Iterative Guided Machine Learning-Assisted Systematic Literature Reviews: A Diabetes Case Study

John Zimmerman (jzimmerman@deloitte.com)
Deloitte LLP https://orcid.org/0000-0001-7241-960X

Robin Soler
Centers for Disease Control and Prevention

Lavinder James
Deloitte LLP

Murphy Sarah
Deloitte LLP

Atkins Charisma
Shanxi Center for Disease Control and Prevention

Hulbert LaShonda
Centers for Disease Control and Prevention

Lusk Richard
Deloitte LLP

Ng Boon Peng
Centers for Disease Control and Prevention

Methodology

Keywords: Machine Learning, Systematic Review Screening, Natural Language Processing, Transfer Learning, Machine Learning Configurations, Applied Case Study

Posted Date: January 11th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-48078/v2

Version of Record: A version of this preprint was published at Systematic Reviews on April 2nd, 2021. See the published version at https://doi.org/10.1186/s13643-021-01640-6.

Abstract

Background: Systematic Reviews (SR), studies of studies, use a formal process to evaluate the quality of scientific literature and determine ensuing effectiveness from qualifying articles to establish consensus findings around a hypothesis. Their value is increasing as the conduct and publication of research and evaluation has expanded and the process of identifying key insights becomes more time consuming. Text analytics and Machine Learning (ML) techniques may help overcome this problem of scale while still maintaining the level of rigor expected of SRs.

Methods: In this article, we discuss an approach that uses existing examples of SRs to build and test a method for assisting the SR title and abstract pre-screening by reducing the initial pool of potential articles down to articles that meet inclusion criteria. Our approach differs from previous approaches to using ML as a SR tool in that it incorporates ML configurations guided by previously conducted SRs, and human confirmation on ML predictions of relevant articles during multiple iterative reviews on smaller tranches of citations. We applied the tailored method to a new SR review effort to validate performance.

Results: the case study test of the approach proved a sensitivity (recall) in finding relevant articles during down selection that may rival many traditional processes.

Conclusions: We believe this iterative method can help overcome bias in initial ML model training by having humans reinforce ML models with new and relevant information, and is an applied step towards transfer learning for ML in SR.

Background

Systematic Reviews (SR), studies of studies, use a formal process to evaluate the quality of scientific literature and determine ensuing effectiveness from qualifying articles to establish consensus findings around a hypothesis. Because SRs involve pooling information, conducting a SR can result in increased power and reduced bias in determining effectiveness, increased generalizability of findings, and can help identify publication trends, research gaps and other indicators of importance.1 Their value is increasing as the conduct and publication of research and evaluation has expanded and the process of identifying key insights becomes more time consuming.2 Text analytics and Machine Learning (ML) techniques may help overcome this problem of scale while still maintaining the level of rigor expected of SRs.

Where SR methods historically depend on human resources for each stage of the process,3 ML is a computer-based technique that uses statistics and pattern recognition to create models to make predictions from data to automate processes. One area of SRs that has seen application of ML is pre-screening title and abstracts for final inclusion in the quality scoring step.4 In this article, we discuss an approach that uses existing examples of SRs to build and test a method for assisting the SR title and abstract pre-screening by reducing the initial pool of potential articles down to articles that meet inclusion criteria. This iterative and guided method is aimed at maintaining sensitivity in finding relevant articles during pre-screening, while reducing the number of articles reviewed compared to traditional SR methods.

We incorporate many of the features of other ML approaches for SRs to our approach including reinforcing training data with human reviews of select articles, using only a small number (<200) of articles for initial ML training, and feature engineering of data for improved performance in selecting articles[5-10] Our approach differs from previous approaches to using ML as a SR tool in that it incorporates testing the proposed process on previously conducted SRs to guide ML parameters and configurations, and attempts to optimize the process of human confirmation on ML predictions of relevant articles during multiple iterative reviews by selecting only articles that will ensure sensitivity is reached. We also incorporate a quality checkpoint and initial determination of when to stop iterations of ML in the process to build confidence that a desired sensitivity has been reached without trusting only the ML effort or reviewing all articles. This guided method is also meant to be portable to new SR topics, and not just the ones that configurations were built upon. To test this, we used our guided configurations on a new SR need as a case study to see how our guided configurations perform on a previously untested topic.

