



SR-Screen-AI: An Explainable Rule-Based AI-Assisted Decision Support System for Human-in-the-Loop Systematic Literature Review Screening

Sruthi R¹, Rajeev P R²

Department of Electrical and Computer Engineering, College of Engineering Kidangoor, Kerala, India¹

Graduate School, Mahatma Gandhi University, Kottayam, Kerala, India²

Abstract: Systematic literature review screening requires assessment of bibliographic records against predefined criteria. This paper presents SR-Screen-AI, an explainable rule-based AI-assisted decision-support system integrating import, duplicate detection, weighted screening, explanations, reviewer validation, and reporting. Eight software tests passed, and a 50-record baseline trial achieved 81.82% precision, 90.00% recall, and 85.71% F1-score, supporting transparent reviewer-controlled screening.

Keywords: Artificial intelligence, Decision support systems, Explainable artificial intelligence, Human-in-the-loop, Literature screening, Rule-based systems, Systematic literature review

I. INTRODUCTION

Systematic literature reviews (SLRs) provide a structured method for identifying, evaluating, and synthesizing evidence according to predefined research questions and eligibility criteria [1]–[5]. Their methodological value depends not only on the completeness of the search strategy but also on the consistency with which retrieved records are screened before detailed assessment. As the volume of scholarly literature increases, title and abstract screening can become a substantial operational burden, motivating the development of computational support for evidence-synthesis workflows [15], [26]–[29].

Early software systems focused on bibliographic management, duplicate identification, collaboration, and documentation. Subsequent approaches introduced interactive machine learning and active learning to prioritize records for reviewer attention. Abstrackr demonstrated interactive machine learning for evidence-based screening [19], ASReview introduced an open-source active-learning framework for systematic-review screening [17], and Rayyan provided a structured environment for semi-automated screening and review collaboration [18]. More recent work has examined broader automation of systematic reviews and the use of artificial intelligence across screening and other evidence-synthesis stages [13]–[16], [22].

These developments establish that computational assistance can be integrated into systematic-review screening, but they also raise a distinct design question: how should a recommendation be represented so that the reviewer can inspect its basis before recording the final decision? In evidence synthesis, a computational label and a methodological eligibility decision are not necessarily interchangeable. The distinction becomes particularly important when a reviewer must justify why a record was included or excluded after applying a protocol-specific set of criteria.

Explainable artificial intelligence (XAI) addresses the broader problem of making computational outputs understandable to their intended users [6]–[12]. The literature distinguishes intrinsically interpretable decision mechanisms from post-hoc explanation techniques applied to complex predictive models [7], [9], [10]. Rule-based approaches are relevant to this distinction because the criteria themselves can be represented explicitly. When the evidence, scoring contributions, and resulting recommendation are preserved, the reviewer can inspect the computational pathway rather than infer it from an opaque model output.

Human-centered AI provides a complementary perspective by emphasizing user control, transparency, appropriate allocation of responsibilities, and meaningful interaction between computational systems and people [23]–[25]. These principles are directly relevant to systematic-review screening because eligibility may depend on contextual characteristics of a study, including its objective, population, methodology, intervention, and interpretation. An AI-assisted system should therefore support reviewer judgment rather than silently replace it.

Against this background, the present study addresses a focused software-engineering problem: how can AI-assisted screening recommendations be generated from explicit evidence while preserving reviewer control over the final



decision? We developed SR-Screen-AI as an explainable rule-based AI-assisted decision-support system for this purpose. The application integrates CSV, RIS, and NBIB bibliographic import, record normalization, duplicate detection, configurable keyword screening, weighted scoring, confidence calculation, explanation generation, reviewer validation, and structured reporting.

The principal contributions are fourfold. First, the study presents a modular screening architecture that separates bibliographic processing, duplicate analysis, recommendation generation, reviewer interaction, and reporting. Second, it implements an explicit weighted keyword mechanism in which inclusion and exclusion evidence remains visible at the record level. Third, it maintains separate computational recommendation and reviewer-decision states, allowing human override without destroying the original recommendation. Fourth, it provides functional verification of the current implementation and establishes an evaluation boundary that distinguishes software behavior from predictive screening validity.

The study combines functional software verification with an initial baseline validation trial. The functional evaluation establishes consistency of the implemented screening, configuration, and reporting mechanisms, whereas the 50-record trial provides an initial empirical comparison against independently established human screening labels. Predictive and human-centered findings are interpreted as preliminary because the baseline sample is limited in size and does not constitute a large-scale comparative benchmark.

II. RELATED WORK

A. Systematic Literature Review Screening

SLR methodologies provide explicit procedures for literature identification, selection, assessment, and synthesis [1]–[5]. Screening is critical because decisions at this stage determine which records progress to subsequent assessment. The methodological literature consequently emphasizes reproducibility, predefined criteria, and transparent reporting. Software can support these requirements by preserving bibliographic information and decision history, but it does not remove the need for protocol-driven human judgment.

The increasing scale of bibliographic retrieval has encouraged the development of tools that reduce repetitive screening operations. Evaluations of screening software show substantial variation in collaboration, duplicate management, automation, and review-management capabilities [18], [21]. This diversity suggests that screening assistance is not a single technical problem; different systems make different trade-offs between automation, interaction, and transparency.

