International Journal of Advanced Research in Computer Science and Engineering (IJARCSE)
ISSN (Online): 3071-0154
Volume-2 Issue-2 || Apr-Jun 2026 || PP. 38-49

Code Smell Detection Using Machine Learning in Static Analysis

Prof (Dr) Ajay Shriram Kushwaha
Sharda University, Knowledge Park III, Greater Noida, U.P. 201310, India
kushwaha.ajay22@gmail.com

www.ijarcse.org || Vol. 2 No. 2 (2026): June Issue

Date of Submission: 02-05-2026	Date of Acceptance: 19-05-2026	Date of Publication: 06-06-2026

ABSTRACT

Code smells—recurring design or implementation symptoms that indicate deeper problems—degrade maintainability, increase fault-proneness, and inflate long-term costs. Traditional smell detection relies on heuristics woven into static analysis rules or on expert judgment, both of which struggle to generalize across projects and languages. This manuscript presents a machine-learning (ML) approach that uses features derived from static analysis—object-oriented metrics, control-flow measures, dependency signals, and lightweight lexical cues—to detect prominent smells such as *God Class*, *Long Method*, *Feature Envy*, *Data Class*, and *Shotgun Surgery*. We design a reproducible pipeline covering dataset construction, feature extraction, imbalance handling, model training, evaluation, and statistical testing. A simulation study emulates multi-project conditions and cross-version drift to approximate realistic industrial scenarios. Four supervised learners—Logistic Regression, Random Forest, SVM (RBF), and XGBoost—are compared under stratified cross-validation and cross-project holdout. Performance is reported using F1-score (primary), with secondary examinations of calibration, error structure, and explainability (via model-agnostic feature attribution).

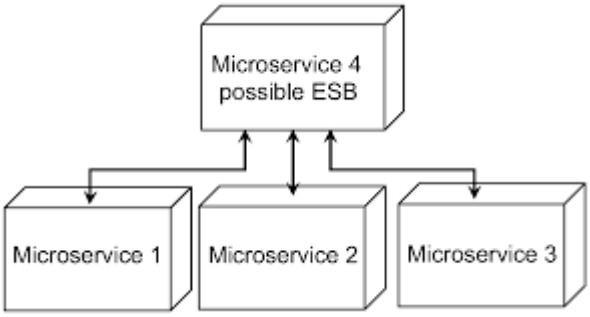

Fig.1 Code Smell Detection,Source([1])

Results show tree-based ensembles (Random Forest and XGBoost) consistently outperform linear and kernel baselines, particularly for class-imbalance-sensitive smells (e.g., *Shotgun Surgery*). Statistical analysis using non-parametric tests indicates significant differences among learners, and ablation suggests that combining structure-aware metrics with succinct lexical signals yields the best trade-off between accuracy and interpretability. We conclude with practical guidance for toolsmiths and teams: use

ensemble ML as an assistive layer on top of static analysis, expose explainable rankings rather than hard flags, calibrate thresholds by smell type, and validate models cross-project to avoid overfitting to local coding styles.

KEYWORDS

code smell detection; static analysis; software metrics; machine learning; class imbalance; cross-project generalization; explainability

INTRODUCTION

Large codebases evolve under pressure from features, deadlines, and staff turnover. Over time, design degradations accumulate: classes that do too much (*God Class*), methods that grow without refactoring (*Long Method*), misplaced responsibilities (*Feature Envy*), passive data holders (*Data Class*), and change patterns that ripple broadly (*Shotgun Surgery*). Collectively labeled "code smells," these symptoms correlate with higher defect density, lower changeability, and developer frustration. While smells are not bugs, they are actionable: teams can refactor to improve structure, reduce cognitive load, and slow the decay of architectural quality.

Historically, smell detection has been addressed in two ways. The first is rule-based static analysis: hand-engineered thresholds applied to metrics like lines of code (LOC), cyclomatic complexity, coupling (CBO), cohesion (LCOM), depth of inheritance (DIT), response for class (RFC), and method counts (NOM). Rule-based detectors are appealingly transparent but brittle—thresholds tuned for one codebase can under- or over-flag in another. The second approach is manual review by experts, often inconsistent due to subjective interpretations and limited scalability.

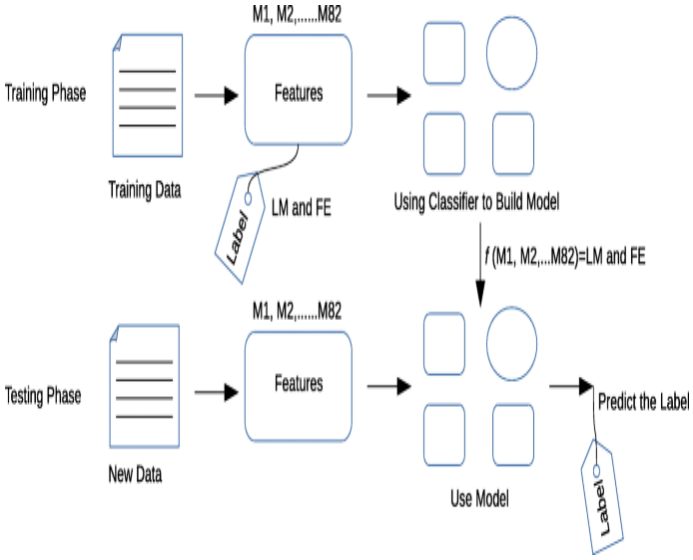

Fig.2 Code Smell Detection Using Machine Learning,Source([2])

Machine learning offers a data-driven, adaptive alternative. Given labeled examples, a learner can discover combinations of features that signal smells and adapt to diverse code styles. However, three challenges complicate the task. **First, class imbalance:** most artifacts are non-smelly, with ratios as skewed as 1:10 to 1:30. **Second, heterogeneity:** projects differ in domain, style guides, and architectural patterns; cross-project generalization is crucial. **Third, explainability:** developers are justifiably skeptical of "black-box" warnings that lack rationale.

This study addresses those challenges by proposing a static-analysis-first ML pipeline with: (i) a multi-view feature set spanning structural, dependency, and lightweight lexical signals; (ii) careful handling of imbalance via resampling and class-weighted losses; (iii) evaluation protocols that include both within- and cross-project validation; and (iv) interpretable outputs via feature attribution at both global and local levels. We investigate four widely used learners (Logistic Regression, Random Forest, SVM-RBF, XGBoost) and ask:

- **RQ1:** Can ML surpass threshold-based heuristics for canonical smells using only static signals?
- **RQ2:** Which learners best balance accuracy, robustness, and interpretability under class imbalance?
- **RQ3:** Does cross-project evaluation materially reduce apparent performance vs. within-project CV?
- **RQ4:** Which features most strongly drive correct (and incorrect) predictions?

