

Explainable Machine Learning for Software Defect Prediction in Enterprise Applications

Rosalyn Ezeako^{1*}, Aisha I. Abdulwahab², Toussida Fatah Tanguy Minoungou³,
Serif Oyindamola Oyesiji⁴

¹ University of Central Missouri, Warrensburg, Missouri, USA

² Cummins Inc., Indiana, USA

³ DonorsChoose, New York, New York, USA

⁴ FrieslandCampina WAMCO, Lagos, Nigeria

* Corresponding author: Rosalyn Ezeako, ezeakononye@gmail.com

Abstract

Defect prediction models are now embedded in enterprise delivery pipelines, and post-hoc explainers such as SHAP and LIME are routinely attached to their risk scores. Whether those explanations are faithful to the model, stable across retraining, and useful to the engineers who must act on them is rarely tested. This critical integrative review examines explainable machine learning for software defect prediction across five evidence streams: process versus product metrics, just-in-time (change-level) prediction, class imbalance treatment, effort-aware evaluation, and cross-project transfer. A structured protocol covering digital libraries and standards published through 2025 was used to appraise studies for temporal validity, label quality, explanation evaluation and decision relevance. The synthesis finds that discrimination metrics overstate operational value when time-ordered splits, effort normalisation and label latency are ignored; that feature-attribution rankings shift under resampling, collinearity and retraining; and that developer benefit depends on whether explanations point to modifiable, change-local factors. The review contributes the FACTS-DP framework, which separates explanation fidelity and stability, actionability, cost-aware evaluation, temporal validity and evidence stewardship; six falsifiable propositions with stated tests; and a validation design that pairs rolling-origin evaluation with a developer decision experiment. The propositions are untested. Enterprise teams should report effort-aware yield and explanation stability alongside AUC, and should release explanations to developers only when documented fidelity and stability thresholds are met.

Keywords: software defect prediction, just-in-time defect prediction, explainable artificial intelligence, SHAP, LIME, effort-aware evaluation, cross-project prediction, class imbalance.

DOI: 10.66592/235ft2g57

1 Introduction

Enterprise applications such as order management platforms, claims engines, payroll systems and customer portals accumulate code faster than any quality assurance function can inspect it. A typical monorepo in such a setting receives hundreds of commits per week, each touching modules whose failure histories differ by orders of magnitude. Software defect prediction (SDP) responds to this imbalance between change volume and inspection capacity by estimating, for a file, class, package or commit, the probability that it contains or will introduce a defect, so that code review, testing and static analysis effort can be directed toward the riskiest units. The idea is old in software engineering, where testing texts have long argued that defects cluster and that test effort should follow risk rather than be spread uniformly (Myers et al., 2011; Kaner et al., 1999; Ammann et al., 2016). What has changed is the modelling machinery. Gradient-boosted trees, random forests and neural encoders of code and change histories now produce risk scores that outperform the simple size and complexity heuristics of earlier decades on many benchmark datasets, and the same learners are increasingly applied to adjacent assurance problems such as test selection and automated regression control (Borah et al., 2022; Mbonu et al., 2021b; Odejebi et al., 2025).

Higher predictive capacity has come with lower transparency. A developer told that a commit carries a 0.71 defect probability has no basis for deciding whether to request a second reviewer, split the change, add tests, or ignore the warning. Model-agnostic explainers have therefore become the default companion to defect models: SHAP decomposes each prediction into additive feature contributions grounded in Shapley values from cooperative game theory, LIME fits a sparse linear surrogate in the neighbourhood of the instance, and rule-based methods such as anchors or counterfactual generators describe minimal

conditions under which a prediction would change. In enterprise tooling these explanations are rendered as bar charts in pull-request comments, dashboards for quality leads, or ranked lists of "risk drivers" attached to release reports. The same pattern appears in other domains where machine-learning scores guide scarce inspection effort, including failure prediction for physical assets, fraud screening and clinical decision support, and the governance questions it raises are similar across them (Komi et al., 2021; Atakpa et al., 2022; Ilodigwe et al., 2025a; Ladapo et al., 2022).

The difficulty is that explanation has been adopted faster than it has been evaluated. Three weaknesses recur in the defect prediction literature and in the practice it informs. First, predictive validity is often estimated with shuffled k-fold cross-validation on data whose labels were assigned retrospectively, so models are trained on information that would not have been available at prediction time and are credited with performance that a deployed system cannot reproduce. Second, explanation quality is judged by visual plausibility: if the top-ranked features look sensible to the authors (lines added, number of prior bug fixes, developer experience), the explanation is accepted without testing whether it is faithful to the model, stable when the model is retrained, or robust to the resampling used to correct class imbalance. Third, usefulness to developers is inferred from accuracy rather than observed in decisions. A model can rank files well by AUC and still waste reviewer time if its top-ranked units are large, if its explanations point to factors developers cannot change, or if its alerts arrive after the change has merged (Wedraogo et al., 2024; Komi, 2023; Ogundairo et al., 2023a; Adebayo et al., 2023).

These weaknesses are not independent. Imbalance correction changes which features dominate SHAP rankings; effort normalisation changes which modules a model should rank first; cross-project transfer changes the distribution against which local explanations are computed; and label latency in just-in-time settings means that recent changes, the ones developers care most about, carry the noisiest labels. Evaluating explanation in isolation from these design choices produces conclusions that do not survive deployment. The problem is one of measurement architecture as much as of algorithm choice, and it resembles the gap between technical detection capability and governed decision use that has been documented for security analytics and risk quantification in critical infrastructure (Adegbite et al., 2025; Dosunmu et al., 2020; Shittu et al., 2023; Atakpa et al., 2023).

1.1 Research Questions and Contributions

The review is organised around one overarching question: under what conditions do explainable machine-learning models for software defect prediction produce explanations that are faithful, stable and actionable enough to improve defect-risk decisions in enterprise development? Four subsidiary research questions structure the synthesis. RQ1 asks how the choice between process and product metrics, and between file-level and change-level granularity, shapes both predictive validity and the actionability of explanations. RQ2 asks how class imbalance treatment, label noise and cross-project transfer alter the fidelity and stability of SHAP and LIME explanations. RQ3 asks which evaluation measures, including effort-aware and cost-sensitive measures, align model assessment with the inspection decisions that enterprise teams actually make. RQ4 asks what governance arrangements are required before explanations are shown to developers as a basis for action. The questions follow the logic of design science, in which an artefact is justified by its utility for a defined class of problems in a defined context rather than by internal elegance (Hevner et al., 2004; Wieringa, 2014).

The article makes four contributions. (1) It provides a cross-stream synthesis that adjudicates conflicting findings on metric families, just-in-time labelling, imbalance correction, effort-aware evaluation and cross-project transfer, and traces the consequence of each for explanation quality rather than for accuracy alone. (2) It formalises explanation fidelity and explanation stability as measurable properties of a deployed defect model, with explicit metrics and thresholds that can be reported next to AUC and effort-aware measures. (3) It proposes the FACTS-DP framework (Fidelity and stability of explanations, Actionability, Cost-aware evaluation, Temporal validity and Stewardship of evidence for Defect Prediction) and derives six falsifiable propositions with stated tests. (4) It specifies a validation design that combines rolling-origin temporal evaluation with a randomised developer decision experiment, and a threat-to-validity analysis specific to defect data. The propositions are conceptual and have not been tested; the article reports no new field data (Kitchenham et al., 2009; Easterbrook et al., 2008).

2 Theoretical Grounding

Two bodies of theory explain why accurate defect models may fail to improve decisions and why explanations may help or harm. Signal detection theory describes the inspection decision as a choice of threshold under uncertainty with asymmetric

costs. Cognitive fit and sensemaking theory describe how a developer integrates an explanation with what they already know about the code. Together they predict when an explanation adds value: when it moves the decision threshold in the right direction for the right units, and when its form matches the task the developer is performing at that moment. The theories are used below to derive constructs and propositions rather than to decorate the argument, following the view that conceptual work in software engineering should state its mechanisms in a form that empirical studies can contradict (Wieringa, 2014; Runeson et al., 2009; Arumosoye et al., 2021).

2.1 Signal Detection and the Economics of Inspection

A defect model outputs a score s for each unit, and the organisation chooses a threshold t above which units are flagged for additional inspection. Signal detection theory separates the discriminating power of the score, captured by AUC or the separation of score distributions for defective and clean units, from the criterion placement that converts scores into actions. The optimal threshold depends on base rates and on the relative cost of a missed defect (an escaped fault that reaches production) and a false alarm (inspection effort spent on clean code). In enterprise settings both costs are unevenly distributed: an escaped defect in a payment-routing service may carry regulatory and financial consequences far larger than one in an internal reporting module, while inspection cost scales with the size and unfamiliarity of the code under review. A model evaluated by AUC alone therefore says nothing about whether its operating point is sensible, and two models with identical AUC can produce very different inspection yield. This reasoning parallels risk-based assurance in cybersecurity and in safety-critical inspection, where detection capability and criterion setting are governed separately (Fadayomi et al., 2019; Gideon et al., 2019; Atta et al., 2021; Adeyelu, 2020).

Explanations enter this model as a second signal. When a developer receives a flagged unit together with an attribution showing, for example, that the risk is driven by a large number of files touched and by the author's low experience with the subsystem, the developer can re-estimate the likelihood of a defect using private knowledge. A faithful explanation should therefore improve the developer's ability to separate true from false alarms among flagged units (raising their own sensitivity), while an unfaithful or unstable explanation adds noise and can shift the developer's criterion in arbitrary directions. The prediction is testable: explanation quality should affect the discrimination of human decisions among flagged units, not only the developer's stated confidence. Human-in-the-loop machine learning research makes the same distinction between the accuracy of the model and the accuracy of the joint human and model system (Ladapo et al., 2022; Fobellah, 2025a).

2.2 Cognitive Fit, Sensemaking and Calibrated Reliance

Cognitive fit theory holds that decision performance improves when the representation of information matches the task. A code reviewer inspecting a pull request performs a local, change-specific task: deciding whether this diff, written by this author, touching these files, needs more scrutiny. A quality lead planning a release performs an aggregate task: deciding which subsystems deserve regression effort. Explanations built on product metrics of whole files (cyclomatic complexity, coupling between objects, lack of cohesion) fit the second task better than the first, while change-level explanations built on process metrics (lines added, entropy of the change across files, number of prior modifications, time since last change) fit the first task better. The mismatch between explanation granularity and task granularity is one reason developers often describe defect-model explanations as correct but unhelpful (Mairiza et al., 2010; Winkler et al., 2010; Adams, 2023).

Sensemaking theory adds that people interpret automated advice through existing mental models and seek cues that confirm or contradict them. Explanations therefore influence reliance: a plausible but unfaithful explanation can increase trust in a wrong prediction, while an accurate but counterintuitive explanation can lead a developer to dismiss a correct one. Calibrated reliance, the goal of explanation from this perspective, requires that the developer's trust in a given prediction track that prediction's actual reliability. Automation-bias findings from clinical and financial decision support, where model advice with explanation shifts human decisions even when the model is wrong, suggest that the same risk applies in code review (Ilodigwe et al., 2024a; Akintola et al., 2025; Adams, 2024). For defect prediction, calibrated reliance implies two requirements that accuracy metrics do not capture: the predicted probability must be calibrated, and the explanation must be faithful enough that a developer who disagrees with it is disagreeing with the model rather than with an artefact of the explainer.

3 Review Protocol

The article is a critical integrative review. It combines the transparency of a systematic protocol with the interpretive latitude needed to integrate evidence from machine learning, empirical software engineering, human factors and model governance, streams that differ in method and vocabulary. The protocol drew on established guidance for systematic reviews in software engineering and on the PRISMA 2020 reporting items that apply to non-clinical reviews, while acknowledging that the integrative aim, theory building, differs from effect estimation (Kitchenham et al., 2009; Page, 2021; Dybå et al., 2008). Standards were consulted for the definition of defects, failures, test processes and quality characteristics, so that terms such as "defect", "fault" and "escaped defect" are used consistently with software engineering bodies of knowledge (IEEE Computer Society, 2014; ISO/IEC/IEEE, 2018).

3.1 Sources, Search Strings and Eligibility

Searches covered IEEE Xplore, the ACM Digital Library, Scopus, Web of Science, SpringerLink and ScienceDirect, supplemented by arXiv for recent explainability work and by backward and forward citation chasing from core studies. The date window ran from January 2000 to December 2025, which captures the emergence of public defect datasets, the rise of change-level prediction and the recent growth of post-hoc explanation. The core search string combined three blocks with AND: ("defect prediction" OR "fault prediction" OR "bug prediction" OR "defect-prone" OR "just-in-time"); ("explainab*" OR "interpretab*" OR SHAP OR LIME OR "feature importance" OR "model-agnostic" OR counterfactual); and (software OR code OR commit OR "pull request"). Additional strings targeted the evidence streams directly, for example ("effort-aware" OR "cost-effective" OR Popt OR "recall at 20%") AND "defect prediction", and ("cross-project" OR "heterogeneous defect prediction" OR "transfer learning") AND defect. Table 1 records the protocol elements, the operational rule applied for each and the safeguard used to limit bias (Borg et al., 2014; Amarahobu et al., 2025a).