B. AI-Assisted Literature Screening

AI-assisted screening has progressed from interactive machine learning to active-learning workflows and broader automation frameworks. Abstrackr demonstrated iterative use of reviewer feedback during evidence-based screening [19]. ASReview subsequently formalized active-learning support for systematic-review screening, combining model-based prioritization with human labeling [17]. Rayyan provides another workflow-oriented approach that incorporates semi-automated screening features within a review-management environment [18].

Recent studies and reviews have expanded the discussion to machine learning, natural-language processing, and broader AI-based automation [13]–[16], [20], [22]. These studies indicate that automated screening can be technically feasible, but also emphasize the importance of task characteristics, review criteria, evaluation design, and transparent reporting. The present work therefore does not claim that rule-based screening is universally preferable to learned screening. Instead, it investigates a complementary design in which recommendation evidence is explicit and reproducible from a configured rule set.

C. Explainable Artificial Intelligence for Decision Support

XAI encompasses methods that expose evidence, model behavior, decision factors, or other information that helps users understand computational outputs [6]–[12], [26]–[28]. The literature also cautions that explanation quality should not be conflated with predictive correctness. An explanation can faithfully describe the computation while the underlying model remains inappropriate for the task [7], [9].

For SR-Screen-AI, explainability is treated as an intrinsic property of the screening mechanism. The recommendation is derived from explicit keyword evidence and weighted rules; the matched terms and resulting scores are retained; and the explanation is generated from these structured fields. This differs from applying a separate post-hoc explainer to an opaque predictive classifier. The objective is not to claim that the rule set captures all semantic aspects of a paper, but to make the implemented reasoning directly inspectable.



D. Human-Centered and Human-in-the-Loop AI

Human-AI interaction research emphasizes user control, communication of system capabilities, support for correction, and appropriate distribution of responsibility [23]–[25]. In evidence synthesis, these principles are relevant because the reviewer remains accountable for eligibility decisions. Existing active-learning systems already provide forms of interaction, but the computational mechanism typically changes as reviewer labels are incorporated into subsequent model behavior [17].

SR-Screen-AI adopts a different interaction pattern. The current screening pipeline does not learn from reviewer decisions during execution. Instead, it generates a deterministic recommendation from the configured keyword dictionary and retains that recommendation as a separate state from the reviewer-confirmed decision. This design allows the reviewer to accept, modify, or reject the recommendation without erasing the computational evidence that produced it.

E. Research Gap and Motivation

The literature therefore establishes three relevant facts. First, computational screening assistance is well established [17]–[22]. Second, explainability and human control are important considerations in AI-assisted decision support [6]–[10], [23]–[25]. Third, recent reviews identify interpretability, evaluation, and workflow integration as continuing challenges for AI-supported systematic-review automation [13]–[16]. The gap addressed by this paper is narrower than a general claim of missing explainability: it concerns the integration of explicit rule-based evidence, inspectable recommendation generation, reviewer-controlled validation, and reporting within one operational screening workflow. SR-Screen-AI addresses this gap as a deterministic decision-support system rather than as another predictive benchmark model. Its research contribution is therefore architectural and methodological: it demonstrates how an interpretable rule mechanism can be embedded in a complete screening workflow while preserving separate computational and human decision states.

III. SR-SCREEN-AI SYSTEM

A. Data Acquisition and Preprocessing

SR-Screen-AI accepts CSV, RIS, and PubMed NBIB records and converts them into a common internal paper representation. The parsers map source-specific fields to title, abstract, authors, keywords, journal, year, persistent identifiers, and other bibliographic attributes. This normalization allows downstream components to operate on a stable representation rather than on the syntax of an individual bibliographic format.

For each record, the application maintains bibliographic, duplicate, screening, and reviewer states. The logical record structure is represented by (1).

$$P_i = \{T_i, A_i, K_i, U_i, D_i, S_i, R_i\} \quad (1)$$

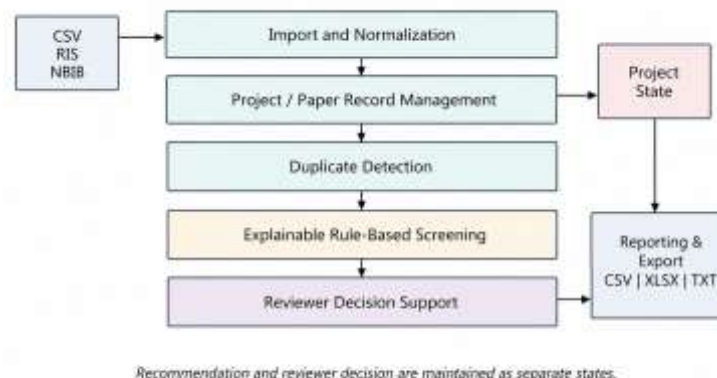


Fig. 1. Layered architecture of SR-Screen-AI.

B. Duplicate Detection

Duplicate detection is treated as a preprocessing stage so that repeated representations of a publication do not independently enter the screening queue. The implementation first compares DOI, PMID, and ISBN values. If these



identifiers do not establish a match, normalized-title similarity is calculated using a sequence-matching method. Author overlap and publication year provide additional evidence for probable duplication when title similarity is very high. Possible duplicates are retained as inspectable candidates rather than being treated as automatically ineligible records.

