International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 30, No. 4, April, 2025, pp. 913-921
Journal homepage: http://iieta.org/journals/isi

An Accurate Model for Text Document Classification Using Machine Learning Techniques

Asraa Safaa Ahmed[1*], Abbas Abd-Alhussein Haddad[2], Rana Sami Hameed[3], Mustafa Sabah Taha[4]

[1] Department of Computer Sciences, College of Science, Diyala University, Diyala 32001, Iraq
[2] College of Education, Computers Department, University of Misan, Misan 62001, Iraq
[3] College of Arts, University of Kirkuk, Kirkuk 36004, Iraq
[4] Missan Oil Training Institute, Ministry of Oil, Missan 62001 Iraq

Corresponding Author Email: asraasafaa@uodiyala.edu.iq

https://doi.org/10.18280/isi.300408

Received: 26 January 2025
Revised: 16 April 2025
Accepted: 24 April 2025
Available online: 30 April 2025

Keywords:
machine learning, text document classifications, K-Nearest Neighbor, XGBoost, Random Forest, Decision Tree

ABSTRACT

Text document classification (TDC) is an approach used for the classification of any kind of document for the target category or out. Text classification algorithms have come across significant challenges recently as a result of the exponential expansion of digital text documents; the large volume of words in each document reduces the effectiveness of these existing text classifiers. A key method for improving classification accuracy and getting rid of redundant data is referred to as feature selection (FS). In this work, several phases have been conducted to test and equip the proposed model. Initially, the applied machine learning algorithms were tested and trained using the Reuters-21578 dataset. Second, data cleaning, label encoding, tokenizing, text cleaning, and last TF-IDF vectorization were done to prepare the dataset. Thirdly, four distinct machine learning algorithms, Extreme Gradient Boosting (XGBoost), K-Nearest Neighbor (KNN), Random Forest (RF), and Decision Tree (DT) were used to build a brand-new machine learning-based text document classification model (ML-TDCM) for document classification. Finally, several metrics, including F1 score, accuracy, precision, and recall, were used to assess the proposed model. With a 91% classification accuracy, XGBoost turned out to be the best-performing algorithm among the others. The obtained results were also matched with results obtained in past studies, verifying the performance of the suggested models and so defining them as possible methods to be applied in the next work concerning document categorization.

1. INTRODUCTION

The ongoing evolution of text classification (TC) can be traced back to the increasing demand for effective management and organization of ever-increasing amounts of text data. Text document classification (TDC) began as a manual procedure based on text and corpus linguistics that included classifying document texts into predetermined genres or topics [1]. However, the creation of automated systems became necessary due to the digital revolution and the exponential rise of textual data, which made human approaches impracticable. The process of classifying documents involves dividing them into groups according to their content. TDC, a fundamental learning challenge, remains the core of many information management and retrieval activities. Documents classification is crucial for many applications that deal with arranging, categorizing, finding, and succinctly expressing large amounts of data. A well-researched and long-standing issue in information retrieval is document classification [2, 3].

In general, automatic TDC falls into three groups -semi-supervised, unsupervised, and supervised. In supervised document classification, information regarding the correct classification of documents is provided by a mechanism external to the classification model, usually a human. In the case of supervised document categorization, this facilitates the testing of the model's accuracy. In semi-supervised document categorization, an external mechanism identifies portions of the document while leaving the remainder unlabeled, whereas in unsupervised document classification, no external mechanism offers any information [4].

Although fundamental, the rule-based algorithms and handmade features used in early TDC approaches were not flexible enough to accommodate new data. Hence, later approaches such as statistical methodologies, nature-inspired algorithms, and graph-based approaches have been developed to improve the accuracy and flexibility of text classification [1]. Moreover, TDC is a difficult undertaking due to two primary factors: (a) feature extraction; and (b) topic ambiguity. Feature extraction is the process of identifying the appropriate set of features that best characterize the document and aid in the development of a strong classification model. Again, it is hard to classify many broad complex topic documents or put them into specific categories. Consider a document that discusses theocracy; it would be difficult to determine if such a document belongs in the political or religious categories. Additionally, broad topic documents may include terminology that can signify different things depending on the context and that may be used more than once throughout a document [5].

