npj | artificial intelligence

Article

https://doi.org/10.1038/s44387-026-00080-8

AIM review tool: artificial intelligence for smarter systematic review screening

Sergio Mena[1,2] ✉, Esther Rituerto-González[2,3,4], Fiona Coutts[1], Jana von Trott[1,2], Grace R. Jacobs[1], Linda Bryant[1], Louise Moles[1], Nicoleta Sirbu[1], Liisi Promet[2], Dominic Oliver[1,5], Muhammad S. Ahmed[6], Paolo Fusar-Poli[1], Marian J. Bakermans-Kranenburg[7,8], Marinus H. van IJzendoorn[8,9], Nikolaos Koutsouleris[1,2,3,4,10] & Paris Alexandros Lalousis[1,2,10]

In this study, we present the *AIM Review Tool*, a modern web-based application that integrates active and supervised machine learning to accelerate the screening of publications for systematic reviews. *AIM Review* combines advanced text vectorization methods with machine learning models executed directly in the web browser, enabling rapid and privacy-preserving analysis. Unlike existing tools, *AIM Review* uniquely incorporates nested cross-validation and semi-automated screening strategies, enhancing both efficiency and precision in evidence synthesis. Using six real-world case studies across various topics, we demonstrate substantial workload reductions through active learning, with the percentage of publications not requiring screening while achieving ≥95% recall ($WSS_{95\%}$) ranging from 20% to 95%. Supervised learning pipelines trained on a subset of screened records predicted the relevance of unscreened publications with balanced accuracies between 75% and 87%. *AIM Review* provides a flexible, scalable, and accessible solution for large-scale literature screening and can be readily integrated into existing manual workflows.

The rapid expansion of scientific knowledge, fueled by an ever-growing number of researchers and publications, has brought both unparalleled opportunities and challenges for academia[1,2]. As a growing number of new studies are added to the scientific literature daily, the task of synthesizing information via systematic reviews—a cornerstone of evidence-based science—has become increasingly challenging[3]. Traditional manual methods of publication screening and data extraction, which are resource-intensive and time-consuming, have struggled to keep up with the rapidly growing volume of published studies. In addition, the growing redundancy of systematic reviews and meta-analyses on the same topics, often based on limited or incomplete searches, creates a significant challenge in accurately assessing the true level of evidence for a given topic[4].

The integration of artificial intelligence (AI) tools has recently become increasingly prominent across various domains of research. AI-powered software is now employed in diverse fields, such as automating data processing and analysis[5], aiding detection, diagnosis and prognosis in healthcare applications[6,7], and assisting in study design and scientific discovery[8]. In recent years, several AI software tools have emerged to speed up systematic review screening[9–14] and automatic data extraction[15–17]. These tools have rapidly gained recognition for their effectiveness in tackling large systematic reviews, positioning themselves as strong competitors to conventional non-AI-powered systematic review software. However, leading AI-powered tools employ either proprietary, closed source algorithms that operate as black boxes and are therefore not configurable and cannot be adapted to different scenarios, or require complex software installations, lacking transparency and precluding model configuration and performance evaluation[18].

One methodology used by cutting-edge applications to prioritize relevant publications in systematic review screening is active learning[19,20]. This is a type of machine learning (ML) where a recursive algorithm learns the natural language patterns of relevant publications already labeled by the screener and sorts the remaining, yet-to-be-labeled publications using their predicted relevance. Active learning is optimal when relevant publications

[1]Department of Psychosis Studies, Institute of Psychiatry, Psychology and Neuroscience, King's College London, London, UK. [2]Department of Psychiatry and Psychotherapy, Ludwig-Maximilians-University, Munich, Germany. [3]Max Planck Institute of Psychiatry, Munich, Germany. [4]German Centre for Mental Health (DZPG), partner site Munich-Augsburg, Munich, Germany. [5]Department of Psychiatry, University of Oxford, Oxford, UK. [6]Department of Psychological Medicine, Institute of Psychiatry, Psychology and Neuroscience, King's College London, London, UK. [7]William James Centre for Research, ISPA, Lisbon, Portugal. [8]Facultad de Psicología y Humanidades, Universidad San Sebastián, Sede Valdivia, Chile. [9]Research Department of Clinical, Education and Health Psychology, Faculty of Brain Sciences, UCL, London, UK. [10]These authors contributed equally: Nikolaos Koutsouleris, Paris Alexandros Lalousis.
✉e-mail: sergio.mena_ortega@kcl.ac.uk

are sparse, since it allows reviewers to identify the few relevant publications by presenting them for screening as soon as possible[12]. However, active learning has inherent limitations: it can be biased by the small number of initial screenings, leading to unrepresentative prioritizations[21]. Moreover, its efficiency is highly dependent on the proportion of relevant publications. When only a small fraction of the literature is relevant (e.g., 0–10%), active learning can significantly reduce screening time. However, maintaining a high recall rate (i.e., identifying most relevant studies) when the proportion of relevant studies is larger often results in minimal time savings, as a substantial portion of literature still requires manual review[22]. Systematic reviews often encounter an overwhelming number of studies that meet the inclusion criteria. Even with the use of active learning, conducting these reviews manually becomes both time-consuming and economically unfeasible due to the sheer volume of publications requiring screening[23,24]. In such cases, semi-automatization of the screening and data extraction processes is essential to maintaining the viability of these systematic reviews.

To address these challenges, nested and cross-validated (NCV) supervised ML offers a robust alternative[25]. By training on a subset of publications (e.g., 10–20%), NCV optimizes and evaluates models to predict relevance in the remaining set with reliable sensitivity and specificity. This approach scales effectively to large systematic reviews where thousands of publications are relevant for full-text screening, enabling partial automation while maintaining accuracy of screening.

Here we introduce the *AIM Review Tool* (link), a freely available web application that integrates active learning and nested cross-validated (NCV) supervised learning to support and accelerate systematic review screening. The application uses advanced text vectorization methods that convert titles and abstracts into embeddings—numerical representations of text—and employs multiple classification models to provide relevance scores and classify publications accordingly. Unlike existing semi-automated screening tools[9,10,14], which primarily use active learning to prioritize records for manual screening, AIM Review additionally includes a dedicated supervised prediction workflow. This workflow applies nested cross-validation (NCV) to a subset of screened publications to robustly train and evaluate models, which are then used to predict the relevance of remaining unscreened records. *AIM Review* provides three modular applications:

- An active learning-powered *labeling application* to screen titles and abstracts of publications using conventional tools, such as highlighting relevant/irrelevant expressions, note-taking, tracking of progress and sorting the order of publications based on publication relevance scores predicted by ML models.
- An *agreement application* that measures the inter-rater and document-level screening agreements as well as discrepancies among screeners
- An NCV-powered *ML prediction application* to train supervised models with a subset of screened publications and predict the relevance of the remaining unscreened publications.

Following, we benchmark the algorithms implemented in *AIM Review* across six real-world systematic reviews[26–32]. In the Methods section, we detail the active learning and NCV supervised learning algorithms available in *AIM Review*, along with the configuration settings and parameter choices used in the benchmarking experiments. The Results section presents evidence of how *AIM Review* can substantially reduce the time and effort required to conduct systematic reviews by prioritizing relevant studies earlier in the screening process, as well as a direct comparison of algorithmic performance in terms of accuracy, efficiency, and reviewer workload reduction. Finally, the Discussion section reflects on the advantages and limitations of the different algorithms and applications within *AIM Review*, including considerations of usability across diverse systematic reviews.

Results

Active learning validation analyses

To assess the efficiency of active learning within the *AIM Review* labeling pipeline, we simulated the title and abstract screening process across all six case studies. Models were iteratively trained on previously screened publications and used to predict the relevance of remaining publications. The active learning cycle was programmed to start when at least one relevant and one irrelevant publication and a minimum initial sample of 25 publications were screened. Models are retrained every 5 new labels are added, and relevance scores of unscreened publications were updated accordingly. On each iteration, unscreened publications were sorted by their predicted relevance score, bringing forward publications with the highest probability of being relevant. Figure 1 shows the mean and standard deviation ($n = 5$ repetitions) percentage of relevant publications found vs. the percentage number of publications screened for the initial three psychology/psychiatry case studies, using LR and L-SVM as classifiers, and stacked generalization or fusion of features from all text vectorization methods: term frequency–inverse document frequency (TF-IDF), latent semantic analysis (LSA), scientific paper embeddings using citation-informed transformers (SPECTER2), universal BERT sentence encoder (univBERT), document to vector (Doc2Vec), and miniature language model, 6-layer, version 2 (all-MiniLM-L6-v2). Additionally, Figs. S2–S7 in the supplementary material shows analogous simulation results where only one text vectorization method is employed. We also evaluated performance using the Work Saved over Sampling (WSS) metric, specifically $WSS_{95\%}$ and $WSS_{100\%}$, which quantify the reduction in screening effort while maintaining a 95% and 100% recall of relevant publications. WSS values for logistic regression (LR) and linear support vector machine (L-SVM) models, utilizing stacking or fusion of features from all text vectorization methods (TF-IDF, LSA of TF-IDF, SPECTER2, univBERT, Doc2Vec and all-MiniLM-L6-v2), are summarized in Table 1. In addition, Table S2 in the supplementary materials shows the WSS values computed from active learning simulations where only features from one text vectorization method are used, and Table S1 shows WSS values for the extended three case studies from other disciplinary fields.