Methods

Common features of the SR process include: Developing a research question, developing inclusion criteria, conducting a literature search, article title and abstract pre-screening, abstracting articles for analysis, and aggregating results to generate summary findings.[2] Our ML application efforts focused on creating efficiency in article title and abstract pre-screening. We refer to this step as “down selection” - reducing the initial pool of potential articles that meet inclusion criteria. To develop a generalizable ML approach to down selection we 1) created a theoretical ML “down selection” process based on goals and constraints of including ML in a down selection process, 2) established ML configuration guides by testing settings with experiment SRs, and 3) applied the process and ML configuration guides to a SR conducted by a team of scientists at the Centers for Disease Control and Prevention (CDC).

Developing a theoretical ML “down selection” process

To operationalize the ML addition to the down selection process,2 we developed a process based on the constraints of ML and SRs. The steps in the proposed process include 1) train ML models and perform ML predictions; 2) conduct human review of articles selected by the ML model and determine accuracy of prediction; 3) incorporate new human reviewed articles into iterative ML training; 4) random sampling to ensure performance (Figure 1).

Step 1. Train the ML Models and Perform ML Prediction. Supervised ML algorithms require training data to build models. These data “teach” the algorithms which articles meet inclusion criteria, and which do not, facilitating the creation of a model.11,12 Our process aimed to show that effective training data can come from a small set of articles that are pre-identified as meeting or not meeting inclusion criteria. In addition, a small volume (less than 200) random sampling of articles can be drawn from a keyword literature search or through unsupervised machine learning[8]. Iterative training will occur later in the proposed process after more articles are reviewed to reinforce learning.

In our proposed process, a ML model makes predictions, after training, on combined title and abstract text review. Through this process, each potential article fed to the model is given a score of how likely it is to fit inclusion criteria based on articles in the training data set. Prediction scores are probabilistic ranging from 0 to 1.0, with 1.0 being a perfect match to inclusion examples.[3]

Step 2. Human Review of ML Selected Articles and Determination. In this step, human reviewers examine articles predicted by the ML model to fit inclusion criteria. During review, humans correct the ML prediction by confirming if an article met inclusion criteria or did not. From this process a new training set (human reviewed) is created for use in a new iteration of training. This iterative training process is proposed to help improve ML prediction by expanding training sets and overcoming potential bias introduced by small training sets in step 1.

Step 3. Incorporate New Human Reviewed Articles into Iterative ML Training. For each iteration, a new model is trained on the set of human reviewed articles in step 2 and any previous training sets. The iterated model is then employed on the remaining unreviewed articles to determine a new predictive score. Iterations should continue until the number of articles predicted as relevant becomes small and human review does not confirm articles predicted to be relevant.

Step 4. Random Sampling to Ensure Accuracy. After exiting step 3, humans select a random sample of articles not predicted as relevant by ML to test sensitivity of the ML process. For our process we suggested a 99% confidence level sample with a 10% margin of error for calculating the total articles for random sampling to ensure confidence. We recommend this process to increase confidence that all inclusion articles have been identified. Humans check this random selection of articles to look for articles that fit review criteria. If more than one or two articles are found that fit inclusion criteria, this would indicate the ML approach has not reached a reasonable sensitivity and should continue for a new iteration (step 2).

Establish ML Configurations for the Down Selection Process

Supervised ML has a superabundant number of configurations for predictive model development. Areas identified that could have multiple configurations include a) cleaning text; b) reducing dimensionality; c) feature engineering; d) developing a training sample; e) initial algorithm selection and assessment; f) creating a soft voting stacked model; and g) choosing thresholds for the iterative modeling steps. To identify which ML configurations should be utilized for the proposed ML down selection process we utilized a hybrid theoretical and results-driven approach by testing on four previously completed SRs hereby referred to as experiment SRs. Configurations for steps a, b, c, and d were selected based on theoretical knowledge, while configurations for steps e, f, and g were selected by testing performance of different configurations for each experiment SR individually.

