# Enhancing systematic reviews: an in-depth analysis on the impact of active learning parameter combinations for biomedical abstract screening.

OFORI-BOATENG, R., TRUJILLO-ESCOBAR, T.G., ACEVES-MARTINS, M., WIRATUNGA, N. and MORENO-GARCIA, C.F.

2024

Artificial Intelligence In Medicine 157 (2024) 102989

Contents lists available at ScienceDirect

Artificial Intelligence In Medicine

journal homepage: www.elsevier.com/locate/artmed

# Enhancing systematic reviews: An in-depth analysis on the impact of active learning parameter combinations for biomedical abstract screening

Regina Ofori-Boateng [a,*], Tamy Goretty Trujillo-Escobar [b], Magaly Aceves-Martins [c], Nirmalie Wiratunga [a], Carlos Francisco Moreno-Garcia [a]

[a] School of Computing, Robert Gordon University, Aberdeen, AB10 7GE, Scotland, UK
[b] Universidad El Bosque, Bogotá, Colombia
[c] The Rowett Institute, University of Aberdeen, Aberdeen, AB25 2ZD, Scotland, UK

## ARTICLE INFO

*Keywords:*
Evidence-based medicine
Abstract screening
Active learning
Machine learning
Human-in-the-loop
Systematic reviews

## ABSTRACT

Systematic Review (SR) are foundational to influencing policies and decision-making in healthcare and beyond. SRs thoroughly synthesise primary research on a specific topic while maintaining reproducibility and transparency. However, the rigorous nature of SRs introduces two main challenges: significant time involved and the continuously growing literature, resulting in potential data omission, making most SRs become outmoded even before they are published. As a solution, AI techniques have been leveraged to simplify the SR process, especially the abstract screening phase. Active learning (AL) has emerged as a preferred method among these AI techniques, allowing interactive learning through human input. Several AL software have been proposed for abstract screening. Despite its prowess, how the various parameters involved in AL influence the software's efficacy is still unclear. This research seeks to demystify this by exploring how different AL strategies, such as initial training set, query strategies etc. impact SR automation. Experimental evaluations were conducted on five complex medical SR datasets, and the GLM model was used to interpret the findings statistically. Some AL variables, such as the feature extractor, initial training size, and classifiers, showed notable observations and practical conclusions were drawn within the context of SR and beyond where AL is deployed.

**Contents**



* Corresponding author.
*E-mail address:* r.ofori-boateng@rgu.ac.uk (R. Ofori-Boateng).

https://doi.org/10.1016/j.artmed.2024.102989
Received 17 October 2023; Received in revised form 16 September 2024; Accepted 22 September 2024
Available online 26 September 2024

## 1. Introduction

*Evidence-Based Medicine* (EBM) involves systematically identifying, evaluating, and applying contemporary research findings to influence policies and bolster clinical decision-making and practices [1]. This is achieved through a well-structured approach known as a *Systematic Review* (SR). An SR aims to provide a comprehensive synthesis of all primary research relevant to a specific research question, ensuring the inclusion of all potentially relevant studies while preserving transparency and reproducibility [2,3]. Thus, an SR is the "heart of EBM" [3]. The SR paradigm typically involves: (i) the formulation of a protocol; (ii) comprehensive exploration of all potential databases for relevant studies; (iii) screening through titles and abstracts of the studies yielded from the search; (iv) meticulous full-text screening; (v) data extraction and synthesis of successful studies from the previous stage; and (vi) interpretation and publication of the findings. We refer readers to work done by Kitchenham et al. [4] for a more comprehensive understanding of SR steps.

However, the rigour of the SR process simultaneously introduces two main substantial challenges [5,6]. Firstly, the complex nature of the SR demands both a significant time investment and considerable mental exertion from researchers given the stringent guidelines governing the process [7]. Empirical evidence reports that an SR typically requires approximately 15 months to reach completion and publication [8]. Secondly, researchers confront the daunting prospect of an ever-expanding body of published literature daily at a rate that poses significant challenges to conducting an SR [9]. Thus, recent published and possibly pertinent studies might be overlooked during the search phase, thereby creating a "missing data" dilemma [10]. Consequently, many SRs become obsolete after completion, necessitating major revisions and updates [10,11].

As a countermeasure to these challenges, artificial intelligence (AI) techniques have been harnessed to alleviate the pressures associated with SRs [5,12]. In the multitude of steps in the SR process, the *abstract/citation screening* phase has been identified as particularly burdensome [5,11,13]. From literature, three widely utilised AI techniques have been proposed to automate this stage—*text classification, screening prioritisation*, and *active learning (AL)*, with AL demonstrating dominance [5]. To cite an example, among the 16 abstract screening SR software on the SR Toolbox [14][1] with detailed and readily available literature explaining its AI methodology, 11 out of these tools use AL. AL, a subset of machine learning (ML), involves a dynamic interaction between ML algorithms and human users, where the algorithm is equipped to actively select its learning data by querying the user for data labelling [15]. As such, it is known as a "human in the loop" method. Various AL strategies have been implemented in abstract screening tools such as AsReview [10], Abstrackr [16], Rayyan [17], Colandr [18], Research Screener [19], SWIFT Active Screener [20], among others [5]. Nonetheless, it remains unclear how these AL techniques, such as the size of the initial training set, query strategies, balancing method, and the classifier chosen by the reviewer, affect the performance of these tools, based on the documented studies and reports on these tools [12,21]. Moreover, in some newer abstract screening such as AsReview and Robot Screener, which allow a variety of different AL strategies, end users often lack clarity on how these varied AL combinations impact the performance for citation automation [22]. Consequently, users often resort to the tool's default configurations without a solid comprehension of how the different combinations influence the tool's performance [21].

Thus, this study aims to comprehensively examine some of the widely adopted AL strategies used for citation automation through an ablation study. We aim to elucidate to the general public, particularly non-technical systematic reviewers, how these different AL combinations impact the performance metrics of screening tools. To help us understand better these AL techniques proposed, we focus on the AsReview software. We selected this tool because it is recent, and current studies attest to its efficacy and ability to surpass other tools in case study analysis [12,23]. Additionally, unlike other AL tools, which offer a restricted selection of classifiers and vectorisation/feature extraction techniques (converting the abstracts to numeric values), AsReview provides greater flexibility by allowing different types of classifiers and vectorisation approaches [10], rendering it particularly advantageous in this ablation study research.

A related study done by Ferdinands et al. [24,25] focused on the effects of AL strategies using the AsReview tool. They investigated how AL strategies such as the impact of two different vectorisation methods, and two different query strategies, different classifiers perform across six SRs datasets. However, this study extends their research scope to evaluate how factors such as the number of "relevant" and irrelevant" instances chosen in the initial training size, the sample size of the initial training data, the query strategy and the chosen data sampling impact the overall performance of the AL model. Ultimately, this paper aims to address the following research questions:

1. *How does the sample size of the initial training set/prior knowledge chosen influence the performance of AL models?*
   To answer this question, we investigate three distinct initial labelled training sample sizes: 20, 60, and 100. These sizes are selected based on estimations from reported studies [26]. For instance, 100 is selected because some existing SR AL software requires a minimum of 100 labelled training samples for training to begin [18]. 20 is selected to represent the smallest feasible training sample size range that can be chosen for training the model, while 60 is selected to depict the intermediate sample.
2. *What is the impact of the number of "relevant" and "irrelevant" instances in the initial labelled training sample size on AL models?*
   Given that SR datasets often exhibit class imbalance ratios, with more irrelevant studies than relevant ones [27], we investigate two distinct divisions (relevant:irrelevant) within the labelled initial training sample. For the 20 sample; 2:18 and 5:15. For 60; 6:54 and 15:45-while with the 100 sample size 10:90 and 25:75, giving a 0.1 and 0.3 class imbalance ratios in all the sets. We investigate these divisions to capture the potential range of relevant studies/abstracts that a user may label for training which are based on estimations from reported literature [28].
3. *What is the impact of different feature extraction methods on AL models?*
   We investigate three possible varieties of feature extraction methods proposed for abstract screening—a traditional method,

[1] http://systematicreviewtools.com/.

Term frequency-inverse document frequency (TF-IDF), a word embedding technique, Document to Vector (Doc2Vec) [29] and a more advanced transformer approach Sentence Bidirectional Encoder Representations from Transformers (S-Bert) [30].

4. *How do abstract screening AL models perform across four traditional classification techniques?* We scrutinise the performance of AL models with the two most common query strategies used in citation screening, namely certainty and uncertainty queries [15, 26]. Additionally, we explore four traditional ML algorithms: Support Vector Machine (SVM) [31], Naive Bayes (NB) [32], Logistic Regression (LR) [33], and Random Forest (RF) [34].
5. *Do data balancing strategies influence the overall performance of AL models?*
   To address this, we explore two different data balancing strategies, undersampling and a combined variant of oversampling and undersampling called Double [28].

In summary, the primary contributions of this paper are as follows: (1) We investigate how the various AL combinations affect the overall performance of the tool for citation screening, (2) we perform a statistical experiment to detail and buttress the influence of the variables raised in the five research questions on the performance metrics of these tools. To the best of our knowledge, this might be the first study to examine the impact of these variables in a statistical context and from this particular perspective, (3) we carry out experiments across a total of five systematic medical review datasets—two private and three public—to evaluate the efficacy of these variable combinations, (4) a comprehensive discussion of experimental findings supported by statistical results. All source codes used for this study are found on the original AsReview software GitHub repository.[2] In this work, we used the latest version at the time of the study, that is v1.4.[3] of the AsReview software. The AsReview GitHub repository cloned which was used for the analysis can be found on Google Colab can be found on this Open Science Framework (OSF)[4]

## 2. Active learning tools for abstract screening

AL is a semi-supervised ML technique predicated on the assumption that a learning algorithm can perform better with less training if it can choose the data from which it learns. Thus, prompting the learner to query a human or an oracle for labelling [15]. Hence, AL is often referred to as a "query learning" or *"human-in-the-loop"* strategy.

Contrary to passive learning methods, such as supervised and unsupervised learning, AL enables the classifier to selectively choose the data points from which it can learn most efficiently instead of learning from a randomly collected dataset [35]. The advantage of this method is underscored by the fact that passive learning techniques typically require a substantial amount of randomly collected data for training, potentially leading to increased time and resource costs. For instance, supervised learning necessitates labelled data for training, which could significantly increase the manual labour required to create the training set. Conversely, AL utilises strategic querying to selectively identify the documents from the pool of unlabelled data that require labelling, thereby necessitating fewer labelled data to make predictions about unseen data [36]. This explains why most SR citation tools opt to implement AL strategies for citation screening over supervised and unsupervised methodologies.