TDC is one of the major issues due to several challenges; in

TDC, model performance is significantly affected by overfitting, where the model learns noise from training data and fails to generalize, as well as underfitting, where the model is too simplistic to capture meaningful patterns. To overcome these challenges, strategies such as regularization, dropout layers, data augmentation, increased model complexity, and the use of advanced architectures like transformers or pre-trained models are employed to improve generalization and ensure reliable classification across diverse text formats [6].

Class imbalance in TDC arises when certain categories dominate the dataset while others are underrepresented, leading to biased predictions and poor generalization, especially in critical tasks like sentiment analysis or spam detection. To address this, techniques such as oversampling (e.g., SMOTE), algorithmic adjustments like class weighting and boosting, and advanced models like BERT and GPT with cost-sensitive learning are employed to enhance fairness and robustness in TDC models [7].

The high-dimensional and sparse nature of the feature space in TDC results from the numerical encoding of unstructured text which poses challenges for model training, generalization, and interpretability. To manage this complexity, techniques such as feature selection, dimensionality reduction (e.g., PCA, autoencoders), word embeddings (e.g., Word2Vec, BERT), and explainable AI tools are employed to improve efficiency, capture contextual meaning, and maintain transparency in complex language tasks [8].

Ambiguity and polysemy in TDC (where words have multiple or context-dependent meanings) pose significant challenges to text classification (TC), as they hinder accurate interpretation and classification. While contextual embeddings (e.g., BERT, GPT), domain-specific models, and ensemble techniques help resolve meaning through surrounding context, these methods remain computationally intensive, highlighting the persistent difficulty of handling linguistic ambiguity in natural language processing [1].

In TDC, a large number of documents is generated every day, and managing document classification accurately is very complex and difficult due to the reliance on an effective recovery of the correct document categories on the number of labeled documents. Moreover, TDC algorithms called classifiers have been developed based on the understanding of the meaning of texts [9].

Among the existing TDC methods are Support Vector Machine, Naïve Bayes classifier, Neural Network, and Decision Trees [10-12]. TDC refers to a data processing technique that routinely assigns a given document to its category or class. There have been many works in the literature dealing with the topic of text classification in different languages such as English [13], Chinese [14], Arabic [11], and many other languages. However, these works have not reached the best degree of accuracy. Hence, this research seeks to develop a model that maintains computational complexity while accurately classifying the text documentation called the machine learning-based text document classification model (ML-TDCM). Four distinct machine learning techniques - RF, KNN, DT, and XGBoost - are applied in this work to classify documents using the Reuters-21578 dataset.

The Organization of this paper is listed as follows. Section 2 addresses the basic principles of text classification, and highlights noteworthy studies from 2020 to 2025. The proposed methodology that achieves the aim of this paper is listed in Section 3, in this section the dataset, the pre-processing actions as well the building of the proposed model have been explained in detail. Section 4 presents the results obtained using the proposed methodology and discusses them in a way that is understandable to researchers interested in this field. The conclusion of this work is listed in section 5.

2. LITERATURE REVIEW

Text categorization (TC) is thoroughly reviewed in the second section, which also covers related subjects and recent advancements in the field. The fundamentals of text classification are covered in this section, along with notable research from 2020 to 2025, important problems, and new developments. The section aims to give a cohesive narrative that connects the theoretical and practical components of text classification by breaking the subject up into logical subsections. Table 1 shows the analysis of relevant studies between 2020 and 2025.

Table 1. The analysis of relevant studies between 2020 and 2025

Ref / Year	Objectives	Insights	Practical Implications
[15] - 2020	-Review of the existing methods of text classification (TC). -Propose future studies for addressing the issues associated with text mining.	Numerous feature selection techniques and classification algorithms have been presented in earlier text classification studies to address issues like scalability caused by the enormous growth in text data. In a variety of academic domains, these works emphasize the significance of efficient information management and organization.	-Significant applications in real-world TC. -Addresses the issues in scalability and text mining.
[16] - 2020	-Compare the performance of DL and ML algorithms. -Explore scalability with larger data instances.	Prior research on text mining used machine learning and deep learning techniques for classification using very small data instances. This study expands on that by evaluating the scalability and performance of these approaches using a bigger dataset of 6000 instances in six different classes.	-DL performed better than conventional TC methods. -Explored scalability of techniques for larger data instances.
[17] - 2020	-Review of the stages of the text classification process. -Overview and comparison of some popular TC algorithms.	The research lacks insight into specifics about the in-text classification of earlier studies; rather, it surveys and contrasts widely-used classification algorithms, concentrating on the classification process, which includes preprocessing, feature engineering, dimension decomposition, model selection, and evaluation.	-There are implications of TC in education, politics, and finance. -Four popular classification algorithms are compared.
[18] – 2022	-Overview of TC research gaps and trends. -Review of research	Nine primary issues are identified, research patterns, data sources, language choices, and applied approaches are analyzed, and important trends and gaps in the field are	-Identifies the patterns and gaps in TC research. -Highlighted nine major TC-