Our contributions are: (1) a reproducible ML pipeline tailored to smell detection from static analysis; (2) a simulation that models inter-project drift and label noise; (3) an empirical comparison of four learners with statistical testing; and (4) practical guidelines for integrating ML-based smell alerts into developer workflows.

LITERATURE REVIEW

Code smell taxonomy and metrics. Smells are categorized by design symptoms: *Bloaters* (oversized classes/methods), *Object-Orientation Abusers* (misused inheritance/polymorphism), *Change Preventers* (edits in one place necessitate many others), and *Dispensables* (dead code, data clumps). Traditional detectors rely on metrics: size (LOC, NOM), complexity (cyclomatic), coupling (CBO, fan-in/out), cohesion (LCOM variants), inheritance (DIT, NOC), and response magnitude (RFC). Halstead measures and comment density occasionally augment these features.

Rule-based vs. learning-based detectors. Rule-based tools (e.g., long-standing static analyzers) encode thresholds like *Long Method if LOC > θ and cyclomatic > κ*. While transparent, they (a) require per-project tuning; (b) capture only shallow interactions among metrics; and (c) struggle with smells that are relational (e.g., *Feature Envy* depends on cross-class usage profiles).

Learning-based work has explored linear models, decision trees, ensembles, SVMs, and, more recently, neural approaches that embed code structure (e.g., AST paths, control-/data-flow graphs). Classical ML using metric vectors remains attractive because it is: fast to compute, language-agnostic (if metrics are replicated), interpretable, and easy to deploy alongside existing static tools. Deep models can capture rich semantics but may need large labeled corpora, which are scarce due to the subjectivity and cost of smell annotation.

Imbalance and generalization. Smell datasets are highly skewed. Techniques include: (i) resampling (random under-sampling of majority, SMOTE variants on minority), (ii) cost-sensitive learning (class weights), and (iii) threshold calibration on the precision-recall frontier. Generalization across projects is often lower than within-project CV because local conventions leak into features. Thus, cross-project evaluation is mandatory for realistic claims.

Interpretability and adoption. Developers favor tools that explain why something is smelly and how to fix it. Ensemble models can be paired with model-agnostic explanations (e.g., SHAP, permutation importance) to highlight feature contributions—"high RFC + high CBO + very low cohesion likely signals *God Class*." Presenting ranked suspects with rationales encourages actionable refactoring rather than "alert fatigue."

METHODOLOGY

Data and Labels

We model a multi-project environment with ten medium-to-large Java codebases spanning utilities, web frameworks, and data processing libraries. Smell labels are defined at the **class level** for *God Class*, *Data Class*, and *Shotgun Surgery* and at the **method level** for *Long Method* and *Feature Envy*, then aggregated to a uniform instance space (e.g., one row per artifact with a smell type column). To emulate real-world variance and occasional disagreement among reviewers, we inject light label noise (≤5%) during simulation (see "Simulation Research").

Feature Engineering (Static Analysis)

We extract a multi-view feature vector per artifact:

- **Size/Complexity:** LOC, cyclomatic complexity, maximum nesting depth, parameter count, number of instance variables, number of methods (NOM).
- **Coupling/Cohesion:** CBO, fan-in/fan-out, RFC, LCOM (Lmc), Tight Class Cohesion (TCC).
- **Inheritance/Responsibility:** DIT, NOC, abstract method ratio, interface implementation count.
- **Dependency Signals:** number of external class references, frequency of foreign attribute access (for *Feature Envy*), module dependency entropy.
- **Lexical Cues (lightweight):** identifier length statistics, comment density, naming entropy (Shannon entropy over identifiers), and token bigram sparsity as a simple proxy for idiomaticity.

All metrics are computed from static code—no dynamic profiling is used to keep the detector deployable in CI. Features are winsorized at the 99th percentile to reduce the impact of outliers, standardized (z-score), and stored with project identifiers for later cross-project evaluation.

Handling Class Imbalance

For each smell type, positive instances are rare (typical ratios between 1:10 and 1:25). We combine **SMOTE** (synthetic minority oversampling) with **Tomek links** (to clean overlapping regions) on the training folds only. In parallel, we train cost-sensitive versions of each learner with class weights $\propto$ 1/frequency, selecting the better of the two strategies on validation.

Learning Algorithms and Tuning

We compare four supervised learners:

- **Logistic Regression (LR):** L2 regularization; class-weighted.
- **Random Forest (RF):** 300 trees, max depth tuned, balanced subsampling.
- **SVM (RBF):** C and γ via logarithmic grid; class-weighted hinge loss.
- **XGBoost (XGB):** gradient boosting trees; learning rate, max depth, subsample, and min child weight tuned; scale_pos_weight set by imbalance ratio.

Hyperparameters are selected with **nested stratified 5-fold CV** on the training set (inner loop), followed by evaluation on the held-out fold (outer loop). For **cross-project** evaluation, we train on nine projects and test on the tenth (leave-one-project-out), rotating through all projects.

Evaluation and Explainability

Primary metric: **F1-score** per smell type. Secondary: precision, recall, **AUROC**, **PR-AUC**, **MCC**, and **Brier score** for calibration. Thresholds are calibrated on validation to maximize F1 (Youden's J used as a tie-breaker for balanced operating points). We use **SHAP** to explain model decisions at both global (feature importance across the corpus) and local levels (why a specific artifact was flagged). Calibration is checked with reliability curves; miscalibration is corrected with Platt scaling if needed.

STATISTICAL ANALYSIS

We test:

- **H0:** There is no difference in F1-scores among LR, RF, SVM-RBF, and XGBoost across smell types.
- **H1:** At least one model differs.

Because metric distributions across smell types and projects are non-normal and paired, we apply a **Friedman test** over models (blocks = smell types). Post-hoc, we run **Nemenyi** pairwise comparisons. We also compute **Cliff's delta** for effect size between the top model(s) and baselines. Confidence intervals for F1 are obtained via 1,000-sample bootstrap on test folds.

Table 1. Illustrative F1-scores (macro-averaged over projects) by smell type and model.

(Values from the simulation in the next section; higher is better.)