The AL cycle typically involves the following steps: (1) data collection, where a given labelled sample data is used as a starting point to train the model, (2) training the model with the initial labelled data, where the model then makes predictions on new, unlabelled data by ranking its relevancy, (3) querying the human from the subset of the predicted unlabelled data points that it is either sure about or has low confidence of its predictions, (4) the human/oracle labels those selected data brought forward by the model, (5) integration of the newly labelled data by the human into the initial training set, (6) repetition of the cycle from step 2 until the desired performance is achieved or the cost of additional querying and labelling outweighs performance improvements [15].

Various tools leveraging these AL steps have been deployed for SR review automation. According to the research done by Yu et al. [26], these tools seek to address the following: (1) when the classifier should commence training, (2) how the classifier should select studies to inquire about (query strategy), and (3) how to balance the training data to solve the class imbalance. A summary of these methodologies and their implementation in existing AL citation screening tools, along with their various classifiers, is presented in Table 1. To begin, *AsReview*[10][5] is a free, open-source desktop application written in Python. As one of the newest tools available, it stands out for its ability to incorporate a wide range of ML techniques, such as classifiers, feature extractors and query strategies, accommodating the varying SRs projects with greater flexibility. This feature sets AsReview apart from other AL SR tools. AsReview implements a variety of classifiers, both traditional and advanced neural networks, such as SVM, NB, LR, RF, Deep Neural Networks (DNNs), and Long Short-Term Memory Networks (LSTM) [37] with NB as the default classifier. The software allows Bag of Words (BoW), TF-IDF, Doc2Vec, sBERT and embedding IDF for vectorisation with TF-IDF as the default. Regarding query strategies, the AsREview software offers uncertainty-based; certainty-based; random sampling; mixed sampling with certainty query strategy as the default allowing the user to label citations that AsReview is most confident about the class, thus ranking the relevant citations first. This decision is fed as the input to the selected/default classifiers, which then rank the remaining citations, and the process continues until the reviewer decides to stop screening. Thus, saving training and human time. Another merit of the AsReview software is that it does not require a substantial number of initial training labelled citations. Users need only to label at least one relevant and one irrelevant citation for the AL cycle to begin.

*Rayyan* [17][6] is a free, cloud-based, closed-source application for both mobile and web, developed using Ruby on Rails and running on Heroku. Feature extraction is conducted using a Bag of Words approach, encompassing unigram and bigram along with MeSH inputs from the user. Rayyan uses only SVM for classifying uploaded citations as relevant or irrelevant. Though the type of query strategy is not explicitly stated, Rayyan uses a five-star ranking to suggest how likely a citation is to be included or excluded, thereby guiding the user on its relevance. *Abstrackr* [16][7] is another free open-source web application. It employs the uncertainty query strategy, presenting the user with citations/abstracts for which the system has the least confidence regarding their classification (as either relevant or irrelevant). The user's annotation (classifying as either relevant, borderline, or irrelevant) forms the training data for the SVM to rank the remaining citations. Abstrackr utilises the concept of N-Gram and TF-IDF for feature extraction. Additionally, the tool incorporates aggressive undersampling to handle dataset imbalance. The AL cycle only begins when the user labels a fair number of citations presented by the uncertainty sampling technique, which can lengthen the classifier's training time to make predictions, unlike AsReview. *Research Screener* [19][8] is a free, open-source, cloud-based recent AL SR tool. It leverages state-of-the-art text mining approaches, notably paragraph embedding, representing

[2] https://github.com/asreview/asreview.
[3] https://zenodo.org/records/10393445.
[4] https://osf.io/kas2c/.
[5] https://asreview.nl.
[6] https://www.rayyan.ai/.
[7] http://abstrackr.cebm.brown.edu.
[8] https://researchscreener.com/.

**Table 1**
Summary of features in existing AL SR citation automation tools.

| AL systematic tools | Year | Feature extractor | Classifiers | Query strategy | Balancing |
|---|---|---|---|---|---|
| Abstrackr | 2012 | N-Gram<br>TF-IDF | SVM | Uncertainty | Undersampling |
| AsReview | 2021 | TF-IDF<br>S-Bert<br>Doc2Vec | NB<br>RF<br>DNN<br>LR<br>LSTM<br>SVM | Certainty<br>Uncertainty<br>Random sampling<br>Mixed sampling | Double<br>Triple<br>Simple<br>Undersampling |
| Colandr | 2018 | Word2Vec | SVM with SGD | Certainty | Weighting |
| EPPI Reviewer | 2010 | BoW | SVM | Not Stated | Not Stated |
| FastRead | 2018 | BoW<br>TF-IDF | SVM | Uncertainty | Mixed |
| Rayyan | 2016 | BoW | SVM | Not stated:<br>Uses a five-star score rating | Not stated |
| Research Screener | 2021 | Doc2vec<br>N-gram<br>TF-IDF | SVM | Certainty | Not stated |
| Robot Analyst | 2018 | TF-IDF | SVM<br>LDA | Uncertainty | Not stated |
| SWIFT Active Screener | 2020 | N-gram<br>TF-IDF | Log-linear model | Uncertainty | Not stated |

abstracts as word embeddings rather than counting sequences of words in the form of n-grams. Similar to AsReview, in Research Screener, the AL begins after manually labelling at least one article relevant to the review, helping to prioritise relevant articles. The tool uses a certainty-based sampling query strategy with AL algorithms, which include SVM, and deep learning models for th classification.

The *SWIFT Active Screener* tool [20][9] on the other hand, is a closed-source web AL SR application. The tool utilises certainty-based sampling as its query selection strategy and with an L2-regularised log-linear model as the classifier. Feature extraction is performed using BoWs in combination with TF-IDF. These extracted features are then used to train the log-linear model after manual annotation by the user. Similarly, *Colandr* [18][10] is a free, closed-source web application offering dual functionality by automating both the screening and data extraction phases. For the screening process, Colandr employs an AL SVM with a Stochastic Gradient Descent (SGD) linear model. SGD is a variant of gradient descent that iteratively uses one observation at a time to minimise the cost function and updates the parameters until all the training data has been utilised. In Colandr, the AL process starts after the user labels 100 citations, thus an initial training size of 100. Feature extraction is performed using a variant of Doc2Vec, and the query strategy is certainty-based sampling. Additionally, *RobotAnalyst* [38][11] is a request-based, closed-source web AL SR application. The tool involves the development of topics using the Latent Dirichlet Allocation (LDA) concept from the abstract document. Feature extraction is performed via BoW and TF-IDF methodologies. These features are then passed to an SVM classifier.

*EPPI-Reviewer* [39,40][12] is a paid, closed-source web application that uses Tri-gram and TF-IDF for rating the importance of each word in the document, with extracted features passed onto an SVM classifier. Lastly, the *SySrEV* [41][13] is a free, closed-source web application with a paid version for private projects. However, it lacks a well-detailed explanation of its framework. Nonetheless, it has been reported to deploy the concept of AL.

[9] https://www.sciome.com/swift-activescreener/.
[10] https://www.colandrapp.com/.
[11] http://nactem.ac.uk/robotanalyst/.
[12] http://eppi.ioe.ac.uk/cms/Default.aspx?tabid=3396.
[13] https://sysrev.com/.

## 3. Methodology design

This section outlines the research methodology used in this study, encompassing the datasets, feature extraction techniques, query strategies, classifiers, training methods, and evaluation metrics implemented. Fig. 1 visually represents the proposed methodology used, drawn from the methodology in the AsReview Software [10].

### 3.1. Dataset

We employed five health-related datasets to train, evaluate, and analyse the diverse AL strategies implemented. Two out of these five datasets are private, while the remaining three are publicly accessible on GitHub.[14] For reproducibility and contribution, the private datasets are available on this OSF repo.[15] The two private datasets used were the Aceves-Martin_2022 dataset, investigating the nutritional status and disparities among imprisoned populations, and the Aceves-Martins_2021 [42], exploring oral health among Mexican children. On the other hand, the three public datasets included the Angiotensin-converting-enzyme (ACE) Inhibitors dataset authored by Cohen et al. [13], the van de Schoot_2017 dataset [22] focusing on post-traumatic stress disorder (PTSD), and the Kwok dataset [43] discussing Virus Metagenomics in animals. Table 2 provides a detailed description of these datasets. Among the five datasets, the Aceves-Martin_2022 has been reported as the most challenging dataset because of how words were used in generating the query for search to capture the prison population, thus capturing a wider number of papers. Different terms were used to define the population and the problem; thus, the search databases dropped many "relevant references", which gave a lot of "irrelevant papers". In all, the total number of papers for this dataset was 13,002. However, in performing the experiments, the number of irrelevant studies was reduced in this study due to computational resources.

To further provide additional details on the private datasets, there exists only one experimental study performed with this dataset, specifically the AM_2021 by [44]. In the study, the authors proposed the

[14] https://github.com/asreview/systematic-review-datasets.
[15] https://osf.io/kas2c/.

**Table 2**
Summary of datasets used in this study with their Imbalance Ratios (IR).

| Dataset | Focus | Total papers | Papers excluded | Papers included | IR |
|---|---|---|---|---|---|
| Aceves-Martins_2022 (AM_2022) | Nutritional status of Prisoners | 5069 | 5000 | 69 | 1:73 |
| Aceves-Martins_2021 (AM_2021) | Oral Health in Mexico children | 807 | 789 | 18 | 1:44 |
| ACEInhibitors_Cohen_2006 (ACE_2006) | ACEInhibitors | 2544 | 2503 | 41 | 1:62 |
| Kirsty_Kwok_2020 (KK_2020) | Virus Metagenomics | 2481 | 2361 | 120 | 1:20 |
| van_de_Schoot_2017 (VS_2017) | PTSD Trajectories | 6189 | 6146 | 43 | 1:143 |

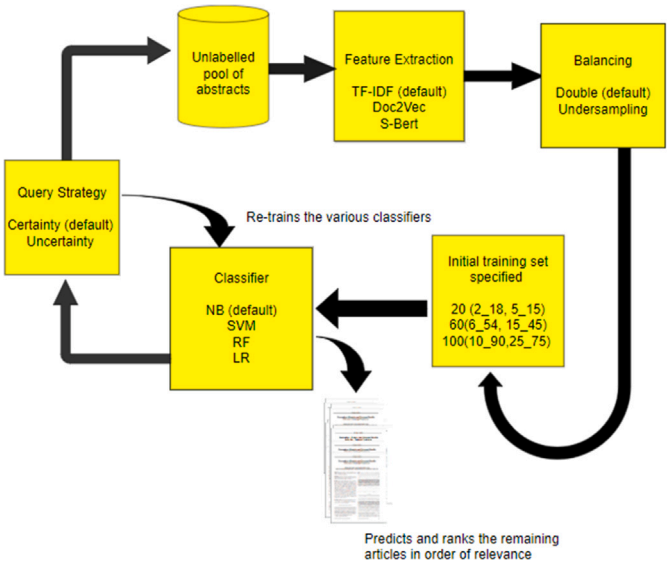


**Fig. 1.** Summary of the methodology design with the default variables proposed in the AsReview tool except for the training size/prior knowledge, which is left open to the end user.

use of an attention based LSTM and Bi-LSTM for the abstract screening classification tasks. However, the study focused on treating this task as a text classification task, hence not allowing humans in the loop as in the AL cycle. On the other hand, experimental studies on the public datasets (ACE_2006, KK_2020 and VS_2017) dataset have ranged from treating the task as a screening prioritisation [45] to text classification [13,44,46,47] and to AL tasks [10,20]. Diving into the AL strategies proposed for VS_2017, prior research highlighted issues such as the class imbalance, computational requirement, and the complexity of accurately modelling PTSD symptoms, which can be highly variable and influenced by numerous external factors [10]. Also, common challenges reported for the KK_2020 include the dataset's limited size and the difficulty in accurately annotating viral sequences due to the vast diversity of viral species. However, our study specifically addresses the underexplored area of how active learning (AL) strategies can optimise the screening process in systematic reviews related to these dataset. We investigate the impact of different AL parameters on the efficiency and accuracy of identifying relevant studies, a topic not thoroughly covered in previous research.