Step a. Cleaning Text. From each of the experiment SRs, we performed standard text cleaning on the combined titles and abstracts,13 removing numbers and common English words, and tokenizing words

into single and bi-grams. We used Python's Natural Language Toolkit (NLTK) version 3.2.4 for this process.

Step b. Reducing Dimensionality. Because our text cleaning process resulted in a data set with a large number of rows and columns (a high-dimensional matrix) that represent the numerical frequency of token occurrences, we performed dimensionality reduction.[14] We also manipulated the cleaned data into a term frequency–inverse document frequency (TF-IDF) matrix.[15] TF-IDF is a statistical weight, meant to show importance of a word is to a document and the entire series of documents in an analysis.

Step c. Feature Engineering. In ML applications, variables for modeling are often referred to as "features" of the data. Feature engineering involves manipulating variables to create new "features" of the data and is often used to boost predictive performance.16 We utilized latent Dirichlet allocation (LDA) on the reduced TF-IDF matrix to create new features based on topics found in the data using a generative probabilistic approach.[17] Using topics instead of just word counts as features creates the ability to identify patterns across articles that does not rely on word token occurrences. We set the LDA topics at 30 new features under the theoretical assumption that 30 would reach topic saturation in the data. We also used truncated singular value decomposition (TSVD) to perform feature decomposition – reducing a matrix to its constituent parts – on the TF-IDF Matrix. This resulted in a condensed TF-IDF matrix containing the 50 most significant features in terms of their representation of the original data.[18] We also know that a literature search will typically have few articles returned that will meet inclusion criteria (imbalanced data). To address this issue, we applied a Synthetic Minority Over-sampling Technique (SMOTE), which creates new feature points aimed at overcoming imbalanced data.[19] We appended the features derived from the LDA, TSVD, and SMOTE to the reduced TF-IDF matrix to get our final matrix for ML modeling.

Step d. Developing Training Sample. Our approach to creating a training set was to mimic the operational approach we outlined in the proposed ML "down-selection" process. We assumed that a small number of articles would be available for training; no more than 60 with examples from both relevant and non-relevant articles. Through this we created our initial training data through stratified random selection of the experiment SRs articles, our test data set was the unreviewed data from experiment SRs.

Step e. Initial Algorithm Selection and Assessment. Many ML algorithms for building models exist. We tested the following algorithms for overall accuracy from the initial training set: Support Vector Machine (SVM) with Stochastic Gradient Descent, K-Nearest Neighbors (KNN), Decision Tree Binary, SVM with a Sigmoidal Kernel, Gradient Boosting Classifier, Random Forest Classifier, and Multinomial Naive Bayes.20,21,22,23. Python's scikit-learn library and the base parameters of these modes were used for implementing models. From these, we chose the four models that performed the best in terms of accuracy (ratio of number of correct predictions to the total number predictions made) to include in a stacked ensemble model,[24,25] which combines the strengths of the best performing models. They were SVM with Stochastic Gradient Decent, KNN, Decision Trees, and Sigmoidal SVM.

Step f. Compiling into a Stacked Ensemble Model. We used a soft voting ensemble classifier to build predictive models to overcome any weakness in each individual model.[25] In a soft voting ensemble, different models are given weights that are applied to their prediction and combined for a final stacked prediction. We evaluated various weight distributions according to their area under receiver operating characteristic (ROC) curves from initial model training (step d).[26] The [10 -SVM, 1 - KNN, 1 – Decision Tree, 1 – Sigmoidal SVM] weighting distribution consistently resulted in the highest area under the curve of the options tested. As with individual models, the stacked ensemble model predicts a value for each article from 0-1.0, where values closer to 1.0 indicate an article being relevant for inclusion.

Step g. Choosing Prediction Thresholds for Iterative Modeling. Prediction thresholds in our scores (0-1.0) can be changed to influence the selection of volume of articles for review in iterations (SR Down Selection Step 3).[27] High thresholds result in lower volumes and vice versa. By comparing actual results of experiment SR data with different prediction thresholds, we were able to identify predictive threshold guides to optimize the selection of articles for review during iterations. Once we determined an optimal threshold for an iteration, we tested multiple thresholds on the next iteration to confirm sensitivity. We were able to accomplish this because we had known results and could simulate a human review (SR Down Selection Step 3). Based on testing of experiment SRs we found that 3 iterations, including the original training round, would reach the optimal trade off in sensitivity versus percent of articles reviewed based on a 98% sensitivity goal of finding relevant articles.