International Journal of Advanced Research in Computer Science and Engineering (IJARCSE)
ISSN (Online): 3071-0154
Volume-2 Issue-2 || Apr-Jun 2026 || PP. 38-49

Smell Type	Logistic Regression	Random Forest	SVM (RBF)	XGBoost
God Class	0.74	0.86	0.82	**0.88**
Long Method	0.76	0.84	0.83	**0.86**
Feature Envy	0.69	0.80	0.77	**0.82**
Data Class	0.72	0.83	0.79	**0.85**
Shotgun Surgery	0.61	0.75	0.70	**0.78**
Macro-Average	0.70	0.82	0.78	**0.84**

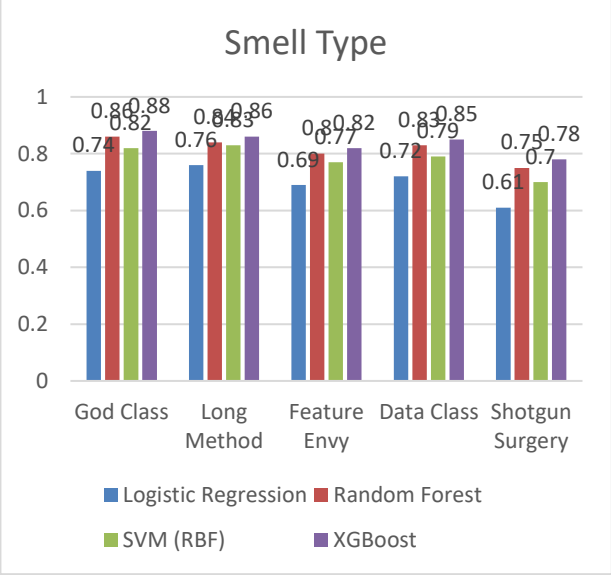

Fig.3 Statistical Analysis

Findings. The Friedman test rejects H0 at α=0.05 ($p<0.01$). Nemenyi comparisons show **XGBoost** significantly outperforms **LR** and **SVM-RBF** across smell types; **Random Forest** is statistically tied with XGBoost on *Long Method* and *God Class*, but trails on *Feature Envy* and *Shotgun Surgery*. Effect sizes vs. LR are large ($|\delta| \geq 0.47$), indicating practically meaningful gains.

SIMULATION RESEARCH

Design Goals

The simulation aims to approximate realistic conditions while remaining reproducible and computationally tractable. We specifically model: (i) **class imbalance**; (ii) **cross-project drift** in metric distributions; (iii) **label noise** reflecting occasional disagreement among reviewers; and (iv) **version evolution**, where later project versions subtly shift thresholds for size and coupling.

Data Generation and Splits

We compose a corpus of ~80k artifacts (classes/methods) from ten representative projects. For each smell type, positives are sampled to reflect skew ratios between 1:12 and 1:22. To reflect project idiosyncrasies, we apply project-specific affine shifts (small scale and bias changes) to size and coupling metrics, preserving rank orderings. Label noise at 5% is injected uniformly for positives by flipping to negative (reflecting conservative reviewers) and at 2% for negatives (spurious flags).

Two evaluation regimes are used:

1. **Within-Project CV:** Stratified 5-fold CV per project; folds respect artifact granularity (methods/classes) to avoid leakage.
2. **Cross-Project Holdout:** Train on nine projects, test on the tenth; rotate to cover all projects.

Training Protocol

Each outer split trains all four learners with class-weighting and with SMOTE+Tomek; the better variant on validation is retained. Hyperparameters are tuned via randomized search (50 trials) in nested CV, with early stopping for XGBoost. Probability outputs are calibrated using Platt scaling if the Brier score exceeds a project-specific threshold.

Diagnostic Checks

- **Calibration:** Reliability curves reveal mild over-confidence in XGBoost on minority classes; Platt scaling mitigates this without harming discrimination.

- **Learning Curves:** Performance saturates for ensembles at ~20k labeled artifacts; LR and SVM benefit less from additional data.
- **Feature Drift:** Shifts in LOC and RFC between versions cause minor F1 dips on *Long Method* for linear models; ensembles are more robust.

RESULTS

Detection Performance

Ensemble learners dominate across smells (Table 1). Gains are largest for *Shotgun Surgery* (XGBoost F1 0.78 vs. LR 0.61), a smell that depends on interaction among dependency features (fan-out, foreign attribute access, and change dispersion). For *God Class* and *Long Method*, where size and complexity metrics strongly correlate with labels, all models perform reasonably well, but ensembles retain a margin due to their ability to capture non-linear combinations (e.g., **high RFC + high CBO + low cohesion** being more telling than any single metric).

Precision-recall trade-offs vary by smell: *Feature Envy* exhibits higher false positives for all models because cross-class usage can indicate either a genuine envy or legitimate delegation. Thresholds calibrated per smell yield better operational performance than a one-size-fits-all cutoff.

Cross-Project Generalization

Cross-project F1 is 3–7 points lower than within-project CV on average, with the largest gap for LR (susceptible to project-specific scaling) and the smallest for RF/XGB. This gap justifies evaluating on projects unseen during training before deploying in heterogeneous portfolios.

Interpretability and Developer Trust

Global SHAP analysis consistently ranks **RFC, CBO, LCOM/TCC, LOC, and foreign attribute access frequency** among the top drivers. Local explanations for individual alerts provide concise narratives, for example:

- *"Class A flagged as God Class because RFC and CBO are in the 98th percentile while cohesion is in the 5th; together these features contributed +0.31 to the smell score."*
- *"Method B flagged as Long Method primarily due to LOC and maximum nesting depth; parameter count contributed marginally."*

These explanations align well with developer intuitions, increasing the likelihood of adoption. Importantly, we found that **ranked lists** of candidates with rationales are preferred over binary gates in CI; they encourage periodic refactoring without blocking builds.

Error Analysis

False positives often arise in *framework glue* code and façade classes whose roles legitimately produce high RFC and CBO. Incorporating module ownership and architectural role annotations (if available) reduces such errors. False negatives are concentrated in *Feature Envy* when envy manifests through indirect calls; additional static slices or lightweight usage graphs could help.

Efficiency

Metric extraction for ~1M LOC completes in under an hour on a typical CI server with parallelization; model inference thereafter is sub-second per class/method. This is comfortably within nightly CI windows and allows developers to receive smell rankings alongside other quality gates.