### 3.2. Feature extraction/vectorisation

The transformation of abstract texts into a machine-readable format is necessary to make them interpretable by ML models. This conversion is accomplished via the vector space model, transforming these texts into vectors, thereby capturing sentence semantics by identifying word similarities. Feature extraction is the method through which these vectors are created, playing a pivotal role in achieving this objective. Vectorisation can be either a classical (traditional) or a distributed representation approach. To answer **RQ3**, we analyse three distinct vectorisation techniques: TF-IDF (traditional), Doc2Vec [29], and Sentence-Bert (Distributed representation) [30].

#### 3.2.1. TF-IDF

It is a statistical technique used to rank the significance of words within a body of text [48]. This method is more advantageous than other traditional methods as it comprises two primary stages: Term Frequency (TF) and Inverse Document Frequency (IDF). The TF stage scrutinises the frequency of words to establish how frequently they appear throughout the document. For instance, if the word "Quality Control" is present 100 times in a 1000-word sample document, the TF score would be determined as $\frac{100}{1000} = 0.1$. Mathematically, the TF core is written as:

$$TF = \frac{number\ of\ times\ term\ appears\ in\ a\ document}{total\ number\ of\ terms\ in\ the\ document} \quad (1)$$

On the other hand, to assess the importance of each word in a document, the IDF Term applies weights to them through a logarithm function. The weight assigned to a word is greater when the word appears more frequently in the corpus. The IDF score ranges from 0 to 1, with higher values indicating that the word is more crucial in the document. Mathematically, it is represented as:

$$IDF = log\frac{total\ number\ of\ documents}{number\ of\ documents\ with\ term\ in\ it} \quad (2)$$

By merging the TF and IDF terms, the resulting TF-IDF probability score falls within the 0 to 1 range. This score is utilised to evaluate the significance of a specific word despite its risk of producing in sparse matrices.

#### 3.2.2. Doc2Vec

Doc2Vec is widely used in NLP applications and has shown promising results in many tasks [49,50]. This sophisticated vectorisation technique creates fixed-length feature representations for variable-length documents, including entire documents, sentences, or paragraphs. It builds on the popular Word2Vec [51] technique, which generates vector representations for individual words in a corpus.

In Doc2Vec, each document is assigned a vector representation that captures its semantic meaning. This is accomplished by training a neural network on a corpus of documents, where the network learns to anticipate the context of a given word within a sentence or document. The document vector is updated during training, along with the word vectors, to reflect the meaning and context of the words in the document. The resulting document vectors can then be used for various downstream tasks such as text classification, document clustering or information retrieval.

#### 3.2.3. S-Bert

It is a modified version of the pre-trained language model BERT [52], which is specifically designed for generating fixed-length vector representations, known as sentence embeddings, that capture the semantic meaning of sentences. To achieve this, S-BERT employs a Siamese network architecture. This architecture consists of two BERT models with shared weights that independently process a pair of input sentences. The final hidden states from both models are concatenated, and the resulting vector is passed through a feedforward neural network to obtain the sentence embeddings. We refer readers to the study done by Reimers et al. [30] for a detailed explanation.

### 3.3. Query strategy

The query strategy (QS) constitutes a critical aspect of AL, facilitating the selection of the most informative or pertinent samples from a collection of unlabelled data for annotation by a human. In AL, QS aims to choose samples that would improve the performance of the ML model when added to its initial training data. From literature, various query strategies have been proposed e.g. Query-by-Committee, Certainty, Uncertainty, Diversity Sampling, and Expected Model Change, among others. For a comprehensive description of the various types of QS, we refer readers to the study conducted by Burr [15].

In the context of SR citation screening, the most popular query strategies proposed in citation screening tools are *Certainty* and *Uncertainty* QS. Thus, in this study, we explore these two QSs to address the first aspect of the research question **RQ4**.

#### 3.3.1. Uncertainty query strategy

Uncertainty QS is a strategy in AL that selects data from the unlabelled pool of examples where the model needs more clarification or is uncertain about its prediction. Uncertainty QS is done either by Maximum Entropy, Least Confidence, and Margin Sampling approaches [15]. However, this research focuses on the Maximum Entropy approach since it the approach proposed in the AsReview software. This approach selects samples that maximise the entropy of the predicted probability distribution over the possible labels. The entropy of a probability distribution is calculated using the formula:

$$H(p) = -\sum_{i=1}^{n} p_i \log p_i \tag{3}$$

where $p = (p_1, p_2, \ldots, p_n)$ is a probability distribution over $n$ possible labels.

The Maximum Entropy approach selects the sample $x$ that maximises the entropy of the predicted probability distribution $p(y \mid x)$ over the possible labels $y$ :

$$x_{\text{maxent}} = \arg\max_{x} H(p(y \mid x)) \tag{4}$$

where in this case, $y$ is the possibility of a piece of abstract being relevant/irrelevant.

#### 3.3.2. Certainty query strategy

Certainty QS, on the other hand, is where the algorithm selects the instances/abstracts that the model is most certain about for annotation by an expert. In certainty sampling, the algorithm selects the examples with the highest predicted probability for the target class. For instance, in SR citation automation where the task is a binary classification, the algorithm selects the cases with the highest predicted probability for the positive class. The formula for certainty sampling can be expressed as follows:

$$x_i^* = argmax_{x_i \in U} P(y_i \mid x_i; \theta)$$

where: $x_i^*$ is the instance with the highest predicted probability, $x_i$ is the instance to be labelled, $U$ is the pool of unlabelled instances, $P(y_i \mid x_i; \theta)$ is the predicted probability of the target class given the instance $x_i$ and the model parameters $\theta$. The algorithm selects the instance $x_i^*$ with the highest predicted probability, which an expert then labels.

### 3.4. Classifiers

To address the second aspect of **RQ4**, the four most used traditional ML algorithms in the AsREview (SVM, RF, LR and NB) [11] are explained in the subsections below.

#### 3.4.1. SVM

SVM is a supervised learning algorithm used for both classification and regression analysis. In classification, SVM separates data into two or more classes (e.g. relevant or irrelevant) based on their features. SVM works by finding the best possible boundary (hyperplane) that can separate different data classes [31]. Data points contributing to discovering the ideal hyperplane are termed support vectors, as the margin or distance between these support vectors and the hyperplane must be maximised. SVM employs a collection of mathematical functions known as kernels, which enable it to handle high-dimensional data efficiently. Such kernels encompass linear, sigmoid, Gaussian, polynomial, nonlinear, and radial basis functions. Nevertheless, linear SVM remains the most prevalent algorithm applied in automating SR citations [11].

#### 3.4.2. LR

LR is a popular algorithm used for binary classification, where the goal is to predict the probability of an input belonging to a certain class [53]. It works by modelling the relationship between the input features and the binary output using a logistic function or a sigmoid function to model the probability of the response variable [54]. The logistic function is defined as follows:

$$p(y = 1 \mid x) = \frac{1}{1 + e^{-z}} \tag{5}$$

where z is a linear combination of the input features and their associated weights

To apply LR, for a given dataset with $n$ observations, as in this case the total number of abstracts to be screened, in Eq. (6), $y_i$ is the binary response variable (0: irrelevant and 1: relevant) and $\mathbf{x}_i$ is the $p$-dimensional vector of the independent variables/abstracts for the $i$th observation. The LR model can be written as follows:

$$P(y_i = 1 \mid \mathbf{x}_i) = \sigma(\mathbf{w}^T \mathbf{x}_i + b) \tag{6}$$

#### 3.4.3. RF

Random forest is another ML algorithm that ensembles multiple decision trees [55] algorithm to improve the accuracy and reduce the overfitting of the model. The algorithm works by building a collection of decision trees, where each tree is trained on a randomly selected subset of the training data and a random subset of the features [56]. The final prediction is made by aggregating the predictions of all the individual trees or by averaging or applying majority voting to the outputs from each individual tree to make a prediction. In theory, the accuracy of RF has been shown to exceed that of the individual decision tree algorithm [57].

#### 3.4.4. NB

NB is a probabilistic algorithm used for classification problems. It is based on Bayes' theorem [32], which describes the probability of a hypothesis given some observed evidence. Naive Bayes is called "naive" because it makes a strong assumption that the features are conditionally independent given the class label [32]. This assumption simplifies the calculations and makes the algorithm computationally efficient. Another assumption made in NB is that each input variable has an equal effect on the output and also that these features do not depend on each other, which in reality is not always so since there is an inter-dependency between these features [32].

### 3.5. Class imbalance techniques

One significant challenge in automating the screening of systematic review abstracts is class imbalance [58]. This arises due to the small number of studies that are actually pertinent to the research topic amidst a large pool of irrelevant studies found during the initial search stage. As a result, this imbalance between relevant and irrelevant papers can detrimentally impact the performance of a classifier trained

**Table 3**
Summary of performance metrics used in this study.

| Metric | Calculation |
|---|---|
| Recall | TP/(TP + FN) |
| Precision | TP/(TP + FP) |
| WSS@95 | ((TN+FN)/(N)) − (0.05) , where N = TP + TN + FP + FN |
| AUC Precision–recall | N/A |
| Run-time | N/A |

with such datasets, leading to a bias towards the abundant, irrelevant class and thus undermining the performance of the less-represented class. In SR automation, some suggestion to compact class imbalance is though the undersampling or oversampling technique [27].

