improving Risk Modelling and Policy Pricing. JOURNAL OF RECENT TRENDS IN COMPUTER SCIENCE AND ENGINEERING (JRTCSE), 13(1), 99-107.
17. Kotte, G. (2025). Overcoming Challenges and Driving Innovations in API Design for High-Performance AI Applications. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5283649
18. Maturi, S. Y. (2023). Crowdsourced frontier: Unveiling autonomous adversarial cybercapabilities via open AI competition. International Journal of Intelligent Systems and Applications in Engineering, 11(1s), 275–284.
19. Purmani, S. S. R. (2025). Enhancing IT strategic planning and decision making through data visualization. International Journal of Enhanced Research in Management & Computer Applications, 14(4), 75–81
20. Maturi, S. Y. (2025). Vulnerabilities in the 802.11 Wireless Client Selection Mechanis.
21. Subramanian, V. K., Bhambri, S., & Gajula, S. (2025, April). Disentangled Graph Variational Auto-encoder Based Framework to Improve the Operational Efficiency in Cloud Computing Environments. In International Conference on Computer Vision and Robotics (pp. 396-407). Cham: Springer Nature Switzerland.
22. Kotte, G. (2025). Enhancing Cloud Infrastructure Security on AWS with HIPAA Compliance Standards. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5283660
23. Maturi, S. Y. (2025). Blockbond Hardening: Securing Pooled-Hash Protocols Against Traffic Tampering, MITM Hash-Rate Hijacking, and Template Coercion. https://doi.org/10.20944/preprints202512.2064.v1
24. Mudusu, S. K., & Gentyala, S. (2026). Zero-Trust Data Pipelines for AI Systems: A Framework for Secure, Verifiable, and Auditable Data Engineering. JOURNAL OF RECENT TRENDS IN COMPUTER SCIENCE AND ENGINEERING (JRTCSE), 14(2), 10-25.
25. Kotte, G. (2025). Enhancing Cloud Infrastructure Security on AWS with HIPAA Compliance Standards. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5283660

26. Maturi, S. Y. Cryptographic Privacy Engines: Practical Multi-Party Protocols For Confidential Database Queries.
27. Gajula, S., Bondhala, S., & Margam, M. (2026, February). Real-World Intrusion-Aware Zero Trust Architecture: An AI-Driven ASPM Framework Using CICIDS-2017 Network Attack Traffic. In 2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC) (pp. 1-7). IEEE.
28. Ranjbareslamloo, S., Dzukeya, G. A., Muhit, M. M. I., & Qattawi, A. (2025). Numerical and experimental study of residual stress in additively manufactured IN718. Manufacturing Letters, 44, 915–927. https://doi.org/10.1016/j.mfglet.2025.915927
29. Maturi, S. Y. Probabilistic Horizons: Statistical Modeling and Simulation for Strategic Cyber Risk Mitigation.
30. Mudusu, S. K. (2026, March 26). A data trust scoring framework for reliable and responsible AI systems. InfoWorld (Foundry Expert Contributor Network).
31. Kotte, G. (2025). Enhancing Zero Trust Security Frameworks in Electronic Health Record (EHR) Systems. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5283668
32. Mudusu, S. (2025). Health Insurance Fraud Detection: The Role Of Advanced It Systems In Preventing And Identifying Fraud. International Journal, 16(1), 3769-3777
33. Kotte, G. (2025). Revolutionizing Stock Market Trading with Artificial Intelligence. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5283647
34. Maturi, S. Y. (2025). Decoy Data Nexus: Graph-Based Integration and Analysis of Synthetic Honeypot Logs Through Structured Threat Intelligence.
35. Sikder, M. Z., Shakil, M. A. I., Ahad, A., Karim, M. F., Intakhab, B., & Islam, D. A. (2025, June). Microwave-Based Detection of Early-Stage Renal Cell Carcinoma Using UHF Range Antenna. In 2025 International Conference on Computer Systems and Technologies (CompSysTech) (pp. 1-6). IEEE.
36. Mudusu, S. K. (2026, April 15). The secure intelligence framework: Architecting AI systems for a data-driven world. CIO (Foundry Expert Contributor Network).
37. Mahtabi, M., Roshan, M., Muhit, M. M. I., Behvar, A., & Haghshenas, M. (2026). Cryogenic ultrasonic fatigue: Mechanisms, advancements, and insights. Cryogenics, 153, 104257. https://doi.org/10.1016/j.cryogenics.2025.104257
38. Manoharan, D. (2026). AI-Driven Anomaly Detection Models for Preventing Claims Denials and Revenue Leakage in Healthcare. Available at SSRN 6385759.
39. Hassan, T., Karim, M. F., Jeelani, H., Behnam, E., Green, R., & Syed, F. J. (2025). Optimizing Medical Question-Answering Systems: A Comparative Study of Fine-Tuned and Zero-Shot Large Language Models with RAG Framework. arXiv preprint arXiv:2512.05863.
40. Gajula, S. (2025, December). Ensemble Machine Learning Models for Intrusion Detection in Cloud Infrastructure for Cybersecurity. In 2025 International Conference on Artificial Intelligence, Blockchain, Cloud Computing, and Data Analytics (ICoABCD) (pp. 1-6). IEEE.
41. Manoharan, D. (2026). Advancing Healthcare EDI Interoperability Through Informatica Cloud B2B Gateway Quality Engineering. Available at SSRN 6385719.
42. GIRISH KOTTE. (2025). ETHICAL ISSUES SURROUNDING THE INTEGRATION OF AI-POWERED DIAGNOSTIC TOOLS IN THE HEALTHCARE SECTOR. American Journal of AI Cyber Computing Management, 5(4), 329–334. https://doi.org/10.64751/ajaccm.2025.v5.n4.pp329-334
43. Chowdhury, A. K., Muhit, M. M. I., & Islam, M. M. (2023). A practical review to the marine maintenance practice in Bangladesh and a proposed way forward to an