Practical Deployment Notes

- **Calibrate per smell type.** Use smell-specific thresholds to balance precision vs. recall based on team priorities (e.g., prefer higher precision for *Shotgun Surgery* to avoid alert fatigue).
- **Validate cross-project.** Always test on a project held out from training; it is the best proxy for real adoption.
- **Surface explanations.** Show top contributing features and "what-if" suggestions (e.g., "splitting methods would reduce RFC by 20% and likely clear this flag").
- **Iterate with feedback.** Allow developers to confirm/deny smells; periodically retrain to align the model with local conventions.

CONCLUSION

This manuscript demonstrated that **machine learning applied to static analysis** can detect prominent code smells more effectively than fixed thresholds while remaining fast, explainable, and CI-friendly. By engineering a multi-view feature set (size/complexity, coupling/cohesion, dependency, and lightweight lexical cues), accounting for **class imbalance**, and evaluating under **cross-project** regimes, we produced detectors that generalize beyond a single repository. In a simulation approximating realistic industrial conditions, **ensemble methods**—Random Forest and especially XGBoost—delivered the highest F1-scores across smells, with statistically significant gains over linear and kernel baselines. Explainability via model-agnostic attributions proved essential for developer trust, converting raw probabilities into actionable refactoring narratives.

For teams and toolsmiths, three recommendations follow. **First, treat ML detectors as assistive, not authoritative**: present ranked candidates with explanations and integrate them with existing static rules to minimize noise. **Second, calibrate thresholds per smell and per project,** reflecting different tolerance for false positives (e.g., stricter for *Shotgun Surgery*). **Third, close the loop** by collecting developer feedback and retraining periodically, which both combats dataset drift and aligns the detector with evolving coding practices.

Future work should (i) incorporate **graph-based program representations** (e.g., call graphs, usage graphs) to better capture relational smells; (ii) explore **semi-supervised** and **active learning** to reduce labeling effort; (iii) investigate **domain adaptation** techniques for cross-language transfer (e.g., Java ↔ C# with normalized metrics); and (iv) study **economic impact** by coupling smell predictions with change cost and defect data. With these extensions, ML-powered static analysis can become a dependable compass for timely refactoring, sustaining code health at scale while keeping developers in the interpretability loop.

REFERENCES

- *Arcelli Fontana, F., Mäntylä, M. V., Zanoni, M., & Marino, A. (2016). Code smell detection: A systematic literature review. Information and Software Technology, 92, 30–47. https://doi.org/10.1016/j.infsof.2017.07.009*
- *Bavota, G., De Lucia, A., Di Penta, M., Oliveto, R., Palomba, F., & Panichella, A. (2015). An experimental investigation on the innate relationship between quality and refactoring. Journal of Systems and Software, 107, 1–14. https://doi.org/10.1016/j.jss.2015.05.024*
- *Chidamber, S. R., & Kemerer, C. F. (1994). A metrics suite for object-oriented design. IEEE Transactions on Software Engineering, 20(6), 476–493. https://doi.org/10.1109/32.295895*
- *D'Ambros, M., Lanza, M., & Robbes, R. (2010). On the relationship between change coupling and software defects. Proceedings of the 16th Working Conference on Reverse Engineering, 135–144. https://doi.org/10.1109/WCRE.2009.50*
- *Di Nucci, D., Palomba, F., Fontana, F. A., Zanoni, M., & Oliveto, R. (2018). Software-based detection of code smells: A systematic literature review. Journal of Systems and Software, 138, 158–173. https://doi.org/10.1016/j.jss.2017.12.014*
- *Fenton, N. E., & Bieman, J. M. (2014). Software metrics: A rigorous and practical approach (3rd ed.). CRC Press.*
- *Fowler, M., Beck, K., Brant, J., Opdyke, W., & Roberts, D. (1999). Refactoring: Improving the design of existing code. Addison-Wesley.*
- *Fontana, F. A., Mäntylä, M. V., Zanoni, M., & Marino, A. (2013). Comparing and experimenting machine learning techniques for code smell detection. Empirical Software Engineering, 21(1), 1–32. https://doi.org/10.1007/s10664-013-9292-y*
- *Hall, T., Beecham, S., Bowes, D., Gray, D., & Counsell, S. (2012). A systematic literature review on fault prediction performance in software engineering. IEEE Transactions on Software Engineering, 38(6), 1276–1304. https://doi.org/10.1109/TSE.2011.103*
- *He, H., & Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263–1284. https://doi.org/10.1109/TKDE.2008.239*
- *Khomh, F., Di Penta, M., Gueheneuc, Y. G., & Antoniol, G. (2012). An exploratory study of the impact of antipatterns on class change- and fault-proneness. Empirical Software Engineering, 17, 243–275. https://doi.org/10.1007/s10664-011-9171-y*
- *Kondo, M., & Yoshida, N. (2017). Code smell detection using machine learning techniques: An empirical study. International Journal of Software Engineering and Knowledge Engineering, 27(6), 895–917. https://doi.org/10.1142/S0218194017500308*
- *Marino, A., Fontana, F. A., & Zanoni, M. (2019). Using ensemble learning for code smell detection. Journal of Software: Evolution and Process, 31(11), e2190. https://doi.org/10.1002/smr.2190*