Table 1. Review protocol for explainable software defect prediction: sources, rules and safeguards

Protocol element	Operational rule applied	Safeguard against bias
Databases and window	IEEE Xplore, ACM DL, Scopus, Web of Science, SpringerLink, ScienceDirect, arXiv; Jan 2000 to Dec 2025	Citation chasing in both directions; grey literature limited to standards and dataset documentation
Search blocks	Defect or fault or JIT terms AND explainability terms AND software artefact terms; stream-specific strings for effort, imbalance and transfer	Strings piloted against a known set of core studies and revised until all were retrieved
Inclusion	Predicts defect-proneness of files, classes, modules or commits; reports an explanation method or an evaluation relevant to explanation, effort or transfer	Two-stage screening (title and abstract, then full text) with reasons recorded for every exclusion
Exclusion	Defect localisation or repair only; test generation without prediction; vendor material; no reproducible evaluation design	Borderline studies retained for the negative-case log rather than silently dropped
Extraction fields	Granularity, metric family, labelling method (e.g., SZZ variant), split design, imbalance treatment, explainer, fidelity and stability measures, effort measure, developer study design	Predefined extraction form; ambiguous fields coded as not reported rather than inferred
Appraisal	Temporal validity, label quality, leakage risk, explanation evaluation, decision relevance, reproducibility (Table 4 criteria)	Claims weighted by appraisal, not by venue or citation count
Synthesis	Mechanism, context and outcome coding; explicit adjudication of conflicting results (Table 2)	Contradictory and null findings retained and reported as boundary evidence

Studies were eligible if they predicted the defect-proneness of a software unit (file, class, package or commit) and either applied an explanation method or evaluated a design choice (metric family, imbalance treatment, effort-aware measure, transfer strategy) whose consequences for explanation could be traced. Studies of defect localisation, automated repair and test generation were excluded unless they reported a prediction component. Screening proceeded in two stages, title and abstract followed by full text, with reasons for exclusion recorded. Because the aim is theory building rather than effect pooling, no numerical count of screened records is reported as an empirical finding; the protocol is described so that another team could reproduce the evidence base and challenge its boundaries (Page, 2021; Kitchenham et al., 2009; Ramesh et al., 2001).

3.2 Appraisal Instrument and Synthesis Logic

Each included study was appraised on six criteria chosen for their relevance to explainable defect prediction rather than taken from a generic checklist. Temporal validity asked whether training data strictly preceded test data in time. Label quality asked how defects were identified (issue tracker links, SZZ variants, manual validation) and whether label latency was handled. Leakage risk asked whether any feature used information unavailable at prediction time, such as post-release churn or counts of later bug reports. Explanation evaluation asked whether fidelity or stability was measured rather than asserted. Decision relevance asked whether results were expressed in units that correspond to inspection decisions, such as defects found per line inspected. Reproducibility asked whether data and code were available. The criteria follow the logic of case study and experimental appraisal in software engineering, which weights construct validity and reproducibility above sample size (Runeson et al., 2009; Easterbrook et al., 2008; De Lucia et al., 2007).

Synthesis used a mechanism, context and outcome logic. For each evidence stream the review asked what mechanism a study proposed (for example, that process metrics capture developer behaviour that product metrics miss), in what context it was observed (open source or industrial, file or commit level, within or across projects), and which outcome was measured (discrimination, effort-aware yield, explanation agreement, developer decision). Conflicting results were tabulated and adjudicated by examining whether the conflict disappeared once context and evaluation design were aligned. Figure 1 depicts this pipeline from retrieval to proposition building. The approach draws on organisational learning perspectives in which contradictory evidence is treated as information about boundary conditions rather than as noise to be averaged away (Arumosoye et al., 2021; Obriki et al., 2025; Ekelemu, 2025b).

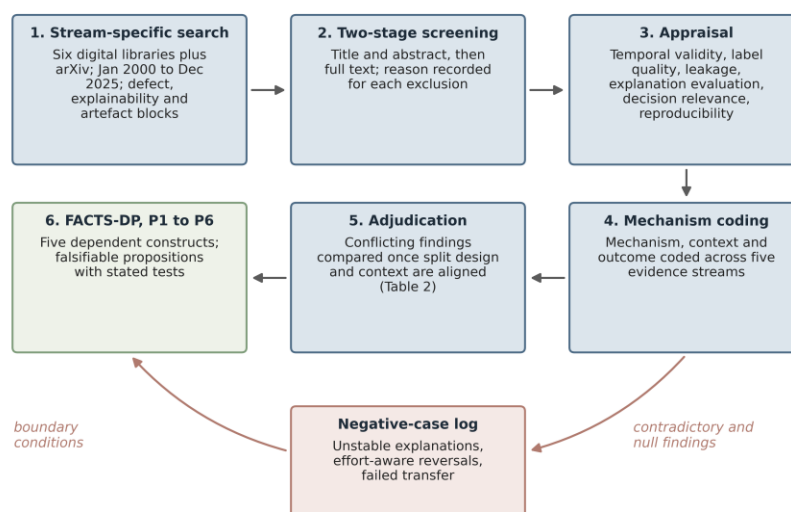


Figure 1. Review pipeline for explainable software defect prediction, from stream-specific search to appraisal, adjudication of conflicting findings and FACTS-DP proposition building

Figure 1 also marks where negative cases enter the synthesis. Studies whose explanations proved unstable, whose effort-aware results reversed their AUC ranking, or whose cross-project models failed were routed to a negative-case log and used to define boundary conditions. Omitting such studies would inflate apparent consistency and hide the conditions under which

explanation misleads, a distortion that safety and reliability reviews have repeatedly warned against when near misses are excluded from learning systems (Obogo et al., 2019; Obriki et al., 2020; Ekelemu, 2025a).

4 Synthesis of the Evidence

The synthesis is organised by evidence stream. Each subsection states the dominant finding, the principal conflict, the adjudication and the consequence for explanation. Table 2 summarises the adjudications so that readers can see where the literature agrees, where it disagrees and why. The streams interact: a decision taken in one (for instance, applying SMOTE to correct imbalance) changes what is observed in another (the SHAP ranking of features), so the subsections should be read as parts of one measurement system rather than as independent topics. The same interdependence between data treatment and downstream interpretation has been observed in predictive analytics for health and marketing, where resampling and attribution choices change the apparent drivers of an outcome (Ilodigwe et al., 2024a; Sanni et al., 2022; Tonoyan et al., 2022c).

4.1 Process Metrics Versus Product Metrics

Product metrics describe the code as it exists: size in lines of code, cyclomatic complexity, depth of inheritance, coupling between objects, response for a class, lack of cohesion, and the Halstead family. Process metrics describe how the code came to be: number of revisions, code churn (lines added and deleted), number of distinct developers, ownership concentration, prior defect fixes, age of the file and recency of change. A consistent pattern in the defect prediction literature is that process metrics give better within-project discrimination than product metrics, and that product metrics add little once process metrics are included. The usual mechanism offered is that defects are introduced by acts of change, and that process metrics record those acts while product metrics record their residue. The pattern is also consistent with testing theory, where coverage and complexity indicate where faults could hide but not where they were recently introduced (Zhu et al., 1997; Ammann et al., 2016; Catal et al., 2013).

The conflict arises in two places. Some studies report that product metrics are more stable across releases because code structure changes slowly, whereas process metrics fluctuate with team composition and release cadence; a model trained on process metrics can therefore degrade faster. Other studies report that the superiority of process metrics shrinks or reverses in cross-project settings, where the meaning of "number of developers" or "churn" depends on repository conventions such as squash merging and monorepo layout. The adjudication is that both findings hold under different conditions. Process metrics dominate when the evaluation is within-project and time-ordered, and when repository conventions are stable; product metrics offer robustness when conventions change or when the model must transfer. For explanation, the choice matters more than accuracy suggests. Process-metric explanations ("this file was changed by five developers in the last month") point to organisational conditions that a single developer cannot alter, whereas product-metric explanations ("this class has very high coupling") point to refactoring targets that are modifiable but rarely addressed within a sprint (Li et al., 2007; Daka et al., 2014; Borah et al., 2022).

A further complication is collinearity. Size correlates with almost every other metric: larger files have more churn, more developers, more complexity and more prior defects. When correlated features enter a model, permutation importance and SHAP values distribute credit among them in ways that depend on the learner and on the background data, so the explanation can attribute risk to complexity in one retraining and to churn in the next without any change in the underlying risk. Feature grouping, correlation-based pruning before training, and grouped SHAP values that attribute contributions to families of metrics rather than to single metrics are the principal mitigations. They trade granularity for stability, a trade-off that must be declared rather than hidden. Similar lessons have emerged in interpretable failure prediction for physical assets, where correlated sensor features produce shifting attributions under retraining (Komi et al., 2021; Tonoyan et al., 2023; Asiedu et al., 2024b).

4.2 Just-in-Time Defect Prediction and Label Latency

Just-in-time (JIT) defect prediction moves the unit of analysis from files to commits and makes a prediction at the moment a change is submitted. Typical features describe the diff (lines added and deleted, number of files, directories and subsystems touched, entropy of the modification across files), its history (number of developers who previously touched the files, age of the files, number of unique prior changes) and the author (experience overall, recent experience, experience in the subsystem),

plus a flag for whether the change itself is a fix. JIT prediction fits the decision point: a reviewer deciding how carefully to examine a pull request is exactly the person who can use a change-level risk score, and the author is still available to explain or restructure the change. Its explanations are also more actionable in principle, because features such as change size and spread can be altered by splitting a commit. This alignment with the moment of review is why JIT has become the default framing for continuous integration settings, and why explainability research has concentrated there (Myers et al., 2011; Kaner et al., 1999; Mbonu et al., 2021b).

The central weakness is labelling. Bug-inducing commits are usually identified with the SZZ algorithm or its variants: a bug-fixing commit is linked to an issue, the lines it deletes or modifies are traced back through version history, and the commits that last touched those lines are marked as bug-inducing. Every step introduces error. Fixing commits may be mislinked or unlinked; blame may land on refactorings, formatting changes or comment edits; and changes that introduced a defect not yet discovered are labelled clean. The last error, label latency, has a systematic structure: the most recent commits are the most likely to be falsely clean, because the time between introduction and discovery of a defect can be months. A model trained and evaluated on a snapshot therefore learns that recent, possibly risky changes are safe, and an evaluation that includes recent commits in the test set scores the model against wrong labels. Verification latency correction, which excludes from training any commit whose observation window has not closed, and waiting-time sensitivity analyses are the main responses (Gotel et al., 1994; Cleland-Huang et al., 2012; Winkler et al., 2010).

For explanation, label noise has a specific and underappreciated consequence. SHAP and LIME explain the model, not the world; if the labels encode a bias (for example, that large refactoring commits are frequently blamed by SZZ because they touch many lines later modified by fixes), the explanation will faithfully report that bias as a risk driver. A faithful explanation of a model trained on noisy labels can therefore be a precise description of the labelling artefact. This is why the FACTS-DP framework treats label quality as part of temporal validity rather than as a data-cleaning detail, and why explanation fidelity is necessary but not sufficient for explanation validity. The general point, that an analytic pipeline can be internally consistent while measuring the wrong construct, recurs in reviews of predictive compliance monitoring and cross-shift information loss in process industries (Adeyelu, 2018; Ekelemu, 2025b; Ekelemu et al., 2025b).

4.3 Class Imbalance and Its Side Effects on Explanation

Defect datasets are imbalanced. At file level, defective files often make up a small minority of a release; at commit level, bug-inducing changes are similarly a minority of all changes, although proportions vary widely by project and labelling method. Imbalance depresses recall for the minority class and inflates accuracy, so most studies apply a correction: random undersampling of clean units, random oversampling of defective units, synthetic oversampling with SMOTE or its variants, cost-sensitive learning that weights minority errors more heavily, or threshold moving after training. The finding that some correction improves recall and F-measure is consistent; the finding about which correction is best is not, and depends on the learner, the dataset and the evaluation measure. Rank-based measures such as AUC are largely insensitive to resampling, whereas threshold-dependent measures such as recall, precision and MCC shift substantially (Wedraogo et al., 2024; Sanni et al., 2023; Atakpa et al., 2022).