Undersampling involves randomly eliminating instances from the majority class to create a balanced dataset. This approach proves beneficial when the majority class possesses numerous instances, and partial removal would not significantly affect the model's learning capacity. Conversely, oversampling increases the number of instances in the minority class through augmentation, suitable when there are insufficient instances of the minority class to train the model adequately. As observed from Table 1, AsReview implements both undersampling and "Double", which is a variant of undersampling + oversampling also referred to as *dynamic sampling* [24]. Therefore, to address **RQ5**, we explore the impact of the Double and the undersampling techniques in this study.

### 3.6. Performance metrics

The performance of the proposed methodology was evaluated using common metrics in citation screening automation, including precision, recall, WSS@R (Work Saved over Sampling@Recall) [13], and the Area Under the Curve (AUC) of the precision–recall curve. WSS@R is an essential metric for evaluating the performance of SR models. It measures how much a classifier can reduce the human burden at a specified recall level [13]. Specifically, WSS@R estimates the reduction in the number of irrelevant articles a researcher will not have to screen manually because the model identified them.

In SRs, a WSS at recall of 95% is considered acceptable [13] even though there may be some "relevant" studies that may not be included (5%). A reason for setting a recall of 0.95 according to Yu et al. [59], is that no algorithm can guarantee 100% recall without examining all potential papers. Therefore, this study reports WSS@95. However, there have been some citation screening studies that have reported WSS@100 [10], which is not necessarily contradicted by this study. To aid in the calculation, these evaluation metrics rely on the concepts of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). A summary of the evaluation metrics is summarised in Table 3. Here, the average of all the metrics after a total of 100 iterations or queries are reported over different runs of the simulation study.

### 3.7. Experimental setup

#### 3.7.1. Active learning methodology

This research encompasses a series of experiments divided into two primary categories: "Double Balancing" and "Undersampling Balancing". In Experiment 1 (Double Balancing), we explored the combinations of the three feature extraction techniques, namely TF-IDF, Doc2Vec, and S-BERT. For each technique, we examined the impact of implementing either Certainty (C) or Uncertainty (UC) as the query strategy. Moreover, in each QS, we evaluated the performance of the four classifiers: SVM, RF, NB, and LR. We repeated these experiments for the six different training size splits. Similarly, we performed the same method in Experiment 2 (Undersampling). In all, there were 48 possible combinations available in each experiment. To provide a comprehensive overview of these experiments and their results, we have summarised them in Table 4.

For example, in Experiment 1 with the 2:18 data split, we used TF-IDF against C or UC with SVM. Thus, the combination would be Double + 2:18 + TF-IDF + C + SVM or Double + 2:18 + TF-IDF + UC + SVM. However, the combination of Doc2Vec + NB and S-Bert + NB could not be experimented with because we realised the Doc2Vec and S-Bert produces a negative feature matrix whilst the NB require a positive feature matrix [24]. To set up the AL cycle described in Section 2, we divided each dataset into a train and test (0.2) split after the application of the feature extractors. Out of the training set, we partitioned to get the initial training sample sizes (20, 60, 100) sets with the number of "relevant" and "irrelevant" with respect to **RQ1** and **RQ2**. The remaining training set after the partitioning was used as the pool of unlabelled abstracts from which the ML algorithm/learner can query the user and add up to the initial training labelled sample. To start our AL loop, we used the three distinct initial labelled training samples to teach the classifier per 100 iteration whilst keeping track of the performance metrics in each iteration. We reported on the average of the performance metrics (recall, precision and WSS@95) obtained from the total iterations.

**Algorithm 1** AL Cycle after feature extraction

**Require:** Test set $(X_{test}, y_{test})$, Pool $(X_{pool}, y_{pool})$, Initial $(X_{initial}, y_{initial})$, $estimator$, $balancing$, $query\_strategy$ , $n_{queries}$ ▷ estimator, balancing, and query_strategy are the different variables of the **RQs** in Section 1

**Ensure:** Create metrics list: precision values, recall values, WSS@95, TP, TN, FP, FN, TPRs, FPRs, AUCs, real predictions, predicted probabilities

**function** AL_LOOP($query\_strategy$, $X_{test}$, $y_{test}$, $X_{pool}$, $y_{pool}$, $X_{initial}$, $y_{initial}$, $E$, $n_{queries}$, $n_{instances}$): ▷ where the $n_{instances}$ denotes the binary class (relevant or irrelevant)
  Perform balancing on initial data: $(X_{initial}, y_{initial}) \leftarrow$ BALANCING$(X_{initial}, y_{initial})$
  Initialise Learner $L$ with $(X_{initial}, y_{initial}, estimator, query\_strategy)$
  Compute initial metrics: $metrics \leftarrow$ SCORE$(L, X_{test}, y_{test})$
  **for** $index$ to $n_{queries}$ **do**
    $(query_{idx}, query_{instance}, prediction_{score}) \leftarrow$ QUERY$(L, X_{pool}, E, n_{instances})$
    Teach $L$ with $(X_{pool}[query_{idx}], y_{pool}[query_{idx}])$
    Remove $(X_{pool}[query_{idx}], y_{pool}[query_{idx}])$ from pool
    append $metrics_{after\_query} \leftarrow$ SCORE$(L, X_{test}, y_{test})$
    **return** $metrics_{after\_query}$
  **end for**
**end function** ▷ the mean of the appended $metrics_{after\_query}$ is reported

#### 3.7.2. Statistical methodology

To statistically analyse the impact of the different research questions on performance metrics, we conducted a series of statistical tests and identified the best models. Among the five metrics reported in Section 3.6, we selected WSS@95 for further analysis. WSS@95 was chosen because it quantifies the extent to which an ML algorithm can reduce the burden on human reviewers, and it has been recognised as an important metric in SLR citation screening [13]. Our main research question for this section was: *"What features or variables in each of the five research questions should be considered to achieve the best WSS@95?"*. By addressing this question, we aim to identify the most appropriate features that contribute to optimising the WSS@95 metric. For each of the five datasets (AM_2022, AM_2021, ACE_2006, KK_2020, VS_2017), the outcome variable was the WSS@95, whilst the independent variables were "Balancing", "Feature extractor", "Query strategy", "Classifiers" and "Training size". That is, these independent variables were used to analyse their effect on the WSS@95 score.

**Table 4**
A summary of the total number of the 48 possible AL combinations that were available for experimentation. However, NB does not work with Doc2Vec and S-Bert due to an incompatibility issue; both Doc2Vec and S-Bert produce a negative feature matrix, while NB requires a positive feature matrix.

| Balancing | Training_size | Feature extractor | Query strategy | Classifier |
|---|---|---|---|---|
| Double Undersampling | 2:18 | TF-IDF<br>Doc2vec<br>S-Bert | C<br>UC | NB<br>SVM<br>RF |
| Double Undersampling | 5:15 | TF-IDF<br>Doc2vec<br>S-Bert | C<br>UC | NB<br>SVM<br>RF |
| Double Undersampling | 6:54 | TF-IDF<br>Doc2vec<br>S-Bert | C<br>UC | NB<br>SVM<br>RF |
| Double Undersampling | 15:45 | TF-IDF<br>Doc2vec<br>S-Bert | C<br>UC | NB<br>SVM<br>RF |
| Double Undersampling | 10:90 | TF-IDF<br>Doc2vec<br>S-Bert | C<br>UC | NB<br>SVM<br>RF |
| Double Undersampling | 25:75 | TF-IDF<br>Doc2vec<br>S-Bert | C<br>UC | NB<br>SVM<br>RF |

Below, we provide descriptions of each independent variable and explain how they were measured or manipulated in the context of our study added to those described in Sections 3.2–3.4. For the training sample size, different subsets from the dataset were selected to match these sizes, providing a basis for comparing model performance across various training set sizes. Similarly, for the balancing variables, each method (Double and Undersampling) was applied to the sampled training size to adjust the relevant to irrelevant instances in each sample size ratio. For the feature extractors, the text data from abstracts were transformed into feature vectors for each feature extractor. Similarly, for the query strategy, instances were selected for labelling based on the chosen strategy, influencing the AL process. Lastly, each classifier was trained on the feature vectors generated from the training data.

The process involved in performing the statistical analysis is described as follows: an initial inspection of each dataset was carried out to determine the behaviour and frequency of each variable. Since all the datasets contain the same number of combinations and variables, we show only one of the summaries, the ACE_2006 dataset in Table 5. Continuous variables are displayed as means (±SD), while categorical variables are presented as frequencies (percentages). The distribution of continuous data was assessed using t-tests, while Fisher's test [60] was employed for categorical variables. To establish the relationship between the outcome variable "WSS95" and the independent variables "Balancing", "Query Strategy", "Classifiers", and "Training Size", two different statistical models were tested: Generalised Linear Models (GLM) and Generalised Additive Models (GAM). The GLM was selected as it provided a simpler explanation for the behaviour of the variables in the model, which was verified using the Akaike Information Criterion (AIC) [61]. The mathematical formula for the fitted model is as follows:

$$\text{Best\_WSS@95} = \beta_0 + \beta_{\text{bal}} \cdot X_{\text{bal}} + \beta_{\text{f.e}} \cdot X_{\text{f.e}} + \beta_{\text{q.s}} \cdot X_{\text{q.s}} + \beta_{\text{cla}} \cdot X_{\text{cla}} + \beta_{\text{t.s}} \cdot X_{\text{t.s}} \quad (7)$$

where $\beta_0$ represents the coefficient of the intercept, $\beta_{\text{bal}}$, $\beta_{\text{f.e}}$, $\beta_{\text{q.s}}$, $\beta_{\text{t.s}}$ are the estimated coefficients of the independent variables. Table 7 presents the corresponding coefficients, 95% confidence intervals (CI), and p-values ($p \leq 0.05$) that indicate the significance of each predictor in the model. Providing a brief description of the interpretation of the regression coefficients and their implication, $\beta_0$, represents the baseline value of the dependent variable (Best_WSS@95) when all independent variables are set to the AsReview reference categories. On the other hand, a positive coefficient of $\beta_{\text{bal}}$, $\beta_{\text{f.e}}$, $\beta_{\text{q.s}}$, $\beta_{\text{t.s}}$, for each dataset in Table 7, indicates that using that independent variable (eg. in terms of balancing, classifiers etc.) increases the Best_WSS@95 dependent metric by that units compared to the reference category of comparison, thus proposes to be a better alternative compared to the reference, especially those with a $p \leq 0.05$ and vice versa for negative coefficients. A more detailed description of how the Table 7 works for the datasets in given in Section 4. The statistical analyses were performed using R version 4.2.2. The detailed methodology of the various statistical methods is provided in Additional file 1 on the OSF.