Overall, the active learning algorithm substantially reduces the screening workload in case study 1 (from $WSS_{95\%}$ of 73.32% with univBERT features and L-SVM, to $WSS_{95\%}$ of 85.35% with LR and model stacking) and case study 2 (from $WSS_{95\%}$ of 36.67% with univBERT features and L-SVM to $WSS_{95\%}$ of 91.60% with LR and model stacking). For case study 3, workload savings are lower due to the high percentage of relevant publications (from $WSS_{95\%}$ of 13.87% with TF-IDF features and L-SVM to $WSS_{95\%}$ of 22.94% with LR and model stacking). Importantly, as shown in Table S1, similar workload savings were also observed in the extended case studies 4–6 from fields of computer science, endocrinology and environmental health, with WSS values appearing to depend on the proportion of relevant studies rather than on the specific disciplinary field of the systematic review. Both ensemble strategies—feature fusion and model stacking—increase WSS95% values compared to using a single text vectorization method, albeit at the cost of higher computational complexity. Considering the total number of publications and optimal configurations found, the active learning algorithm would reduce the number of publications that need to be screened from 16,660 original publications to 2448 publications in case study 1, from 5536 to 457 publications in case study 2, and from 729 to 561 publications in case study 3.

Supervised learning validation analyses

To evaluate the efficacy of the supervised learning approach we divided the publications into a model discovery out-of-training sample (20%, OOT) and out-of-cross-validation sample (80%, OOCV) in each case study. We evaluated the performance of three classifiers—LR, L-SVM and sequential neural networks (SeqNN)—and one text vectorization method (TF-IDF, LSA of TF-IDF, SPECTER2, univBERT, Doc2Vec and all-MiniLM-L6-v2), as well as stacked generalization and fusion of all vectorization methods combined. Figure 2A–C show the ROC curves at OOT and OOCV level of models trained using either stacked generalization or fusion of features from all text vectorization methods for the initial three case studies from the fields of psychology and psychiatry. Figure 2D–F shows balanced accuracies (BACs) at the OOT level (mean [SD], $n = 5$ outer permutations) and OOCV

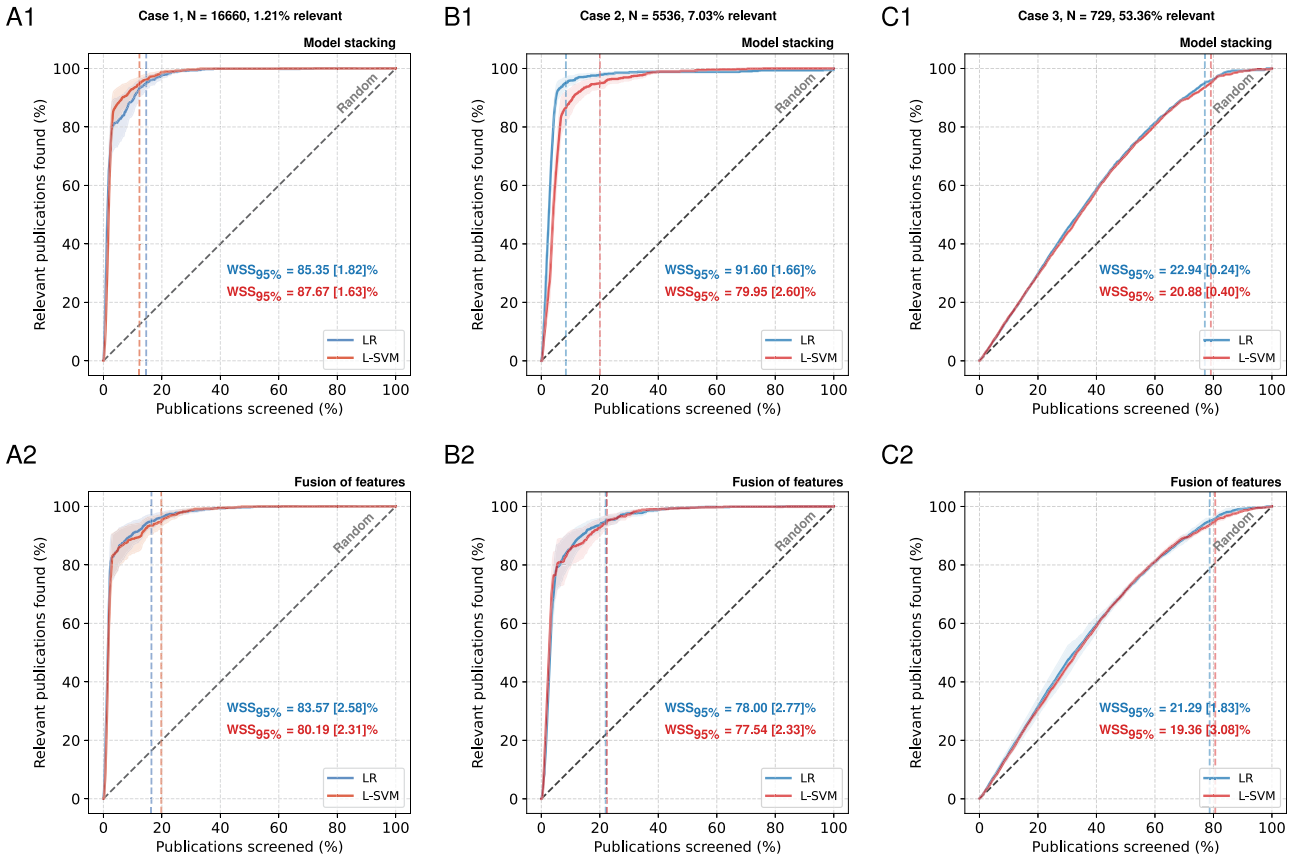

Fig. 1 | Active learning simulation results. Mean and standard deviation of relevant publications identified during the screening simulation versus the total publications screened in **A** case study 1, **B** case study 2, and **C** case study 3. The top row (A1, B1, and C1) presents results using a stacked generalization model, while the bottom row (A2, B2, and C2) shows results using feature fusion. In the stacked generalization approach, the selected classifiers (LR or L-SVM) are trained independently using features extracted from each text vectorization method (TF-IDF, LSA of TF-IDF, SPECTER2, univBERT, Doc2Vec and all-MiniLM-L6-v2). A meta-model (LR with regularization parameter C = 1) is then trained on the predictions from the base models to generate the final predictions. In the feature fusion approach, features extracted from all text vectorization methods are concatenated and used to train a single model (LR or L-SVM) to predict publication relevance. Insets in the figures display the mean and standard deviation of $WSS_{95\%}$ for each classifier type. The mean $WSS_{95\%}$ is also visually represented by the dashed lines. The gray dashed line shows the expected mean number of relevant publications found if the order of publications was randomized and no active learning was used. Figures S2–S7 in the supplementary materials present the results of repeated simulations using individual text vectorization methods. Abbreviations: WSS work saved over sampling.

level computed using a relevance probability threshold of 0.5, while Figure S8 in the supplementary materials shows area under the receiver operating characteristic curve (AUROC) and P4 score for models employing an ensemble approach (model stacking or fusion of features). In addition, sensitivity, specificity, BAC and AUROC metrics are summarized in Table 1. Figures S9–S14 in the supplementary materials display ROC curves, AUROC values, and BAC for individual text vectorization methods, while Figure S15 and Table S2-S3 in the supplementary materials illustrates and summarizes the AUROC, BAC, sensitivity, specificity and P4 score of models using features derived from a single text vectorization method. Finally, Table S1 in the supplementary materials shows the performance metrics for the extended three case studies from other disciplinary fields.