The consequence for explanation has received less attention than the consequence for accuracy. Resampling changes the training distribution and therefore the model; synthetic oversampling in particular creates interpolated defective instances in regions of feature space that may be sparsely populated in reality, which can shift decision boundaries and alter which features the model relies on. Studies that compare explanations across imbalance treatments report that global importance rankings can reorder, and that local attributions for the same instance can change sign, even when AUC is nearly identical. The adjudication offered here is that rank-preserving treatments (cost-sensitive learning and post-hoc threshold moving) are preferable when explanations will be shown to developers, because they change the operating point without changing the training distribution from which background data for SHAP are drawn. Where resampling is used, explanations should be computed against a background sample drawn from the original, unresampled distribution, and the stability of explanations across treatments should be reported. The same caution applies to anomaly detection for fraud and to financial time series, where synthetic balancing can manufacture spurious drivers (Atakpa et al., 2023; Tonoyan et al., 2024b; Oluwo et al., 2025).

4.4 Effort-Aware and Cost-Sensitive Evaluation

Classification measures treat every unit as equal, but inspecting a 2,000-line module costs more than inspecting a 20-line one. Effort-aware evaluation reorders units by predicted defect density rather than predicted probability, or equivalently assesses how many defects are found as a function of lines of code inspected. The two most common measures are recall at a fixed effort budget, usually the proportion of defective units found when 20 percent of total lines have been inspected (Recall@20%LOC), and Popt, which compares the area under the model's cumulative lift chart with the areas under the optimal and worst orderings. With Area(m) denoting the area under the cumulative curve of defects found against LOC inspected for ordering m, the normalised measure is:

$$Popt = 1 - [Area(optimal) - Area(model)] / [Area(optimal) - Area(worst)]$$

Popt equals 1 when the model reproduces the optimal density ordering and approaches 0 when it performs like the worst ordering. Figure 3 illustrates the construction. A recurrent and important finding in the effort-aware literature is that simple unsupervised orderings, for example ranking changes in ascending order of size, can match or exceed complex supervised models on effort-aware measures, because small changes are cheap to inspect and a meaningful share of them are defective. Other studies counter that such orderings exploit the measure rather than the task, since reviewers who inspect only tiny changes miss the large, risky ones and incur a context-switching cost per change that LOC does not capture. The adjudication is that effort-aware measures are necessary but must be paired with a per-unit overhead term and with a check on the size distribution of flagged units (Tonoyan et al., 2022b; Okonkwo et al., 2024e; Adesuyi et al., 2023).

A cost-sensitive alternative is to specify the expected cost of an inspection policy directly. If C_FN is the cost of an escaped defect, C_FP the cost of inspecting a clean unit, c_0 a fixed per-unit review overhead and c_1 the marginal cost per line inspected, the expected cost of flagging a unit i of size L_i with calibrated defect probability p_i is:

$$E[cost_i | flag] = c_0 + c_1 \cdot L_i + (1 - \rho) \cdot p_i \cdot C_{FN}, \quad E[cost_i | no flag] = p_i \cdot C_{FN}$$

where ρ is the probability that inspection detects a defect that is present. A unit should be flagged when the first expression is smaller than the second, that is when $\rho \cdot p_i \cdot C_{FN}$ exceeds $c_0 + c_1 \cdot L_i$. The rule shows why calibration matters: the flagging decision depends on the absolute probability, not on the rank, so a model with excellent AUC but poor calibration will flag the wrong units. It also shows why explanations matter economically. A faithful explanation can raise ρ by directing the reviewer to the part of a change that drives risk, and can lower c_1 by narrowing the region that needs close reading. Explanation value can therefore be expressed in the same currency as model value. Enterprise cost governance and KPI-design research make the parallel argument that a metric without a cost model and a named decision owner is reporting rather than control (Sakyi et al., 2022b; Sakyi et al., 2023; Adesuyi et al., 2023).

4.5 Cross-Project and Heterogeneous Transfer

Enterprise organisations frequently need defect predictions for new services with little history, which motivates cross-project defect prediction (CPDP): train on projects with labelled history and apply to a target project with none. The consistent finding is that naive CPDP performs worse than within-project prediction, because metric distributions differ across projects in scale, skew and meaning. Remedies include per-project normalisation such as rank or z-score transformation, instance selection that retains source units most similar to the target, transfer component analysis and other domain adaptation methods that align feature distributions, and heterogeneous defect prediction that matches metrics across projects with different feature sets using correlation or distribution similarity. Results are mixed: several remedies close much of the gap on some benchmark pairs and fail on others, and simple normalisation is often competitive with sophisticated adaptation. The stream mirrors the broader experience of transferring predictive models across organisations and channels, where performance depends on how closely source and target contexts align (Wedraogo et al., 2024; Michael et al., 2023c; Sanni et al., 2023).

Transfer poses a distinct problem for explanation. An explanation of a cross-project prediction describes the source model's reasoning, expressed in transformed features, applied to a target instance whose feature values may lie outside the source distribution. SHAP values computed against a source background sample answer the question "why does this unit differ from a typical source-project unit", which is not the question a target-project developer is asking. Computing attributions against a target background, once enough target units exist, changes the reference point and can change the explanation even though the model is unchanged. Domain adaptation adds a further layer: features after transfer component analysis are linear

combinations of original metrics and cannot be explained to developers without back-projection, which introduces its own approximation. The adjudication is that cross-project explanations should be labelled as provisional, computed against a declared background, and retired once within-project data suffice. The analogy with infrastructure resilience planning is instructive: models transferred from other systems are useful for initial triage but must be recalibrated against local failure data before they justify operational commitments (Aminu-Ibrahim et al., 2025b; Shittu et al., 2023; Dagodzo et al., 2020).

4.6 Fidelity and Stability of SHAP and LIME Explanations

Fidelity is the degree to which an explanation reflects the model's actual behaviour. For SHAP, the additivity (local accuracy) property guarantees that, for a model f and instance x with M features, the attributions ϕ_j sum to the difference between the prediction and the expected prediction over the background distribution:

$$f(x) = \phi_0 + \sum_{j=1..M} \phi_j(x), \text{ with } \phi_0 = E_{\text{background}}[f(X)]$$

Additivity is a property of exact Shapley values; it does not guarantee that the attributions are meaningful. Two sources of infidelity matter for defect models. The first is the choice between interventional and conditional value functions: interventional SHAP breaks feature dependence by sampling features independently from the background, which can evaluate the model on impossible instances (a commit that adds 5,000 lines to one file with zero entropy), while conditional SHAP respects dependence but can attribute importance to features the model does not use. The second is approximation: KernelSHAP estimates values by sampling coalitions, and small sample budgets produce noisy attributions, whereas TreeSHAP computes exact values for tree ensembles efficiently. LIME is more exposed. Its surrogate is fitted to perturbed samples drawn around the instance, weighted by a kernel whose width is a free parameter, and both the random sampling and the kernel width change the explanation. Local fidelity for LIME can be measured as the weighted squared error between the surrogate g and the model f over the perturbation neighbourhood, and should be reported for every explanation shown to a user (Komi et al., 2021; Jimoh et al., 2024; Ominyi et al., 2025a).

A practical fidelity test for defect models is deletion or perturbation: remove or neutralise the top- k features by attribution, replacing them with background values, and measure the drop in predicted risk. A faithful explanation should produce a larger drop than removing k random features or the k lowest-ranked features. Stability is a separate property: the degree to which explanations for the same or similar instances remain consistent under changes that should not matter. Three forms are relevant. Run-to-run stability measures agreement between repeated explanations of the same instance by the same model (relevant for LIME and KernelSHAP). Retraining stability measures agreement between explanations of the same instance by models retrained on slightly different data, such as successive weekly training windows or different random seeds. Neighbourhood stability measures agreement between explanations of near-identical instances. For the top- k feature sets E_a and E_b of two explanations, stability can be expressed as Jaccard overlap, and rank agreement as Kendall's tau over the full attribution ranking:

$$S_k(E_a, E_b) = |E_a \cap E_b| / |E_a \cup E_b|, \quad \tau = (n_c - n_d) / [M(M-1)/2]$$

where n_c and n_d are the numbers of concordant and discordant feature pairs. Reports on explanation stability in defect prediction indicate that global rankings from tree ensembles and TreeSHAP are fairly stable for the top few features, that LIME explanations vary noticeably between runs at default settings, and that all methods become less stable under strong collinearity and after resampling. Local explanations are generally less stable than global ones, which matters because developers see local explanations. The adjudication in Table 2 is that fidelity and stability should be treated as release criteria for explanations, measured routinely, and reported next to model performance. Explanations that fail thresholds should be withheld or shown with a stated uncertainty rather than presented as authoritative. Governance research on multi-agent AI quality control reaches the same conclusion for automated document review: an explanation layer requires its own acceptance test (Ominyi et al., 2025a; Ominyi et al., 2024b; Nwakamma et al., 2024b).

Table 2. Adjudication of conflicting findings across the five evidence streams and consequences for explanation

Stream	Dominant finding	Conflicting finding	Adjudication	Consequence for explanation
Process vs product metrics	Process metrics discriminate better within project	Product metrics more stable across releases and projects	Both hold; condition on split design and repository conventions	Process explanations point to team conditions; group correlated metrics

Just-in-time prediction	Change-level scores fit the review decision point	SZZ labels noisy; recent commits falsely clean	Close observation windows before training; report waiting-time sensitivity	Faithful explanations may describe labelling artefacts
Class imbalance	Some correction improves recall and MCC	No correction best across learners and datasets	Prefer cost-sensitive learning or threshold moving when explanations are shown	Resampling reorders rankings; use unresampled background
Effort-aware evaluation	Popt and Recall@20%LOC reorder models ranked by AUC	Size-ordered baselines exploit the measure	Add per-unit overhead; report size of flagged units	Explanations can raise detection per inspected line
Cross-project transfer	Naive transfer loses accuracy	Simple normalisation rivals complex adaptation	Normalise first; adapt only with evidence of gain	Background choice changes attributions; label as provisional
SHAP and LIME	TreeSHAP exact and fairly stable at global top-k	Local and LIME explanations vary across runs and retraining	Treat fidelity and stability as release criteria	Withhold or qualify explanations below thresholds

4.7 Developer Actionability

Actionability is the extent to which an explanation supports a concrete, feasible and beneficial action by the person who receives it. It is not the same as plausibility. An explanation that ranks "number of prior defects" first is plausible and probably faithful, but a developer reviewing a pull request cannot change a file's history. Actionability depends on three properties of the features an explanation highlights. Modifiability asks whether the feature can be changed by the recipient in the relevant time frame (change size and spread can; file age cannot). Locality asks whether the feature refers to the unit under decision rather than to its environment (the entropy of this change is local; the number of developers who have ever touched the file is not). Directionality asks whether changing the feature in the indicated direction is known to reduce risk rather than merely to reduce the score, since some features are proxies whose manipulation would game the model. Requirements engineering research on non-functional qualities makes a comparable point that quality attributes are only operational when they are decomposed into verifiable, owned actions (Mairiza et al., 2010; Ramesh et al., 2001; Borg et al., 2014).

Studies of developer perception report a consistent pattern: practitioners value defect predictions that come with reasons, prefer explanations at the granularity of the change or function rather than the file, and are sceptical of explanations dominated by history features they cannot influence. Counterfactual and rule-based explanations ("this change would fall below the review threshold if it touched fewer than three subsystems") are often rated more useful than attribution bar charts because they state an action. Two cautions temper the finding. Most evidence is perceptual, gathered by survey or interview rather than by observing decisions, and perceptions of usefulness do not reliably predict decision quality. And actionable explanations can induce gaming: if splitting commits lowers risk scores, developers may split commits without reducing defect introduction, which erodes the model's validity over time. Actionability must therefore be evaluated by observed behaviour and by downstream escaped defects, not by satisfaction ratings (Eyetezmitan et al., 2023b; Akhigbe et al., 2024; Falemi et al., 2024).

Actionability also depends on the recipient's authority. A reviewer can request changes but may not be able to delay a release; a quality lead can reallocate regression testing but not rewrite a module; an architect can schedule refactoring but not within the current sprint. Explanations should therefore be routed to the role that can act on them: change-local process features to authors and reviewers, subsystem-level product features to architects, and aggregate trends to release managers. This routing principle is standard in safety and operational risk management, where leading indicators are useful only when they reach a person with authority to intervene before the lagging outcome materialises (Obriki et al., 2019; Arumosoye et al., 2024; Obogo et al., 2023).

5 The FACTS-DP Framework

The synthesis motivates a framework that separates five properties which the literature has tended to conflate. FACTS-DP holds that an explainable defect prediction system improves defect-risk decisions only when (T) its predictions are temporally

valid, (F) its explanations are faithful and stable, (A) those explanations are actionable by the role that receives them, (C) the system is evaluated and operated with cost-aware criteria, and (S) evidence about all four is stewarded and governed through owned thresholds and feedback. The properties are ordered by dependency rather than maturity: a faithful, stable explanation of a temporally invalid model is a faithful description of an artefact, and an actionable explanation evaluated without cost-aware criteria can direct effort to cheap but unimportant units. Figure 2 shows the framework and its feedback path. The framework is a design artefact in the sense of design science, specified so that its components can be evaluated separately and jointly (Hevner et al., 2004; Wieringa, 2014; Akhigbe et al., 2025; Arumosoye et al., 2022).