**Table 5**
Statistical summary of variables in the training ACE_2006 dataset.

| Variable | N = 240[a] |
|---|---|
| **Balancing** | |
| Double | 120 (50%) |
| Undersamplng | 120 (50%) |
| **Feature_Extractor** | |
| TF-IDF | 96 (40%) |
| Doc2Vec | 72 (30%) |
| S-Bert | 72 (30%) |
| **Query_Strategy** | |
| Certainty | 120 (50%) |
| Uncertainty | 120 (50%) |
| **Classifiers** | |
| NB | 24 (10%) |
| SVM | 72 (30%) |
| RF | 72 (30%) |
| LR | 72 (30%) |
| **Training_size** | |
| 2:18 | 40 (17%) |
| 5:15 | 40 (17%) |
| 6:54 | 40 (17%) |
| 15:45 | 40 (17%) |
| 10:90 | 40 (17%) |
| 25:75 | 40 (17%) |
| **Best_Precision** | 0.97 (0.97, 0.98) |
| **Best Recall** | 0.96 (0.90, 0.98) |
| **AUC Precision–recall mean** | 0.06 (0.03, 0.09) |
| **Best_WSS95** | 0.92 (0.85, 0.94) |

[a] Mean (SD) or Frequency (%).

## 4. Results and discussion

Table 6 summarises the results from the combinations of the experiments that achieved the best precision, recall, AUC, running time, and WSS@95. At a glance, it is difficult to conclusively address the various research question posed, as the various variables of consideration are widely spread with respect to the performance metrics across all

datasets. Nonetheless, certain individual observations can be made and deduced.

Discussing the results of **RQ1**, which aims to evaluate the impact of various initial training sample sizes (20 (2:18, 5:15), 60 (6:54, 15:45), 100(10:90, 25:75)) on the performance of AL models across the five datasets. The results from Table 6 reveal diverse outcomes for different sample sizes across each dataset, implying a possible impact of the sample size on the performance of AL models. The 20-sample size appears 11 out of 25 times (most frequently) across the five performance metrics, followed by the 100-sample size (9/25 times) and the least frequent is the 60-sample size (5/25 times). A clear observation from the table is that the smallest training sample size, 20, provides the best result in at least one of the five performance metrics in each dataset. For instance, this sample size achieves the best running time metrics in the AM_2022, ACE_2006, KK_2020, and VS_2017 datasets, which is expected due to its smaller initial training size. Moreover, the 20-sample size offers the best WSS@95 metric across three of the datasets (KK_2020, ACE_2006, AM_2022). This may suggest that utilising a smaller initial training sample to train classifiers in an AL tool for citation automation, like the AsReview could potentially reduce human effort more quickly compared to the other training sample sizes. However, this observation with respect to the dataset used primarily applied to datasets within the 2000–5000 range (AM_2022, KK_2020, ACE_2006) and did not perform the same for the dataset that was not within the range like the AM_2021 (800) and VS_2017 (6000). Therefore, addressing **RQ1** practically, a user employing AsReview, or in a broader context, a traditional AL approach, may consider using a smaller number of initial training sizes/prior knowledge to train the model if the dataset falls within this particular range.

However, it is crucial to consider the costs associated with obtaining more training data. Gathering additional training data often requires significant financial and resource investments. For instance, acquiring larger datasets may involve costs related to data collection, storage, and preprocessing. This is particularly relevant for private datasets where data access might require permissions, licenses, or purchasing agreements. Moreover, increasing the initial training sample size necessitates more computational resources for processing and analysis, which can be costly in terms of time and money. For instance, in the case of the AM_2022 dataset, which initially included 13,002 papers, substantial human and computational resources were required to filter out irrelevant studies. Reducing the dataset to a manageable size necessitated both time and computational power, highlighting the hidden costs in the data preparation stages. These costs can impact the feasibility of using larger training samples in real-world applications. Understanding these trade-offs between the volume of training data and the resources required to obtain it is essential. Researchers and practitioners must balance the benefits of larger training samples with the available resources and budget constraints. This broader context is essential for guiding future research and application, helping stakeholders make informed decisions about data acquisition and model training strategies.

Moving on to the 100-sample training size, it also yielded the best results for at least one metric in each dataset except for the ACE_2006 dataset. This size was best in terms of the precision metric score for three out of the five datasets (AM_2022, VS_2017, KK_2020). This is of another great significance because, according to Cohen at al. [62], SR automation tools must aim to achieve a $recall \geq 95\%$ to include all potentially relevant literature [13]. However, as it is well-known in classification problems, a rise in recall results in a fall in precision, and vice versa. Nonetheless, achieving high precision is similarly important because it assures that the articles flagged as relevant are indeed relevant to the SR topic. Generalising the inference of **RQ1**, a deduction to be drawn may be that the initial training size that a user may choose or begin with to train an AL classifier for citation screening (such as AsReview) will depend entirely on the overall dataset size. However, in cases of larger datasets (approximately 6000 or higher) or smaller datasets (below 2000), it might be beneficial for users to choose a larger initial training size. Nonetheless, this is an interesting area for future investigation.

In addressing **RQ2**, which examines how the proportion of the number of relevant to irrelevant articles in an initial training sample (2:18, 6:54, 10:90 all at a 0.1 ratio and 5:15, 15:45, 25:75 all at a 0.3 ratio) impacts the performance of the AL model across the five datasets, some notable observations is seen. It is evident from Table 6 that the imbalanced ratio of 0.1 (2:18, 6:54, 10:90) in the initial training set resulted in the best performance metrics across the five datasets 15 out of 25 times. A vivid observation to be made is that an imbalance ratio of 0.1 (2:18 and 10:90) works well in terms of the best running time across 4/5 of the datasets, AM_2021 (10:90), AM_2022 (2:18), ACE_2006(2:18) and the largest dataset, VS_2017(2:18). One general inference drawn from this is that for every 100 queries presented to the user, the AL tool will likely operate faster if the initial training set has a lower imbalanced ratio compared to a higher ratio. Furthermore, another generalisation that can be made from the table is that the user selecting a smaller number of relevant: irrelevant ratio to train a classifier in Asreview may work best for datasets approximately 6000 or a lower dataset below 2000 if the initial training dataset has a lower imbalanced ratio and vice versa for smaller or extremely larger datasets.

Addressing **RQ3**, where we examined the impact of different feature extraction techniques — traditional (TF-IDF), word embedding (Doc2Vec), and transformer-based (S-Bert) — when used as input for the four different classifiers (SVM, RF, NB, and LR), one striking observation from the results table is that TF-IDF consistently delivered the best results for at least one of the five metrics across all datasets. Generally, across all datasets, the use of TF-IDF yields the highest mean AUC for precision recall. Additionally, except for the AM_2022 dataset, TF-IDF provided the best results for recall and WSS@95, both of which are crucial metrics in SR automation. Even though Doc2Vec stood out as the best metric for running time across all datasets, one general inference can be drawn: the various traditional classifiers in the AsReview tool or, in a broader AL context, performed better with TF-IDF in abstract screening compared to using a word-embedding or transformer-based method at the sentence level. In conclusion, for **RQ3**, the traditional feature extraction (TF-IDF) outperforms the Doc2Vec and S-Bert and works well with the two AL strategies explored in this paper (Certainty and Uncertainty). This confirms why the original authors of AsReview set TF-IDF as the default feature extraction technique.

Discussing **RQ4**, which asks, *"What is the performance of the AL models across four traditional classification techniques?"*, we approach this question in two parts. (1) We examine the performance of the two active query strategies (QS), Certainty (C) and Uncertainty (UC), and (2) We explain the impact of the two QS on the effectiveness of the four classifiers explored. Firstly, it can be observed from Table 6 that the effects of the two QS strategies varied across the different datasets. For instance, across all 25 metrics for each of the five datasets, C QS influenced 12/25 while UC influenced 13/25. It is, however, clear from the table that all of these results were most effective with the TF-IDF vectorisation method. Except for the dataset with the smallest total sample, AM_2021, which is <1000, one observation is that the use of UC strategy resulted in a high recall, especially with TF-IDF and S-Bert. Similarly, aside from the KK_2020 dataset, the UC strategy also yielded a good WSS@95 score across all datasets. On the other hand, C QS resulted in a higher recall for the AM_2022 dataset, but this was not consistent for the other metrics. Therefore, an inference to be drawn is that the performance of the two QS strategies can vary depending on the specific dataset being used, and the choice of strategy may depend on the priority metric in a wider contextual setting. Nonetheless, it can be emphasised that these two QS strategies integrate well with TF-IDF.

Similarly to the QS, the potential of the various classifiers explored was broadly distributed for each dataset. As illustrated in Table 6, using

**Table 6**
Summary of the best performing AL combinations for the performance metrics in the experiments done.

| Datasets | Balancing type | Sample_size | Metrics | Combination | Results |
|---|---|---|---|---|---|
| AM_2022 | Undersampling | 2:18 | Best running time | Doc2Vec+ C + LR | 1.1312 |
| | Double | 25:75 | Best Precision | S-BERT+ UC +LR | 0.9901 |
| | Undersampling | 10:90 | Best Recall | S-BERT+ UC +LR | 0.9905 |
| | Double | 6:54 | AUC Precision–recall mean | TF-IDF + C + SVM | 0.6559 |
| | Double | 2:18 | Best WSS@95 | S-BERT+ UC+ LR | 0.9880 |
| AM_2021 | Double | 10:90 | Best running time | Doc2Vec+ UC + LR | 1.0448 |
| | Double | 5:15 | Best Precision | TF-IDF + C + SVM | 0.9817 |
| | Double/Undersampling | 5:15 | Best Recall | TF-IDF + C+ SVM | 0.9814 |
| | Double | 6:54 | AUC Precision–recall mean | TF-IDF + UC + SVM | 0.4758 |
| | Double | 25:75 | Best WSS@95 | TF-IDF + UC+ RF | 0.9612 |
| VS_2017 | Undersampling | 2:18 | Best running time | Doc2Vec+ C + LR | 1.1185 |
| | Double | 25:75 | Best Precision | TF-IDF+ C+LR | 0.9947 |
| | Undersampling | 6:54 | Best Recall | TF-IDF + UC+ RF | 0.9939 |
| | Double | 25:75 | AUC Precision–recall mean | TF-IDF+ C+SVM | 0.7269 |
| | Double | 15:45 | Best WSS@95 | TF-IDF+ UC+RF | 0.9863 |
| ACE_2006 | Undersampling | 2:18 | Best running time | Doc2Vec+ UC + LR | 0.9385 |
| | Undersampling | 15:45 | Best Precision | Doc2Vec+ C + RF | 0.9849 |
| | Double | 2:18 | Best Recall | TF-IDF + UC+ SVM | 0.9846 |
| | Undersampling | 2:18 | AUC Precision–recall mean | TF-IDF+ UC+RF | 0.6680 |
| | Undersampling/Double | 2:18 | Best WSS@95 | TF-IDF+ UC+RF | 0.9500 |
| KK_2020 | Undersampling | 5:15 | Best running time | Doc2Vec+ C+ LR | 1.5702 |
| | Double | 10:90 | Best Precision | TF-IDF + C+ SVM | 0.9595 |
| | Undersampling | 25:75 | Best Recall | TF-IDF + UC+ SVM | 0.9582 |
| | Double | 10:90 | AUC Precision–recall mean | TF-IDF + C+ SVM | 0.5863 |
| | Double | 2:18 | Best WSS@95 | TF-IDF + C+ RF | 0.9500 |