Overall, the supervised learning models predict the relevance of unseen publications with high accuracy. In case study 1, OOCV BACs range from 78.50% with SeqNN and Doc2Vec features to 86.51% with LR and model stacking. In case study 2, OOCV BACs range from 62.30% with SeqNN and UnivBERT features to 86.68% with LR and model stacking. In case study 3, OOCV BACs range from 64.84% with LR and UnivBERT features to 74.63% with SeqNN and fusion of features. Across all case studies, ensemble strategies—whether feature fusion or model stacking—enhance predictive performance compared to using a single text vectorization method, as demonstrated in the active learning benchmarking. Model stacking achieves a higher OOT BAC compared to feature fusion but comes at the cost of increased overfitting, as indicated by the decrease in OOCV performance. In contrast, feature fusion exhibits the opposite trend, with a higher OOCV BAC than OOT BAC, demonstrating its superior generalizability to unseen publications. Importantly, analogous performances and patterns were observed in the extended benchmarking for case studies 4–6 from other scientific fields.

Application workflow and documentation

As part of this report, we have developed comprehensive documentation detailing the applications and methods implemented in *AIM Review*. This documentation includes step-by-step guides on how to use each application, video tutorials of the applications, descriptions of active learning and supervised learning pipelines, and insights into the underlying algorithms. Additionally, we have created a short workshop that includes mock data to facilitate hands-on learning for users, enabling them to run the application, understand its functionalities, and apply it to their own systematic review processes. Figure 3 also illustrates a proposed workflow of the *AIM Review* applications based on systematic review characteristics and the findings from the case studies (see below). The full documentation, including access to workshop materials and audiovisual content can be accessed in the documentation tab of AIM Review.

Discussion

In this work, we have developed the *AIM Review Tool*, a highly customizable web application that integrates active and NCV supervised ML with cutting-

Table 1 | Performance metrics of active learning and supervised learning pipelines using model stacking and fusion of features

			Active learning		Supervised learning							
Case	**Model**	**Ensemble**	**$WSS_{95\%}$ (%)**	**$WSS_{100\%}$ (%)**	**BAC OOT (%)**	**Sens. OOT (%)**	**Spec. OOT (%)**	**AUROC OOT**	**BAC OOCV (%)**	**Sens. OOCV (%)**	**Spec. OOCV (%)**	**AUROC OOCV**
Case 1	LR	Model stacking	85.35 [1.82]	74.60 [5.52]	90.54 [1.05]	97.95 [1.92]	83.13 [0.43]	0.95 [0.003]	86.51	89.02	84.00	0.93
		Fusion of features	83.57 [2.58]	65.48 [2.58]	84.30 [1.86]	92.82 [3.77]	75.79 [0.26]	0.92 [0.01]	85.41	88.41	82.41	0.92
	L-SVM	Model Stacking	87.67 [1.63]	77.21 [5.14]	91.03 [1.00]	95.38 [1.92]	86.67 [0.25]	0.95 [0.002]	86.10	92.07	80.12	0.93
		Fusion of features	80.19 [2.31]	68.99 [4.03]	81.98 [1.60]	93.85 [2.61]	70.11 [0.84]	0.90 [0.01]	82.91	83.54	82.28	0.90
	SeqNN	Model Stacking	-	-	83.07 [1.55]	77.95 [4.75]	88.18 [2.18]	0.90 [0.03]	82.97	80.49	85.45	0.86
		Fusion of features	-	-	80.69 [2.81]	79.50 [8.10]	81.89 [4.46]	0.90 [0.02]	84.71	83.54	85.89	0.93
Case 2	LR	Model stacking	91.60 [1.66]	78.70 [1.66]	90.14 [1.98]	95.56 [3.62]	84.73 [1.32]	0.95 [0.01]	86.68	87.73	85.62	0.93
		Fusion of features	78.00 [2.77]	59.75 [6.48]	82.64 [2.70]	85.19 [5.23]	80.09 [0.42]	0.91 [0.01]	86.62	86.50	86.74	0.93
	L-SVM	Model Stacking	79.95 [2.60]	59.80 [5.28]	88.96 [1.24]	95.56 [2.77]	82.53 [1.56]	0.95 [0.01]	84.76	83.44	86.09	0.92
		Fusion of features	77.54 [2.33]	64.07 [6.56]	80.22 [1.52]	75.56 [2.96]	84.88 [0.87]	0.88 [0.02]	83.68	82.21	85.16	0.92
	SeqNN	Model Stacking	-	-	85.99 [1.69]	82.22 [2.77]	89.76 [0.87]	0.92 [0.01]	81.19	71.78	90.60	0.90
		Fusion of features	-	-	82.62 [2.91]	83.70 [6.86]	81.55 [3.51]	0.90 [0.02]	85.46	84.66	86.25	0.93
Case 3	LR	Model stacking	22.94 [0.24]	14.01 [0.07]	76.29 [2.03]	76.62 [1.97]	73.04 [3.25]	0.85 [0.02]	73.65	74.04	73.26	0.78
		Fusion of features	21.29 [1.83]	11.68 [0.17]	70.16 [1.54]	62.35 [1.66]	77.97 [4.03]	0.76 [0.01]	72.87	79.81	65.93	0.78
	L-SVM	Model Stacking	20.88 [0.40]	10.99 [0.12]	77.10 [2.38]	76.23 [2.91]	77.97 [2.96]	0.85 [0.02]	73.03	80.13	65.93	0.79
		Fusion of features	19.36 [3.08]	7.55 [0.09]	68.18 [2.67]	73.18 [2.72]	63.19 [3.25]	0.73 [0.01]	69.18	85.26	53.11	0.75
	SeqNN	Model Stacking	-	-	76.18 [1.21]	72.94 [1.49]	79.42 [3.22]	0.84 [0.01]	74.45	75.64	73.26	0.81
		Fusion of features	-	-	70.37 [1.46]	66.82 [5.38]	73.91 [5.02]	0.77 [0.01]	74.63	83.33	65.93	0.80

AUROC area under the receiver operating characteristic curve, *BAC* balanced accuracy, *L-SVM* linear support vector machine, *LR* logistic regression, *OOCV* out-of-cross-validation, *OOT* out-of-training, *Sens* sensitivity, *SeqNN* sequential neural networks, *Spec* specificity, *WSS* work saved over sampling.