From our predictive threshold testing, we used the weighted average of best thresholds from each experiment SR as a guide for non-experimental application. These thresholds are shown in Table 1. These thresholds should be thought of as guides. Volume of articles selected for review from different thresholds should also be considered when selecting which threshold to proceed with.

Using these ML configurations, we examined the percent of total articles needed to reach 95% and 98% sensitivity of what human reviewers selected for inclusion in the experiment SRs. On average only 21% of articles would have to be reviewed to find 95% of what the human SR selected, while 30% would have to be reviewed to find 98%, including initial training articles (Table 1).

Table 1. Average Post-Hoc Model Performance

Average Post-Hoc Model Performance						
Data Set	Prediction Threshold for 1st Iteration	Prediction Threshold for 2nd Iteration	Prediction Threshold for 3rd Iteration	Total Articles	% of total human-reviewed articles needed to return 95% relevant articles	% of total human-reviewed articles needed to return 98% relevant articles
1st SR Review	50.0%	20.0%	20%	14,655	19.3%	24%
2nd SR Review	50.1%	30.3%	44%	15,234	18.9%	25%
3rd SR Review	75.0%	20.0%	20%	7,670	10.0%	34%
4th SR Review	70.0%	27.5%	19.5%	1,820	30.0%	41.8%
Weighted Average	57.6%	26.0%	29.5%	N/A	20.9%	29.8%

Case Study: Applying the Process and ML Configuration Guides

In May of 2018, CDC's Division of Diabetes Translation initiated a SR designed to describe the effectiveness of incentives in increasing enrollment and retention in chronic disease prevention and management programs.

To initiate our case study, we started with the articles identified in the CDC SR team's literature search phase. A total of 3,137 articles were returned from the literature search (following deduplication across searched databases). We applied the ML down selection process described above to this same set of 3,137 articles. Four team members participated in the iterative review of ML selected articles and their inclusion in the SR. Two team members independently reviewed each assigned abstract. If there were any conflicts, they were discussed and resolved. Once the team completed the ML down selection process, unreviewed article titles were scanned to determine if our case produced acceptable results when compared to a traditional approach.

Results

Table 2 shows a breakdown of the articles reviewed during the ML assisted process and how they compared to totals after final quality checks.

Table 2. Results for Each Iteration and Random Sample

	Relevant Training Sample (iteration)	Non-Relevant Sample (iteration)	Threshold Selected (iteration)	Articles for Review (iteration)	Relevant Articles (iteration)	Non-relevant Articles (iteration)	Total Articles Reviewed (cumulative)	Total Articles Not Reviewed (cumulative)
First Iteration	15	40	0.4	458	155	303	513	2,624
Second Iteration	170	343	0.3	260	43	217	773	2,364
Third Iteration	213	560	0.3	45	0	45	818	2,319
Fourth Iteration	213	605	Not Selected	N/A	N/A	N/A	818	2,319
Random Sample	N/A	N/A	N/A	156	1	155	974	2,163

SR ML Down Selection Step 1. To develop the first iteration of our case study, the CDC SR team identified a training set of 15 articles that met inclusion criteria, and 40 that did not. We built the trained models using the ML configuration approach identified from the experiment SRs.

SR ML Down Selection Step 2. We examined different prediction thresholds after the first iteration model predictions. We selected a threshold of 0.4, which resulted in 458 articles as priority for review. Although lower than the prediction threshold identified from experiment SRs, the guide threshold resulted in a number of articles thought to be too low (250) to produce enough of a robust new training set. For more information on threshold selection of each iteration, please view the supplemental material in Table S1. From the 458 articles, human review identified 155 as relevant and 303 as not relevant. We added this new set of human reviewed articles to the initial training articles for the second iteration of model training and predictions.