	patterns, issues, and problem-solving techniques in TC.	highlighted as the report thoroughly examines 96 papers on text classification conducted between 2006 and 2017.	related problems.
[19] - 2023	-Use deep networks to design text representation and classification models. - Improve the accuracy of text feature classification and representation.	The study emphasizes how conventional text classification approaches, such as the bag-of-words model and vector space model, struggle with issues like sparsity, high dimensionality, and context loss, leading to a move toward deep learning approaches for better results.	-DL models enhance the performance of TC models compared to conventional models. -The accuracy of the proposed BRCNN and ACNN models in terms of text feature classification and representation is improved.
[20] - 2023	Reduce workload in systematic reviews by automating citation classification in systematic reviews.	The study focuses on automating citation categorization to reduce workload in systematic review preparation, is cited in the report. This study serves as a basis for the datasets used in the current text classification research on drug class reviews.	Reduce workload in drug class studies by automating citation classification in systematic reviews.
[21]- 2022	-Explore ML algorithms for effective TC. -Compare various algorithms such as SVM, KNN, CNN, and RNN for performance in TC.	Specific information about earlier text classification research is not included in the paper. For text classification problems, it focus on assessing and contrasting the effectiveness of several machine learning algorithms, including DT, SVM, KNN, CNN, and RNN.	-Text classification accuracy and efficiency are proved using ML. -Complex and large datasets are effectively handled by the algorithms.
[22] - 2024	-Review of DL in TC. -Evaluate research patterns and technical approaches.	The study lacks the in-depth details of TC of existing studies; rather, it reviews and compares widely used classification techniques.	There are implications of TC in education, politics, and finance.
[23] - 2024	-Evaluate the need for simple vs complex models. -Assess the performance of different simple and complex models on different datasets and classification tasks.	By pointing out that prior research mostly compares similar sorts of procedures without a thorough benchmark, the study draws attention to a gap in the body of current work. The goal of this study is to present a thorough assessment across a range of tasks, datasets, and model architectures.	-In certain tasks, simple methods can perform better than complex models. -F1 performance and complexity have a negative relationship for small datasets.
[24] - 2025	-Explore the effectiveness of ML models for TC. -Compare the effectiveness of different algorithms, such as BiLSTM and CNN.	Specific information about earlier text classification research is not included in the paper. It focuses on assessing and contrasting how well different machine learning algorithms, like CNN, BiLSTM, and DT, perform in TC tasks.	ML enhances the TC efficiency and accuracy. -Complex and large datasets are effectively handled by the algorithms.

3. THE PROPOSED METHOD

This section of the research proposal presents information about the Reuters-21578 dataset that we have used in detail, as well as the fundamental ideas and model architecture of our suggested (ML-TDCM). The general framework of the suggested model is shown in Figure 1.