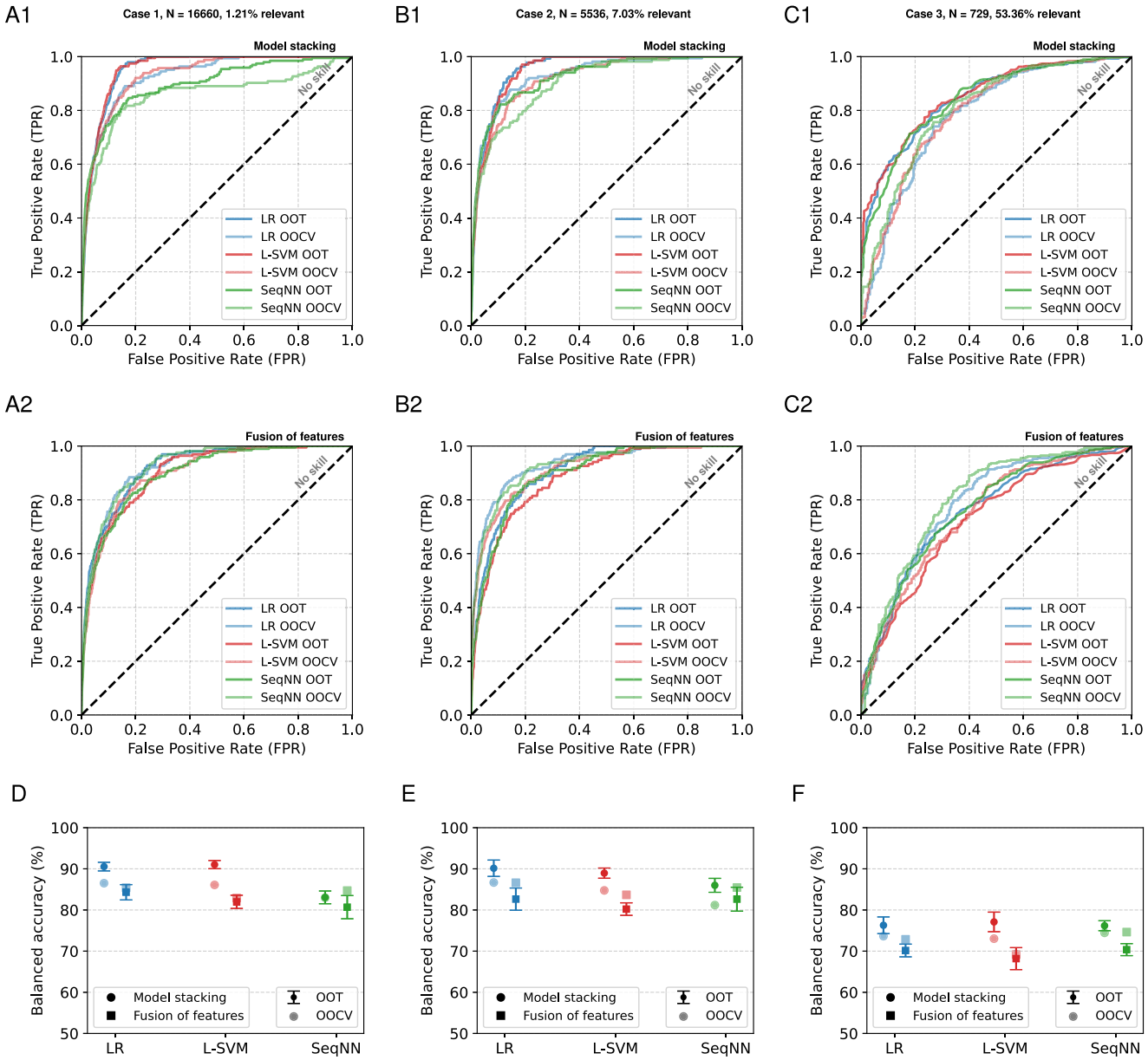

Fig. 2 | Supervised learning results. ROC curves at OOT (20% of publications) and OOCV level (80% of publications) are shown for **A** case study 1, **B** case study 2 and **C** case study 3. The top row (A1, B1, and C1) presents results using a stacked generalization model, while the bottom row (A2, B2, and C2) shows results using feature fusion. In the stacked generalization approach, the chosen classifiers (LR, L-SVM or SeqNN) are independently trained using features derived from each text vectorization method (TF-IDF, LSA of TF-IDF, SPECTER2, univBERT, Doc2Vec and all-MiniLM-L6-v2). A meta-model (LR with regularization parameter C = 1) is then trained on the predictions of the base models to produce the final predictions. In the feature fusion approach, the features extracted from all text vectorization methods are concatenated and used to train a single model (LR, L-SVM, or SeqNN) for predicting publication relevance. Models are trained and validated at the OOT level using a NCV framework of 5 outer permutations, five outer folds and five inner folds. The final optimum model is trained with all training publications and tested in the OOCV sample. Balanced accuracies at the OOT (mean and standard deviation, $n = 5$ outer permutations) and OOCV levels are shown for **D** case study 1, **E** case study 2 and **F** case study 3. The BAC values presented here are calculated by applying a threshold probability of relevance of 0.5. Figure S8 in the supplementary materials presents the area under the ROC curve of the models using an ensemble method (stacking or fusion of features). Figure S9–S14 in the supplementary materials shows the ROC curves, area under the ROC curves and balanced accuracies using individual text vectorization methods. Finally, Fig. S15 in the supplementary materials shows the area under the ROC curve and balanced accuracies of models using features from one vectorization method. Abbreviations: L-SVM linear support vector machine, OOCV out-of-cross-validation, OOT out-of-training, SeqNN sequential neural network.

edge browser technologies to accelerate systematic review publication screening. Using three real-world systematic reviews as case studies and simulating manual screening, we have evaluated the capabilities of *AIM Review*'s active learning strategies in saving screening workload and resource allocation.

Our benchmark assessments demonstrate that the active learning strategies implemented in *AIM Review* yield substantial reduction in screening workload (up to $WSS_{95\%}$ of 95%) across the six case studies. Workload savings varied with dataset characteristics, with larger studies containing fewer relevant publications yielding higher savings, and those with more relevant publications yielding lower savings. There are two main reasons for this occurrence. First, in cases where half of the publications are relevant, a hypothetical ideal predictor—perfectly ranking all relevant publications first—would achieve a maximum possible workload savings of

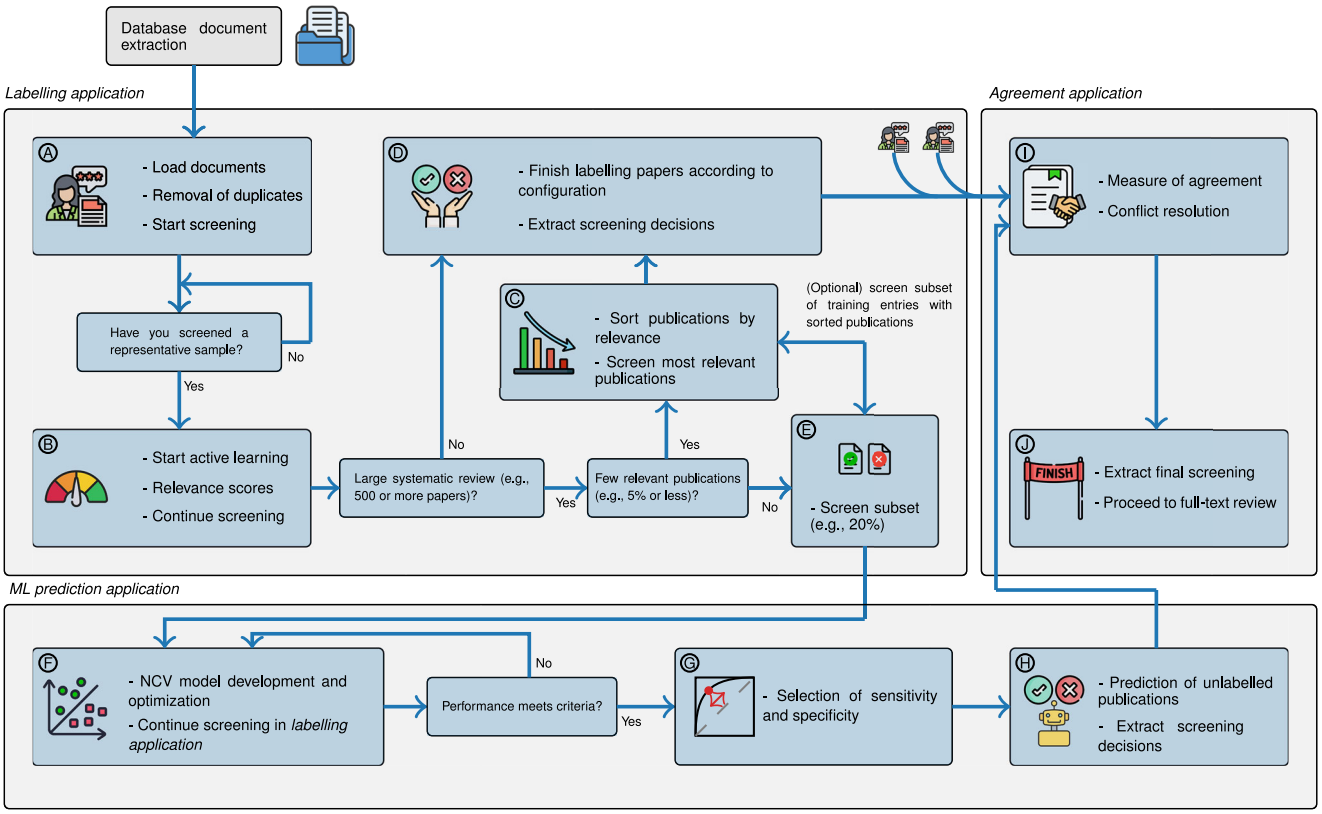

Fig. 3 | Proposed workflow of AIM Review applications. This workflow is provided to guide users in applying the different modules of the application. Given the flexibility of *AIM Review*, uncertainty about how to configure the available options may arise; the workflow illustrates recommended practices to support consistent and effective use. Following database searches, users can import the results into the labeling application, which automatically removes duplicates, allowing them to seamlessly begin screening titles and abstracts. Once a representative number of relevant and irrelevant publications have been screened, users can activate the active learning engine to generate relevance scores, guiding their manual labeling. For large systematic reviews with few expected relevant publications, the optimal strategy is to sort entries by relevance using the active learning engine and label the most relevant publications until none remain. Conversely, if many publications are expected to be relevant, users can screen a subset of publications—either sorted by relevance or not—and then develop and evaluate NCV models in the ML prediction application. While models are being developed, the modular design of *AIM Review* applications allows users to continue screening titles and abstracts in the labeling application. If models meet the user-defined performance criteria, they can be used to predict the relevance of unscreened publications and extract screening decisions (both manual and AI-generated). Finally, the agreement application enables multiple users to assess inter-rater and document-level agreement, resolve conflicts, and extract the final screening labels to proceed to full-text review. Artwork was created using Flaticon designs. Abbreviations: NCV nested cross-validation. Notably, this is a proposed workflow, but users can decide to use the application in different ways. Artwork was created using Flaticon icons.

approximately 50%. Second, highly imbalanced datasets benefit from the use of active learning with certainty-based sorting of publications, as models can focus on the highly similar predictive pattern of the few positive instances, minimizing the effort to refine their decision boundary. Cases with high percentages of inclusions have been underrepresented in previous benchmarking analyses of active learning software[9–11,33]. Previous benchmark studies showed that substantial workload reductions with active learning mostly occur in large sample sizes (> 2000 publications) where approximately less than 5% of publications are relevant[13,20]. Our findings extend this understanding by showing that active learning strategies are highly effective in large systematic reviews where relevant publications are scarce—enabling researchers to cast a broad range of publications and still feasibly identify all relevant studies, even when resources are limited. In systematic reviews with many relevant studies, while active learning offers comparatively smaller gains, modern ML strategies like multiple text vectorization and ensemble methods can still reduce workload by approximately 20% which, although modest, can still lead to significant time savings.