SVM and LR yielded 9/25 of the best-ranking metrics. In contrast, RF accounted for 7/25 of the best metric results, whilst NB did not achieve any top scores, although Additional file II provided some notable results with the use of NB. A detailed observation from the table concerning **RQ3** and **RQ4** is that using the UC QS pairs well with TF-IDF + RF. For instance, in the AM_2021 and the VS_2017 datasets, UC + TF-IDF+ RF provided the best WSS@95 score, which is highly significant for datasets that are extremely small or large. This combination also achieved the best WSS@95 score for the ACE_2006 dataset. Another observation is that UC + TF-IDF+ SVM yielded a strong recall score for the AM_2021 and KK_2020 datasets. Furthermore, the table suggests that UC + LR integrates well with word embedding or transformer-based feature extraction methods. As for C QS, it appears to work well with SVM (TF-IDF + C + SVM) showing up in 6/12 cases. For example, except for the AM_2021 dataset, this combination offered the best results regarding the AUC Precision–recall mean. Though a generalised inference cannot be conclusively drawn for **RQ4**, an important observation is that using TF-IDF pairs well with both Certainty and Uncertainty strategies. Finally, for **RQ5**, similar to other research questions, a generalised inference cannot be explicitly drawn as both Double and Undersampling are seen to be equally spread across the metric in each dataset.

### *4.1. Statistical analysis of the results*

Each dataset exhibited distinct performance results, as observed in Table 6. In this section, we explain the statistical results of each variable on the WSS@95 metric from our experiments stated in Section 3.7.2. To ensure a fair comparison, we establish the default settings in AsReview as the reference. Specifically, the following reference categories in each of the independent variables associated with the five research questions were considered: a training size of 2:18 for **RQ1 and RQ2**, TF-IDF for feature extraction in **RQ3**, NB as the classifier in **RQ4**, Double for query strategy in **RQ 4**, and Double for balancing in **RQ5**.

Discussing the ACE_2006 dataset from Table 7, Undersampling was found to be associated with an increase in WSS@95 ($\beta$ = 0.039, p < 0.001). The $p$-value < 0.05 and the positive beta coefficient indicate that using Undersampling as a balancing technique will likely result in a better WSS@95 than that of the reference balancing, Double. Additionally, a noticeable observation is that the CI non-inclusion of zero, which shows a statistically significant association between balancing and WSS@95. Regarding feature extraction, Doc2Vec was also associated with a decrease in WSS@95 ($\beta$ = –0.028, p = 0.006) with respect to the reference, TF-IDF. An inference is that the negative beta coefficient for Doc2Vec and a $p$-value < 0.05 indicate that the reference category, TF-IDF, is statistically likely to be better than Doc2Vec at achieving a WSS@95. S-Bert, on the other hand, did not show any significance (p = 0.957). Comparing the two query strategies categories: certainty(reference) against uncertainty, it is observed that the latter has an increase of 0.082 in the value of the beta coefficient, thus, indicating that the use of uncertainty is probably better for obtaining a WSS@95. Similarly, all three classifiers (SVM, RF, and LR) were positively associated with WSS@95: SVM ($\beta$ = 0.038, p = 0.017), RF ($\beta$ = 0.037, p = 0.019), and LR ($\beta$ = 0.054, p = 0.001) respectively. As such, all three categories showed an increase in the beta coefficient, indicating their association compared to the reference category NB, with the largest increase for the LR classifier. On the other hand, all the training size categories were negatively associated with WSS@95: 5:15 ($\beta$ = –0.051, p = 0.001), 6:54 ($\beta$ = –0.048, p = 0.001), 15:45 ($\beta$ = –0.067, p = 0.001), 10:90 ($\beta$ = –0.039, p = 0.004), 25:75 ($\beta$ = –0.062, p = 0.001). The decreasing beta values compared to the reference training size category, 2:18, indicates an association with WSS@95, based on the $p$-value < 0.05, although the reference category has the best statistical significance. Thus, using a smaller training size and a minimal ratio number of "relevant" to "irrelevant" samples gives a better WSS@95. This same inference and deductions in the ACE_2006 can be made for the KK_2020 and VS_2017 datasets as seen in Table 7 but with the S-Bert showing a negative association in WSS@95. In the VS_2017dataset, S-BERT ($\beta$ = –0.040, p < 0.001) and in the KK_2020 dataset, S-BERT ($\beta$ = –0.030, p < 0.039).

In the least sampled dataset, AM_2021, except for the training size, all the other independent variables showed the same association as the ACE_2006 dataset, except S-Bert, which showed a negative association with WSS@95 ($\beta$ = –0.028, p = 0.001). Additionally, with respect to the training size, the 5:15 and 6:54 were not statistically significant as their $p$-value < 0.05, i.e. the null hypothesis of no association between

**Table 7**
Summary of statistical results.

| ACE_2006 | | | | AM_2021 | | | AM_2022 | | | VS_2017 | | | KK_2020 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Variable | Beta | 95% $CI_l$ | p-value | Beta | 95% $CI_l$ | p-value | Beta | 95% $CI_l$ | p-value | Beta | 95% $CI_l$ | p-value | Beta | 95% $CI_l$ | p-value |
| **(Intercept)** | 0.839 | 0.807, 0.871 | <0.001 | 0.880 | 0.853, 0.907 | <0.001 | 0.680 | 0.644, 0.716 | <0.001 | 0.857 | 0.829, 0.885 | <0.001 | 0.760 | 0.714, 0.806 | 0.000 |
| **Balancing** | | | | | | | | | | | | | | | |
| Double | – | – | | – | – | | – | – | | – | – | | – | – | |
| Undersampling | 0.039 | 0.023, 0.054 | <0.001 | 0.024 | 0.011, 0.037 | <0.001 | 0.001 | −0.016, 0.019 | 0.870 | 0.019 | 0.006, 0.033 | 0.006 | 0.031 | 0.009, 0.053 | 0.006 |
| **Feature_Extractor** | | | | | | | | | | | | | | | |
| TF-IDF | – | – | | – | – | | – | – | | – | – | | – | – | |
| Doc2Vec | −0.028 | −0.048, −0.008 | 0.006 | −0.107 | −0.124, −0.090 | <0.001 | −0.060 | −0.083, −0.038 | <0.001 | −0.070 | −0.088, −0.052 | <0.001 | −0.083 | −0.112, −0.055 | 0.000 |
| S-Bert | 0.001 | −0.020, 0.021 | 0.957 | −0.028 | −0.045, −0.011 | 0.001 | −0.009 | −0.031, 0.013 | 0.420 | −0.040 | −0.058, −0.023 | <0.001 | −0.030 | −0.058, −0.002 | 0.039 |
| **Query_Strategy** | | | | | | | | | | | | | | | |
| Certainty | – | – | | – | – | | – | – | | – | – | | – | – | |
| Uncertainty | 0.082 | 0.066, 0.097 | <0.001 | 0.057 | 0.044, 0.070 | <0.001 | 0.104 | 0.087, 0.122 | <0.001 | 0.065 | 0.051, 0.078 | <0.001 | 0.119 | 0.097, 0.141 | 0.000 |
| **Classifiers** | | | | | | | | | | | | | | | |
| NB | – | – | | – | – | | – | – | | – | – | | – | – | |
| SVM | 0.038 | 0.007, 0.068 | 0.017 | 0.035 | 0.009, 0.061 | 0.008 | 0.191 | 0.156, 0.225 | <0.001 | 0.058 | 0.031, 0.085 | <0.001 | 0.105 | 0.061, 0.149 | 0.000 |
| RF | 0.037 | 0.006, 0.067 | 0.019 | 0.065 | 0.039, 0.091 | <0.001 | 0.204 | 0.170, 0.239 | <0.001 | 0.092 | 0.065, 0.119 | <0.001 | 0.120 | 0.076, 0.164 | 0.000 |
| LR | 0.054 | 0.023, 0.085 | 0.001 | 0.032 | 0.007, 0.058 | 0.015 | 0.164 | 0.129, 0.198 | <0.001 | 0.064 | 0.037, 0.091 | <0.001 | 0.112 | 0.068, 0.156 | 0.000 |
| **Training_size** | | | | | | | | | | | | | | | |
| 2:18 | – | – | | – | – | | – | – | | – | – | | – | – | |
| 5:15 | −0.051 | −0.078, −0.024 | <0.001 | −0.007 | −0.030, 0.015 | 0.538 | −0.016 | −0.046, 0.014 | 0.302 | −0.026 | −0.049, −0.002 | 0.034 | −0.054 | −0.092, −0.016 | 0.006 |
| 6:54 | −0.048 | −0.075, −0.021 | 0.001 | −0.014 | −0.037, 0.008 | 0.219 | 0.011 | −0.019, 0.041 | 0.486 | −0.024 | −0.047, 0.000 | 0.050 | −0.046 | −0.084, −0.008 | 0.019 |
| 15:45 | −0.067 | −0.094, −0.040 | <0.001 | −0.053 | −0.076, −0.031 | <0.001 | 0.004 | −0.026, 0.034 | 0.785 | −0.035 | −0.059, −0.012 | 0.004 | −0.080 | −0.118, −0.042 | 0.000 |
| 10:90 | −0.039 | −0.066, −0.012 | 0.004 | −0.042 | −0.064, −0.019 | <0.001 | 0.027 | −0.004, 0.057 | 0.085 | −0.029 | −0.052, −0.005 | 0.018 | −0.042 | −0.080, −0.004 | 0.032 |
| 25:75 | −0.062 | −0.089, −0.035 | <0.001 | −0.039 | −0.061, −0.016 | 0.001 | 0.005 | −0.025, 0.035 | 0.737 | −0.039 | −0.063, −0.016 | 0.001 | −0.091 | −0.130, −0.053 | 0.000 |

sample size and WSS@95 cannot be rejected. The 15:45 ($\beta$ = –0.053, p < 0.001), 10:90 ($\beta$ = –0.042, p < 0.001) and 25:75 ($\beta$ = –0.039, p = 0.001) were negatively associated with WSS@95 of in AM_2021. Similar to the ACE_2006 dataset, the decreasing beta values compared to 2:18, the reference category, indicates an association with WSS@95, although 2:18 proves the statistical significance.