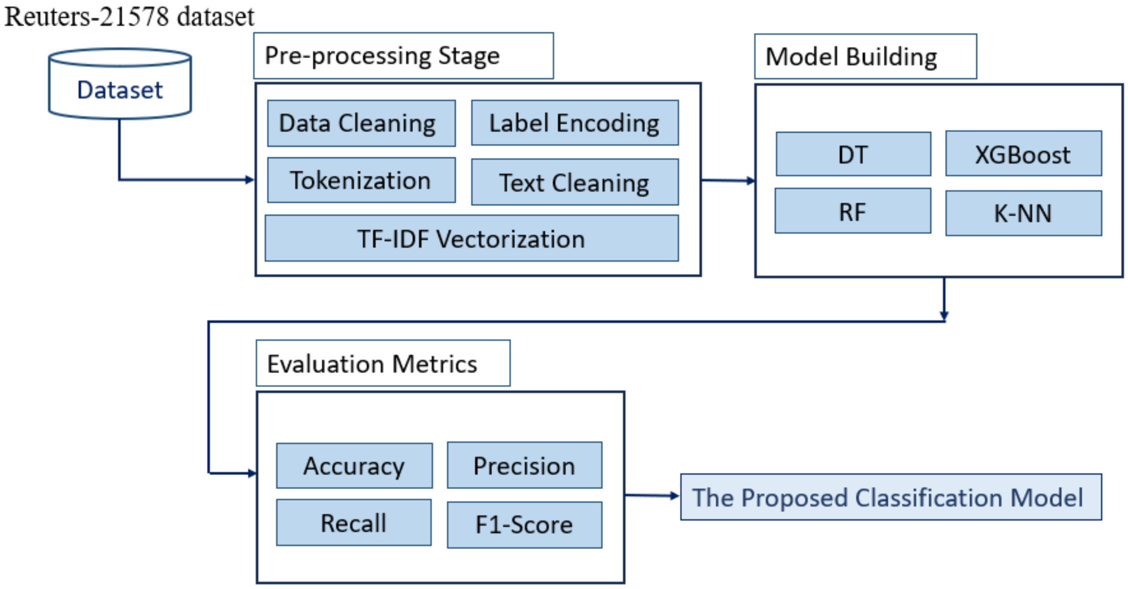

Figure 1. The proposed ML-TDCM framework

To be more specific, the first step in this investigation is to obtain the Reuters-21578 dataset. Then, for these data to be accepted by the anticipated (ML-TDCM) that will be put into place, they must be cleaned and processed. The data pre-processing stage involves several techniques such as tokenization, label encoding, text cleaning, TF-IDF vectorization, and data cleaning. After the data is pre-processed, it is ready to be fed to the proposed models that will be implemented to perform document classification. In the proposed research, four machine learning algorithms were implemented XGBoost, DT, KNN, and RF. Once these machine learning algorithms are trained, they are tested on a fraction of the dataset, after which their performance is assessed according to certain evaluation metrics.

3.1 Dataset

The Reuters-21578 dataset, which is pertinent to this study, is an initial compilation that was published in the Reuters Newswire in 1987 [25]. The collection known as "Reuters-21578" consisted of a compilation of papers that were indexed by hand in the same year by personnel from Reuters Ltd. and Carnegie Group, Inc. Steve Weinstein, Mike Topliss, and Sam Dobbins were the representatives of Reuters Ltd., while Monica Cellio, Irene Nirenburg, Phil Hayes, Peggy Andersen, and Laura Knecht were the representatives of Carnegie Group, Inc. The remarkable work that was done by these individuals was shared in 1990 with the research community through the Information Retrieval Laboratory in Amherst. Next, the documents are formatted by David D. Lewis and Stephen Harding before producing their related data files.

Later on, David Lewis in collaboration with Peter Shoemaker continued formatting the documents in 1991 and 1992 at the University of Chicago [26]. This version of the documents was also released in 1993 and was known as "Reuters-22173, Distribution 1.0". In subsequent years, some researchers concluded that a new version of Distribution 1.0 was necessary to make the results produced by using it more similar. In addition to fixing typographical problems in the categories and formats of the collection, the new suggested collection must have clearer formatting and explain common usage instructions. Based on Finch's SGML-tagged version of the collection from a previous study, Steve Finch and David D. Lewis created this updated version. Consequently, in the cleaning process, 595 documents were removed due to having duplicates. This means that the new cleaned version had 21,578 documents in total, hence the name Reuters-21578 collection. The "Reuters-21578, Distribution 1.0" consists of clean documents without typographical errors and topic errors. A subset of this collection, known as “ModApte” including 9,603 documents only for training and 3,299 documents for testing is often used. With further adjustments, by focusing on categories that only have 1 document in both the training and testing sets, the dataset becomes comprised of 7769 training documents and 3019 testing documents falling under 90 categories [25].