Evidently, scenarios exist in which a 20% reduction in screening workload remains insufficient, particularly in large-scale systematic reviews with a high acceptance rate of publications. In such cases, the absolute number of studies requiring manual assessment remains prohibitively large, imposing a significant burden on researchers and resource allocation[3,23], or leading to the unfortunate outcome where systematic reviews are not conducted due to a lack of available resources. Further advancements in automation are imperative, with a focus on transitioning from active learning-driven prioritization to fully supervised learning approaches.

To address this limitation, using a subset of publications (20%), we trained supervised models with NCV and evaluated their ability to predict the relevance of unscreened publications (80%) in the three case studies presented. The results demonstrated that supervised models could predict the relevance of unscreened publications with moderate to high balanced accuracies in the three case studies, with top balance accuracies from 75% to 87% across multiple text vectorization methods, classifiers and ensemble methods. These results were obtained using an NCV framework that provides reliable estimates of generalization error, as evidenced by similar external (OOCV) performances. A key finding with ensemble methods is the trade-off between feature fusion and stacked generalization. Model stacking generally outperformed feature fusion in both OOT and OOCV, although often showing a mild susceptibility to overfitting, as indicated by a drop in performance from OOT to OOCV. In contrast, feature fusion consistently demonstrated better generalizability to unseen publications, with OOCV performance often exceeding OOT performance, suggesting a trade-off between generalization and predictive power when integrating

multiple representations of paragraph contexts. Regarding classifiers, shallow learners (LR and L-SVM) outperformed the deep learner (SeqNN) in unbalanced samples (case studies 1 and 2), while the deep learner performed best in balanced samples (case study 3) and benefits from highly dimensional feature spaces (fusion of features), improving their generalizability. When using single text vectorization methods alone, pre-trained, transformer models (SPECTER2, all-MiniLM-L6-v2) outperformed simpler text vectorization techniques (e.g., TF-IDF and LSA of TF-IDF), highlighting the benefits of contextualized embeddings[34]—though at the cost of higher computational cost in large-scale screenings.

Beyond optimal configurations, the results of this study highlight the complimentary trade-offs between active learning, which achieves high recall in all cases but may be less efficient in scenarios where many relevant studies are found, and NCV-supervised learning, which can greatly reduce the screening burden (e.g., 20%) but may result in lower recall of relevant papers found. In contrast, NCV-supervised learning is computationally more intensive than active learning and may require additional time to generate predictions. However, since the process is fully automatic, generating predictions requires minimal manual effort and, with modest configurations, can typically be completed within a few minutes. To our knowledge, this is the first application that introduces NCV supervised learning for semi-automated systematic review screening, aimed at complementing the established active learning strategies. With the growing popularity of large language models (LLMs), recent benchmark studies have evaluated their performance in title and abstract systematic review screening[35–37]. Initial studies indicate that LLM performance in identifying relevant publications is highly dependent on the prompt and often results in high differences between sensitivity and specificity, reflecting the tendency of LLMs toward overconfidence or underconfidence. In contrast, our supervised models achieve a balanced trade-off between sensitivity and specificity using the default probability threshold of 0.5. Additionally, users can tune the optimization metric and threshold probability, allowing them to adjust the model's conservatism and balance the sensitivity and specificity of predictions to their needs. Unlike fully automated pre-trained LLM-based approaches—an advantage in terms of scalability and speed—our NCV-supervised models use manual labels to be trained and estimate accuracy before predicting unscreened publications. This requires labeling a subset of publications, making the process semi-automatized, but also ensures that users have a measurable assessment of model performance obtained using NCV. The reliability of these estimates depends on the representativeness of the training set, which should be sufficiently large; in our case studies, a randomized 20% of publications provided an adequate representation. When the predicted model performance does not meet the requirements, *AIM Review* users can seamlessly transition to traditional manual screening, with or without incorporating active learning techniques, thereby maintaining full oversight and control over the systematic review process.

In summary, the findings reported here have important implications for planning and resource allocations in evidence synthesis. The demonstrated workload reductions using *AIM Review* active learning pipelines suggest that *AIM Review* could accelerate literature screening across all research disciplines, potentially reducing the time required to conduct comprehensive systematic reviews and meta-analyses. Finally, the robust and precise predictions produced by *AIM Review*'s supervised learning pipeline enables a semi-automated automated and thus more replicable knowledge extraction process for systematic reviews and meta-analyses. Future developments of *AIM Review* will focus on automating full-text data extraction and synthesis, as well as advanced sorting of publications—based on criteria, such as sample size, methodological quality and bias. In addition, systematic software comparisons would help clarify relative strengths, limitations, and scalability of software available in evidence synthesis.

The *AIM Review Tool* and the benchmark results reported here are not without limitations. First, while we leverage modern web technologies, including Web Workers, WebAssembly and WebGL, to enable the processing of large datasets and the execution of ML tasks directly in the browser with minimal latency, we acknowledge the inherent constraints of computational resources within front-end web environments. To address these limitations, we have optimized our algorithms to ensure efficient execution even with large-scale data. Despite these constraints, we have rigorously tested the application using large systematic reviews and complex ML workflows (e.g., >45,000 studies and stacked generalization of multiple text vectorization methods), demonstrating the robustness and scalability of the system in handling demanding tasks within the browser environment of a conventional desktop PC. We also anticipate that these computational constraints will diminish over time with the continued advancement and adoption of in-browser AI technologies, which are expected to substantially expand the scope of feasible computations. A limitation of the benchmarking reported here is the lack of a large systematic review (e.g., >10,000 entries) with a balanced percentage of significant publications. In practice, such large systematic reviews with many relevant papers are rarely available due to the substantial costs associated with the screening and data extraction. As a result, none of the case studies included in the benchmark were drawn from studies of this scale, as these types of resources are typically not accessible. Umbrella review paradigms that combine results from single studies across multiple systematic reviews could serve as a viable alternative for testing the application in large samples, and we will explore this in future benchmarking steps. Additionally, due to computational constraints, we did not benchmark the algorithms across different devices and browsers, nor did we evaluate every possible combination of classifiers, text vectorization methods, and ensemble approaches in *AIM Review*. Notably, deep learning models were not included in the active learning benchmarking due to their high computational and time costs during training, and we only tested logistic regression as a meta-model in stacked generalization. In future implementations, we plan to fine-tune deep learning models at each active learning interaction rather than fully retraining them, making them more computationally efficient in this context. Finally, when measuring performance, we did not account for the time real users spend learning to use *AIM Review*, computational cost, browser memory usage, algorithmic biases or disagreement with traditional screening. We acknowledge this as a limitation of our simulation-based approach, and in the future, we plan to evaluate *AIM Review* also against traditional manual screening.

In this work, we have presented The *AIM Review Tool*, a modern web application integrated with static state-of-the-art web technologies to accelerate systematic review screening and make large-scale systematic review projects more feasible. *AIM Review* combines active learning and NCV-supervised ML pipelines to prioritize relevant publications and semi-automatize the systematic review process. Additionally, *AIM Review* incorporates a robust set of diverse text vectorization methods, ranging from simple TF-IDF to advanced language models like sentence transformers, as well as classifiers of varying complexities. Future implementations could expand on these capabilities by incorporating question-answering language models and pre-trained LLMs to semi-automatize the full-text data extraction step of systematic reviews. Moreover, *AIM Review* could benefit from implementation of a wider array of text vectorization methods (e.g., domain-specific and fine-tuned transformers) and multi-label classifiers to further enhance the precision and scalability of the tool. Finally, future benchmarking should include browser memory usage and computational time as primary performance metrics. *AIM Review* sets the foundation for more efficient, scalable, and accurate systematic reviews and meta-analyses.

Methods

An extended description of the methods can be found in the supplementary materials.