On the other hand, there was statistical significance with Undersampling (p = 0.870) in the AM_2022, the most arduous/difficult dataset in the study. Consistent with the other datasets, the remaining variables showed the same association on the WSS@95. For example, Doc2Vec was negatively associated with WSS@95 ($\beta$ = –0.060, 95% CI = −0.083 to −0.038, p < 0.001), and uncertainty strategy showed positive association in WSS@95 ($\beta$ = 0.104 < 0.001). All three classifiers, SVM, RF, and LR, were positively associated with WSS@95: SVM ($\beta$ = 0.191, p = 0.001), RF ($\beta$ = 0.204, p = 0.001), and LR ($\beta$ = 0.164, p = 0.001). None of the training sizes showed were statistically significant.

Generalising the results of all the statistical results, it can clearly be seen that the WSS@95 depend highly on the specific dataset and problem at hand. However, a general inference is that undersampling was consistently associated with a positive impact on the WSS@95, except the most challenging dataset, AM_2022, which was inconsistent with the reference category, Double. A practical application could be that Double might be beneficial for extremely challenging datasets and may explain why Double is set as the reference balancing in AsReview. TF-IDF, on the other hand, generally showed better or at least comparable performance to others. Therefore, it could be considered a robust first choice. The uncertainty strategy consistently performed better across all the datasets, suggesting that it might be the preferred method in many cases. While the specific best-performing classifier varied across datasets, SVM, RF, and LR often outperformed NB. Therefore, trying these three classifiers in initial models could be beneficial. Smaller training sizes often showed better performance. However, all these inference may largely depend on the specifics of your dataset and might not hold for all datasets.

## 5. Conclusion and future works

In conclusion, this comprehensive study provides significant insights into the optimal use of AL combinations in the context of systematic reviews. It sought to address five key research questions, investigating how the AL variables such as choice of the initial training size, feature extraction method, classifier, query strategy, and data balancing can impact the performance of AL tools for abstract screening, enabling us to identify trends, recommend best practices, and contribute to the broader understanding of the practical application of AL in SRs.

For **RQ1**, addressing the effects of different initial training sample sizes, our findings suggest that smaller training sample sizes, specifically around 20, were associated with improved performance metrics across various datasets, particularly those ranging from 2000 to 5000 abstracts. However, this trend was not universally applicable and, in practicality, should be carefully considered based on the dataset's specific characteristics. In response to **RQ2**, we observed using the smaller imbalanced ratio of, in this study, 0.1 in the initial training sample size (2:18, 6:54, 10:90) led to optimal performance across most datasets. This insight could influence a reviewer's selection of the number of "relevant" and "irrelevant" that are selected to train the model. Future work may be to look at the effect of having an equal number of relevant: irrelevant. In **RQ3**, we compared the effects of different feature extraction techniques. The results indicated that TF-IDF consistently outperformed the other techniques across all datasets, offering the best results for key metrics such as AUC Precision–recall mean, Recall, and WSS. These findings suggest that traditional feature extraction methods may still be preferable over more modern approaches when using AsReview or similar AL tools. Also, in **RQ4**, SVM, RF, and LR classifiers frequently outperformed NB. This indicates the necessity of choosing an appropriate classifier based on the specifics of the dataset, but SVM, RF, or LR would be reasonable starting points. It was also observed that the two query strategies (Certainty and Uncertainty) interacted effectively with TF-IDF. Lastly, in **RQ5**, we observed that there was no explicit generalisation possible regarding the impact of the sampling methods, undersampling, and double sampling. Their performance seemed to be fairly balanced across the metrics and datasets. The experiments and statistical analysis results indicate that the impact of AL variables in SR automation is highly dependent on the specific dataset, highlighting the importance of tailoring the AL process to the specifics of each problem or dataset.

In a nutshell reiterating the salient points, to assist users in implementing these findings in their systematic review processes without needing to fully understand the underlying technical complexities, we offer the following distilled guidelines: for the initial training sample size, starting with <20 abstracts, especially for datasets ranging from 2000 to 5000 abstracts to train the model can enhance performance metrics. For the imbalanced ratio in the training samples, choosing a few more irrelevant abstracts compared to relevant ones (e.g., 2 relevant and 18 irrelevant) can help improve performance. Also, the use of traditional feature extraction methods like TF-IDF may offer better results than modern techniques. Lastly, both undersampling and double-sampling methods, are generally balanced across metrics and datasets, as such users may best experiment with both undersampling and double sampling methods to see which works best for your specific dataset as both methods are viable options. Despite providing valuable insights, this study had some limitations. Its conclusions were drawn from a limited number of datasets and one type of AL tool, AsReview, v1.4 at the time of the study, though we do acknowledge that the software undergoes regular updates. Thus, it may not be universally applicable. In addition to the limitations of the work, exploring the

impact of a minimum training set of records(eg one relevant and one irrelevant (1:1)) could have aided in establishing a baseline from a theoretical point. However, our study aimed to evaluate the practical and applicable scenarios for AL models in SRs in terms of the initial training sizes. As such, future works may be explored the impact of minimum or extreme cases on such models. Also, this study focused on exploring traditional AL classifiers. Future research may investigate the impact across wider datasets and use more than one AL SR tool with additional parameters and conditions in the AL process across other domains. Another area of investigation may be exploring the impact of deep learning models with these AL combinations in SR citation screening. This could lead to more refined guidelines for applying AL in SRs.

## CRediT authorship contribution statement

**Regina Ofori-Boateng:** Writing – review & editing, Writing – original draft, Visualization, Methodology, Data curation. **Tamy Goretty Trujillo-Escobar:** Writing – review & editing, Visualization, Methodology. **Magaly Aceves-Martins:** Validation, Supervision. **Nirmalie Wiratunga:** Supervision. **Carlos Francisco Moreno-Garcia:** Supervision, Project administration, Conceptualization.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References


[1] Khan KS, Kunz R, Kleijnen J, Antes G. Five steps to conducting a systematic review. J R Soc Med 2003;96(3):118–21, [Online]. Available: https://doi.org/10.1258/jrsm.96.3.118.

[2] Clarke J. What is a systematic review? Evid-Based Nurs 2011;14(3):64, [Online]. Available: https://doi.org/10.1136/ebn.2011.0049.

[3] Stevens K. Systematic reviews: The heart of evidence-based practice. AACN Clin Issues 2001;12:529–38, [Online]. Available: https://doi.org/10.1097/00044067-200111000-00009.

[4] Kitchenham B, Brereton OP, Budgen D, Turner M, Bailey J, Linkman S. Systematic literature reviews in software engineering – A systematic literature review. Inf Softw Technol 2009;51(1):7–15, [Online]. Available: https://doi.org/10.1016/j.infsof.2008.09.009.

[5] Marshall IJ, Wallace BC. Toward systematic review automation: A practical guide to using machine learning tools in research synthesis. Syst Rev 2019;8(1):1–10, Avaiable: https://doi.org/10.1186/s13643-019-1074-9.

[6] Bannach-Brown A, Przybyła P, Thomas J, Rice ASC, Ananiadou S, Liao J, Macleod MR. Machine learning algorithms for systematic review: reducing workload in a preclinical review of animal studies and reducing human screening error. Syst Rev 2019;8(1):1–12, [Online]. Available: https://doi.org/10.1186/s13643-019-0942-7.

[7] Bastian H, Glasziou P, Chalmers I. Seventy-five trials and eleven systematic reviews a day: How will we ever keep up? PLoS Med 2010;7(9):e1000326, [Online]. Available: https://doi.org/10.1371/journal.pmed.1000326.

[8] Borah R, Brown AW, Capers PL, Kaiser KA. Analysis of the time and workers needed to conduct systematic reviews of medical interventions using data from the PROSPERO registry. BMJ Open 2017;7(2):1–7, [Online]. Available: https://doi.org/10.1136/bmjopen-2016-012545.

[9] 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 2015;66(11):2215–22, [Online]. Available: https://doi.org/10.1002/asi.23329.

[10] van de Schoot R, de Bruin J, Schram R, Zahedi P, de Boer J, Weijdema F, Kramer B, Huijts M, Hoogerwerf M, Ferdinands G, Harkema A, Willemsen J, Ma Y, Fang Q, Hindriks S, Tummers L, Oberski DL. An open source machine learning framework for efficient and transparent systematic reviews. Nat Mach Intell 2021;3(February):125–33, [Online]. Available: http://dx.doi.org/10.1038/s42256-020-00287-7.

[11] O'Mara-Eves A, Thomas J, McNaught J, Miwa M, Ananiadou S. Using text mining for study identification in systematic reviews: a systematic review of current approaches. Syst Rev 2015;4(1):1–22, [Online]. Available: https://doi.org/10.1186/2046-4053-4-5.

[12] Blaizot A, Veettil SK, Saidoung P, Moreno-Garcia CF, Wiratunga N, Aceves-Martins M, Lai NM, Chaiyakunapruk N. Using artificial intelligence methods for systematic review in health sciences: A systematic review. Res Synth Methods 2022;13(3):353–62, [Online]. Available: https://doi.org/10.1002/jrsm.1553.

[13] Cohen AM, Hersh WR, Peterson K, Yen P-Y. Reducing workload in systematic review preparation using automated citation classification. J Am Med Inform Assoc 2006;13(2):206–19, [Online]. Available: http://dx.doi.org/10.1197/jamia.m1929.

[14] Johnson EE, O'Keefe H, Sutton A, Marshall C. The systematic review toolbox: keeping up to date with tools to support evidence synthesis. Syst Rev 2022;11(1). [Online]. Available: http://dx.doi.org/10.1186/s13643-022-02122-z.

[15] Settles B. Active learning literature survey. 2010.

[16] Deploying an interactive machine learning system in an evidence-based practice center: Abstrackr. In: IHI'12 - proceedings of the 2nd ACM SIGHIT international health informatics symposium. 2012, p. 819–23, Available: https://doi.org/10.1145/2110363.2110464.

[17] Ouzzani M, Hammady H, Fedorowicz Z, Elmagarmid A. Rayyan-a web and mobile app for systematic reviews. Syst Rev 2016;5(1):1–10, Available: https://doi.org/10.1186/s13643-016-0384-4.