In this paper, the collection of text document comprises 8,000 documents and 15,818 words. We trained the models on a random subset including 90% of texts, categorized as science, social studies, snacks, and tours. Through model training, we acquire estimates for the word-topic distributions φ, topic-document distributions θ, the allocation of word tokens to topics, and the hyperparameter γ pertaining to the topic-category distribution. We assess generalization performance on the remaining papers in the science, snack, and tour genres, as well as on a selection of documents categorized as social studies. By examining these classified papers, we initially assess the models' capacity to generalize within the same genre, followed by an evaluation of their performance across genres in other categories. For each test document, we utilize a random 50% of the document's lexicon to estimate document-topic and topic-category distributions. Utilize the calculated distributions to assess perplexity on the remaining 50% of words.

3.2 Data preprocessing

One of the most important steps in the approach used is data pre-processing, which is the stage where data are ready to be fed into machine learning algorithms. Data typically has to be modified to meet the requirements of the proposed ML-TDCM since machine learning algorithms require varied qualities from input data. Numerous sub-steps may be included in the data pre-processing stage; they are outlined in this section.

Data Cleaning: In this stage rows containing null values were removed from the dataset. This step comes to guarantee that the dataset does not contain any incomplete data.

Label Encoding: In this stage labels within the dataset were converted into a form that is understandable by the proposed (ML-TDCM). In this case, the proposed (ML-TDCM) cannot comprehend string labels. As a result, label encoding transforms the string labels into a numerical value.

Tokenization: In this stage, the text within the dataset was divided into smaller units, which are referred to as tokens. Tokens can vary in size, where some can be words, thus small, while others can be phrases, thus large.

Text Cleaning: In this stage, the unnecessary words, symbols, and characters within the text must be removed. The process of removing these unwanted items is known as text cleaning. Text cleaning takes place automatically by removing stopwords such as “the”, “is” and many other words as well as removing punctuation.

Term Frequency-Inverse Document Frequency vectorization (TF-IDF): This stage represents the final step in the data pre-processing for the proposed study. During tasks such as text classification and summarization, TF-IDF is helpful for reducing unnecessary terms [1]. Hence, this study used TF-IDF, a popular feature weighting technique in the vector space model that treats all documents equally in its computation and successfully highlights the significance of a term within a document collection. Moreover, this study uses the TF-IDF vectorization method for several reasons. First: Term Frequency (TF). It measures how often a term appears in a document. A higher frequency indicates the term's importance within that specific document. While Inverse Document Frequency (IDF) adjusts the term frequency by considering how common or rare the term is across all documents in the dataset. Common words (like "the," "is," and "and") that appear in many documents are given lower weight, while rare terms are weighted higher. Second, reducing the impact of common words means common words (also known as stopwords) like "the," "a," and "and" are frequent across most documents but often carry little useful information for classification. TF-IDF reduces their weight, preventing them from dominating the feature set. Moreover, to calculate the TF-IDF value, the values of two terms known as term frequency “TF” and inverse document frequency “IDF” must be multiplied. Through this process, the text is converted into a numerical format that is understandable by the proposed ML-TDCM. In TF-IDF vectorization, each document is represented as a vector whose dimension reflects a single term. On the other hand, the value of the vector reflects how important the term is within the document.

3.3 The proposed machine learning algorithms

3.3.1 DT algorithm

One of the most used algorithms for classification jobs is the DT algorithm [27]. DT uses decision rules that are taught to the model through training it on input data to forecast the target's value. By recognizing features in the data, the model picks up the rules. The binary form that makes up the Decision Tree's structure has internal nodes that correspond to attribute tests; test results are displayed on branches, and class labels are displayed on leaf nodes. The top node in the Decision Tree model is referred to as the "root node." The attribute values are used by the root node to acquire segmentation knowledge. In addition, the tree is often split using a technique known as "recursive partitioning."

3.3.2 XGBoost

Another commonly used method for loss function reduction is the XGBoost algorithm [28]. The XGBoost technique introduces models into communities iteratively by utilizing loops within loops. One of the noteworthy aspects is that to address this specific problem, XGBoost concentrates on instances in which the model is unable to predict anything accurately, or at least not easily. This is typically accomplished by skewing the distribution to increase the likelihood of these inaccurate prediction occurrences in a sample. This focus, together with its ability to make basic predictions, aids the algorithm in making more reliable results forecasts. Consequently, XGBoost has been applied in this study.