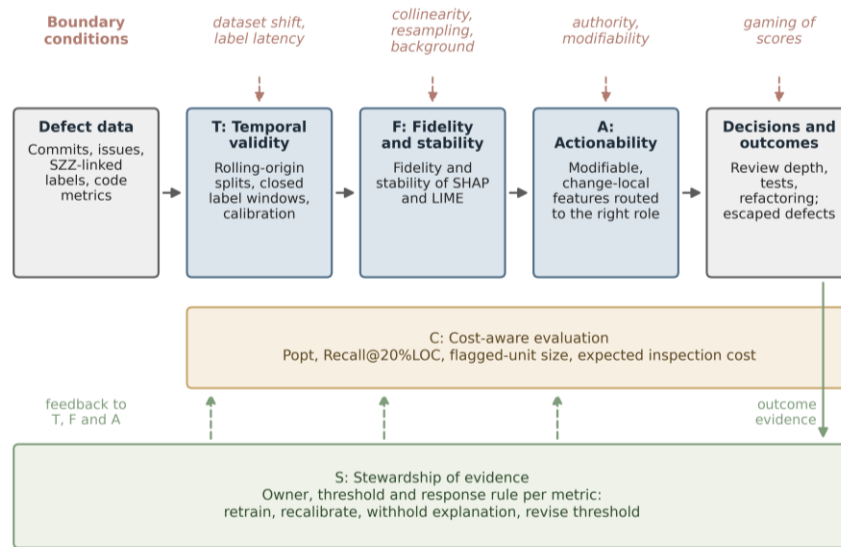


Figure 2. The FACTS-DP framework: five dependent properties linking defect data to developer decisions, with evidence governance closing the feedback loop

Figure 2 locates the boundary conditions identified in the synthesis. Dataset shift and label latency act on temporal validity; collinearity, resampling and background choice act on explanation fidelity and stability; the recipient's authority and the modifiability of features act on actionability; and gaming acts on the link between actions and outcomes. The stewardship layer receives evidence from each stage and can trigger retraining, recalibration, withholding of explanations or threshold revision. This structure reflects a general principle from continuous-control and cyber risk governance: detection and explanation capabilities must be linked to named owners and response rules before their outputs acquire operational authority (Adebayo et al., 2025; Adegbite et al., 2025; Anioke et al., 2023a; Ahmed et al., 2021).

5.1 Constructs and Operational Measures

The five constructs are operationalised in Table 3. Temporal validity is measured by performance under rolling-origin evaluation, calibration error, and the sensitivity of results to the label observation window. Fidelity and stability are measured by deletion-based fidelity, run-to-run and retraining stability of top-k feature sets, and agreement between explainers. Actionability is measured by the share of explanation mass on modifiable and local features, and behaviourally by the rate at which explanations lead to a documented action. Cost-aware evaluation is measured by Popt, Recall@20%LOC, the size distribution of flagged units and expected inspection cost under a declared cost model. Stewardship of evidence is measured by whether each metric has an owner, a threshold and a response rule, and by the latency between a threshold breach and the response. Each measure is defined at a stated unit and window, with the numerator and denominator made explicit, because ambiguity in these choices is a common source of irreproducible defect prediction results (Zhu et al., 1997; Tonoyan et al., 2024b; Sakyi et al., 2022b).

Table 3. FACTS-DP constructs, operational measures, release thresholds and governance responses

Construct	Operational measures	Illustrative threshold	release	Response when breached
-----------	----------------------	------------------------	---------	------------------------

Construct	Operational measures	Illustrative threshold	Response when breached
Temporal validity (T)	Rolling-origin AUC and MCC; expected calibration error; change in results when observation window varies	Calibration error below an agreed bound; results stable across window lengths	Retrain with closed windows; recalibrate; suspend scoring for new units
Fidelity and stability (F)	Deletion fidelity gain over random; top-5 Jaccard across seeds and weekly retrains; SHAP and LIME agreement	Top-5 Jaccard at or above an agreed floor; fidelity gain clearly above random	Withhold local explanations; show grouped features; increase sampling budget
Actionability (A)	Share of attribution on modifiable, change-local features; action rate per flagged unit	Majority of shown attribution on features the recipient can change	Re-route explanation to a role with authority; add counterfactual form
Cost-aware evaluation (C)	Popt; Recall@20%LOC; median size of flagged units; expected cost under declared C_FN, C_FP, c_0, c_1	Model beats size-ordered and history-ordered baselines on expected cost	Revise threshold; add per-unit overhead; review cost parameters
Stewardship of evidence (S)	Owner, threshold and response recorded per metric; breach-to-response latency; escaped-defect trend	Every metric owned; breaches answered within the review cycle	Escalate to quality board; retire model or explanation layer

The thresholds in Table 3 are illustrative and must be set locally. The point of the table is that each threshold is attached to a response, so that a breach changes what the system does rather than appearing only in a report. Enterprise KPI and real-time performance monitoring research consistently finds that indicators without response rules decay into dashboards that are observed but not acted upon (Tonoyan et al., 2024b; Sakyi et al., 2022b; Aminu-Ibrahim et al., 2025a).

5.2 Propositions and Stated Tests

Six propositions follow from the framework. Each is stated so that it could be falsified, and each is paired with the test that would contradict it. P1 (temporal inflation): the gap between shuffled cross-validation performance and rolling-origin performance of JIT models increases with the proportion of test commits whose label observation window has not closed. Test: vary the observation window on the same repository and regress the performance gap on the open-window share; P1 is contradicted if the gap is flat. P2 (resampling instability): for equal AUC, models trained with synthetic oversampling produce lower retraining stability of local top-5 explanations than models trained with cost-sensitive weighting. Test: paired comparison across repositories of top-5 Jaccard under the two treatments; P2 is contradicted if stability is equal or higher under oversampling. P3 (fidelity and reviewer discrimination): among flagged changes, reviewers given explanations that pass a fidelity threshold separate truly defective from clean changes better than reviewers given no explanation, while reviewers given explanations that fail the threshold do no better, or worse, than those given none. Test: randomised three-arm decision experiment measuring reviewer sensitivity (d') on held-out labelled changes (Ladapo et al., 2022; Fobellah, 2025a; Kitchenham et al., 2009).

P4 (actionability and effect): the reduction in escaped defects attributable to an explainable JIT model is larger when the majority of shown attribution falls on modifiable, change-local features than when it falls on history features. Test: compare escaped-defect rates across teams or periods that differ in explanation composition, controlling for change volume; P4 is contradicted if composition does not moderate the effect. P5 (effort-aware reversal): model rankings by AUC and by expected inspection cost with a non-zero per-unit overhead disagree in a substantial share of comparisons, and the disagreement shrinks as per-unit overhead rises relative to per-line cost. Test: re-evaluate published model comparisons under a family of cost parameters. P6 (governed feedback): systems in which explanation stability and calibration have named owners and response rules show smaller performance decay over successive releases than systems that monitor the same metrics without response rules. Test: interrupted time series across teams adopting the governance layer at different times. The propositions permit null and adverse findings, and P3 in particular predicts harm from unfaithful explanation (Runeson et al., 2009; Uchechi et al., 2022; Ilodigwe et al., 2024b).

5.3 Rival Explanations and Boundary Tests

Several simpler accounts could explain improvements attributed to explainable defect prediction. The first is attention: any tool that directs reviewers to some changes may raise defect detection simply by increasing review effort, regardless of the model or explanation. A design that includes a random-flagging arm or a size-ordered arm controls for this. The second is selection: teams that adopt explainable prediction may already have stronger review cultures, lower change volume or simpler architectures. Recording baseline review depth, change volume and architectural complexity before adoption, and staging adoption across comparable teams, addresses this. The third is labelling drift: an apparent drop in defects after adoption may reflect slower discovery of defects or changes in issue triage rather than fewer defects. Holding the labelling pipeline constant and extending observation windows before comparing periods addresses this. Leadership and team-composition effects documented in project performance research are further candidates that must be measured rather than assumed away (Amayo et al., 2023c; Obogo et al., 2019a; Arumosoye et al., 2023).

Reverse causality also deserves attention. Teams that experience a spike in escaped defects may adopt defect prediction in response, so that adoption coincides with a period of regression to the mean; a naive before-and-after comparison would then credit the tool with an improvement that would have happened anyway. Baseline trends over several releases, comparison teams without adoption, and process tracing that confirms reviewers changed their behaviour in response to explanations before outcomes changed provide stronger evidence. Figure 3 shows why effort-aware analysis is itself a boundary test: a model that appears superior on AUC may fall below a size-ordered baseline once inspection effort is counted. The mechanism is supported only when the intermediate decision (which changes were reviewed more carefully, and what reviewers did) shifts before the downstream outcome (Obogo et al., 2019b; Uchechi et al., 2018; Uchechi, 2018; Amadi et al., 2025).

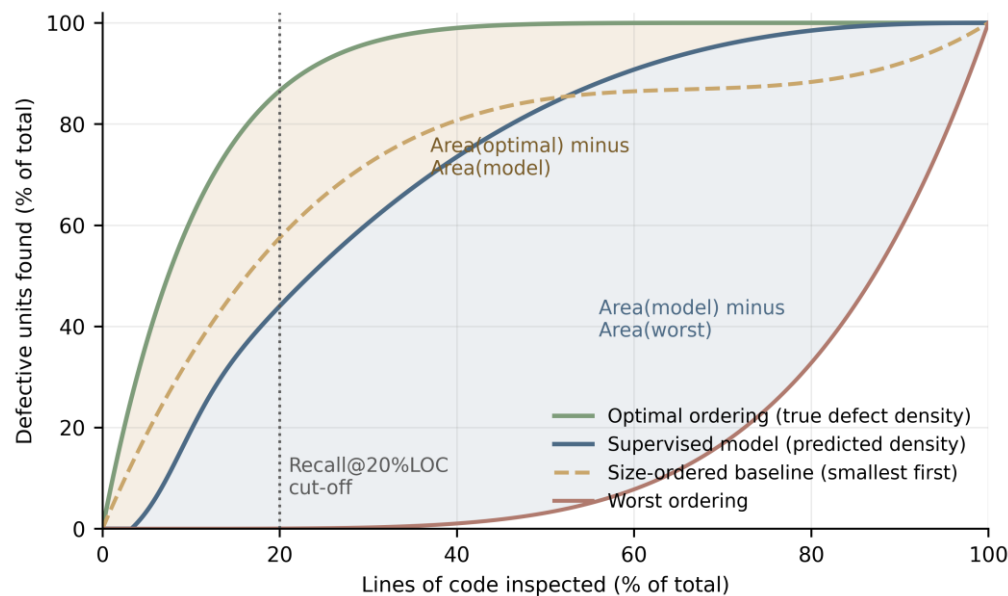


Figure 3. Schematic effort-aware cumulative lift curves (illustrative, not empirical), showing the areas used to compute P_{opt} and the Recall@20%LOC cut-off

Figure 3 is a schematic rather than a data plot. It shows how the optimal ordering (by true defect density), a supervised model, a size-ordered baseline and the worst ordering produce different cumulative curves of defects found against lines inspected, and how P_{opt} normalises the model's area between the optimal and worst curves. In the illustrated case the supervised model and the size-ordered baseline cross: the baseline finds more defects in the first few percent of inspected lines because it ranks many small changes first, while the model overtakes it later. Where the organisation's effort budget falls determines which is preferable, which is why the budget itself must be stated. Negative cases of this kind carry as much analytical weight as positive ones, a lesson familiar from alarm rationalisation where standing alarms that look informative in aggregate fail operators at the decision point (Ekelemu, 2025a; Jones et al., 2024a; Tawose et al., 2023).

6 Validation Design

The propositions call for a validation programme that tests model, explanation and decision together. The design proposed here has two linked components: a rolling-origin temporal evaluation of models and explanations on enterprise repository data, and a randomised decision experiment with practising developers. The two components share a dataset, a labelling pipeline and a set of pre-registered measures, so that findings about explanation stability in the first can be connected to findings about decision quality in the second. The programme follows design science evaluation logic, in which an artefact is assessed first for technical properties and then for utility in context, and it borrows staged-rollout and matched-comparison designs from evaluation research in organisational and policy settings (Wieringa, 2014; Miller et al., 2015; Hevner et al., 2004).

6.1 Rolling-Origin Evaluation and Developer Decision Experiment

In the temporal component, commits are ordered by timestamp and divided into successive windows. For each origin t , the model is trained on commits whose label observation window closed before t and evaluated on commits in the following window, with labels assigned only after that window's own observation period has closed. Hyperparameters are tuned within the training window using an inner time-ordered split. At each origin the study records AUC, MCC, expected calibration error, Popt, Recall@20%LOC and expected cost under a declared cost model; computes TreeSHAP and LIME explanations for a fixed audit sample of test commits; and measures deletion fidelity, run-to-run stability and retraining stability relative to the previous origin. Imbalance treatment (none, cost-sensitive weighting, SMOTE) and metric family (process only, product only, combined) are crossed as factors. Figure 4 shows the structure. The design reports how explanation properties evolve over time, which a single train-test split cannot reveal (Li et al., 2007; Catal et al., 2013; Daka et al., 2014).