The title similarity measure is defined by (2), where $N(\cdot)$ converts the title to lowercase, removes non-alphanumeric characters, and normalizes whitespace. The current implementation classifies a pair as a probable duplicate when similarity is at least 0.95 and author overlap and publication year also agree; similarity of at least 0.85 without these additional conditions is treated as possible duplication.

$$S_i(P_i, P_j) = \text{SeqMatch}(N(T_i), N(T_j)) \quad (2)$$

C. Explainable Screening Engine

The operational screening pipeline uses a configurable keyword dictionary containing separate inclusion and exclusion terms. The current default dictionary contains seven inclusion terms and four exclusion terms.

TABLE I.
DEFAULT SCREENING RULE CATEGORIES

Category	Term	Weight
Inclusion	machine learning	5
Inclusion	artificial intelligence	5
Inclusion	deep learning	4
Inclusion	explainable AI	8
Inclusion	systematic review	3
Domain relevance	electronic health record	8
Domain relevance	clinical decision support	6
Exclusion	editorial	5
Exclusion	book review	5
Exclusion	conference announcement	5
Exclusion	preface	5

Each term has an explicit integer weight. The exclusion side contains editorial, book review, conference announcement, and preface, each with weight five.

For a record P_i , matched inclusion and exclusion keywords form the sets I_i and E_i . The weighted evidence scores are defined in (3). The recommendation is INCLUDE when $S_i^+ > S_i^-$, EXCLUDE when $S_i^- > S_i^+$, MAYBE when $S_i^+ = S_i^- > 0$, and UNDECIDED when both scores are zero. The rule is deterministic for a fixed dictionary and record. The decision path is illustrated in Fig. 2.

$$S_i^+ = \sum_{k \in I_i} w_k^+ ; \quad S_i^- = \sum_{k \in E_i} w_k^- \quad (3)$$

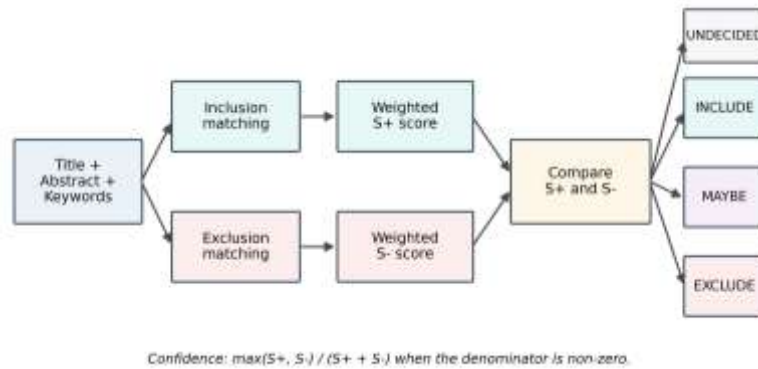


Fig. 2. Rule-based evidence and scoring path.

D. Confidence and Explanation

The screening score is accompanied by the evidence-derived confidence value defined in (4) when the denominator is non-zero; otherwise $C_i = 0$. The value summarizes the dominance of the stronger evidence category and is stored as part of the screening result. It is not a calibrated probability of eligibility and is not interpreted as predictive accuracy.

$$C_i = \max(S_i^+, S_i^-) / (S_i^+ + S_i^-) \quad (4)$$

The explanation engine constructs a human-readable report from the recommendation, confidence, matched inclusion and exclusion terms, triggered rules, and score information. The explanation is therefore generated from the same structured evidence used by the screening pipeline rather than from a separate opaque model. This implements the principle of intrinsic inspectability discussed in the XAI literature [6], [7], [9], [10].



Fig. 3. Example of a human-readable screening explanation generated by SR-Screen-AI.

An example of the resulting human-readable explanation output is shown in Fig. 3.

E. Human-in-the-Loop Review

AI-assisted recommendation and reviewer decision are stored as separate states. The reviewer interface permits explicit INCLUDE, EXCLUDE, MAYBE, or UNDECIDED decisions and records a reason or comment where appropriate. A record can consequently contain a computational recommendation while remaining pending final review. This state separation is central to the human-in-the-loop design and follows the principle that computational assistance should remain subject to meaningful user control [23]–[25].

F. Reporting and Auditability

The project-level reporting logic distinguishes imported records, duplicate records, unique records, AI recommendations, explicit reviewer decisions, reviewed records, and pending final review. Duplicate records are marked



as non-screenable for the aggregate unique-record statistics. AI recommendation counts are calculated only over unique records, while final reviewer counts require an explicit recorded decision. This prevents an untouched record from being represented as a completed reviewer decision.

Record-level exports preserve bibliographic identifiers, duplicate status, AI recommendation, inclusion and exclusion scores, confidence, matched terms, triggered rules, explanation, reviewer decision, reviewer comment, and review status. The complete operational sequence is summarized in Fig. 4.