SR ML Down Selection Step 3. We examined different prediction thresholds after the second iteration predictions (Table S1). We selected a threshold of 0.3 based on experiment SR guides and number of articles which identified 260 articles as a priority for review. Human review identified 43 as relevant and 217 as non-relevant. We added this new set of human-reviewed articles to the existing training set for the third iteration of model training and predictions.

Results of third iteration. We examined different prediction thresholds after the third iteration predictions (Table S1). We selected a threshold of 0.3 based on experiment SR guides and number of articles, which resulted in 45 articles as priority for review. Human review identified no relevant articles. We added this new set of 45 human-reviewed articles to the existing training set for the fourth iteration of model training and predictions.

Results of fourth iteration. During experiment SR testing we found three iterations to reach an acceptable level of sensitivity performance. During our case we ran a fourth iteration to understand if the newly trained model selected a large number of articles for review. After reviewing potential inclusion article volume at different prediction thresholds, we decided to move on to Step 4—reviewing a random sample for an error check due to small articles volumes at low thresholds.

SR ML Down Selection Step 4. After three training iterations, we identified 2,319 citations as non-relevant. To test for saturation in sensitivity, we randomly selected 6.7% (155 citations) of these articles for human abstract review. This created a 10% margin of error with a 99% confidence level. Results from the random sample review returned 154 articles not meeting inclusion criteria and 1 potential article that met inclusion criteria, which was subsequently identified as background material and excluded.

As a final sensitivity verification after the ML process, we conducted a human review of only the titles of all 2,172 unreviewed articles. Of these we identified 6 articles for abstract review. We determined these were not relevant, which aligned with the ML process predictions.

Using recognized machine learning performance statistics to evaluate our approach, we achieved a sensitivity of 99.5% (213 out of 214) of total relevant articles while only conducting a human review of 31% of total articles returned from the search (Table 3). Sensitivity of the model was paramount to model acceptability, as a missed relevant article could jeopardize consensus findings of a full review of results. These results are consistent with experiment SR results in terms of sensitivity and percent of articles reviewed to reach that sensitivity.

Table 3. Final Results after Quality Check

	Number of Articles	Percent of Total N (3,137)
Total Articles Reviewed During ML Down Selection Process	974	31.0%
Total Articles Not Reviewed	2,163	69.0%
Total Relevant Articles Meeting Inclusion Criteria During ML Down Selection Process ML – Iterative Review Only	213	6.8%
Total Relevant Articles Meeting Review after Random Sampling Error Check and Iterative Review	214	6.8%

Discussion And Limitations

Our case study was conducted on a topic with a certain volume of articles for down selection. The question of efficiency for smaller and larger volume reviews remains unanswered. ML approaches require not only the expertise needed for normal SR down selection, but also the time of persons familiar with the ML process. In addition, between each iteration a brief period is needed for training and applying the ML models. Thus, time saved by ML exclusion of irrelevant articles may be lost while carrying out technical processes. For a low-volume down selection, the potential risk of not predicting correctly with the ML process may not be justifiable given the skillset and model training time needed. However, it is likely that the benefits outweigh the risks for high-volume reviews.

To develop our approach, we trained the model using data from four completed experiment SRs, each to answer different questions. This only provided guides for certain configurations as seen by us choosing different thresholds during our case study. Our approach could be fine-tuned using a larger number of completed reviews. In addition, other approaches and configurations in NLP and text processing such as transformer models built on entire literature corpuses will likely improve ML support for systematic reviews in terms of accuracy as they have outperformed common NLP tasks compared to n-gram based approaches, but were not applied at the time of writing. In the future, it is also possible that the approach could be conducted not just on title and abstract data, but on full text data. However, many literature

search engine capabilities and copyright restrictions only allow for a title and abstract review to be conducted without extra monetary costs.

Conclusions

In this paper we described the creation and testing of an approach to use guided ML to support SRs. To our knowledge, our case study is the first test of ML-supported SR that incorporates ML guides from previous reviews and multiple iterative human reviews. This approach to using guides represents some basic steps towards transfer learning approaches for ML in systematic reviews. The case study achieved sensitivity in finding relevant articles that rivals that of traditional SR. We believe this iterative method can help overcome bias in initial model training by having humans reinforce models with new information, however ensuring multiple reviewers may still be necessary to overcome human bias in the process. We also think this iterative approach is applicable to real-world scenarios where large initial training sets are not likely to be found. Others can look to our ML configurations as guides, but more experiments can be done to fine-tune configurations.