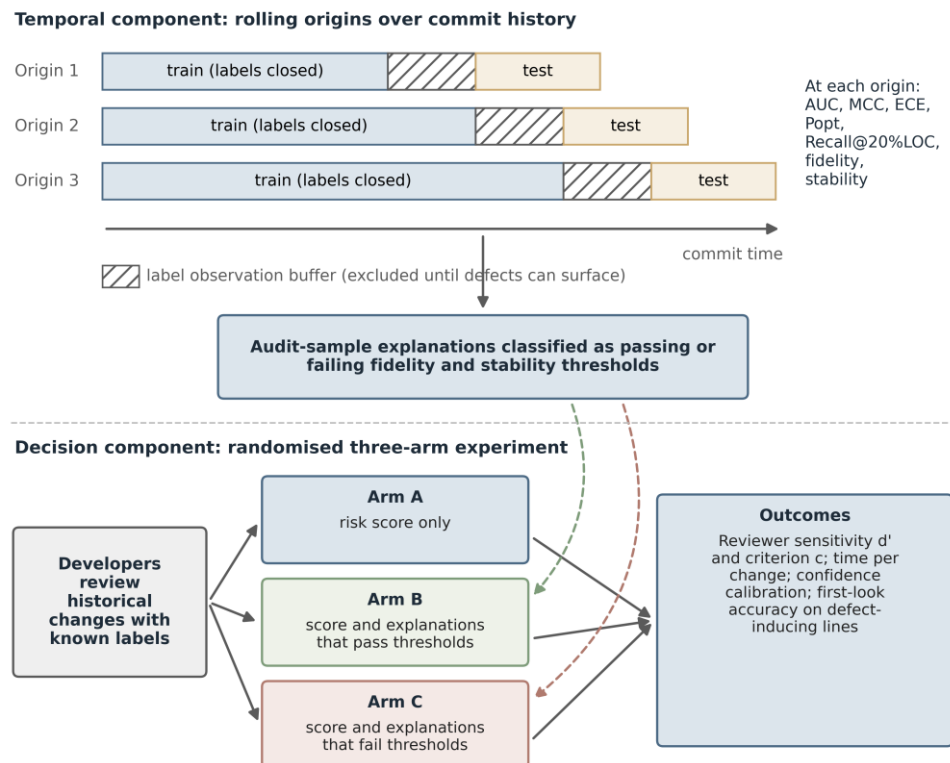


Figure 4. Proposed validation design: rolling-origin temporal evaluation with closed label windows (top) feeding a three-arm randomised developer decision experiment (bottom)

In the decision component, practising developers from the participating organisation review a set of historical changes whose true defect status is known from closed observation windows. Changes are drawn so that defect prevalence among them is

fixed and known to the analyst but not to participants. Participants are randomised to three arms: risk score only; risk score with explanations that passed the fidelity and stability thresholds; and risk score with explanations that failed them (drawn from unstable configurations). For each change, participants decide whether to approve, request changes or escalate, state their confidence, and indicate which part of the diff they would examine first. Primary outcomes are reviewer sensitivity (d') and criterion (c) computed from approve and escalate decisions against true labels; secondary outcomes are time per change, calibration of confidence, and whether the part of the diff examined first contains the defect-inducing lines. Ethics review, informed consent and anonymisation of participant data are required, since review performance data could otherwise be used for individual appraisal (Kumuyi et al., 2024b; Badmus et al., 2024; Ominyi, 2021).

Analysis uses mixed-effects models with random intercepts for participants and changes, which accounts for variation in reviewer skill and change difficulty. Robustness checks vary the fidelity and stability thresholds used to assign explanations to arms, test whether effects differ for large versus small changes and for history-dominated versus change-local explanations, and exclude changes whose labels depend on low-confidence SZZ links. Pre-registration of primary outcomes and of the thresholds that define passing explanations prevents the analyst from choosing thresholds after seeing results. Human-in-the-loop and cognitive-bias research in audit and risk management supports measuring decisions directly, because stated confidence and satisfaction diverge from accuracy under automated advice (Fobellah, 2025a; Ladapo et al., 2022; Eze et al., 2024b).

6.2 Threats to Validity

Table 4 identifies threats specific to explainable defect prediction and the mitigation each requires. Construct threats centre on the gap between SZZ-derived labels and true defect introduction and on the gap between attribution and causal influence. Internal threats centre on leakage and on confounding between adoption and team characteristics. External threats centre on the difference between open-source benchmarks and enterprise repositories, whose conventions for branching, squash merging, code ownership and issue tracking differ. Conclusion threats centre on multiple comparisons across learners, treatments and explainers, which can manufacture significant differences, and on low defect counts in individual windows. Ethical threats centre on the use of author-experience features and review performance data, which could be repurposed for individual performance management. The categories are kept separate because more data does not repair a weak label, and a well-calibrated model does not repair an explanation that points developers to factors they cannot change (Runeson et al., 2009; Mbonu et al., 2018; Aliliele et al., 2024a).

Table 4. Threats to validity specific to explainable defect prediction and required mitigations

Validity type	Specific threat	Consequence	Mitigation
Construct	SZZ mislabels refactorings and misses undiscovered defects	Model and explanation learn labelling artefacts	Refactoring-aware SZZ; closed observation windows; manual validation of an audit sample
Construct	Attribution read as causal influence	Developers act on correlates that do not reduce risk	Deletion fidelity tests; counterfactual checks; wording that states association
Internal	Time-travel leakage from shuffled splits or post-release features	Inflated performance and misleading drivers	Rolling-origin splits; feature timestamp audit
Internal	Adoption confounded with review culture and change volume	Tool credited with pre-existing strength	Staged adoption; baseline trends; random-flag control arm
External	Open-source benchmarks differ from enterprise repositories	Findings fail to transfer	Report branching, merge policy, ownership and tracker conventions
Conclusion	Many learners, treatments and explainers compared	Spurious significant differences	Pre-registered primary outcomes; correction for multiplicity; effect sizes with intervals
Ethical	Author-experience features and review data used for appraisal	Surveillance and distorted behaviour	Aggregate reporting; purpose limitation; exclude features from individual reports

Two threats deserve emphasis because they are specific to explanation. The first is explainer-model coupling: evaluating TreeSHAP fidelity on a tree ensemble favours that pairing, so fidelity comparisons across explainers should use the same model and a model-agnostic fidelity test. The second is audience coupling: explanations judged by the researchers who built

the model are assessed by the people least likely to find them confusing. Decision experiments with practitioners unfamiliar with the model are the corrective. Data protection governance frameworks that classify the sensitivity of enterprise data and trace regulatory obligations offer a template for handling developer-level features responsibly (Aliliele et al., 2024a; Aliliele et al., 2025b; Mbonu et al., 2018).

7 Discussion

The review set out to establish when explainable defect prediction improves defect-risk decisions. Its answer is conditional but specific. Explanations help when they are computed from temporally valid models, when they are faithful and stable enough that a developer's disagreement with them is informative, when they highlight features the recipient can change, and when the system is judged by the cost of the inspection decisions it drives. Each condition is frequently violated in published work and in practice, and the violations interact: resampling to fix imbalance destabilises explanations, label latency corrupts what faithful explanations describe, and cross-project transfer changes the reference against which attributions are computed. The discussion below considers what this means for theory, for enterprise practice and for the counter-arguments that a sceptical reader should raise (Borah et al., 2022; Odejobi et al., 2025; Ilodigwe et al., 2025a).

7.1 Theoretical Implications

The first implication is that explanation quality is a property of a measurement system, not of an explainer. Debates about whether SHAP or LIME is better for defect prediction are underdetermined unless they specify the labelling pipeline, the split design, the imbalance treatment and the background distribution. FACTS-DP makes these dependencies explicit and orders them, which turns an open-ended comparison into a set of conditional claims. The second implication concerns signal detection. Treating explanation as a second signal that affects reviewer sensitivity, rather than as a presentation layer that affects trust, yields sharper predictions: faithful explanations should raise d' among flagged units, and unfaithful ones should lower it or shift the criterion without improving discrimination. The third implication concerns cognitive fit. The match between explanation granularity and task granularity predicts that change-local process explanations will be more useful in review, and subsystem-level product explanations more useful in release planning, a prediction that existing perception studies support but decision studies have not yet tested (Wieringa, 2014; Ladapo et al., 2022; Mairiza et al., 2010).

A fourth implication links explanation to organisational learning. When explanations are logged together with the actions they prompted and the outcomes that followed, the organisation accumulates evidence about which risk drivers are real and which are artefacts. Over time this evidence can feed back into feature engineering, labelling rules and review policy. Organisational learning models from safety management describe the same cycle, in which leading indicators are validated against lagging outcomes and revised when they fail, and they caution that the cycle breaks when feedback is not owned (Arumosoye et al., 2021; Obriki et al., 2025; Ilodigwe et al., 2024b).

7.2 Practical Implications for Enterprise Teams

For enterprise teams the implications are concrete. Evaluate every candidate model with rolling-origin splits and closed label windows before comparing it with anything else. Report calibration, Popt, Recall@20%LOC and expected inspection cost next to AUC, and compare against a size-ordered baseline and a history-ordered baseline, since a model that cannot beat both on expected cost does not justify its maintenance. Prefer cost-sensitive weighting or threshold moving over synthetic oversampling when explanations will be shown to developers. Compute explanations against a declared background drawn from the unresampled, in-project distribution; group correlated metrics; and measure fidelity and stability on an audit sample at every retraining. Withhold local explanations that fall below thresholds, or show them with a visible uncertainty marker. Route change-local explanations to authors and reviewers, and subsystem-level ones to architects and release managers (Sakyi et al., 2023; Tonoyan et al., 2022b; Adesuyi et al., 2023).

Governance should be proportionate but real. Each metric in Table 3 needs an owner, a threshold and a response rule, and breaches should be reviewed in the same forum that reviews escaped defects. Author-experience features should never appear in individual performance reports, and developers should be told which features the model uses and why. Teams in regulated sectors should treat the defect model as a component within their AI governance inventory, with documentation of training data, intended use and known failure modes, in line with the organisational readiness obligations emerging under AI regulation in multinational enterprises. Reviews of algorithmic accountability and model risk management in financial

services offer a mature template for such an inventory, pairing independent validation and ongoing performance monitoring with a named owner accountable for each model's use (Ominyi & Anichukwueze, 2023). These arrangements follow the direction of enterprise risk and compliance research that links analytics to accountable decision owners (Ominyi et al., 2024b; Adebayo et al., 2025; Falemi et al., 2024; Anichukwueze et al., 2022; Arumosoye et al., 2025).

7.3 Counter-Arguments and Boundary Conditions

Sceptics raise three objections. The first holds that explanation is unnecessary: if a model is accurate and well calibrated, reviewers should simply inspect what it flags. This view neglects the reviewer's private information. Reviewers know about design intent, pending migrations and fragile integrations that no metric captures, and a faithful explanation lets them combine that knowledge with the model's evidence. It also neglects accountability: in regulated enterprises a decision to block or approve a change may need to be justified to auditors. The second counter-argument holds that simple heuristics such as change size capture most of the value, so machine learning adds complexity without benefit. The effort-aware evidence gives this view real support, and FACTS-DP accepts it as a baseline that any model must beat on expected cost; where it is not beaten, the heuristic should be used (Kaner et al., 1999; Myers et al., 2011; Onche et al., 2023).

The third counter-argument holds that explanations will be gamed. If developers learn that small, focused commits lower risk scores, they may split changes artificially. This is a real boundary condition, and the framework addresses it by evaluating actionability through escaped defects rather than scores, and by monitoring feature distributions for abrupt shifts after explanations are introduced. Other boundary conditions limit transfer. Very small teams may lack enough defects per window for stable estimates. Repositories with heavy squash merging lose the commit granularity on which JIT features depend. Systems dominated by configuration, infrastructure-as-code or low-code artefacts may not be described well by conventional code metrics. And organisations whose issue tracking is inconsistent will produce labels too noisy for any explanation to be trustworthy. In each case the framework predicts reduced or absent benefit, which is a claim that can be checked (Jones et al., 2021; Oluwadele et al., 2025; Abah et al., 2025; Michael et al., 2025c).

8 Limitations and Future Research Agenda

The review has limitations that bound its conclusions. Terminology in the field is inconsistent: "defect", "fault", "bug" and "failure" are used interchangeably, and explanation terms such as "interpretability", "explainability" and "transparency" are rarely defined, so some relevant studies may have been missed by the search strings despite piloting. Most primary evidence comes from open-source projects, whose development practices differ from enterprise ones, and industrial studies are fewer and less often reproducible. Publication bias probably favours studies in which explanations looked sensible, so the true rate of unstable or misleading explanations may be higher than the literature suggests. The propositions are derived by reasoning from heterogeneous evidence and have not been tested; the article offers a framework and a validation design, not an effect estimate. The effort-aware illustration in Figure 3 is schematic and should not be read as reporting results (Dybå et al., 2008; Page, 2021; Atta et al., 2022).