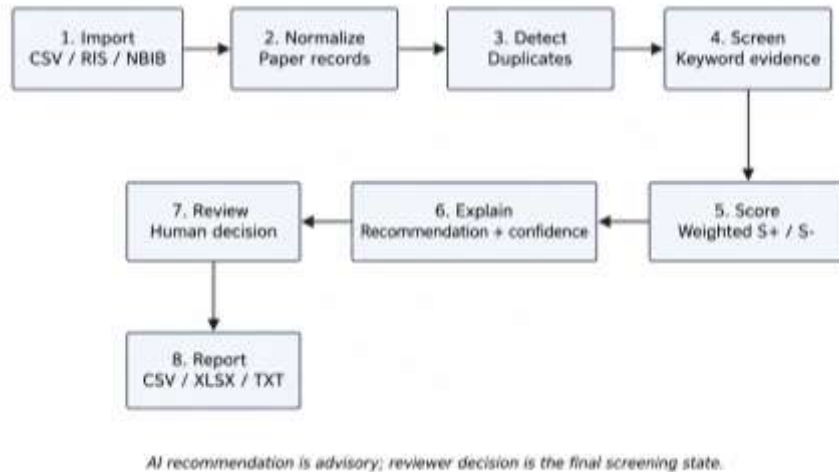


Fig. 4. End-to-end operational workflow.

IV. IMPLEMENTATION

A. Prototype Configuration

The current release is implemented in Python as a modular desktop application. The package structure separates core data models, bibliographic parsers, screening components, graphical-user-interface views, project management, and report export. The graphical interface is organized around dedicated views for import, records, duplicate detection, screening, reviewer decisions, reporting, and settings.

TABLE II.
CORE SR-SCREEN-AI SOFTWARE MODULES

Module	Primary responsibility	Output
Import/parsers	CSV, RIS, NBIB parsing and field mapping	Normalized records
Project/Paper model	Record and project state management	Structured paper objects
Duplicate detector	Identifier and title-similarity analysis	Duplicate status
Keyword engine	Evidence matching and rule activation	Matched terms/rules
Scoring engine	Weighted inclusion/exclusion evaluation	Scores, recommendation, confidence
Explanation engine	Human-readable representation	Screening explanation



Module	Primary responsibility	Output
Reviewer view	Human validation and override	Final reviewer state
Report exporter	CSV, Excel, and text output	Review reports

B. Explainability Representation

Each screening result stores the matched inclusion keywords, matched exclusion keywords, triggered rules, recommendation, confidence, and explanation. The resulting object can be serialized and reconstructed, which supports persistence and testability. The explanation engine also provides a concise summary containing the decision, confidence, and counts of matched evidence. An example of the human-readable explanation generated from these fields is shown in Fig. 3.

The current architecture also contains a separate RuleEngine and ConfidenceEngine used by the project test fixtures. The operational ScreeningPipeline, however, invokes KeywordEngine, ScoringEngine, and ExplanationEngine directly. This distinction matters for reproducibility: the paper reports the behavior of the current screening pipeline rather than treating every helper module in the repository as part of the active execution path.

C. Interface and Workflow

The Dashboard provides project statistics; the Paper Records view exposes imported metadata; Duplicate Detection presents duplicate matches; the Screening view executes recommendation generation; the Reviewer view records human decisions; and the Reports view exports project and record-level information. The interface therefore mirrors the logical processing pipeline and allows users to inspect a record before and after recommendation generation without directly interacting with the Python implementation.

The reporting implementation uses the project-level state model to compute non-overlapping aggregate counts. In particular, unique-record counts exclude duplicates; AI recommendation counts are restricted to unique records; and reviewed counts require the reviewer-decision flag to be explicitly recorded. These rules preserve the distinction between a computational recommendation and a completed review action.

V. EXPERIMENTAL METHODOLOGY

A. Test Strategy

The evaluation treats SR-Screen-AI as a software system and defines a Functional Behavior Profile (FBP) rather than a predictive benchmark. The latest project release was first enumerated with `pytest --collect-only -q` and then executed with `pytest -q`. The collection step produced eight executable tests, and the complete run produced eight passes and zero failures.

The eight collected tests comprise five tests for confidence and configuration behavior and three tests for project reporting. The confidence tests cover the no-evidence case, an equal-evidence tie, two asymmetric evidence cases, and configuration round-trip behavior. The reporting tests verify separation of AI recommendation and reviewer decision states, duplicate and explicit-undecided reporting invariants, and the treatment of duplicate records in exported status.

B. Functional Behavior Profile

TABLE III.
FUNCTIONAL TEST RESULTS

Test group	Checks	Passed	Failed
Confidence and configuration	5	5	0
Project reporting	3	3	0
Total	8	8	0



The verified confidence behavior follows the configured scoring relation. With no matched evidence, the recommendation is UNDECIDED and confidence is zero. Equal non-zero evidence produces MAYBE at confidence 0.50. In the asymmetric test cases, inclusion evidence of 10 versus exclusion evidence of 2 produces INCLUDE with 0.83 confidence, while 1 versus 9 produces EXCLUDE with 0.90 confidence. These values are evidence-dominance scores, not probabilities of screening correctness.