Abbreviations

CDC –Centers for Disease Control and Prevention

KNN – K Nearest Neighbors

LDA – Latent Dirichlet Allocation

ML – Machine Learning

NLP – Natural Language Processing

ROC – Receiver Operator Curve

SMOTE - Synthetic Minority Over-sampling Technique

SR – Systematic Reviews

SVM – Support Vector Machine

TF-IDF - Term frequency–inverse document frequency

TSVD – Truncated Singular Value Decomposition

Declarations

Ethical Approval and Consent to participate – not applicable

Consent for publication - All authors have confirmed approval of the manuscript for submission. No study subjects were part of the data so consent is not applicable

Availability of supporting data – The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Competing interests – No competing interest of conflicts of interest

Funding – no special interest funding was received for this

Authors' contributions -

Zimmerman John.,MPH - Is the lead author who developed initial code and approach to using previous reviews to guide ML configurations and developed the manuscript for this application. He also oversaw the adoption of the method to the case study referenced in the manuscript

Soler Robin E., PhD - Identified the case study application and confirmed the approach of applying the guide ML approach to the case study. Provided editing on the manuscript, and deep subject knowledge of Systematic reviews throughout the process

Lavinder James. - Developed portions of code used for the approach. Also reviewed the manuscript and contributed to its development and participated in the case study as a reviewer.

Murphy Sarah, MA. - Also reviewed the manuscript and contributed to its development and participated in the case study as a reviewer.

Atkins Charisma, MPH - Also reviewed the manuscript and contributed to its development and participated in the case study as a reviewer.

Hulbert LaShonda., MPH - Also reviewed the manuscript and contributed to development and participated in the case study as a reviewer.

Lusk Richard., MS - Developed portions of code used for the approach. Also reviewed the manuscript and contributed to

Ng Boon Peng, PhD - Also reviewed the manuscript and contributed to the overall approach of application of the method to the case study. Participated in final editorial reviews of the manuscript and participated in background and discussion development

Acknowledgements – no other acknowledgements

Authors' information

Zimmerman John.,MPH1, Soler Robin E., PhD[2], Lavinder James.[1], Murphy Sarah, MA.[1], Atkins Charisma, MPH[1], Hulbert LaShonda., MPH[2], Lusk Richard., MS1, & Ng Boon Peng, PhD[2,3]

[1] Deloitte Consulting, LLP, 191 Peachtree Street, Atlanta, GA

2 Centers for Disease Control and Prevention, National Center for Chronic Disease Prevention and Health Promotion, Division of Diabetes Translation, 1600 Clifton Rd, Atlanta, GA

[3] College of Nursing & Disability, Aging and Technology Cluster, University of Central Florida, 12201 Research Pkwy Suite 300, Orlando, FL

[1] Centers for Disease Control and Prevention, National Center for Chronic Disease Prevention and Health Promotion, Division of Diabetes Translation, 1600 Clifton Rd, Atlanta, GA

[3] College of Nursing & Disability, Aging and Technology Cluster, University of Central Florida, 12201 Research Pkwy Suite 300, Orlando, FL