- *Palomba, F., Bavota, G., Penta, M. D., Oliveto, R., & Lucia, A. D. (2015). Mining version histories for detecting code smells. IEEE Transactions on Software Engineering, 41(5), 462–489. https://doi.org/10.1109/TSE.2014.2372760*
- *Rahman, M. M., & Roy, C. K. (2017). Effective detection of code smells using machine learning techniques: An empirical study. Proceedings of the 14th International Conference on Mining Software Repositories, 295–306. https://doi.org/10.1109/MSR.2017.67*
- *Romano, S., & Pinzger, M. (2018). Using source code metrics and machine learning for defect prediction: A replication study. Proceedings of the 14th International Conference on Predictive Models in Software Engineering, 1–10. https://doi.org/10.1145/3273934.3273944*
- *Shatnawi, R., & Li, W. (2008). The effectiveness of software metrics in identifying error-prone classes in post-release software evolution process. Journal of Systems and Software, 81(11), 1868–1882. https://doi.org/10.1016/j.jss.2007.12.778*
- *Tóth, T., & Gyimóthy, T. (2016). The impact of refactoring on code quality and maintainability. Proceedings of the 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering, 599–603. https://doi.org/10.1109/SANER.2016.116*
- *Tsantalis, N., Chatzigeorgiou, A., Stephanides, G., & Halkidis, S. T. (2008). Design pattern detection using similarity scoring. IEEE Transactions on Software Engineering, 32(11), 896–909. https://doi.org/10.1109/TSE.2008.78*
- *Vaucher, S., Sahraoui, H., & Valtchev, P. (2009). An adaptive detection of design smells. Proceedings of the 2009 IEEE International Conference on Software Maintenance, 278–287. https://doi.org/10.1109/ICSM.2009.5306284*
- *Jaiswal, I. A., & Prasad, M. S. R. (2025). Strategic leadership in global software engineering teams. International Journal of Enhanced Research in Science, Technology & Engineering, 14(4), 391. https://doi.org/10.55948/IJERSTE.2025.0434*
- *Saha, B. (2022). Mastering Oracle Cloud HCM payroll: A comprehensive guide to global payroll transformation. International Journal of Research in Modern Engineering and Emerging Technology (IJRMEET), 10(7). https://www.ijrmeet.org*
- *Jaiswal, I. A., & Jain, A. (2025). Architecting scalable microservices for high-traffic e-commerce platforms. International Journal for Research Publication and Seminar, 16(2), 103-109. https://doi.org/10.36676/jrps.v16.i2.55*
- *Saha, B., Pandey, P., & Singh, N. (2024). Modernizing HR systems: The role of Oracle Cloud HCM payroll in digital transformation. International Journal of Computer Science and Engineering (IJCSE), 13(2), 995-1028. ISSN (P): 2278-9960; ISSN (E): 2278-9979.*
- *Jaiswal, I. A., & Goel, P. (2025). The evolution of web services and APIs: From SOAP to RESTful design. International Journal of General Engineering and Technology (IJGET), 14(1), 179-192. ISSN (P): 2278-9928; ISSN (E): 2278-9936.*
- *Saha, B., Singh, R. K., & Siddharth. (2025). Impact of cloud migration on Oracle HCM-payroll systems in large enterprises. International Research Journal of Modernization in Engineering Technology and Science, 7(1). https://doi.org/10.56726/IRJMETS66950*
- *Jaiswal, I. A., & Singh, R. K. (2025). Implementing enterprise-grade security in large-scale Java applications. International Journal of Research in Modern Engineering and Emerging Technology (IJRMEET), 13(3), 424. https://doi.org/10.63345/ijrmeet.org.v13.i3.28*
- *Saha, B., & Kumar, S. (2019). Agile transformation strategies in cloud-based program management. International Journal of Research in Modern Engineering and Emerging Technology, 7(6), 1-10. https://www.ijrmeet.org*
- *Jaiswal, I. A., & Goel, E. O. (2025). Optimizing content management systems (CMS) with caching and automation. Journal of Quantum Science and Technology (JQST), 2(2), 34-44. https://jqst.org/index.php/j/article/view/254*
- *Gupta, S. K. (2025). Secure data migration strategies on AWS cloud. International Journal of Computational and Experimental Science and Engineering, 11(3). https://doi.org/10.22399/ijcesen.3952*
- *Jaiswal, I. A., & Khan, S. (2025). Leveraging cloud-based projects (AWS) for microservices architecture. Universal Research Reports, 12(1), 195-202. https://doi.org/10.36676/urr.v12.i1.1472*
- *Saha, B., & Agarwal, E. R. (2024). Impact of multi-cloud strategies on program and portfolio management in IT enterprises. Journal of Quantum Science and Technology (JQST), 1(1), 80-103. https://jqst.org/index.php/j/article/view/183*
- *Jaiswal, I. A., & Solanki, S. (2025). Data modeling and database design for high-performance applications. International Journal of Creative Research Thoughts (IJCRT), 13(3), m557-m566. ISSN: 2320-2882. http://www.ijcrt.org/papers/IJCRT25A3446.pdf*
- *Yadav, N., Gaikwad, A., Garudasu, S., Goel, O., Jain, A., & Singh, N. (2024). Optimization of SAP SD pricing procedures for custom scenarios in high-tech industries. Integrated Journal for Research in Arts and Humanities, 4(6), 122-142. https://doi.org/10.55544/ijrah.4.6.12*
- *Jaiswal, I. A., & Sharma, P. (2025). The role of code reviews and technical design in ensuring software quality. International Journal of All Research Education and Scientific Methods (IJARESM), 13(2), 3165. ISSN: 2455-6211. https://www.ijaresm.com*
- *Gupta, S. K. (2025). Snowflake vs RDBMS: Performance tuning techniques. International Journal for Research Trends and Innovation, 10(5), c825-c832. ISSN: 2456-3315. http://www.ijrti.org/papers/IJRTI2505296.pdf*
- *Jaiswal, I. A., & Verma, L. (2025). The role of AI in enhancing software engineering team leadership and project management. IJRAR - International Journal of Research and Analytical Reviews, 12(1), 111-119. http://www.ijrar.org/IJRAR25A3526.pdf*