The reporting tests verify that a paper can carry an AI recommendation without acquiring an implicit reviewer decision. They also verify that duplicate records are not counted as pending final review and that explicit undecided reviewer states remain distinguishable from the absence of a reviewer decision.

C. Initial Baseline Validation Trial

To establish an initial empirical baseline prior to broader evaluation, a dedicated validation subset of 50 unique bibliographic records was screened independently by two human reviewers. The reference screening identified 20 eligible records and 30 ineligible records. The independently established labels were used as the reference standard for evaluating the deterministic recommendation mechanism of SR-Screen-AI. Comparison with the system output yielded 18 true positives (TP), 4 false positives (FP), 2 false negatives (FN), and 26 true negatives (TN). Accordingly, precision was 81.82%, recall was 90.00%, F1-score was 85.71%, and specificity was 86.67%.

The human-centered component of the trial also recorded screening time and inter-rater agreement. Mean manual screening time was 42 s per record, whereas the SR-Screen-AI-assisted condition required 24 s per record, corresponding to a 42.86% reduction in mean time-on-task. Prior to final conflict resolution, the two reviewers achieved a Cohen's kappa of 0.81, indicating substantial agreement under the specified protocol. These findings provide an initial empirical characterization of the system; broader multi-reviewer and cross-domain studies are required to assess generalizability.

TABLE IV.
EMPIRICAL CRITERIA FRAMEWORK

Metric	Definition	Current Status
Precision	$TP / (TP + FP)$	81.82%
Recall	$TP / (TP + FN)$	90.00%
F1-score	$2PR / (P + R)$	85.71%
Specificity	$TN / (TN + FP)$	86.67%
Reviewer agreement	Cohen's kappa	0.81
Screening time	Average time per record	24 s/record (manual: 42 s; 42.86% lower)

D. Evaluation Boundary

The 382-record IEEE Xplore set established for the project remains a demonstration corpus rather than a gold-standard benchmark. The initial 50-record validation subset provides the first independently reviewed empirical reference used in this study, but it is not intended to represent the full diversity of systematic-review topics or bibliographic sources. The baseline findings should therefore be interpreted as an initial validation anchor rather than as definitive evidence of general predictive performance.

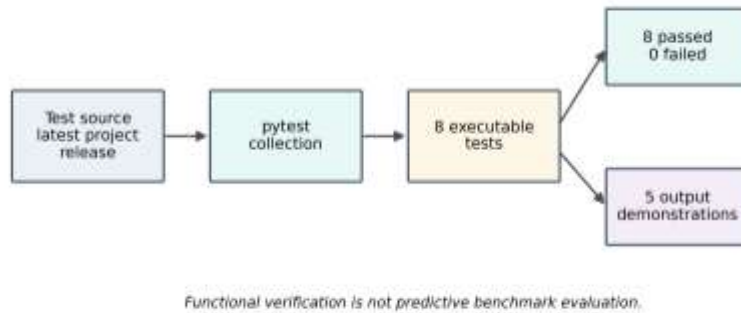


Fig. 5. Functional validation pathway used for the prototype.

VI. RESULTS AND DISCUSSION

A. Functional Test Results

The latest project release completed all eight collected automated tests without failure. The result establishes that the tested confidence, configuration, and reporting invariants hold under the supplied conditions. In addition to these software-level checks, the initial 50-record baseline validation yielded 18 true positives, 4 false positives, 2 false negatives, and 26 true negatives, corresponding to 81.82% precision, 90.00% recall, 85.71% F1-score, and 86.67% specificity. The pilot therefore provides an initial empirical baseline in addition to the functional verification results.

The project archive also contains generated review outputs that demonstrate the reporting semantics across several input scenarios. A 10-record demonstration produced 8 AI INCLUDE recommendations and 2 AI UNDECIDED recommendations with no reviewer decisions. A 100-record RIS demonstration produced 100 AI INCLUDE recommendations. A 6-record demonstration produced 5 INCLUDE and 1 UNDECIDED recommendation. A 146-record NBIB demonstration contained 2 duplicates and 144 unique records, with 144 AI INCLUDE recommendations. A 73-record demonstration produced 67 INCLUDE and 6 UNDECIDED recommendations. These counts are software-behavior demonstrations; they are not accuracy estimates and must not be interpreted as predictive performance.

B. Screening-Pipeline Behavior

The screening pipeline provides a deterministic computational pathway from record text to recommendation. A fixed title, abstract, keyword set, and scoring configuration yields the same recommendation and evidence-derived confidence on repeated execution. This reproducibility is useful for auditing because a reviewer can reconstruct the computational output from the configured rule dictionary without relying on a hidden model state.

The results also demonstrate that the system can preserve evidence at the record level. Matched inclusion and exclusion terms remain available alongside the recommendation, and the explanation engine incorporates these values into a human-readable report. The resulting representation is suitable for review workflows in which the reviewer must inspect why a record was recommended for further consideration.

C. Human Oversight

The reporting invariants reinforce the human-in-the-loop interpretation of the system. AI recommendation counts and final reviewer counts are maintained independently, and an untouched record remains pending rather than being silently classified as a final undecided decision. This behavior is consistent with human-AI interaction guidance emphasizing user control and the ability to correct system outputs [23]–[25].