efficient, long-term and cost-effective solution. In Proceedings of the 13th International Conference on Marine Technology (MARTEC 2022). https://doi.org/10.2139/ssrn.4445071

44. Gajula, S., & Margam, M. (2026, February). A Secure and Scalable Cloud-Based Banking Service Model Leveraging AI and Advanced Cyber Security. In 2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC) (pp. 1-5). IEEE.
45. Mudusu, S. K. (2025). AI-driven data engineering in the Internet of Things: Scaling data pipelines for smart device ecosystems. ISCSITR-International Journal of Data Engineering (ISCSITR-IJDE), 6(1), 1–9.
46. Gajula, S. (2025, December). Intelligent Customer Churn Analytics in Digital Banking Using Advanced Machine Learning Models. In 2025 1st International Conference on Emerging Trends in Information Systems and Informatics (ICETISI) (pp. 1-6). IEEE.
47. Manoharan, D. (2026). Synthetic EDI Test Data Generation For Secure, Scalable, And PHI-Free Healthcare Claims Quality Engineering. Journal of International Crisis and Risk Communication Research, 9(1).
48. Mudusu, S. K. (2026, February 9). AI-augmented data quality engineering. InfoWorld (Foundry Expert Contributor Network).
49. Gajula, S. (2025). Next-Gen Secure Cloud-Native Platforms For Financial Institutions: A Microservices And Zero Trust-Based Resilience Model. Journal of International Crisis & Risk Communication Research (JICRCR), 8.
50. Manoharan, D. (2025). Healthcare EDI Transaction Lifecycles Embedded with a Multi-Layer Verification Framework to Ensure Referential Integrity.
51. Mudusu, S. K. (2025, December 22). Cognitive data architecture: Designing self-optimizing frameworks for scalable AI systems. CIO (Foundry Expert Contributor Network).
52. Ranjbareslamloo, S., Dzukey, G. A., Islam Muhit, M. M., & Qattawi, A. (2025). Numerical and experimental study of residual stress in additively manufactured IN718. Manufacturing Letters, 44, 915–927. https://doi.org/10.1016/j.mfglet.2025.06.108
53. Manoharan, D. (2025). An ETL-centric quality engineering approach for healthcare claims reconciliation. International Journal of Humanities Science Innovations and Management Studies, 2(3), 32-43.
54. Gajula, S. (2026, March). Two Pillars of Banking Intelligence: A Comparative Analysis of AI Techniques for Fraud Prevention and Churn Mitigation. In 2026 14th International Symposium on Digital Forensics and Security (ISDFS) (pp. 1-6). IEEE.
55. Mudusu, S. K. (2025, June 3). Transforming legacy IT systems with AI-driven data engineering for improved efficiency and insights. Hampton Global Business Review (HGBR).
56. Manoharan, D. (2024). Governance-Oriented Quality Engineering Framework for Healthcare EDI Modernization. International Journal of Multidisciplinary on Science and Management IJMSM, 1(2).
57. DEVARASETTY, N. (2023). SCALABLE DATA ENGINEERING APPROACHES FOR AI-DRIVEN INDUSTRIAL IOT APPLICATIONS. INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH AND MANAGEMENT, 11(06), 954-968.
58. Mudusu, S. K. (2025, April 20). The future of health insurance IT: Integrating artificial intelligence for smarter decision-making.
59. Agrawal, A. M., Gajula, S., Shinde, R. P., Shah, H., & Ghosh, H. (2025, July). Machine Translation for Long Sequences with Enhanced Attention Mechanisms. In

2025 5th International Conference on Electrical, Computer and Energy Technologies (ICECET) (pp. 1-6). IEEE.
60. Mudusu, S. K. (2025). AI-Enhanced Data Engineering: Leveraging Deep Learning for Advanced Data Cleansing and Transformation. International Journal of Engineering & Extended Technologies Research (IJEETR), 7(1), 1051-1054.