3.3.3 RF algorithm

RF algorithm [29] gained its name from the real-life structure that it resembles, a forest, thus a collection of tree-type classifiers. The properties relative to RF are that it selects the strongest features and uses them as an input. This process takes place through simple probability. Originally, the RF algorithm was crafted by Breiman in 2001 by merging subsets of data and mapping random feature subspace samples, consequently, the multiple Decision Trees format was created. In this paper, we did experiments with different values for the number of trees (e.g., 50, 100, 200) and evaluated performance using cross-validation to find the optimal number. Since increasing the number of trees may not significantly improve performance but will increase computation time, we have used 100 trees as start points.

3.3.4 KNN algorithm

KNN [30] is regarded as a crucial method in machine learning that is used to forecast the collection of datasets that are most similar to one another. It is frequently used to classify a quantified request based on both the text and the text region closest to it (where K is the number of neighbors in the datasets that are available). The KNN method is based on the similarity learning method, which is used in numerous text classification and data analytics applications. While the KNN classifier finds the closest neighbors in the learning texts, a test document is used to forecast the category. The classes of the k neighbors are then utilized to assign the value to the respected class. The idea behind the KNN technique and the clusters concept is depicted in Figure 2 using different colors and shapes.

In this study, KNN was used to categorize the TDC using the Reuters-21578 dataset. The classification performance showed that KNN performed better than other classification techniques in both classification and clustering. The concept KNN is built on the fact that the result is computed from the input by whichever neighbor is the nearest to the input in terms of features and characteristics, hence, the name, “Nearest neighbor”. As a result, the output results are directly influenced by the nearest neighbors concluded from the training data. Often, KNN is used in pattern recognition tasks, as well as regression tasks and outlier detection.

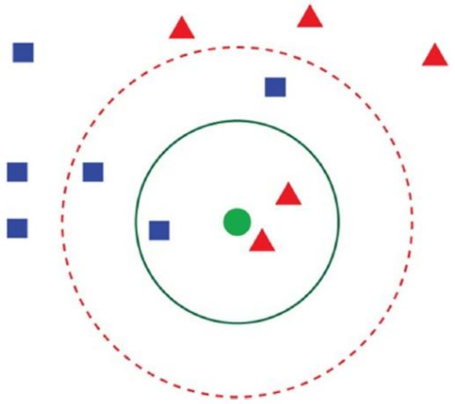

Figure 2. KNN -based classification approaches [31]

4. RESULTS AND DISCUSSION

The suggested ML-TDCM's implementation achievements were assessed using several evaluation metrics, including precision, recall, F1 score, and precision [32-34]. Some terms are utilized frequently in the calculation of all the metrics: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Real positives (TP) are those in which the model correctly predicts the real instant to be true. Likewise, a true negative, or TN, is what happens when the system classifies a negative instance as negative. Conversely, a false negative, or FN, occurs when the system incorrectly classifies a positive occurrence as untrue. On the other hand, a false positive, or FP, occurs when the model incorrectly forecasts a negative event as positive.

i. Accuracy Metric: In this study, the accuracy metric has been used to represent the percentage of correctly predicted samples. It is determined by dividing the number of true predictions (TP and TN) by the total number of predictions generated by the model. Obviously, in any classification task, the model must produce as few false negatives as possible. The equation for accuracy is as follows:

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

ii. Precision Metric: Another metric has been used in this research, in this study, Precision is calculated depending on the 4 terms, but specifically on TP and FP. Precision measures the fraction of the correctly predicted positives from the combination of true positives and false positives. Thus, precision can be calculated as follows:

$$\text{Precision} = \frac{TP}{TP+FP} \qquad (2)$$

iii. Recall Metric: Recall on the other hand of this paper relies on TP and FN, which in reality are all positive instances. Thus, recall measures how well the model identifies the true positives that are accurately classified as positives. Another term that is often related to Recall is sensitivity. The following equation depicts how recall can be calculated from TP and FN.