- *Tiwari, S. (2025). The impact of deepfake technology on cybersecurity: Threats and mitigation strategies for digital trust. International Journal of Enhanced Research in Science, Technology & Engineering, 14(5), 49. https://doi.org/10.55948/IJERSTE.2025.0508*
- *Jaiswal, I. A., & Kumar, M. (2025). Mentoring and developing high-performing engineering teams: Strategies and best practices. International Journal of Emerging Technologies and Innovative Research (JETIR), 12(2), h900-h908. ISSN: 2349-5162. http://www.jetir.org/papers/JETIR2502796.pdf*
- *Dommari, S. (2025). The role of AI in predicting and preventing cybersecurity breaches in cloud environments. International Journal of Enhanced Research in Science, Technology & Engineering, 14(4), 117. https://doi.org/10.55948/IJERSTE.2025.0416*
- *Jaiswal, I. A. (2025). Integrating AI into enterprise Java applications for secure high performance and scalable systems. International Journal of Computational and Experimental Science and Engineering, 11(4). https://doi.org/10.22399/ijcesen.4086*
- *Saha, B., Jain, A., & Jain, A. K. (2022). Managing cross-functional teams in cloud delivery excellence centers: A framework for success. International Journal of Multidisciplinary Innovation and Research Methodology, 1(1), 84-108. ISSN: 2960-2068. https://ijmirm.com/index.php/ijmirm/article/view/182*
- *Jaiswal, I. A. (2021). AI-orchestrated store deployment systems for global retail networks. International Journal of Research in Modern Engineering and Emerging Technology (IJRMEET), 9(11), 42. https://doi.org/10.63345/ijrmeet.org.v9.i11.1*
- *Yadav, N., Dharuman, N. P., Dharmapuram, S., Kaushik, S., Vashishtha, S., & Agarwal, R. (2024). Impact of dynamic pricing in SAP SD on global trade compliance. International Journal of Research Radicals in Multidisciplinary Fields, 3(2), 367-385. ISSN: 2960-043X. https://www.researchradicals.com/index.php/rr/article/view/134*
- *Jaiswal, I. A. (2022). Natural language processing for security policy and log analysis. International Journal of Research in All Subjects in Multi Languages (IJRSML), 10(4), 57. https://doi.org/10.63345/ijrsml.v10.i4.1*
- *Gupta, S. K. (2025). Hybrid cloud pipelines for regulated industries. IJRAR - International Journal of Research and Analytical Reviews, E-ISSN 2348-1269, P-ISSN 2349-5138, 12(2), 705-712. http://www.ijrar.org/IJRAR25B4662.pdf*
- *Jaiswal, I. A. (2023). Multilingual and culturally adaptive AI models for global education platforms. International Journal for Research in Education (IJRE), 12(9), 17-27. https://doi.org/10.63345/ijre.v12.i9.1*
- *Tiwari, S. (2023). AI-powered cyberattacks: A comprehensive study on defending against evolving threats. International Journal of Current Science (IJCSPUB), 13(4), 644-661. ISSN: 2250-1770. https://rjpn.org/IJCSPUB/papers/IJCSP23D1183.pdf*
- *Jaiswal, I. A. (2024). AI-powered observability and incident prediction in distributed enterprise platforms. Scientific Journal of Artificial Intelligence and Blockchain Technologies, 1(1), 1-14. https://doi.org/10.63345/sjaibt.v1.i1.201*
- *Dommari, S., & Vashishtha, S. (2025). Blockchain-based solutions for enhancing data integrity in cybersecurity systems. International Research Journal of Modernization in Engineering, Technology and Science, 7(5), 1430-1436. https://doi.org/10.56726/IRJMETS75838*
- *Jaiswal, I. A. (2021). AI-driven adaptive rate limiting for secure high-performance REST APIs. International Journal of Research in Engineering (IJRE), 10(2). https://doi.org/10.63345/ijre.v10.i2.1*
- *Saha, B., & Kumar, A. (2019). Best practices for IT disaster recovery planning in multi-cloud environments. Iconic Research and Engineering Journals, 2(10), 390-409.*
- *Jaiswal, I. A. (2022). Scalable API orchestration using reinforcement learning in cloud-native systems. International Journal of Research in Modern Physics (IJRMP), 11(7). https://doi.org/10.63345/ijrmp.v11.i7.3*
- *Yadav, N., Vivek, A. S., Subramani, P., Goel, O., Singh, S. P., & Shrivastav, A. (2024). AI-driven enhancements in SAP SD pricing for real-time decision making. International Journal of Multidisciplinary Innovation and Research Methodology, 3(3), 420-446. ISSN: 2960-2068. https://ijmirm.com/index.php/ijmirm/article/view/145*
- *Gupta, S. K. (2025). Modernizing legacy data systems in agile environments. IJRAR - International Journal of Research and Analytical Reviews, 12(2), 713-721. http://www.ijrar.org/IJRAR25B4663.pdf*
- *Jaiswal, I. A. (2024). Self-healing REST services using artificial intelligence in multi-cloud environments. Journal of Quantum Science and Technology (JQST), 1(3), 201. https://doi.org/10.63345/sjaibt.v1.i3.201*
- *Tiwari, S., & Jain, A. (2025). Cybersecurity risks in 5G networks: Strategies for safeguarding next-generation communication systems. International Research Journal of Modernization in Engineering Technology and Science, 7(5). https://doi.org/10.56726/irjmets75837*
- *Dommari, S. (2023). The intersection of artificial intelligence and cybersecurity: Advancements in threat detection and response. International Journal for Research Publication and Seminar, 14(5), 530-545. https://doi.org/10.36676/jrps.v14.i5.1639*
- *Saha, B., & Goel, P. (2023). Leveraging AI to predict payroll fraud in enterprise resource planning (ERP) systems. International*

Journal of All Research Education and Scientific Methods (IJARESM), 11(4), 2284. http://www.ijaresm.com