The system nevertheless does not solve the more difficult methodological problem of determining whether a rule-based recommendation captures the full semantic meaning of a publication. A reviewer may therefore disagree with a recommendation even when the computational reasoning is internally consistent. The present architecture is designed to preserve that disagreement as a valid part of the workflow rather than treating it as an exception to the system.

D. Comparison With Existing Approaches

ASReview, Abstrackr, and related approaches establish strong precedents for integrating computational assistance with human review [17], [19]. The main distinction of SR-Screen-AI lies in its current screening mechanism and explanation representation. The active-learning approach used by ASReview continually exploits reviewer labels to



prioritize subsequent records [17], whereas SR-Screen-AI applies a configured keyword dictionary and deterministic weighted scoring. Rayyan focuses on review management and semi-automated screening within a collaborative environment [18]. SR-Screen-AI instead emphasizes a locally inspectable rule path and record-level evidence representation.

This comparison is deliberately complementary rather than competitive. The present pilot is limited to 50 independently reviewed records and was not designed as a common benchmark against competing platforms. Accordingly, the study does not establish that one approach is more accurate, faster, or less burdensome than another. The defensible comparison remains architectural: SR-Screen-AI provides an explicit rule-based evidence path and separates recommendation state from reviewer state within the same workflow.

E. Software Testing, Fixtures, and Reproducibility

The use of deterministic fixtures has two benefits. First, expected behavior is specified independently of the graphical interface, allowing core logic to be tested programmatically. Second, a fixed fixture can be rerun after software changes to detect unintended behavioral changes in recommendation, confidence, or reporting state. The current tests provide a starting regression suite, while broader property-based, malformed-input, and integration testing would be appropriate for subsequent releases.

The repository also contains standalone console-oriented test scripts for the CSV parser, duplicate detector, paper model, rule engine, screening pipeline, and screening-result model. These scripts provide additional implementation checks but were not included in the eight pytest-collected tests reported in this paper. Keeping this distinction explicit avoids inflating the formal automated-test count.

F. Practical Significance

The practical significance of SR-Screen-AI is its integration of several repetitive screening operations into one evidence-preserving workflow. The application imports heterogeneous bibliographic records, normalizes them, identifies duplicate relationships, generates configurable recommendations, records the supporting evidence, permits reviewer confirmation or override, and exports structured results. For reviews in which eligibility criteria can be represented reasonably well through explicit terminology, the design offers a reproducible way to operationalize those criteria while retaining reviewer authority.

The same design also exposes a limitation that is important for deployment. Because the screening engine relies on explicit vocabulary, its coverage depends on how well the configured terms represent the concepts of the review protocol. A transparent rule set can therefore be reproducible and still be incomplete. The appropriate practical role of the current release is assistance with inspection and prioritization, not autonomous eligibility adjudication.

G. Limitations and Threats to Validity

Several limitations constrain the interpretation of the present results. First, the initial validation sample contains only 50 records and therefore provides a preliminary rather than comprehensive assessment of predictive behavior. Second, the demonstration corpus remains broader than the independently adjudicated pilot and should not be treated as a gold-standard benchmark. Third, the rule vocabulary is topic-dependent and may miss semantically relevant records expressed using terminology absent from the configured dictionary. Fourth, the eight-test pytest suite is focused rather than exhaustive. Fifth, the human-centered trial involved two reviewers and therefore does not establish general usability or population-level workload effects. Finally, the evidence-derived confidence score is not statistically calibrated and should not be interpreted as a probability of eligibility. The present study is primarily a functional validation of a novel rule-based software architecture prototype. The 50-record trial represents a localized pipeline validation, while a larger statistical deployment benchmark is planned as Phase 2.

H. Future Work

A broader Phase 2 evaluation is proposed to extend the initial 50-record validation across independent review topics and larger adjudicated corpora. Two or more reviewers should independently screen the same fixed record set under a prespecified eligibility protocol, with disagreement resolved through adjudication after independent screening. The resulting benchmark would permit more stable estimates of precision, recall, F1-score, specificity, reviewer agreement, and subgroup behavior.

The human-centered component of Phase 2 should use a counterbalanced within-subject design, where feasible, with manual screening and SR-Screen-AI-assisted screening treated as separate conditions. For each record, screening time should be measured from task presentation to the reviewer's recorded decision, allowing mean and median time-on-task



to be compared across conditions. Reviewer agreement should be estimated before adjudication using Cohen's kappa, or an appropriate multcategory extension when INCLUDE, EXCLUDE, MAYBE, and UNDECIDED states are retained. Agreement, disagreement counts, confidence intervals, recommendation-override rates, and explanation-usefulness measures should be reported together rather than relying on a single performance statistic. This protocol would permit the effect of computational assistance on efficiency, agreement, and reviewer control to be examined without conflating these constructs with predictive accuracy. Further software development may investigate semantic matching, synonym expansion, natural-language processing, or active-learning components while preserving the existing explanation and independent recommendation/reviewer states. Comparative experiments should use identical corpora, eligibility criteria, and evaluation protocols before any superiority claim is considered.