$$Recall \ = \frac{TP}{TP+FN} \qquad (3)$$

iv. F1 Score: The last metric used in this research is the F1 score, this metric relies on both precision and recall metrics. This score evaluates the performance of the model by relying on false positives and false negatives. The following equation reveals how the F1 score can be calculated:

$$F1 = \frac{2\times Precision \times Recall}{Precision + Recall} \qquad (4)$$

XGBoost, KNN, RF, and DT algorithms were used in the proposed study to classify documents. These algorithms are more appropriate for document classification because they are faster to learn and produce results in comparison with such Deep Learning algorithms as CNN or RNN, and have fewer layers. Thus, on the Reuters-21578 dataset, these four algorithms were evaluated, and the following results were achieved:

i. XGBoost algorithm: This algorithm was able to achieve a 0.9115 accuracy with the F1 score being 0.9055. The F1 score was a result of the 0. 9038 precision and 0. 9115 recall. These results indicate the highest performance of the XGBoost model in its classification of the documents.

ii. KNN algorithm: The accuracy of the utilized KNN algorithm was the second-best in the proposed study; the KNN algorithm achieved a 0.8711 accuracy and 0.8622 F1 score while the precision was 0.8629 and its recall was 0.8711.

iii. RF algorithm: This algorithm has achieved a value of 0.8298 in terms of accuracy and 0.8046 in terms of F1 score. On the other hand, the scored precision was 0.8189, and the recall was 0.8298. These results put the DT model in a lower position relative to XGBoost and KNN models, however, these results still reflect a good performance of the RF model.

iv. DT algorithm: The lowest achieving algorithm with respect to accuracy and F1 score was the DT algorithm which achieved a 0.8172 accuracy level, and 0.8145 F1 score. As for the precision, it was 0.8160, and the recall was 0.8172. These results put the DT model in a lower position relative to XGBoost and KNN models, however, these results still reflect a good performance of the DT model.

In this research, we presented the performance results of the used parameters—precision, accuracy, recall, and F1—for the machine learning algorithms used in two ways. First, we presented them in a tabular form to make it easier for the reader to understand the results. Second, we presented them in a graphical form.