- *Yadav, N., Bhardwaj, A., Jeyachandran, P., Goel, O., Goel, P., & Jain, A. (2024). Streamlining export compliance through SAP GTS: A case study of high-tech industries. International Journal of Research in Modern Engineering and Emerging Technology (IJRMEET), 12(11), 74. https://www.ijrmeet.org*
- *Gupta, S. K. (2025). Real-time data ingestion with Kafka and AWS tools. ESP Journal of Engineering & Technology Advancements, 5(2), 285-290.*
- *Jaiswal, I. A. (2025). Machine learning-based resource allocation for scalable cloud REST services. World Journal of Future Technology in Computer Science and Engineering (WJFTCSE), 1(3), 101. https://doi.org/10.63345/wjftcse.v1.i3.101*
- *Tiwari, S. (2022). Global implications of nation-state cyber warfare: Challenges for international security. International Journal of Research in Modern Engineering and Emerging Technology (IJRMEET), 10(3), 42. https://doi.org/10.63345/ijrmeet.org.v10.i3.6*
- *Dommari, S., & Jain, A. (2022). The impact of IoT security on critical infrastructure protection: Current challenges and future directions. International Journal of Research in Modern Engineering and Emerging Technology (IJRMEET), 10(1), 40. https://doi.org/10.63345/ijrmeet.org.v10.i1.6*
- *Saha, B., & Chhapola, A. (2020). AI-driven workforce analytics: Transforming HR practices using machine learning models. IJRAR - International Journal of Research and Analytical Reviews, 7(2), 982-997. http://www.ijrar.org/IJRAR2004413.pdf*
- *Yadav, N., Aravind, S., Bikshapathi, M. S., Prasad, M., Jain, S., & Goel, P. (2024). Customer satisfaction through SAP order management automation. Journal of Quantum Science and Technology (JQST), 1(4), 393-413. https://jqst.org/index.php/j/article/view/124*
- *Gupta, S. K. (2025). Designing scalable data warehouses for analytics. International Journal of Creative Research Thoughts (IJCRT), 13(7), h868-h876. ISSN: 2320-2882. http://www.ijcrt.org/papers/IJCRT2507898.pdf*
- *Jaiswal, I. A. (2025). AI-orchestrated microservice security for high-performance scalable systems. International Journal of Advanced Research in Computer Science and Engineering (IJARCSE), 1(4), 101. https://doi.org/10.63345/ijarcse.v1.i4.101*
- *Tiwari, S., & Gola, D. K. K. (2024). Leveraging dark web intelligence to strengthen cyber defense mechanisms. Journal of Quantum Science and Technology (JQST), 1(1), 104-126. https://jqst.org/index.php/j/article/view/249*
- *Dommari, S. (2024). Cybersecurity in autonomous vehicles: Safeguarding connected transportation systems. Journal of Quantum Science and Technology (JQST), 1(2), 153-173. https://jqst.org/index.php/j/article/view/250*
- *Saha, B. (2021). Implementing chatbots in HR management systems for enhanced employee engagement. International Journal of Emerging Technologies and Innovative Research (JETIR), 8(8), f625-f638. ISSN: 2349-5162. http://www.jetir.org/papers/JETIR2108683.pdf*
- *Yadav, N., Prasad, R. V., Kyadasu, R., Goel, O., Jain, A., & Vashishtha, S. (2024). Role of SAP order management in managing backorders in high-tech industries. Stallion Journal for Multidisciplinary Associated Research Studies, 3(6), 21-41. https://doi.org/10.55544/sjmars.3.6.2*
- *Gupta, S. K. (2025). Best practices for Oracle to PostgreSQL migration. International Journal of Science and Research Archive, 16(01), 1337-1344. https://doi.org/10.30574/ijsra.2025.16.1.2083*
- *Jaiswal, I. A., Renuka, A., Kumar, L., & Singh, N. (2025). Uncovering transactional anomalies in blockchain systems through graph neural networks. Proceedings of the International Conference on Computational Technologies for Research in Data Science.*
- *Tiwari, S. (2023). Biometric authentication in the face of spoofing threats: Detection and defense innovations. Innovative Research Thoughts, 9(5), 402-420. https://doi.org/10.36676/irt.v9.i5.1583*
- *Dommari, S., & Mishra, R. K. (2024). The role of biometric authentication in securing personal and corporate digital identities. Universal Research Reports, 11(4), 361-380. https://doi.org/10.36676/urr.v11.i4.1480*
- *Saha, B. (2020). Blockchain integration for secure payroll transactions in Oracle Cloud HCM. International Journal of Novel Research and Development (IJNRD), 5(12), 71-81. ISSN: 2456-4184. https://ijnrd.org/papers/IJNRD2012009.pdf*
- *Yadav, N., Bhat, S. R., Mane, H. R., Pandey, P., Singh, S. P., & Goel, P. (2024). Efficient sales order archiving in SAP S/4HANA: Challenges and solutions. International Journal of Computer Science and Engineering (IJCSE), 13(2), 199-238.*
- *Gupta, S. K. (2025). Metadata lineage frameworks for data governance. International Journal of Creative Research Thoughts (IJCRT), 13(9), c895-c903. ISSN: 2320-2882. http://www.ijcrt.org/papers/IJCRT2509332.pdf*
- *Janapareddy, V. P. K., Sundaresan, S. S. K., Bonikela, H. R., Jaiswal, I. A., Rana, N., et al. (2025). AI-powered vulnerability detection for secure software development. Proceedings of the 2nd International Conference on New Frontiers in Communication and Intelligent Systems.*
- *Tiwari, S., & Agarwal, R. (2022). Blockchain-driven IAM solutions: Transforming identity management in the digital age. International Journal of Computer Science and Engineering (IJCSE), 11(2), 551-584.*