References

1. Larsen PO & von Ins M. The rate of growth in scientific publication and the decline in coverage provided by Science Citation Index. *Scientometrics*, 2010;84(3):575-6
2. Munn Z, Stern C, Lockwood C & Jordan Z. What kind of systematic review should I conduct? A proposed typology and guidance for systematic reviewers in the medical and health sciences. *BMC Medical Research Methodology,* 2018:18:5. https://doi.org/10.1186/s12874-017-0468-4
3. Chen JJ, Tsai CA, Moon H, Ahn H, Young JJ, Chen CH. *Decision threshold adjustment in class prediction.* SAR QSAR Environ Res. 2006 Jun;17(3):337-52.
4. Tsafnat G, Glasziou P, Choong MK, Dunn A, Galgani F & Coiera E. Systematic review automation technologies. *Systematic Reviews*, 2014:3, 1-15.
5. Thomas J, Noel-Storr A, Marshall I, Wallace B, McDonald S, Mavergames S et al. Living systematic reviews: 2. Combining human and machine effort. *J of Clin Epi*; 2017:91(31:37).
6. Carlos Francisco Moreno-Garcia, Magaly Aceves-Martins, Francesc Serratosa; Unsupervised machine learning application to perform a systematic and meta-analysis in medical research. *Computación y Sistemas,* 2016:20(1), 7-17. doi: 10.13053/CyS-20-1-2360
7. Jaspers S, De Troyer E, & Aerts M. Machine learning techniques for the automation of literature reviews and systematic reviews in EFSA. *EFSA supporting publication*; 2018:EN-1427, 83pp. doi:10.2903/sp.efsa.2018.EN-1427
8. Xiong Z, Liu T, Tse G, Gong M, Gladding PA, Smaill BH, Stiles MK, Gillis AM & Zhao J. A Machine learning aided systematic review and meta-Analysis of the relative risk of atrial fibrillation in patients with diabetes mellitus. *Physiol*, 2018:9:835. doi: 10.3389/fphys.2018.00835
9. Wallace, B.C., Trikalinos, T.A., Lau, J. et al. Semi-automated screening of biomedical citations for systematic reviews. *BMC Bioinformatics*11, 55 (2010). https://doi.org/10.1186/1471-2105-11-55
10. Bannach-Brown, A., Przybyła, P., Thomas, J., Rice, A. S. C., Ananiadou, S., Liao, J., & Macleod, M. R. Machine learning algorithms for systematic reviews: reducing workload in a preclinical review of

animal studies and reducing human screening error. *Systematic Reviews* 8(23) 2019. https://doi.org/10.1101/255760

11. Kosiantis SB. Supervised machine learning: A review of classification techniques. *Informatica*, 2007:31;249:268.
12. James, G. An introduction to statistical learning: with applications in R. New York, NY. Springer. 2013:21-23.
13. Boudin F, Mougard H, & Cram D. How document pre-processing affects keyphrase extraction performance. *International Conference on Intelligent Text Processing and Computational Linguistics*; 2014:April. doi:10.1007/978-3-642-54906-9_14.
14. Mao Y, Balasubramanian K & Lebanon G. *Dimensionality reduction for text using domain knowledge*. *COLING;* 2010:801-809.
15. Ramos, J. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*. 2013: 242, pp. 133-142.
16. Joachims T. *Text categorization with support vector machines: Learning with many relevant features. ECML'98 Proceedings of the 10th European Conference on Machine Learning. 1998:137-142.*
17. *Blei, D. M., Ng, A. Y., & Jordan, M. I.. Latent dirichlet allocation. Journal of machine Learning research, 2003:3(Jan), 993-1022.*
18. Dhillon, I. S., & Modha, D. S. Concept decompositions for large sparse text data using clustering. *Machine learning*. 2001:(1-2), 143-175.
19. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P.. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*. 2002:16, 321-357.
20. Cortes C, Vapnik V. Support-vector networks. Kluwer Academic Publishers. 1995: 273-297.
21. Quinlan, JR. Induction of decision trees. Kluwer Academic Publishers. 1986:81-106.
22. Lu S & Jin, Z. Improved stochastic gradient descent algorithm for SVM. *International Journal of Recent Engineering Science*. 2017.
23. Cover T, Hart P. Nearest neighbor pattern classification. *IEEE - Transactions on Information Theory*. 1967:21–27.
24. Hsiang-Fu Y, Hung-Yi L, et al. Feature engineering and classifier ensemble for KDD Cup 2010. *Journal of Machine Learning Research Workshop and Conference Proceedings*. 2010:1-16.
25. Onan, A., Korukoğlu, S., & Bulut, H. A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification. *Expert Systems with Applications.* 2016;62, 1-16.
26. Bradley AP. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern recognition*, 1997:30(7),1145-1159.
27. Lipton Z, Elkan C & Naryanaswamy B. Optimal thresholding of classifiers to maximize F1 measure. *Machine Learning Knowledge Discovery Databases*. 2014:225-239.