VII. CONCLUSION

This paper presented SR-Screen-AI, an explainable rule-based AI-assisted decision-support system for human-in-the-loop systematic literature review screening. The system integrates bibliographic import, normalization, duplicate detection, configurable keyword evidence, weighted recommendation generation, confidence calculation, explanation generation, reviewer validation, and structured reporting within a modular desktop workflow.

The latest software release was verified with eight executable pytest tests; all eight passed. The verified behaviors include evidence-derived confidence calculation, configuration persistence, separation of AI-assisted and reviewer decision states, and reporting invariants for duplicates and pending review. Additional generated review reports demonstrated consistent handling of unique records, duplicates, recommendations, and review status across several input scenarios.

The principal contribution is not autonomous literature selection or predictive benchmarking. It is the integration of an explicitly inspectable rule mechanism with a reviewer-controlled screening workflow. By retaining the evidence and computational recommendation separately from the human decision, SR-Screen-AI provides a reproducible basis for subsequent review and audit without obscuring the role of expert judgment.

The current study provides an initial empirical baseline but does not establish generalized predictive accuracy, workload reduction, or superiority over existing screening platforms. The 50-record trial is best interpreted as a first empirical validation of the proposed recommendation mechanism and human-in-the-loop workflow. Larger independent corpora, additional reviewers, and controlled comparative studies are required before the observed results can be generalized.

ACKNOWLEDGMENT

The authors gratefully acknowledge Prof. Dr. P. S. Raghupathi for his encouragement, guidance, and support during the development of this research.

REFERENCES

- [1] M. J. Page et al., "The PRISMA 2020 statement: An updated guideline for reporting systematic reviews," *BMJ*, vol. 372, p. n71, 2021, doi: 10.1136/bmj.n71.
- [2] M. J. Page et al., "PRISMA 2020 explanation and elaboration: Updated guidance and exemplars for reporting systematic reviews," *BMJ*, vol. 372, p. n160, 2021, doi: 10.1136/bmj.n160.
- [3] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," *EBSE Technical Report EBSE-2007-01*, Keele University, 2007.
- [4] J. P. T. Higgins et al., Eds., *Cochrane Handbook for Systematic Reviews of Interventions*, 2nd ed. Chichester, U.K.: Wiley, 2019.
- [5] H. Snyder, "Literature review as a research methodology: An overview and guidelines," *J. Bus. Res.*, vol. 104, pp. 333-339, 2019, doi: 10.1016/j.jbusres.2019.07.039.
- [6] A. Barredo Arrieta et al., "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82-115, 2020, doi: 10.1016/j.inffus.2019.12.012.
- [7] G. Vilone and L. Longo, "Notions of explainability and evaluation approaches for explainable artificial intelligence," *Inf. Fusion*, vol. 76, pp. 89-106, 2021, doi: 10.1016/j.inffus.2021.05.009.
- [8] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138-52160, 2018, doi: 10.1109/ACCESS.2018.2870052.
- [9] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artif. Intell.*, vol. 267, pp. 1-38, 2019, doi: 10.1016/j.artint.2018.07.007.
- [10] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nat. Mach. Intell.*, vol. 1, no. 5, pp. 206-215, 2019, doi: 10.1038/s42256-019-0048-x.