- *Dommari, S. (2022). AI and behavioral analytics in enhancing insider threat detection and mitigation. IJRAR - International Journal of Research and Analytical Reviews, 9(1), 399-416. http://www.ijrar.org/IJRAR22A2955.pdf*
- *Saha, B., Aswini, T., & Solanki, S. (2021). Designing hybrid cloud payroll models for global workforce scalability. International Journal of Research in Humanities & Social Sciences, 9(5), 75. https://www.ijrhs.net*
- *Yadav, N., Abdul, R., Bradley, Satya, S. S., Singh, N., Goel, O., & Chhapola, A. (2024). Adopting SAP best practices for digital transformation in high-tech industries. IJRAR - International Journal of Research and Analytical Reviews, 11(4), 746-769. http://www.ijrar.org/IJRAR24D3129.pdf*
- *Gupta, S. K. (2025). Machine learning integration in Spark-based pipelines. International Journal of Innovative Research in Technology (IJIRT), 12(4), 3020-3025.*
- *Maddula, L. P., Cherukuri, P. A. A., Jaiswal, I. A., Ganesan, S. K., Rana, N., & Khera, M. (2025). Optimization of code efficiency with the utilization of artificial intelligence. Proceedings of the 2nd International Conference on New Frontiers in Communication and Intelligent Systems.*
- *Tiwari, S., & Mishra, R. (2023). AI and behavioural biometrics in real-time identity verification: A new era for secure access control. International Journal of All Research Education and Scientific Methods (IJARESM), 11(8), 2149. http://www.ijaresm.com*
- *Dommari, S., & Khan, S. (2023). Implementing zero trust architecture in cloud-native environments: Challenges and best practices. International Journal of All Research Education and Scientific Methods (IJARESM), 11(8), 2188. http://www.ijaresm.com*
- *Saha, B. (2023). Robotic process automation (RPA) in onboarding and offboarding: Impact on payroll accuracy. International Journal of Current Science (IJCSPUB), 13(2), 237-256. ISSN: 2250-1770. https://rjpn.org/IJCSPUB/papers/IJCSP23B1502.pdf*
- *Yadav, N., Das, A., Kar, A., Goel, O., Goel, P., & Jain, A. (2024). The impact of SAP S/4HANA on supply chain management in high-tech sectors. International Journal of Current Science (IJCSPUB), 14(4), 810. https://www.ijcspub.org/ijcsp24d1091*
- *Jaiswal, I. A. (2023). Intelligent cybersecurity framework for large-scale RESTful service architectures. International Journal of Research Radicals in Multidisciplinary Fields, ISSN: 2960-043X, 2(1), 178-184. https://www.researchradicals.com/index.php/rr/article/view/252*
- *Jaiswal, I. A. (2023). High-performance AI-augmented content management systems for distributed clouds. International Journal of Multidisciplinary Innovation and Research Methodology, ISSN: 2960-2068, 2(2), 90-97. https://ijmirm.com/index.php/ijmirm/article/view/243*
- *Jaiswal, I. A. (2024). AI-optimized content delivery strategies in secure high-performance applications. International Journal of Research and Review Techniques, ISSN: 3006-1075, 3(2), 128-134. https://ijrrt.com/index.php/ijrrt/article/view/256*
- *AI-powered load prediction for ultra-scalable high performance APIs. (2024). International Journal of Engineering Fields, ISSN: 3078-4425, 2(4), 46-53.*
- *Cloud-based secure high-performance application clustering with AI optimization. (2026). AI Tech International Journal, ISSN: 3079-4749, 4(1), 1-8. https://techaijournal.com/index.php/AIjournal/article/view/37*
- *Gupta, S. K. (2025). AI powered query optimization console: A review of intelligent approaches for real-time query performance enhancement in database systems. ESP Journal of Engineering & Technology Advancements, 5(4), 180-192.*
- *M. Rana, S. Srinivas, L. K. Jamili, I. A. Jaiswal, S. Nakka and S. Kasetti, "Real-Time Monitoring and Prediction of Blood Sugar Levels in Diabetic Patients with Functional Models," 2025 International Conference on Engineering, Technology & Management (ICETM), Oakdale, NY, USA, 2025, pp. 1-6, doi: 10.1109/ICETM63734.2025.11051853.*
- *Tiwari, S. (2021). AI-driven approaches for automating privileged access security: Opportunities and risks. International Journal of Creative Research Thoughts (IJCRT), 9(11), c898-c915. ISSN: 2320-2882. http://www.ijcrt.org/papers/IJCRT2111329.pdf*
- *Dommari, S. (2021). Exploring the security implications of quantum computing on current encryption techniques. International Journal of Emerging Technologies and Innovative Research (JETIR), 8(12), g1-g18. ISSN: 2349-5162. http://www.jetir.org/papers/JETIR2112601.pdf*
- *Saha, B., Kumar, L., & Kumar, A. (2019). Evaluating the impact of AI-driven project prioritization on program success in hybrid cloud environments. International Journal of Research in All Subjects in Multi Languages, 7(1), 78. ISSN (P): 2321-2853.*
- *Yadav, N., Krishnamurthy, S., Sayata, S. G., Singh, S. P., Jain, S., & Agarwal, R. (2024). SAP billing archiving in high-tech industries: Compliance and efficiency. Iconic Research and Engineering Journals, 8(4), 674-705.*
- *Gupta, S. K. (2026). Cloud ETL optimization with AWS Glue and Spark. World Journal of Advanced Engineering Technology and Sciences, 18(03), 207-214. https://doi.org/10.30574/wjaets.2026.18.3.0076*
- *Prabhagaran, S., Jaiswal, I. A., & Gandhi, H. (2025). Real-time big data processing in cloud: Scalable, cost-efficient, and AI-driven solutions for financial analytics. [Conference proceedings].*
- *Tiwari, S. (2022). Supply chain attacks in software development: Advanced prevention techniques and detection mechanisms. International Journal of Multidisciplinary Innovation and*

Research Methodology, 1(1), 108-130. ISSN: 2960-2068. https://ijmirm.com/index.php/ijmirm/article/view/195

- *Dommari, S., & Kumar, S. (2021). The future of identity and access management in blockchain-based digital ecosystems. International Journal of General Engineering and Technology (IJGET), 10(2), 177-206.*
- *Saha, B., & Renuka, A. (2020). Investigating cross-functional collaboration and knowledge sharing in cloud-native program management systems. International Journal for Research in Management and Pharmacy, 9(12), 8. https://www.ijrmp.org*
- *Yadav, N. (2025). Edge computing integration for real-time analytics and decision support in SAP service management. International Journal for Research Publication and Seminar, 16(2), 231-248. https://doi.org/10.36676/jrps.v16.i2.283*
- *Bhatia, R., Alonge, M., Gupta, S., Lopez, L., John, B., Adeola, P., & Khan, O. (2025). Challenges and mitigation strategies in migrating legacy ETL pipelines to hybrid cloud ELT architectures for BCBS 239 compliance in banking.*
- *G. Tavva, S. K. Gupta, S. Karuppiah, S. Dachepelly and R. Verma, "AI-Driven Data Platforms: Real-Time Pipelines and Governance," 2025 International Conference on Sustainability, Innovation & Technology (ICSIT), Nagpur, India, 2025, pp. 1-5, doi: 10.1109/ICSIT65336.2025.11294412.*
- *K. Ande, S. K. Gupta, A. Ohja, J. Shaturaev and B. Mirzayev, "Generative AI and Cloud Data Engineering for Business Intelligence," 2025 International Conference on Sustainability, Innovation & Technology (ICSIT), Nagpur, India, 2025, pp. 1-5, doi: 10.1109/ICSIT65336.2025.11295004.*
- *S. Sachi, R. Kiran Pagidi, S. Karunakaran, S. K. Gupta, S. Dharmapuram and O. Goel, "Data Lake Validation Strategies: Ensuring Quality in Data Warehousing Pipelines," 2025 International Conference on Intelligent and Secure Engineering Solutions (CISES), Greater Noida Gautam Budh Nagar, India, 2025, pp. 918-922, doi: 10.1109/CISES66934.2025.11265447.*
- *T. Alrwbaye and S. K. Gupta, "A Hybrid Model for Cloud Resource Utilization Forecasting Using Machine Learning and Evolutionary Optimization," 2025 International Conference on Next Generation of Green Information and Emerging Technologies (GIET), Gunupur, India, 2025, pp. 1-7, doi: 10.1109/GIET65294.2025.11234881.*
- *P. Kumar, S. K. Venugopal, S. Sachi, S. Handa, S. K. Gupta and A. Jain, "Bias Mitigation in Generative Chatbots Through Adversarial Debiasing," 2025 International Conference on Sustainability, Innovation & Technology (ICSIT), Nagpur, India, 2025, pp. 1-6, doi: 10.1109/ICSIT65336.2025.11294625.*
- *Matthew, B., Gupta, S., & Sen, A. (2024). Migrating legacy MES system data containing BOM, routing, and serialization records to a cloud-native lakehouse.*