Five lines of future research follow. First, decision experiments with practising developers are the largest gap; perception surveys should give way to studies that measure reviewer sensitivity, time and escaped defects under randomised explanation conditions, as P3 specifies. Second, explanation stability should be reported as routinely as AUC, using shared definitions such as top-k Jaccard and Kendall's tau across seeds and retraining windows, so that studies can be compared. Third, labelling research should quantify how SZZ errors propagate into explanations, for example by comparing attributions from models trained on SZZ labels and on manually validated labels for the same repository. Fourth, explanations for defect models built on code representations from large language models need fidelity tests suited to token-level and embedding inputs, where feature attribution is harder to interpret than for tabular metrics. Fifth, longitudinal studies should examine whether explanation exposure changes developer behaviour, including gaming, and whether governance with owned thresholds slows model decay, as P6 predicts (Ladapo et al., 2023b; Sanni et al., 2025; Sanni et al., 2024; Aminu-Ibrahim et al., 2025a; Ogbete et al., 2023; Yusuf et al., 2018; Eze et al., 2022b).

9 Conclusion

Explainable machine learning can make software defect prediction more useful to enterprise development teams, but only when explanation is treated as an engineered component with its own acceptance tests. Predictions must be temporally valid,

with labels whose observation windows have closed. Explanations must be faithful to the model and stable across runs and retraining, which favours cost-sensitive over synthetic resampling and declared, in-project background data. Explanations must point to features the recipient can change, and the whole system must be judged by the cost of the inspection decisions it drives rather than by discrimination alone. The FACTS-DP framework organises these requirements into five dependent properties, six falsifiable propositions and a validation design that couples rolling-origin evaluation with a developer decision experiment. Adopting it would give enterprise teams a defensible basis for deciding when a defect model and its explanations have earned a place in the review workflow.

Declarations

Funding: This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

Competing interests: The authors declare no competing interests.

Data availability: No new data were generated or analysed in this study; all sources are cited in the reference list.

References

- Abah, M. A., Okwah, M. N., Oladosu, M. A., Yohanna, N. R., Blessing, C. T., & Vanessa, O. I. (2025). The impact of lifestyle modifications on cardiovascular health: A review of diet, exercise, and stress reduction intervention. *Drug and Pharmaceutical Science Archives*, 5(3), 26-34. <https://doi.org/10.47587/DAP.2025.5301>
- Adams, V. (2024). Assessment integrity in the age of artificial intelligence: Redesigning learning evaluation for authentic human competency. *World Journal of Innovation and Modern Technology*, 8(6), 185-195. <https://doi.org/10.56201/wjimt.v8.no6.2024.pg185.195>
- Adams, V. (2023). Preserving critical thinking in the era of generative artificial intelligence: Evidence-based frameworks for human-centered learning. *International Journal of Social Sciences and Management Research*, 9(11), 321-332. <https://doi.org/10.56201/ijssmr.v9.no11.2023.pg321.332>
- Adebayo, A., Adegbite, M. P., & Ahmed, M. O. (2023). AI augmented threat detection in industrial control systems: A systematic review of machine learning approaches for ICS anomaly detection. *International Journal of Engineering and Modern Technology*, 9(3), 287-340. <https://doi.org/10.56201/ijcsmt.v9.no3.2023.pg287.340>
- Adebayo, A., Amadi, C., & Ijagbemi, A. O. (2025). From audit to continuous control: Implementation pathways, governance, and organizational readiness for AI-enabled compliance in critical infrastructure. *International Journal of Computer Science and Mathematical Theory*, 11(8), 88-134. <https://doi.org/10.56201/ijcsmt.vol.11.no8.2025.pg88.134>
- Adebayo, A., Anunagba, C. O., & Ozowara, D. E. (2025). A review of zero trust security models and cost effectiveness in healthcare infrastructure. *International Journal of Advanced Multidisciplinary Research and Studies*, 5(6), 2269-2283. <https://doi.org/10.62225/2583049X.2025.5.6.6051>
- Adegbite, M. P., Adebayo, A., & Ahmed, M. O. (2025). A cyber risk quantification and governance architecture for critical infrastructure: From posture measurement to executive reporting. *International Journal of Computer Science and Mathematical Theory*, 11(12), 232-293. <https://doi.org/10.56201/ijcsmt.vol.11.no12.2025.pg232.293>
- Adesuyi, M. O., Kalu, A., & Walawalkar, G. (2023). Data-led cost governance in technology-intensive enterprises. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 9(3), 877-896. <https://doi.org/10.32628/IJSRCSEIT>
- Adeyelu, O. O. (2020). An artificial intelligence-driven conceptual model for wildlife strike risk quantification and aircraft airworthiness impact assessment at high-traffic African airports. *Iconic Research and Engineering Journals*, 4(1), 339-368. <https://doi.org/10.64388/IREV4I1-1718482>



- Adeyelu, O. O. (2018). A predictive compliance monitoring framework for detecting systemic safety risks through aviation consumer complaint data. *Iconic Research and Engineering Journals*, 1(8), 250-276. <https://doi.org/10.64388/IREV118-1718478>
- Ahmed, M. O., Adegbite, M. P., & Adebayo, A. (2021). Zero trust architecture for operational technology in North American critical infrastructure: A framework for implementation and resilience optimization. *International Journal of Engineering and Modern Technology*, 7(1), 66-113. <https://doi.org/10.56201/ijemt.vol.7.no1.2021.pg66.113>
- Akhigbe, R., Falemi, A., & Akin-Oluyomi, O. T. (2024). An advanced framework for technology adoption that supports employee readiness in modern organizations. *International Journal of Advanced Multidisciplinary Research and Studies*, 4(1), 1637-1659. <https://doi.org/10.62225/2583049X.2024.4.1.5344>
- Akhigbe, R., Falemi, A., & Akin-Oluyomi, O. T. (2025). An advanced governance model for improving transparency accountability and strategic execution in complex enterprises. *International Journal of Advanced Multidisciplinary Research and Studies*, 5(6), 1169-1188. <https://doi.org/10.62225/2583049X.2025.5.6.5341>
- Akintola, A. S., Fawehinmi, Y. O., Aiyenitaju, O., Chinemerem, B., & Emmanuella, O. (2025). Digital transformation in telehealth: A systematic review of user trust, privacy, and regulatory governance in AI-powered remote monitoring systems. *Journal of Scientific Research and Reports*, 31(11), 163-182. <https://doi.org/10.9734/jsrr/2025/v31i113658>
- Aliliele, C., Mbonu, I. S., & Iwuanyanwu, U. (2024a). A conceptual framework for enterprise data sensitivity classification and regulatory traceability mechanisms. *International Journal of Advanced Multidisciplinary Research and Studies*, 4(6), 3103-3124. <https://doi.org/10.62225/2583049X.2024.4.6.5991>
- Aliliele, C., Mbonu, I. S., Uzoka, E., & Iwuanyanwu, U. (2025b). Advances in data lakehouse governance architectures for enterprise data loss prevention and compliance assurance. *Shodhshauryam, International Scientific Refereed Research Journal*, 8(4), 171-213. <https://doi.org/10.32628/SHISRRJ258474>
- Amadi, C. E., Okorafor, U. C., Okorafor, C. I., Okwah, M. N., Akanbi, P. O., Okam, O. V., Udenze, C. I., Mbakwem, A. C., & Ajuluchukwu, J. N. (2025). Association between triglyceride-glucose index and estimated 10-year risk of cardiovascular disease in Nigerian non-diabetic hypertensives: A cross-sectional study. *PLOS Global Public Health*, 5(7), e0004760. <https://doi.org/10.1371/journal.pgph.0004760>
- Amarahobu, C., Ekelemu, O., Akano, O. A., & Adikwu, F. E. (2025a). Structured mentorship models in on-the-job technician development: A review of advancement and promotion coaching schemes, workplace learning theory and the leadership interface in continuous processing environments. *International Journal of Social Sciences and Management Research*, 11(8), 626-656. <https://doi.org/10.56201/ijssmr.vol.11no8.2025.pg626.656>
- Amayo, E. B., Owulade, O. A., & Isi, L. R. (2023c). The role of diversity and inclusion in enhancing project team performance and delivery in multinational environments. *Journal of Frontiers in Multidisciplinary Research*, 4(1), 48-58. <https://doi.org/10.54660/IJFMR.2023.4.1.48-58>
- Aminu-Ibrahim, A. Y., Ogbete, J. C., & Iwuanyanwu, O. C. (2025a). Cost control and financial accountability frameworks for national healthcare construction programs. *International Journal of Advanced Multidisciplinary Research and Studies*, 5(6), 1970-1990. <https://doi.org/10.62225/2583049X.2025.5.6.5700>
- Aminu-Ibrahim, A. Y., Ogbete, J. C., & Iwuanyanwu, O. C. (2025b). Infrastructure resilience planning for national diagnostic systems under public health stress conditions. *Gyanshauryam, International Scientific Refereed Research Journal*, 8(1), 340-381. <https://doi.org/10.32628/GISRRJ2582311>



Ammann, P., & Offutt, J. (2016). *Introduction to Software Testing* (2nd ed.). Cambridge University Press.

Anichukwueze, C. C., Ominyi, M., Uzougbo, N. S., & Jones, B. C. (2022). Cross-functional legal education as risk mitigation: Empirical analysis of contract error reduction in engineering teams. *International Journal of Engineering and Modern Technology*, 8(5), 142-180. <https://doi.org/10.56201/ijemt.v8.no5.2022.pg142.180>

Anioke, S. C., & Atima, M. E. (2023a). Public health governance models using process optimization and performance metrics for regulatory oversight. *International Journal of Advanced Multidisciplinary Research and Studies*, 3(6), 2534-2548. <https://doi.org/10.62225/2583049X.2023.3.6.5491>

Arumosoye, O. M., & Obriki, O. D. (2023). Conceptual model for emergency response readiness and capability in energy and process facilities. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 9(3), 897-917. <https://doi.org/10.32628/CSEIT25112791>

Arumosoye, O. M., & Obriki, O. D. (2024). Conceptual model of safety leadership influence in large temporary project organizations. *International Journal of Advanced Multidisciplinary Research and Studies*, 4(6), 3034-3047. <https://doi.org/10.62225/2583049X.2024.4.6.5895>

Arumosoye, O. M., & Obriki, O. D. (2022). Conceptual risk pathway model for lifting and rigging operations in heavy industrial construction. Gyanshauryam, *International Scientific Refereed Research Journal*, 5(5), 346-369. <https://doi.org/10.32628/GISRRJ2256237>

Arumosoye, O. M., & Obriki, O. D. (2021). Organizational learning-based conceptual maturity model for continuous safety performance improvement. *International Journal of Multidisciplinary Research and Growth Evaluation*, 2(1), 970-980. <https://doi.org/10.54660/IJMRGE.2021.2.1.970-980>

Arumosoye, O. M., Obriki, O. D., & Ozobu, C. O. (2025). Conceptual framework for environmental risk control and ESG performance through waste handling and operational discipline. Gyanshauryam, *International Scientific Refereed Research Journal*, 8(4), 177-198. <https://doi.org/10.32628/GISRRJ258367>

Asiedu, C. S., & Asiedu, A. A. (2024b). Reconceptualizing quality assurance in pharmaceutical manufacturing as a determinant of supply reliability. *International Journal of Health and Pharmaceutical Research*, 9(5), 148-173. <https://doi.org/10.56201/ijhpr.v9.no5.2024.pg148.173>

Atakpa, M. I., & Abetoh, N. F. (2022). A systematic review of machine learning advances in financial fraud detection for banking systems. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 8(2), 771-801. <https://doi.org/10.32628/CSEIT23906220>

Atakpa, M. I., & Fobellah, A. N. (2023). Anomaly detection in financial time-series data: A conceptual model for healthcare and banking applications. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 9(2), 967-997. <https://doi.org/10.32628/CSEIT2342441>

Atta, S., Adjaklo, E., Asomani, E., & Adenuga, O. M. (2022). Advances in welding inspection standards and quality assurance practices in large-scale power generation projects. *International Journal of Engineering and Modern Technology*, 8(5), 101-124. <https://doi.org/10.56201/ijemt.v8.no5.2022.pg101.124>

Atta, S., Tofah, P., & Adenuga, O. M. (2021). Advances in non-destructive testing methods for weld integrity evaluation in high-capacity power infrastructure. *International Journal of Engineering and Modern Technology*, 7(1), 20-47. <https://doi.org/10.56201/ijemt.vol.7.no1.2021.pg20.47>