Application design, structure and technology stack

AIM Review is a client-side web application developed using Google's Firebase platform. The application operates without a server-side connection, meaning that all the data and algorithms run in the user's local browser. *AIM Review* has been tested across major browsers—including Google Chrome, Mozilla Firefox, and Microsoft Edge—and on multiple devices, such as Windows PCs, Android phones, and tablets.

The web application has a modular structure with three independent components. The first is the *labeling application*, which allows users to upload publications from a variety of databases in multiple formats (e.g., Excel spreadsheet, RIS files and text files), remove duplicates, manually screen them as relevant or irrelevant based on a given inclusion and exclusion criteria, and track screening progress with interactive plots. It includes an active learning algorithm that iteratively predicts relevance scores for unscreened publications and sorts them based on relevance. The second is the *agreement application*, used to assess inter-rater and document-level agreement between screeners by importing labels from the *labeling application* and resolving disputes among raters (see supplementary materials for a description of the agreement measures). The third is the *ML prediction application*, which applies NCV-based ML pipelines to classify a subset of manually labeled publications based on their relevance, and predict the relevance of unscreened publications. Importantly, each module of the application can be run in parallel. Publication screening, agreement measures and model predictions can be saved into a standardized spreadsheet file.

The *AIM Review* web application framework is built with HTML, CSS, and object-oriented JavaScript. ML computations run in the background using JavaScript web workers. Python-based ML libraries, such as scikit-learn[38], are executed utilizing Pyodide's WebAssembly implementation[39]. TensorFlow.js[40] powers the training of sequential neural networks and Doc2Vec sentence encoders, while HuggingFace.js and the HuggingFace inference API[41] were used to generate embeddings from pretrained sentence transformers.

Vectorization of text

AIM Review incorporates multiple text vectorization methods to convert titles and abstracts into numerical representations for ML classification:

- Term frequency-inverse document frequency (TF-IDF; scikit-learn)[42] — default option
- Latent semantic analysis (LSA; scikit-learn) of TF-IDF using truncated singular value decomposition
- Universal sentence encoder (univBERT; tensorflow.js)[43]
- Document to vector embedding model (Doc2Vec; tensorflow.js native implementation)[44]
- Pretrained sentence transformers (e.g., all-MiniLM-L6-v2 and SPECTER2; transformers.js)[45,46].

AIM Review allows users to choose multiple vectorization methods and combine the embeddings either via concatenation of embeddings (fusion) or via stacked generalization[47]. Each method balances speed and complexity —simpler models like TF-IDF are faster, while deep learning models like sentence transformers are slower but able to capture more complex semantic relationships.

Classification models

AIM Review includes various classifier algorithms that can be trained on user labels to predict the relevance of unseen publications:

- Logistic regression (LR; scikit-learn) — default option
- Linear and non-linear support vector machines (SVM; scikit-learn)
- Decision tree classifier (DT; scikit-learn)
- Multi-layer perceptron (MLP; scikit-learn)
- Sequential neural networks (SeqNN; tensorflow.js).

The application supports both shallow (e.g., LR) and deep learners (e.g., SeqNN), but shallow learners were prioritized in the benchmarking to maximize efficiency and minimize latency, in line with previously developed applications. The model parameters can be configured by users. Additionally, using the NCV framework, selected model hyperparameters can be optimized (see supplementary materials for further details on model parameters).

Active learning strategies

The active learning methodologies in *AIM Review* are inspired by state-of-the-art implementations of active learning for systematic review screening[9–11], and applied into a web environment. Figure 4A illustrates the active learning strategies used in the labeling application of *AIM Review*. Briefly, once the user begins screening titles and abstracts, an active learning cycle can be activated in which a classifier model is trained using the numerical representations of the screened papers to predict the assigned label. The classifier is then applied to predict the relevance of unlabeled entries. The predicted relevance scores are displayed using a color-coded scheme. Users can choose to sort the publications based on their relevance score (high or low relevance first, or uncertain first; see supplementary materials), as well as add random percentage of publications to avoid unrepresentative prioritizations. This cycle can be reactivated every time the user clicks a button or, alternatively, at regular time intervals. Classifiers are applied without hyperparameter tuning; all parameters are pre-defined by the user.

Nested cross-validated supervised machine learning strategies

The NCV methodologies implemented in *AIM Review* are inspired by NeuroMiner, a ML pattern recognition software developed to design, train and validate clinical diagnostic and prognostic models. NCV allows to obtain a reliable assessment of the models' performance in unseen publications while simultaneously tuning hyperparameters of models (see supplementary methods in supplementary materials). Figure 4B illustrates the supervised learning pipeline in *AIM Review*. In short, a subset of screened titles and abstracts are used to train and optimize models to predict the relevance of publications using NCV. Data is split into folds within an outer cross-validation (CV2) for model evaluation and an inner cross-validation (CV1) for hyperparameter tuning. The dataset is split into N outer folds, where one fold serves as a validation set while the remaining $N-1$ folds undergo inner cross-validation. Within CV1, the data is further split into M inner folds for training and testing, identifying optimal hyperparameters. Once these are found, models are retrained with the optimal parameters and are validated on the held-out outer fold. An ensemble prediction that aggregates results across all validation folds is generated to produce performance estimates. To enhance reliability, users can apply multiple permutations of inner and outer loops, averaging performance across iterations. This approach provides a more robust and unbiased performance estimate compared to conventional cross-validation, which lacks a dedicated loop for hyperparameter tuning and may lead to overfitting. Model performance is assessed via the receiver operating characteristic (ROC) curve generated using the aggregated validation predictions (predictions for publications when they serve as the validation sample). This enables users to set probability thresholds for classifying relevant publications in the unscreened subset, with the default threshold being 0.5. A higher threshold (e.g., 0.75) is more conservative, prioritizes specificity, reducing false positives but lowering recall. A lower threshold (e.g., 0.25) is less conservative, increases recall at the cost of specificity. Once a threshold is selected in the application, various performance metrics, including accuracy, BAC, sensitivity, specificity, precision, F_1 score and P_4 score (see metrics definition in supplementary materials) are provided, and the model predicts the relevance of the unscreened publications.

Ensemble strategies

AIM Review employs ensemble learning strategies when multiple text vectorization modalities are selected, allowing the combination of complementary textual representations to improve prediction robustness. Two ensemble approaches are supported: stacked generalization (default) and modality fusion.

In stacked generalization, an independent base classifier is trained for each selected vectorization modality. The predictions of these base models are then aggregated and used as input features for a meta-learner, which generates the final relevance prediction. This layered architecture enables the model to capture different semantic and lexical aspects of the data across modalities. Both base learners and the meta-learner are trained and evaluated within the NCV framework, ensuring consistent model selection and reliable performance estimation. Although stacking can enhance predictive