- [11] G. Schwalbe and B. Finzel, "A comprehensive taxonomy for explainable artificial intelligence: A systematic survey of surveys on methods and concepts," *Data Min. Knowl. Discov.*, vol. 38, no. 5, pp. 3043-3101, 2024, doi: 10.1007/s10618-022-00867-8.
- [12] M. R. Islam, M. U. Ahmed, S. Barua, and S. Begum, "A systematic review of explainable artificial intelligence in terms of different application domains and tasks," *Appl. Sci.*, vol. 12, no. 3, p. 1353, 2022, doi: 10.3390/app12031353.
- [13] F. Bolaños, A. Salatino, F. Osborne, and E. Motta, "Artificial intelligence for literature reviews: Opportunities and challenges," *Artif. Intell. Rev.*, vol. 57, p. 259, 2024, doi: 10.1007/s10462-024-10902-3.
- [14] J. de la Torre-López, A. Ramírez, and J. R. Romero, "Artificial intelligence to automate the systematic review of scientific literature," *Comput.*, vol. 105, no. 10, pp. 2171-2194, 2023, doi: 10.1007/s00607-023-01181-x.
- [15] B. Tóth, L. Berek, L. Gulácsi, M. Péntek, and Z. Zrubka, "Automation of systematic reviews of biomedical literature: A scoping review of studies indexed in PubMed," *Syst. Rev.*, vol. 13, p. 174, 2024, doi: 10.1186/s13643-024-02592-3.
- [16] L. Affengruber et al., "An exploration of available methods and tools to improve the efficiency of systematic review production: A scoping review," *BMC Med. Res. Methodol.*, vol. 24, p. 210, 2024, doi: 10.1186/s12874-024-02320-4.
- [17] R. van de Schoot et al., "An open source machine learning framework for efficient and transparent systematic reviews," *Nat. Mach. Intell.*, vol. 3, no. 2, pp. 125-133, 2021, doi: 10.1038/s42256-020-00287-7.
- [18] M. Ouzzani, H. Hammady, Z. Fedorowicz, and A. Elmagarmid, "Rayyan-A web and mobile app for systematic reviews," *Syst. Rev.*, vol. 5, p. 210, 2016, doi: 10.1186/s13643-016-0384-4.
- [19] B. C. Wallace, K. Small, C. E. Brodley, and T. A. Trikalinos, "Deploying an interactive machine learning system in an evidence-based practice center: Abstrackr," in *Proc. 2nd ACM SIGHIT Int. Health Inform. Symp.*, 2012.
- [20] I. J. Marshall, A. Noel-Storr, J. Kuiper, J. Thomas, and B. C. Wallace, "Machine learning for identifying randomized controlled trials: An evaluation and practitioner's guide," *Res. Synth. Methods*, vol. 9, no. 4, pp. 602-614, 2018, doi: 10.1002/jrsm.1287.
- [21] H. Harrison, S. J. Griffin, I. Kuhn, and J. A. Usher-Smith, "Software tools to support title and abstract screening for systematic reviews in healthcare: An evaluation," *BMC Med. Res. Methodol.*, vol. 20, p. 7, 2020, doi: 10.1186/s12874-020-0897-3.
- [22] J. A. van der Pol et al., "Is AI-assisted active learning software able to reliably speed-up systematic literature reviews in rheumatology? A real-time comparison of AI-assisted and manual abstract selection," *RMD Open*, vol. 10, no. 4, p. e005024, 2024, doi: 10.1136/rmdopen-2024-005024.
- [23] S. Amershi et al., "Guidelines for Human-AI Interaction," in *Proc. 2019 CHI Conf. Human Factors Comput. Syst.*, 2019, Art. no. 3, pp. 1-13, doi: 10.1145/3290605.3300233.
- [24] B. Shneiderman, "Human-Centered Artificial Intelligence: Reliable, Safe & Trustworthy," *Int. J. Human-Computer Interaction*, vol. 36, no. 6, pp. 495-504, 2020, doi: 10.1080/10447318.2020.1741118.
- [25] U. Ehsan et al., "Operationalizing human-centered perspectives in explainable AI," in *Proc. 2021 CHI Conf. Human Factors Comput. Syst.*, 2021, Art. no. 106, doi: 10.1145/3411764.3445083.
- [26] C. Molnar, *Interpretable Machine Learning*, 2nd ed. 2022. [Online]
- [27] R. Guidotti et al., "A survey of methods for explaining black box models," *ACM Comput. Surv.*, vol. 51, no. 5, p. 93, 2018, doi: 10.1145/3236009.
- [28] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, "Machine learning interpretability: A survey on methods and metrics," *Electronics*, vol. 8, no. 8, p. 832, 2019, doi: 10.3390/electronics8080832.
- [29] E. Beller, J. Clark, G. Tsafnat, et al., "Making progress with the automation of systematic reviews: Principles and recommendations," *Syst. Rev.*, vol. 7, p. 77, 2018, doi: 10.1186/s13643-018-0740-7.
- [30] Rajeev P R and K. Aravinthan, "A Generic Approach for Software Metrics Based Software Defect Prediction," *Indian Journal of Natural Sciences*, vol. 15, no. 86, 2024.
- [31] Rajeev P R and K. Aravinthan, "A Novel Approach for Metrics-Based Software Defect Prediction Using Genetic Algorithm," *Scientific Temper*, vol. 15, no. 3, 2024.
- [32] Rajeev P R and Sruthi R Varrier, "A Deep Learning Enabled Hybrid Genetic Rule Mining Framework for Software Defect Prediction," *Technix International Journal for Engineering Research*, Volume 13, Issue 6, 2026, doi : <https://doi.org/10.56975/tijer.v13i6.163155>.



BIOGRAPHY



Sruthi R received her B.Tech in Computer Science and Engineering in 2018 and M.Tech in Computer Science and Information Systems in 2021. Her major field of study is artificial intelligence, machine learning and data science.

She is currently an Adhoc Assistant Professor at College of Engineering Kidangoor, Kerala, India. She has six years of teaching experience. Her publication includes: “A Deep Learning Enabled Hybrid Genetic Rule Mining Framework for Software Defect Prediction,” *Technix International Journal for Engineering Research*, 2026. Her research interests include artificial intelligence, machine learning, data science, cybersecurity, Internet of Things, data analytics and software engineering.

She is engaged in academic and research activities. E-mail: sruthirvarrier@gmail.com. ORCID iD: 0009-0009-9585-1791.



Rajeev P R received his MCA and MPhil in Computer Science in 2014 and 2019, respectively, and his PhD in Computer Science in 2025. His major field of study is software defect prediction, software metrics, machine learning and artificial intelligence.

He is currently an Assistant Professor at Graduate School, Mahatma Gandhi University, Kottayam, India. He has six years of teaching and six years of research experience. His publications include “A Novel Approach for Metrics-Based Software Defect Prediction Using Genetic Algorithm,” *The Scientific Temper*, 2024. His research interests include software defect prediction, machine learning and software metrics.

He is a Professional Member of the Association for Computing Machinery (ACM) and a Member of the International Institute of Research in Multidisciplinary Skill Development Trust (IIRMDSDT). E-mail: prrajeev1904@gmail.com. ORCID iD: 0000-0001-5009-0989.