- Badmus, O., Jooda, D., & Anunagba, C. O. (2024). A privacy-by-design framework for role-based security architecture in Salesforce environments handling sensitive personal data. *International Journal of Advanced Multidisciplinary Research and Studies*, 4(6), 3244-3254. <https://doi.org/10.62225/2583049X.2024.4.6.6243>
- Borah, S., Aliliele, K. C., Rakshit, S., & Vajjhala, N. R. (2022). Applications of artificial intelligence in software testing. In P. K. Mallick et al. (Eds.), *Cognitive informatics and soft computing (Lecture Notes in Networks and Systems, Vol. 375)*, pp. 727-736. Springer. https://doi.org/10.1007/978-981-16-8763-1_60
- Borg, M., Runeson, P., & Ardö, A. (2014). Recovering from a decade: A systematic mapping of information retrieval approaches to software traceability. *Empirical Software Engineering*, 19, 1565-1616. <https://doi.org/10.1007/s10664-013-9255-y>
- Catal, C., & Mishra, D. (2013). Test case prioritization: A systematic mapping study. *Software Quality Journal*, 21, 445-478. <https://doi.org/10.1007/s11219-012-9181-z>
- Cleland-Huang, J., Gotel, O., & Zisman, A. (Eds.). (2012). *Software and Systems Traceability*. Springer. <https://doi.org/10.1007/978-1-4471-2239-5>
- Dagodzo, D., & Ahiaeké Patrick, M. C. (2020). UAV-based pipeline and corridor monitoring: A review of current practices and emerging technologies. *Iconic Research and Engineering Journals*, 3(10), 574-597. <https://doi.org/10.64388/IREV3110-1716084>
- Daka, E., & Fraser, G. (2014). A survey on unit testing practices and problems. *IEEE International Symposium on Software Reliability Engineering Workshops*, 201-211. <https://doi.org/10.1109/ISSREW.2014.11>
- De Lucia, A., Fasano, F., Oliveto, R., & Tortora, G. (2007). Recovering traceability links in software artifact management systems using information retrieval methods. *ACM Transactions on Software Engineering and Methodology*, 16(4), Article 13. <https://doi.org/10.1145/1276933.1276934>
- Dosunmu, A. A., & Ogundele, P. O. (2020). Intrusion detection and prevention models for enhancing organizational cyber defense effectiveness. *Iconic Research and Engineering Journals*, 4(6), 310-324. <https://doi.org/10.64388/IREV4I6-1713226>
- Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9-10), 833-859. <https://doi.org/10.1016/j.infsof.2008.01.006>
- Easterbrook, S., Singer, J., Storey, M., & Damian, D. (2008). Selecting empirical methods for software engineering research. In *Guide to Advanced Empirical Software Engineering* (pp. 285-311). Springer.
- Ekelemu, O. (2025a). Control room alarm management and human factors in large liquefied natural gas facilities: A comprehensive review of rationalization frameworks, operator response, and the elimination of standing and inhibited alarms. *International Journal of Engineering and Modern Technology*, 11(8), 352-396. <https://doi.org/10.56201/ijemt.vol.11.no8.2025.pg352.396>
- Ekelemu, O., & Akano, O. A. (2025b). Loss detection and integrity surveying in liquefaction header systems: Flare hunting, attribution and programme design for unintended flow to flare. *International Journal of Engineering and Modern Technology*, 11(8), 458-487. <https://doi.org/10.56201/ijemt.vol.11.no8.2025.pg458.487>
- Ekelemu, O. (2025b). Cross-shift information loss and the standardization of logbook handovers: A critical review of communication failure and continuity protocols in continuous processing operations. *International Journal of Engineering and Modern Technology*, 11(8), 397-424. <https://doi.org/10.56201/ijemt.vol.11.no8.2025.pg397.424>



- Eyetsemitan, R. A., Ambali, K. B., Oyeleye, A. O., & Fadayomi, O. (2023b). User acceptance testing in small business technology deployment: A structured validation framework for lean operational environments. *International Journal of Multidisciplinary Research and Growth Evaluation*, 4(6), 1512-1531. <https://doi.org/10.54660/IJMRGE.2023.4.6.1512-1531>
- Eze, F. I., Akinleye, O. K., & Anene, U. N. (2024b). Development of a risk-based accelerated approval decision model balancing speed, safety, and evidence quality in pharmaceutical regulatory science. *Research Journal of Pure Science and Technology*, 7(6), 120-161. <https://doi.org/10.56201/rjpst.v7.no6.2024.pg120.161>
- Eze, F. I., Anene, U. N., & Akinleye, O. K. (2022b). Design of an integrated regulatory readiness maturity model for pharmaceutical companies seeking global market entry. *International Journal of Health and Pharmaceutical Research*, 7(1), 77-112. <https://doi.org/10.56201/ijhpr.v7.no1.2022.pg77.112>
- Fadayomi, O., Abolaji, T. O., Edivri, J., Ogbole, J. I., Okoruwa, P. O., & Akeju, B. (2019). Risk-based cybersecurity assurance and data availability: Limitations, advances and future research opportunities. *Iconic Research and Engineering Journals*, 2(12), 602-617. <https://doi.org/10.64388/IREV2I12-1713779>
- Fadayomi, O., Abolaji, T. O., Edivri, J., Ogbole, J. I., Okoruwa, P. O., & Akeju, B. (2019). Risk-based cybersecurity assurance and data availability limitations, advances and future research opportunities. *Iconic Research and Engineering Journals*, 2(12), 602–617. <https://doi.org/10.64388/IREV2I12-1713779>
- Fadayomi, O., Abolaji, T. O., Edivri, J., Ogbole, J. I., Okoruwa, P. O., & Akeju, B. (2019). Risk-based cybersecurity assurance and data availability: Limitations, advances and future research opportunities. *IRE Journals*, 2(12). <https://doi.org/10.64388/IREV2I12-1713779>
- Falemi, A., Akhigbe, R., & Akin-Oluyomi, O. T. (2024). A conceptual enterprise risk management model that integrates HR strategy with operational analytics. *International Journal of Multidisciplinary Research and Growth Evaluation*, 5(6), 1730-1753. <https://doi.org/10.54660/ijmrge.2024.5.6.1730-1753>
- Fobellah, A. N. (2025a). Cognitive biases in financial decision-making: Implications for audit and risk management in large corporations: A conceptual review. *International Journal of Science and Research Archive*, 16(2), 17-22. <https://doi.org/10.30574/ijrsra.2025.16.2.2273>
- Gideon, E. N., Adaramola, T. S., & Fadero, S. (2019). Reliability engineering and failure analysis in complex engineering systems. *Iconic Research and Engineering Journals*, 2(11), 703-727. <https://doi.org/10.64388/IREV2I11-1722490>
- Gotel, O. C. Z., & Finkelstein, A. C. W. (1994). An analysis of the requirements traceability problem. *IEEE International Conference on Requirements Engineering*, 94-101. <https://doi.org/10.1109/ICRE.1994.292398>
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75-105. <https://doi.org/10.2307/25148625>
- IEEE Computer Society. (2014). Guide to the Software Engineering Body of Knowledge Version 3.0. IEEE.
- Ilodigwe, L., & Adesemoye, A. C. (2024a). Advances in real-world evidence, predictive analytics, and data-driven decision making for improving health system performance and patient outcomes. *International Journal of Health and Pharmaceutical Research*, 9(5), 174-197. <https://doi.org/10.56201/ijhpr.v9.no5.2024.pg174.197>
- Ilodigwe, L., & Adesemoye, A. C. (2025a). Advances, risks, and implementation challenges of artificial intelligence as a force multiplier for clinical decision making and health system efficiency. *International Journal of Medical Evaluation and Physical Report*, 9(7), 165-185. <https://doi.org/10.56201/ijmepr.v9.no7.2025.pg165.185>



- Ilodigwe, L., & Adesemoye, A. C. (2024b). Beyond technology: A strategic framework for building sustainable and resilient health systems through organizational strategy, operational excellence, and workforce capability. *International Journal of Medical Evaluation and Physical Report*, 8(6), 299-319. <https://doi.org/10.56201/ijmepr.v8.no6.2024.pg299.319>
- ISO/IEC/IEEE. (2018). ISO/IEC/IEEE 29119-2 Software and systems engineering: Software testing processes. International Organization for Standardization.
- ISO/IEC/IEEE. (2018). ISO/IEC/IEEE 29148 Systems and software engineering: Life cycle processes and requirements engineering. International Organization for Standardization.
- Jimoh, H. O., & Ahmed, M. O. (2024). Analyzing Network Time Protocol (NTP) based amplification DDoS attack and its mitigation techniques. *Journal of Digital Innovations and Contemporary Research in Science, Engineering and Technology*, 12(2), 17-24. <https://doi.org/10.22624/AIMS/DIGITAL/V11N2P2x>
- Jones, B. C., & Ominyi, M. (2024a). Applications of artificial intelligence in customs administration and trade facilitation: A review. *World Journal of Innovation and Modern Technology*, 8(6), 181-217. <https://doi.org/10.56201/wjimt.v8.no6.2024.pg181.217>
- Jones, B. C., & Ominyi, M. (2021). Developing a conceptual framework linking West African agricultural value chains to household economic stability. *International Journal of Economics and Financial Management*, 6(2), 89-114. <https://doi.org/10.56201/ijefm.v6.no2.2021.pg89.114>
- Kaner, C., Falk, J., & Nguyen, H. Q. (1999). *Testing Computer Software* (2nd ed.). Wiley.
- Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering: A systematic literature review. *Information and Software Technology*, 51(1), 7-15. <https://doi.org/10.1016/j.infsof.2008.09.009>
- Komi, N. M., & Adamolekun, A. (2021). Interpretable machine learning for early failure prediction in distributed renewable energy assets. *International Journal of Multidisciplinary Research and Growth Evaluation*, 2(6), 1015-1038. <https://doi.org/10.54660/IJMRGE.2021.2.6.1015-1038>
- Komi, N. M. (2023). When accurate forecasts are not fair forecasts: Deep learning load prediction and equity across districts. Shodhshauryam, *International Scientific Refereed Research Journal*, 6(6), 476-546. <https://doi.org/10.32628/SHISRRJ247102>
- Kumuyi, O., Uzoka, E., Akeju, B., & Ozowara, D. E. (2024b). Framework for privacy-focused digital identity verification supporting financial inclusion in Africa. *International Journal of Advanced Multidisciplinary Research and Studies*, 4(6), 3150-3162. <https://doi.org/10.62225/2583049X.2024.4.6.6005>
- Ladapo, O. O., Dosunmu, A. A., Jooda, D., & Abolaji, T. O. (2022). Human-in-the-loop machine learning: A state of the art. *Journal of Frontiers in Multidisciplinary Research*, 3(1), 656-669. <https://doi.org/10.54660/JFMR.2022.3.1.656-669>
- Ladapo, O. O., Jooda, D., Dosunmu, A. A., & Abolaji, T. O. (2023b). Systematic literature review on security access control policies and techniques based on privacy requirements in a BYOD environment. *International Journal of Advanced Multidisciplinary Research and Studies*, 3(6), 2890-2904. <https://doi.org/10.62225/2583049X.2023.3.6.6206>
- Li, Z., Harman, M., & Hierons, R. M. (2007). Search algorithms for regression test case prioritization. *IEEE Transactions on Software Engineering*, 33(4), 225-237. <https://doi.org/10.1109/TSE.2007.38>
- Mairiza, D., Zowghi, D., & Nurmuliani, N. (2010). An investigation into the notion of non-functional requirements. *ACM Symposium on Applied Computing*, 311-317. <https://doi.org/10.1145/1774088.1774153>