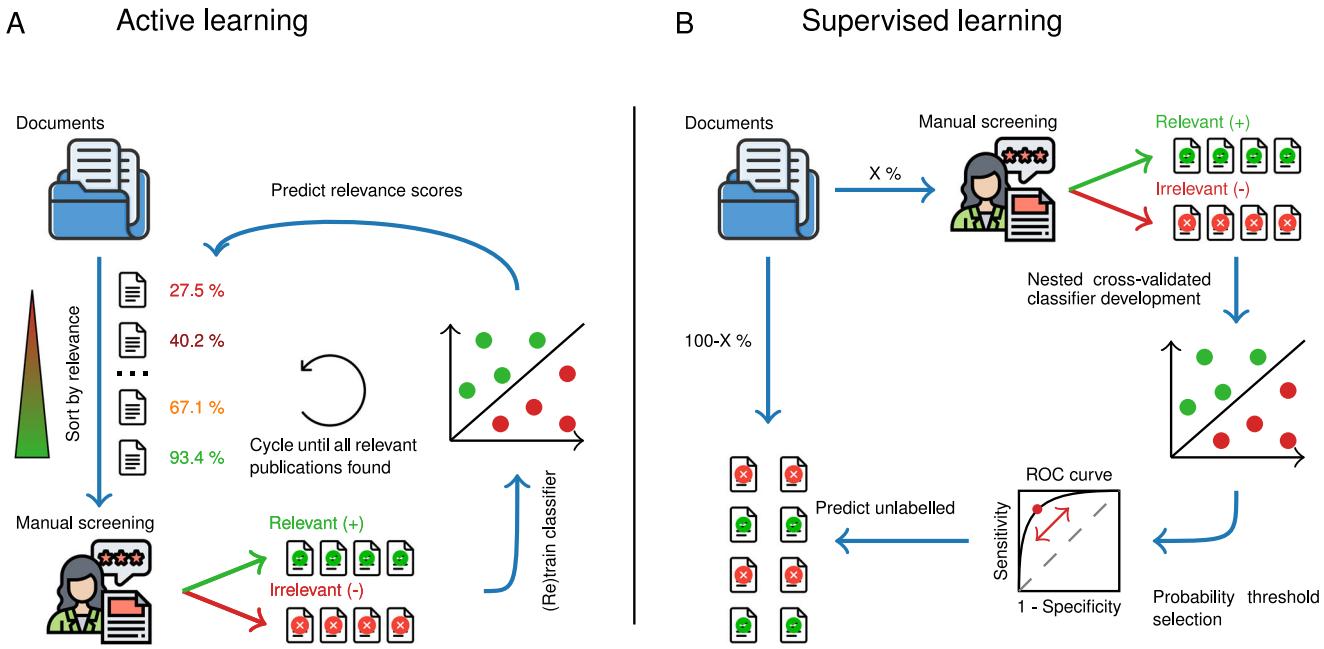

Fig. 4 | Active learning and supervised learning frameworks in AIM Review. **A** Active learning pipeline for labeling. Documents (titles and abstracts from publications) are screened by the researcher. Once active learning is activated, titles and abstracts are converted into numerical representations using selected vectorization methods, and a classifier is trained on publications screened by the user to predict relevance. The classifier is then applied to predict the relevance scores of unscreened publications, which are displayed in the application. The researcher can sort publications by relevance (heist or lowest relevance first) or uncertainty (closeness to 50% relevance). This process is repeated with each button press or at set intervals. Sorting by highest relevance allows researchers to stop screening once no relevant papers appear. **B** Supervised learning pipeline for ML prediction. A subset of documents (X%) is manually screened to train and optimize model parameters via nested cross-validation. After finding the optimal hyperparameters, the model is applied to all screened samples, and an ROC curve is created to assess sensitivity and specificity at different relevance thresholds. The remaining publications (100-X%) are labeled using a user-defined threshold, with higher thresholds (e.g., 0.5–0.75) labeling only highly probable relevant papers, and lower thresholds (e.g., 0.25-0.5) allowing more papers to be labeled as relevant. Artwork was created using Flaticon designs. Abbreviations: ROC receiver operating characteristic.

performance, it increases computational demands due to the training of multiple models.

In contrast, modality fusion combines all selected vectorization modalities into a single feature space by concatenating their representations, upon which a single classifier is trained. This approach reduces model complexity and computational overhead, making it suitable when resources are limited or when the modalities provide complementary information that jointly capture diverse textual patterns. Figure S1 in the supplementary materials illustrates both ensemble strategies.

Case studies

We assessed the *AIM Review* pipelines using six systematic review case studies with varying search sizes and acceptance rates. Three case studies[26–28] were accessed through professional networks from the fields of psychology/psychiatry. An additional three case studies[29–31] were later accessed from Synergy[32], a free and open dataset on study selection in systematic reviews, to evaluate the generalizability of AIM Review to other disciplinary fields, such as computer science, environmental health and endocrinology; the results of these analyses are provided in the supplementary materials. Case study 1 evaluated universal prevention strategies for affective and psychotic disorders (16,661 publications, 1.21% included for full-text screening)[28]. Case study 2 examined data-driven biological subtypes in schizophrenia and bipolar disorder (5536 publications, 7.03% included)[27]. Case study 3 identified studies using the Adult Attachment Interview (729 publications, 53.36% included)[26]. Case study 4 examined fault prediction performance in software engineering (8793 publications, 1.18% included)[29]. Case study 5 examined the effects of postmenopausal hormone replacement therapy (366 publications, 21.86% included)[31]. Case study 6 studied the transgenerational inheritance of health effects (48,375 publications, 1.57% included)[30]. The inclusion and exclusion criteria of each study are described in the supplementary materials. All tasks were performed using *AIM Review* on Google Chrome v136 in a PC equipped with an Intel i7-4770 processor and 32 GB RAM.

To ensure consistency in the evaluation of vectorization and classification methods in *AIM Review*, we maintained fixed configuration parameters for all three case studies across active learning and supervised learning pipelines. We utilized TF-IDF, LSA of TF-IDF, Doc2Vec, univBERT, all-MiniLM-L6-v2 and SPECTER2 as text vectorization methods, and LR, L-SVM and SeqNN as classifiers (SeqNN used only in supervised learning; see definitions and configuration parameters in the supplementary materials). Additionally, we also utilized stacked generalization (with a LR meta-model) and fusion of modalities to combine all vectorization methods. Within each case study, we also maintained the same NCV structure across all experiments to ensure fair and directly comparable performance evaluations. For case studies 4–6, conducted later to validate AIM Review in other disciplinary fields, we limited the evaluation to the combination of all vectorization methods.

For supervised learning, we used 20% of publications as our model training sample and 80% of publications as our OOCV sample. Within the training sample, a nested 5×5 cross-validation (CV2) framework with 5 outer permutations was used for model hyperparameter tuning. For LR and L-SVM, the regularization parameter C was optimized over the range [2^{-4}, 24] to improve the BAC of predictions. The structure of SeqNN was optimized between three structures: a single 500-unit layer, two 50-unit layers, and two 100-unit layers. These neural network structures were selected to balance model complexity and overfitting risk, ranging from a large-capacity single-layer model to smaller, deeper architectures better suited for capturing non-linear patterns in data. Performance of the optimal model was assessed using the AUROC metric of the model applied to the ensembled validation CV2 sets (OOT) and on the 80% held-out set (OOCV). Additionally, BAC, sensitivity, specificity and P_4 score are measured by applying a default threshold of 0.5. We did not use F_1 scores as our

model optimization metric since the score does not account for true negatives, and we aimed to optimize models with a robust ability to correctly reject irrelevant papers—reflecting a key functionality of the application in efficiently filtering out non-relevant literature.

For active learning, models were trained using all available screened publications and applied to the unscreened publications to predict a probability of relevance without optimization of parameters, as the goal was iterative low-latency ranking of records rather than identifying a globally optimal classifier. This pipeline was evaluated using the WSS metric, which quantifies the reduction in screening effort compared to random sampling while achieving a target recall level[13]. Specifically, we used the WWS that measures the reduction of workload while identifying 95% ($WSS_{95\%}$) and 100% ($WSS_{100\%}$) of the relevant records.

$$WSS_{X\%} = \frac{N_T - N_{X\%recall}}{N_T} \times 100 \quad (1)$$

Where N_T is the total number of publications in the case study and $N_{X\%recall}$ is the number of publications that need to be reviewed to achieve X% recall. To obtain a reliable WSS, we simulated the title and abstract screening of all three case studies ($n = 5$ repetitions with shuffling) and sorted the publications by their relevance (highest relevance first) every five new publications were screened and assigned the correct label. The screening process was simulated by waiting 2 s and assigning the observed relevance of the publication. The 2-second delay simulated the time required for model retraining, relevance scoring, and sorting of publications—not human screening time, as labels were already known.

Data availability

The results described in this publication are stored in an Open Science Framework repository (https://doi.org/10.17605/OSF.IO/GH3UK). The manual screenings of the three case studies are available upon request.

Code availability

Code used to reproduce the results and figures in this publication are available in an Open Science Framework repository (https://doi.org/10.17605/OSF.IO/GH3UK). The code for the AIM Review web application is proprietary and not publicly available. Access can be discussed upon request for research collaborations or evaluation purposes. Please contact us for further inquiries.

Received: 6 June 2025; Accepted: 10 February 2026;
Published online: 21 February 2026