[18] Cheng SH, Augustin C, Bethel A, Gill D, Anzaroot S, Brun J, DeWilde B, Minnich RC, Garside R, Masuda YJ, Miller DC, Wilkie D, Wongbusarakum S, McKinnon MC. Using machine learning to advance synthesis and use of conservation and environmental evidence. Conserv Biol 2018;32(4):762–4, Blackwell Publishing Inc. [Online]. Available: https://doi.org/10.1111/cobi.13117.

[19] Chai KEK, Lines RLJ, Gucciardi DF, Ng L. Research screener: a machine learning tool to semi-automate abstract screening for systematic reviews. Syst Rev 2021;10(1):1–13, Systematic Reviews, [Online]. Available: https://doi.org/10.1186/s13643-021-01635-3.

[20] Howard BE, Phillips J, Tandon A, Maharana A, Elmore R, Mav D, Sedykh A, Thayer K, Merrick BA, Walker V, Rooney A, Shah RR. SWIFT-active screener: Accelerated document screening through active learning and integrated recall estimation. Environ Int 2020;138(2019):105623, [Online]. Available: https://doi.org/10.1016/j.envint.2020.105623.

[21] van Dijk SHB, Brusse-Keizer MGJ, Bucsán CC, van der Palen J, Doggen CJM, Lenferink A. Artificial intelligence in systematic reviews: promising when appropriately used. BMJ Open 2023;13(7):e072254, BMJ, [Online]. Available: https://doi.org/10.1136/bmjopen-2023-072254.

[22] van de Schoot R, Sijbrandij M, Depaoli S, Winter SD, Olff M, van Loey NE. Bayesian PTSD-trajectory analysis with informed priors based on a systematic literature search and expert elicitation. Multivariate Behav Res 2018;53(2):267–91, Informa UK Limited. [Online]. Available: https://doi.org/10.1080/00273171.2017.1412293.

[23] Hughes M. Hasta la vista, baby - will machine learning terminate human literature hasta la vista, baby - will machine learning terminate human literature reviews in entrepreneurship?. 2021.

[24] Ferdinands G, Schram R, de Bruin J, Bagheri A, Oberski DL, Tummers L, van de Schoot R. Active learning for screening prioritization in systematic reviews - a simulation study. Center for Open Science; 2020, [Online]. Available: https://doi.org/10.31219/osf.io/w6qbg.

[25] Ferdinands G, Schram R, de Bruin J, Bagheri A, Oberski DL, Tummers L, Teijema JJ, van de Schoot R. Performance of active learning models for screening prioritization in systematic reviews: a simulation study into the average time to discover relevant records. Syst Rev 2023;12(1). [Online]. Available: http://dx.doi.org/10.1186/s13643-023-02257-7.

[26] Yu Z, Kraft NA, Menzies T. Finding better active learners for faster literature reviews. Empir Softw Eng 2018;23(6):3161–86, [Online]. Available: https://doi.org/10.1007/s10664-017-9587-0.

[27] Almeida H, Meurs M-J, Kosseim L, Tsang A. Data sampling and supervised learning for HIV literature screening. IEEE Trans Nanosci 2016;15(4):354–61, IEEE. [Online]. Available: https://doi.org/10.1109/bibm.2015.7359733.

[28] van Dinter R, Tekinerdogan B, Catal C. Automation of systematic literature reviews: A systematic literature review. Inf Softw Technol 2021;136:106589, [Online]. Available: https://doi.org/10.1016/j.infsof.2021.106589.

[29] Le QV, Mikolov T. Distributed representations of sentences and documents. 2014, arXiv preprint. Available: arXiv:1405.4053.

[30] Reimers N, Gurevych I. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing. EMNLP-IJCNLP, Association for Computational Linguistics; 2019, [Online]. Available: https://doi.org/10.18653/v1/d19-1410.

[31] Cortes C, Vapnik V, Saitta L. Support-vector networks editor. Mach Learn 1995;20:273–97, Kluwer Academic Publishers.

[32] Lewis DD. Naive (Bayes) at forty: The independence assumption in information retrieval. In: ECML. 1998.

[33] Hans C. Elastic net regression modeling with the orthant normal prior. J Amer Statist Assoc 2011;106(496):1383–93, [Publisher: Taylor & Francis]..

[34] Breiman L. Random forests. Mach Learn 2001;45(1):5–32, [Online]. Available: https://doi.org/10.1023/A:1010933404324.

[35] Thrun SB. Exploration in active learning. In: Handbook of brain and cognitive science. 1995, p. 381–4, [Online]. Available: http://robots.stanford.edu/papers/thrun.arbib-handbook.ps.gz.

[36] Lewis DD, Gale WA. A sequential algorithm for training text classifiers. In: Proceedings of the 17th annual international ACM SIGIR conference on research and development in information retrieval. SIGIR, Vol. 1994, 1994, p. 3–12, [Online]. Available: https://doi.org/10.1007/978-1-4471-2099-5_1.

[37] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9(8):1735–80, [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735.

[38] Przybyła P, Brockmeier AJ, Kontonatsios G, Le Pogam MA, McNaught J, von Elm E, Nolan K, Ananiadou S. Prioritising references for systematic reviews with RobotAnalyst: A user study. Res Synth Methods 2018;9(3):470–88, [Online]. Available: https://doi.org/10.1002/jrsm.1311.

[39] EPPI-reviewer 3.5: software for research synthesis. EPPI-Centre, Social Science Research Unit, Institute of Education, University of London; 2007.

[40] This B. The Rct, Cochrane Crowd, N. H. S. Eed, and Cochrane Rct. Machine learning functionality in EPPI-Reviewer. [s.l.], 1–9.

[41] Bozada T, Borden J, Workman J, Del Cid M, Malinowski J, Luechtefeld T. Sysrev: A FAIR platform for data curation and systematic evidence review. Front Artif Intell 2021;4(August):1–18, Available: https://doi.org/10.3389/frai.2021.685298.

[42] Aceves-Martins M, López-Cruz L, García-Botello M, Gutierrez-Gómez YY, Moreno-García CF. Interventions to treat obesity in mexican children and adolescents: Systematic review and meta-analysis. Nutr Rev 2022;80(3):544–60, Oxford University Press. [Online]. Available:.

[43] Kwok KTT, Nieuwenhuijse DF, Phan MVT, Koopmans MPG. Virus metagenomics in farm animals: A systematic review. Viruses 2020;12(1):107, 022. [Online]. Available: https://doi.org/10.3390/v12010107.

[44] Ofori-Boateng R, Aceves-Martins M, Jayne C, Wiratunga N, Moreno-Garcia CF. Evaluation of attention-based LSTM and bi-LSTM networks for abstract text classification in systematic literature review automation. Procedia Comput Sci 2023;222:114–26, [Online]. Available: http://dx.doi.org/10.1016/j.procs.2023.08.149.

[45] Howard BE, Phillips J, Miller K, Tandon A, Mav D, Shah MR, Holmgren S, Pelch KE, Walker V, Rooney AA, Macleod M, Shah RR, Thayer K. SWIFT-review: a text-mining workbench for systematic review. Syst Rev 2016;5(1). [Online]. Available: http://dx.doi.org/10.1186/s13643-016-0263-z.

[46] Timsina P, Liu J, El-Gayar O. Advanced analytics for the automation of medical systematic reviews. Inf Syst Front 2015;18(2):237–52, [Online]. Available: https://doi.org/10.1007/s10796-015-9589-7.

[47] Olorisade BK, Brereton P, Andras P. The use of bibliography enriched features for automatic citation screening. J Biomed Inform 2019;94:103202, [Online]. Available: https://doi.org/10.1016/j.jbi.2019.103202.

[48] Singh AK, Mogalla S. Vectorization of text documents for identifying unifiable news articles. Int J Adv Comput Sci Appl 2019;10(7):305–10, [Online]. Available: https://doi.org/10.14569/ijacsa.2019.0100742.

[49] Dharma EM, Gaol FL, Leslie H, Warnars HS, Soewito B. The accuracy comparison among word2vec, glove, and fasttext towards convolution neural network (cnn) text classification. J Theor Appl Inf Technol 2022;100(2):31.

[50] Toshevska M, Stojanovska F, Kalajdjieski J. Comparative analysis of word embeddings for capturing word similarities. 2020, arXiv preprint arXiv:2005.03812.

[51] Mikolov Tomas, Chen Kai, Corrado Greg, Dean Jeffrey. Efficient estimation of word representations in vector space. In: 1st international conference on learning representations, ICLR 2013 - workshop track proceedings. 2013, p. 1–12.

[52] Devlin J, Chang M-W, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. 2019, ArXiv, abs/1810.04805.

[53] Kleinbaum David G. Modeling strategy guidelines. Logist Regres 1994;161–89, [Online]. Available: https://doi.org/10.1007/978-1-4757-4108-7_6.

[54] Sarker Iqbal H. Machine learning: Algorithms, real-world applications and research directions. SN Comput Sci 2021;2(3):1–21, [Online]. Available: https://doi.org/10.1007/s42979-021-00592-x.

[55] Quinlan JR. Induction of decision trees. Mach Learn 1986;1(1):81–106, [Online]. Available: https://doi.org/10.1007/bf00116251.

[56] Rokach L, Maimon O. Decision trees. Lecture notes in mathematics, vol. 1928, 2008, p. 67–86, [Online]. Available: https://doi.org/10.1007/978-3-540-75859-4_5.

[57] Sarker IH. A machine learning based robust prediction model for real-life mobile phone data. Internet Things (Netherlands) 2019;5:180–93, [Online]. Available: https://doi.org/10.1016/j.iot.2019.01.007.

[58] Moreno-García CF, Jayne C, Elyan E. Class-decomposition and augmentation for imbalanced data sentiment analysis. In: International joint conference on neural networks. IJCNN, IEEE; 2021, p. 1–7.

[59] Yu Z, Kraft NA, Menzies T. Finding better active learners for faster literature reviews. Empir Softw Eng 2018;23(6):3161–86, [Online]. Available: https://doi.org/10.1007/s10664-017-9587-0.

[60] Fisher RA. On the interpretation of $\chi^2$ from contingency tables, and the calculation of P. J R Stat Soc 1922;85(1):87, [Online]. Available: https://doi.org/10.2307/2340521.

[61] Claeskens G, Hjort NL. Akaike's information criterion. In: Model selection and model averaging, cambridge series in statistical and probabilistic mathematics. Cambridge: Cambridge University Press; 2008, p. 22–69, [Online]. Available: https://doi.org/10.1017/CBO9780511790485.003.

[62] Automated confidence ranked classification of randomized controlled trial articles: An aid to evidence-based medicine. J Am Med Inform Assoc 2015;22(3):707–17, [Author not provided]. [Online]. Available: https://doi.org/10.1093/jamia/ocu025.