- Mbonu, I. S., Aliliele, C., Iwuanyanwu, U., & Oluoha, O. M. (2018). A conceptual framework for legal and ethical risk modeling in enterprise data protection governance systems. *Iconic Research and Engineering Journals*, 2(2), 207-226. <https://doi.org/10.64388/IREV2I2-1714911>
- Mbonu, I. S., Aliliele, C., Iwuanyanwu, U., & Uzoka, E. (2021b). Advances in artificial intelligence techniques for secure software testing and automated regression control mechanisms. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 7(5), 468-496. <https://doi.org/10.32628/CSEIT217565>
- Michael, O. N., & Ogunsola, O. E. (2025c). Assessing the role of artificial intelligence in transforming decision-making across modern agricultural systems. *Engineering and Technology Journal*, 10(12), 8027-8048. <https://doi.org/10.47191/etj/v10i12.06>
- Michael, O. N., & Ogunsola, O. E. (2023c). Development of predictive data models for enhancing crop resilience and climate adaptation in agrarian economies. *International Journal of Advanced Multidisciplinary Research and Studies*, 3(1), 1510-1527. <https://doi.org/10.62225/2583049X.2023.3.1.5143>
- Miller, M., Reichelstein, J., Salas, C., & Zia, B. (2015). Can you help someone become financially capable? *World Bank Research Observer*, 30(2), 220-246. <https://doi.org/10.1093/wbro/lkv009>
- Myers, G. J., Sandler, C., & Badgett, T. (2011). *The Art of Software Testing* (3rd ed.). Wiley.
- Nwakamma, S., Ojukwu, J., & Oyesiji, S. O. (2024b). Securing agentic AI enterprise workflows against prompt injection, tool poisoning, memory manipulation, and excessive agency threats. *World Journal of Innovation and Modern Technology*, 8(6), 217-248. <https://doi.org/10.56201/wjimt.v8.no6.2024.pg217.248>
- Obogo, S. F., Nwafor, M. I., & Ozobu, C. O. (2023). Conceptual leadership model for safety culture development in construction and engineering projects. *International Journal of Scientific Research in Civil Engineering*, 7(6), 121-153. <https://doi.org/10.32628/IJSRCE237554>
- Obogo, S. F., Ozobu, C. O., & Uduokhai, D. O. (2019a). Advances in leadership driven safety culture transformation in large construction workforces. *Iconic Research and Engineering Journals*, 3(5), 507-523. <https://doi.org/10.64388/IREV3I5-1715497>
- Obogo, S. F., Ozobu, C. O., & Uduokhai, D. O. (2019b). Development of a construction safety risk governance model for multi contractor high rise projects. *Iconic Research and Engineering Journals*, 2(9), 502-522. <https://doi.org/10.64388/IREV2I9-1715495>
- Obogo, S. F., Uduokhai, D. O., & Ozobu, C. O. (2019). Systematic review of hazard identification and risk control practices in urban construction projects. *Iconic Research and Engineering Journals*, 2(12), 666-681. <https://doi.org/10.64388/IREV2I12-1715496>
- Obriki, O. D., & Arumosoye, O. M. (2019). A conceptual framework linking management safety walkthrough frequency and coverage to safety culture outcomes in mega projects. *Iconic Research and Engineering Journals*, 2(8), 355-374. <https://doi.org/10.64388/IREV2I8-1714416>
- Obriki, O. D., & Arumosoye, O. M. (2020). Conceptual framework for human error causation in high-risk construction and industrial activities. *International Journal of Multidisciplinary Research and Growth Evaluation*, 1(5), 715-727. <https://doi.org/10.54660/IJMRGE.2020.1.5.715-727>
- Obriki, O. D., Arumosoye, O. M., & Ozobu, C. O. (2025). Conceptual model linking leading safety signals to sustained injury-free project performance. *International Journal of Scientific Research in Humanities and Social Sciences*, 2(3), 233-254. <https://doi.org/10.32628/IJSRHSS252342>



- Odejobi, O. D., Okonkwo, C. S., Ahiaeke Patrick, M. C., Okeke, O. T., & Mayo, W. (2025). AI-augmented secure software engineering: Leveraging deep learning for autonomous threat detection and mitigation. *International Journal of Engineering and Modern Technology*, 11(12), 101-121. <https://doi.org/10.56201/ijemt.vol.11.no12.2025.pg101.121>
- Ogbete, J. C., Aminu-Ibrahim, A. Y., & Ambali, K. B. (2023). Lifecycle performance evaluation of purpose built diagnostic laboratories supporting long term healthcare delivery. *International Journal of Advanced Multidisciplinary Research and Studies*, 3(6), 2605-2621. <https://doi.org/10.62225/2583049X.2023.3.6.5698>
- Ogundairo, K. M., & Ayivi-Donkor, S. S. (2023a). Machine learning approaches to customer acquisition and retention: A framework for predictive lifecycle management. *IIARD International Journal of Economics and Business Management*, 9(10), 210-246. <https://doi.org/10.56201/ijebm.v9.no10.2023.pg210.246>
- Okonkwo, C. S., Agbabiaka, J., Mayo, W., & Okeke, O. T. (2024e). Review of digital supply chain models for cost control and operational continuity. *International Journal of Advanced Multidisciplinary Research and Studies*, 4(6), 2836-2846. <https://doi.org/10.62225/2583049X.2024.4.6.5496>
- Oluwadele, J. F., Samuel, O., Tawose, O. M., Ekeocha, A. H., & Adika, A. O. (2025). Evaluation of alternative energy sources to replace maize in Marshall broiler diets: Effects on growth performance, meat quality, and serum biochemistry. *EUREKA: Life Sciences*. <https://doi.org/10.21303/2504-5695.2025> [Author list truncated in source; verify full authorship.]
- Oluwo, K., Bola-Sadipe, D., & Aliliele, C. (2025). Conceptual framework for data analytics driven financial decision making in banking sector organizations. *Shodhshauryam, International Scientific Refereed Research Journal*, 8(5), 259-302. <https://doi.org/10.32628/SHISRRJ258529>
- Ominyi, M., & Anichukwueze, C. C. (2023). A review of algorithmic accountability and model risk management approaches in financial services. *Journal of Accounting and Financial Management*, 9(12), 218-238. <https://doi.org/10.56201/jafm.v9.no12.2023.pg218.238>
- Ominyi, M., Anichukwueze, C. C., & Uzougbo, N. S. (2025a). A conceptual framework for multi-agent AI quality control in the review of regulated documents. *International Journal of Social Sciences and Management Research*, 11(8), 534-557. <https://doi.org/10.56201/ijssmr.vol.11no8.2025.pg534.557>
- Ominyi, M., Anichukwueze, C. C., & Uzougbo, N. S. (2024b). A systematic review of organizational readiness for the EU Artificial Intelligence Act in multinational enterprises. *World Journal of Innovation and Modern Technology*, 8(6), 185-203. <https://doi.org/10.56201/wjimt.v8.no6.2024.pg185.203>
- Ominyi, M. (2021). Toward a conceptual framework for embedding privacy by design in corporate restructuring and transaction structuring. *Journal of Accounting and Financial Management*, 7(5), 152-173. <https://doi.org/10.56201/jafm.vol.7.no5.2021.p152.173>
- Onche, V. O., Adegbite, M. P., & Ogbonna, C. S. (2023). BYOD adoption and information-security risk in university administrative offices: Evidence from Nigerian federal universities. *International Journal of Multidisciplinary Research and Growth Evaluation*, 4(6), 1549-1563. <https://doi.org/10.54660/IJMRGE.2023.4.6.1549-1563>
- Page, M. J., et al. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ*, 372, n71. <https://doi.org/10.1136/bmj.n71>
- Ramesh, B., & Jarke, M. (2001). Toward reference models for requirements traceability. *IEEE Transactions on Software Engineering*, 27(1), 58-93. <https://doi.org/10.1109/32.895989>



- Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14, 131-164. <https://doi.org/10.1007/s10664-008-9102-8>
- Sakyi, J. K., Filani, O. M., Nnabueze, S. B., Okojie, J. S., & Ogedengbe, A. O. (2022b). Developing KPI frameworks to enhance accountability and performance across large-scale commercial organizations. *Journal of Frontiers in Multidisciplinary Research*, 3(2), 81-93. <https://doi.org/10.54660/IJFMR.2022.3.2.81-93>
- Sakyi, J. K., Nnabueze, S. B., Filani, O. M., & Okojie, J. S. (2023). Revenue assurance strategies leveraging artificial intelligence and big data in service-intensive organizations. *International Journal of Multidisciplinary Evolutionary Research*, 4(2), 58-75. <https://doi.org/10.54660/IJMER.2023.4.2.58-75>
- Sakyi, J. K., Nnabueze, S. B., Filani, O. M., & Okojie, J. S. (2023). Revenue assurance strategies leveraging artificial intelligence and big data in service-intensive organizations. *International Journal of Multidisciplinary Evolutionary Research*, 4(2), 58-75. <https://doi.org/10.54660/IJMER.2023.4.2.58-75>
- Sanni, J. O., Atima, M. E., & Attah, A. (2022). Systematic review of attribution modeling methods resolving bias in multi-touch journeys. *Shodhshauryam, International Scientific Refereed Research Journal*, 5(1), 304-334. <https://doi.org/10.32628/SHISRRJ247135>
- Sanni, J. O., Iwuanyanwu, U. A., Essien, M. A., & Attah, A. (2023). Lifecycle-aware marketing automation using federated learning for secure cross-organizational data management. *Gyanshauryam, International Scientific Refereed Research Journal*, 6(6), 337-364. <https://doi.org/10.32628/GISRRJ236642>
- Sanni, J. O., Iwuanyanwu, U. A., & Essien, M. A. (2025). Problem-oriented process mining for auditable marketing automation lifecycle control. *International Journal of Advanced Multidisciplinary Research and Studies*, 5(6), 1933-1947. <https://doi.org/10.62225/2583049X.2025.5.6.5650>
- Sanni, J. O., & Wedraogo, L. (2024). Data-centric funnel optimization frameworks resolving revenue leakage in high-value services. *International Journal of Advanced Multidisciplinary Research and Studies*, 4(6), 2859-2874. <https://doi.org/10.62225/2583049X.2024.4.6.5641>
- Shittu, M. A., Shittu, H. A., Adeleke, O. J., & Adedokun, O. J. (2023). Digital twin modeling for real time monitoring and fault detection in smart substations. *International Journal of Industrial Engineering Research and Development*, 14(2), 25-44. https://doi.org/10.34218/IJIERD_14_02_003
- Tawose, O. M., Oluwadele, J. F., Ekeocha, A. H., Odumboni, A. A., & Egbeyemi, O. S. (2023). Aflatoxin detection and quantification in poultry feeds available in selected areas in Ekiti State. *FUOYE Journal of Agriculture and Human Ecology*, 7(2). <https://doi.org/10.62923/fuojahe.v7i2.302>
- Tonoyan, A., Dada, O., & Ayivi-Donkor, S. S. (2023). A conceptual framework for retail asset valuation: Integrating quantitative pricing and multi-unit financial performance prediction. *Journal of Accounting and Financial Management*, 9(12), 218-250. <https://doi.org/10.56201/jafm.v9.no12.2023.pg218.250>
- Tonoyan, A., Dada, O., & Ayivi-Donkor, S. S. (2022b). Algorithmic process optimization and cost reduction: A review of simulation modeling and financial impact assessment. *Journal of Accounting and Financial Management*, 8(8), 139-169. <https://doi.org/10.56201/jafm.vol.8.no8.2022.p139.169>
- Tonoyan, A., Dada, O., & Ayivi-Donkor, S. S. (2024b). Real-time KPI tracking systems: A review of automated performance monitoring and data-driven decision making. *World Journal of Innovation and Modern Technology*, 8(6), 247-281. <https://doi.org/10.56201/wjimt.v8.no6.2024.pg247.281>



- Tonoyan, A., Dada, O., & Ayivi-Donkor, S. S. (2022c). Conceptual model for predictive cost optimization: Machine learning applications in business transformation analysis. *International Journal of Economics and Business Management*, 8(6), 68-102. <https://doi.org/10.56201/ijebm.v8.no6.2022.pg68.102>
- Uchechi, N. (2018). Mitigating enterprise financial exposure: Standardization of stamp duty policies and class action defense protocols. *Iconic Research and Engineering Journals*, 2(5). <https://doi.org/10.64388/IREV2I5-1722788>
- Uchechi, N., Ominyi, M., Anichukwueze, C. C., & Uzougbo, N. S. (2022). Advances in quantitative evaluation models for measuring corporate diversity and inclusion investments. *World Journal of Finance and Investment Research*, 6(2), 56-101. <https://doi.org/10.56201/wjfir.v6.no2.2022.pg56.101>
- Uchechi, N., Ominyi, M., & Jones, B. C. (2018). Procedural standardization and workflow optimization in high-volume dispute resolution clinics: A systematic review. *Iconic Research and Engineering Journals*, 2(6). <https://doi.org/10.64388/IREV2I6-1722789>
- Wedraogo, L., & Sanni, J. O. (2024). Machine learning models addressing uncertainty in cross-channel campaign performance forecasting accuracy. *International Journal of Advanced Multidisciplinary Research and Studies*, 4(6), 2875-2890. <https://doi.org/10.62225/2583049X.2024.4.6.5648>
- Wieringa, R. J. (2014). *Design Science Methodology for Information Systems and Software Engineering*. Springer. <https://doi.org/10.1007/978-3-662-43839-8>
- Winkler, S., & von Pilgrim, J. (2010). A survey of traceability in requirements engineering and model-driven development. *Software and Systems Modeling*, 9, 529-565. <https://doi.org/10.1007/s10270-009-0145-0>
- Yusuf, H. D., & Abdulrazaq, K. (2018). A conceptual framework for strengthening diagnostic quality systems in resource-limited clinical laboratories. *Iconic Research and Engineering Journals*, 1(11), 145-179. <https://doi.org/10.64388/IREV1I11-1723090>
- Zhu, H., Hall, P. A. V., & May, J. H. R. (1997). Software unit test coverage and adequacy. *ACM Computing Surveys*, 29(4), 366-427. <https://doi.org/10.1145/267580.267590>