References

1. Bornmann, L. & Mutz, R. Growth rates of modern science: a bibliometric analysis based on the number of publications and cited references. *J. Assoc. Inf. Sci. Technol.* **66**, 2215–2222 (2015).
2. Bornmann, L., Haunschild, R. & Mutz, R. Growth rates of modern science: a latent piecewise growth curve approach to model publication numbers from established and new literature databases. *Humanit Soc. Sci. Commun.* **8**, 224 (2021).
3. Borah, R., Brown, A. W., Capers, P. L. & Kaiser, K. A. Analysis of the time and workers needed to conduct systematic reviews of medical interventions using data from the PROSPERO registry. *BMJ Open* **7**, e012545 (2017).
4. Ioannidis, J. P. A. The mass production of redundant, misleading, and conflicted systematic reviews and meta-analyses. *Milbank Q.* **94**, 485–514 (2016).
5. Acosta, J. N., Falcone, G. J., Rajpurkar, P. & Topol, E. J. Multimodal biomedical AI. *Nat. Med.* **28**, 1773–1784 (2022).
6. Kraljevic, Z. et al. Foresight; a generative pretrained transformer for modelling of patient timelines using electronic health records: a retrospective modelling study. *Lancet Digit. Health* **6**, e281–e290 (2024).
7. Rasmy, L., Xiang, Y., Xie, Z., Tao, C. & Zhi, D. Med-BERT: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *npj Digit. Med.* **4**, 86 (2021).
8. Luo, X. et al. Large language models surpass human experts in predicting neuroscience results. *Nat. Hum. Behav.* **9**, 305–315 (2025).
9. van de Schoot, R. et al. An open source machine learning framework for efficient and transparent systematic reviews. *Nat. Mach. Intell.* **3**, 125–133 (2021).
10. Przybyła, P. et al. Prioritising references for systematic reviews with RobotAnalyst: a user study. *Res Synth. Methods* **9**, 470–488 (2018).
11. Wallace, B. C., Small, K., Brodley, C. E., Lau, J. & Trikalinos, T. A. Deploying an interactive machine learning system in an evidence-based practice center: abstrackr. In *Proc. 2nd ACM SIGHIT International Health Informatics Symposium* 819–824 (ACM, 2012).
12. Miwa, M., Thomas, J., O'Mara-Eves, A. & Ananiadou, S. Reducing systematic review workload through certainty-based screening. *J. Biomed. Inf.* **51**, 242–253 (2014).
13. Cohen, A. M., Hersh, W. R., Peterson, K. & Yen, P.-Y. Reducing workload in systematic review preparation using automated citation classification. *J. Am. Med. Inform. Assoc.* **13**, 206–219 (2006).
14. Ouzzani, M., Hammady, H., Fedorowicz, Z. & Elmagarmid, A. Rayyan—a web and mobile app for systematic reviews. *Syst. Rev.* **5**, 210 (2016).
15. Jonnalagadda, S. R., Goyal, P. & Huffman, M. D. Automating data extraction in systematic reviews: a systematic review. *Syst. Rev.* **4**, 78 (2015).
16. Marshall, I. J., Kuiper, J. & Wallace, B. C. RobotReviewer: evaluation of a system for automatically assessing bias in clinical trials. *J. Am. Med. Inform. Assoc.* **23**, 193–201 (2016).
17. Pradhan, R. et al. Automatic extraction of quantitative data from ClinicalTrials.gov to conduct meta-analyses. *J. Clin. Epidemiol.* **105**, 92–100 (2019).
18. Khalil, H. et al. Automation tools to support undertaking scoping reviews. *Res Synth. Methods* **15**, 839–850 (2024).
19. Boetje, J. & van de Schoot, R. The SAFE procedure: a practical stopping heuristic for active learning-based screening in systematic reviews and meta-analyses. *Syst. Rev.* **13**, 81 (2024).
20. Ferdinands, G. et al. Performance of active learning models for screening prioritization in systematic reviews: a simulation study into the average time to discover relevant records. *Syst. Rev.* **12**, 100 (2023).
21. Harmsen, W. et al. Machine learning to optimize literature screening in medical guideline development. *Syst. Rev.* **13**, 177 (2024).
22. Callaghan, M. W. & Müller-Hansen, F. Statistical stopping criteria for automated screening in systematic reviews. *Syst. Rev.* **9**, 273 (2020).
23. Michelson, M. & Reuter, K. The significant cost of systematic reviews and meta-analyses: a call for greater involvement of machine learning to assess the promise of clinical trials. *Contemp. Clin. Trials Commun.* **16**, 100443 (2019).
24. Kolev, S. D. *Solution of Mathematical Models of Flow Systems Used in Analytical Chemistry and Process Analysis in the Case of Slug and Time Injection*. *Analytrca Chrmrca Acfa* **229**, 183–189 (1990).
25. Varma, S. & Simon, R. Bias in error estimation when using cross-validation for model selection. *BMC Bioinforma.* **7**, 91 (2006).
26. Bakermans-Kranenburg, M. J., Dagan, O., Cárcamo, R. A. & van IJzendoorn, M. H. Celebrating more than 26,000 adult attachment interviews: mapping the main adult attachment classifications on personal, social, and clinical status. *Attach Hum Dev* 1–38 https://doi.org/10.1080/14616734.2024.2422045.
27. Promet, L., Mena, S., Lalousis, P. A. & Koutsouleris, N. Data-driven biological subtypes in schizophrenia spectrum disorders and bipolar disorder: a systematic review. https://www.crd.york.ac.uk/prospero/display_record.php?ID=CRD42024572364 (CRD, 2024).
28. Brodeur, S. et al. Why we need to pursue both universal and targeted prevention to reduce the incidence of affective and psychotic disorders: systematic review and meta-analysis. *Neurosci. Biobehav. Rev.* **161**, 105669 (2024).

29. Hall, T., Beecham, S., Bowes, D., Gray, D. & Counsell, S. A systematic literature review on fault prediction performance in software engineering. *IEEE Trans. Softw. Eng.* **38**, 1276–1304 (2012).
30. Walker, V. R. et al. Human and animal evidence of potential transgenerational inheritance of health effects: An evidence map and state-of-the-science evaluation. *Environ. Int* **115**, 48–69 (2018).
31. Nelson, H. D., Humphrey, L. L., Nygren, P., Teutsch, S. M. & Allan, J. D. Postmenopausal hormone replacement therapyscientific review. *JAMA* **288**, 872–881 (2002).
32. De Bruin, J., Ma, Y., Ferdinands, G., Teijema, J. & de Schoot, R. SYNERGY - Open machine learning dataset on study selection in systematic reviews. Preprint at https://doi.org/10.34894/HE6NAQ (2023).
33. Yu, Z., Kraft, N. A. & Menzies, T. Finding better active learners for faster literature reviews. *Empir. Softw. Eng.* **23**, 3161–3186 (2018).
34. Garrido-Merchan, E. C., Gozalo-Brizuela, R. & Gonzalez-Carvajal, S. Comparing BERT against traditional machine learning models in text classification. *J. Comput. Cogn. Eng.* **2**, 352–356 (2023).
35. Delgado-Chaves, F. M. et al. Transforming literature screening: the emerging role of large language models in systematic reviews. *Proc. Natl. Acad. Sci. USA* **122**, e2411962122 (2025).
36. Syriani, E., David, I. & Kumar, G. Screening articles for systematic reviews with ChatGPT. *J. Comput Lang.* **80**, 101287 (2024).
37. Tran, V.-T. et al. Sensitivity and specificity of using GPT-3.5 turbo models for title and abstract screening in systematic reviews and meta-analyses. *Ann. Intern. Med.* **177**, 791–799 (2024).
38. Pedregosa, F. et al. Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
39. Chatham, H. et al. pyodide/pyodide: 0.27.4. Preprint at https://doi.org/10.5281/zenodo.15046691 (2025).
40. Martín A. et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. In *Proc. 12th USENIX Conference on Operating Systems Design and Implementation* 265–283 (USENIX Association, 2016).
41. Jain, S. M. Hugging face. in *Introduction to transformers for NLP: With the hugging face library and models to solve problems* 51–67 (Springer, 2022).
42. Sparck Jones, K. A statistical interpretation of term specificity and its application in retrieval. *J. Documentation* **28**, 11–21 (1972).
43. Cer, D. et al. Universal sentence encoder for English. In *Proc. Conference on Empirical Methods in Natural Language Processing: System Demonstrations* 169–174 (NLP, 2018).
44. Le, Q. & Mikolov, T. Distributed representations of sentences and documents. In *International conference on machine learning* 1188–1196 (PMLR, 2014).
45. Wang, W. et al. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Adv. Neural Inf. Process Syst.* **33**, 5776–5788 (2020).
46. Cohan, A., Feldman, S., Beltagy, I., Downey, D. & Weld, D. S. Specter: Document-level representation learning using citation-informed transformers. *arXiv preprint arXiv:2004.07180* (2020).
47. Wolpert, D. H. Stacked generalization. *Neural Netw.* **5**, 241–259 (1992).

Acknowledgements

The authors would like to thank Google's Firebase support team who helped solve software compatibility limitations. In addition, the authors would like to thank researchers that have provided feedback on the application.

Author contributions

S.M. developed the code that runs the application and manages the code repository. S.M., N.K., and P.A.L. conceptualized and designed the application and wrote the first draft of the manuscript. E.R.G., F.C., G.R.J., J.v.T, L.M., L.B., N.S., and P.F.P. provided conceptual feedback on the design of the application, as well as in the later usability, visual appearance and computational performance. S.M., D.O., M.S.A., L.P., M.J.B.K., and M.H.v.I. generated the title and abstract screenings from the systematic reviews used to benchmark the application. All authors contributed to writing the manuscript.

Competing interests

P.A.L. has received honoraria for talks presented at educational meetings organized by Boehringer-Ingelheim outside of the submitted work. The other authors do not have competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s44387-026-00080-8.

Correspondence and requests for materials should be addressed to Sergio Mena.