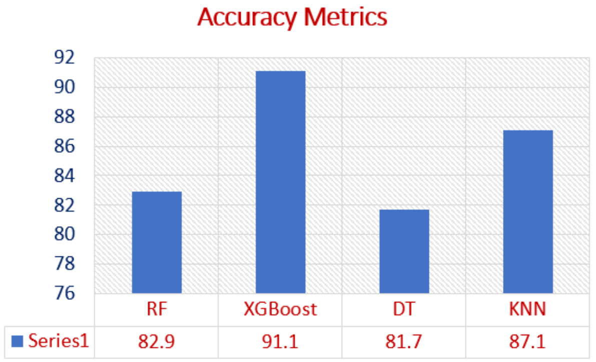

Figure 3. The accuracy metrics of the implemented machine learning algorithms

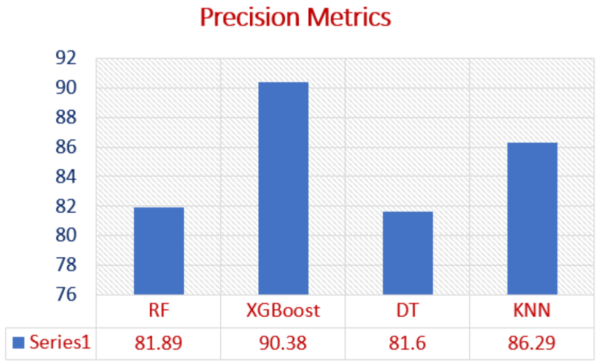

Figure 4. The precision metrics of the implemented machine learning algorithms

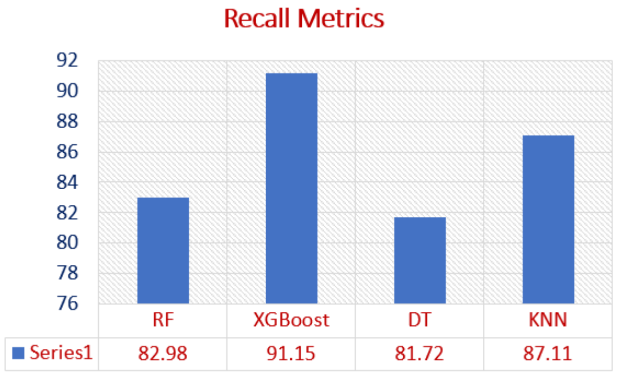

Figure 5. The recall metrics of the implemented machine learning algorithms

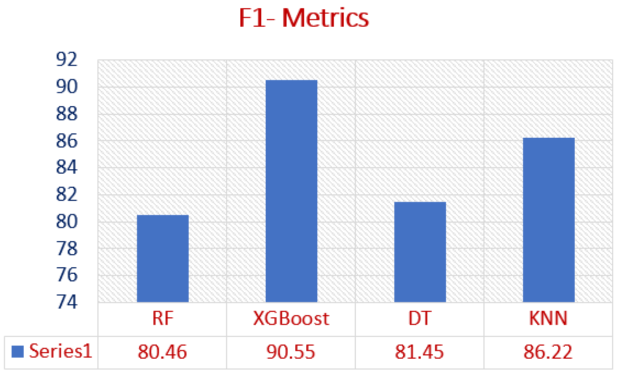

Figure 6. The F1 score metrics of the implemented machine learning algorithms

While Figures 3-6 reflect the evaluation performances of the proposed ML-TDCM in terms of accuracy, precision, recall, and F1 score. Figures below show that the XGBoost is the superior algorithm, followed by the KNN algorithm, whereas the least performing algorithm is the DT algorithm.

To gain more insight into the performance of the proposed ML-TDCM model, it is important to perform a comprehensive comparison with models that have been published in the literature. As mentioned earlier, the proposed method comprising four machine learning algorithms namely DT, RF, XGBoost, and KNN performed well in classifying documents, with the best-performing algorithm being XGBoost, achieving a 91% accuracy. In a study presented [35], the authors introduced the gLDA algorithm as a method for classifying documents. Their concept was to add a topic-category distribution parameter to Latent Dirichlet Allocation (LDA) to classify texts. Their methodology relied on Gibbs sampling to approximate inferences. In addition, they tested their model on 2 different datasets (Reuters-21578 dataset and Stanford Sentiment Treebank dataset). Their model was able to achieve 89.1% accuracy. Even though this proposition is an enhancement to the original LDA algorithm, it is still surpassed by the proposed XGBoost algorithm in this study.

Furthermore, the study [36] presented neural networks as a method for classifying text documents. They introduced the BiLSTM model with a simple architecture incorporating regularization techniques to classify documents. The authors chose to train and test the BiLSTM model on 4 different datasets, including the Reuters-21578 dataset. In terms of results, the LSTMbase model scored 87.6% accuracy, whereas the LSTMreg model scored 89.1% accuracy. These results are comparable to the accuracies achieved by the proposed KNN model (87.1%). Both of these studies not only validate the results achieved by this study but also show the possibility of increasing document classification accuracy by using the XGBoost algorithm. Table 2 shows the accuracies reached by the proposed ML-TDCM model as well as the models from the other studies. Table 3 shows that the XGBoost algorithm achieved the highest accuracy, followed by the LSTMreg [36], the LDA [35], and the KNN models.

Table 2. The performance results of the parameters used

Algorithm	Accuracy	Presession	Recall	F1
RF	82.9	81.89	82.98	80.46
XGBoost	91.1	90.38	91.18	90.55
DT	81.7	81.6	81.72	81.45
KNN	87.1	86.29	87.11	86.22

Table 3. The numerical values of the accuracies reached by the implemented ML algorithms

Model / Reference	Accuracy Metric
Our Model / RF algorithm	82.9%
Our Model / XGBoost algorithm	91.1%
Our Model / DT algorithm	81.7%
Our Model / KNN algorithm	87.1%
gLDA / [18]	89.1%
LSTMbase / [19]	87.6%
LSTMreg / [19]	89.1%

5. CONCLUSION AND FUTURE WORK

Text categorization (TC) has advanced significantly and is now a key component of ML and NLP. The processing, organization, and analysis of large scale textual data across multiple domains has been transformed by the shift from human classification to scalable, machine learning-driven techniques. One of the main components of many modern programs that deal with spam filtration, content organization, information retrieval, search and recommendation systems, and customer assistance is document classification. Document classification entails assigning labels to each document according to one or more categories that correspond to its content. Various models were used in numerous research to classify documents. In this study, document classification was carried out by 4 distinct machine learning algorithms, namely KNN, XGBoost, RF, and DT. The methodology started by acquiring the Reuters-21578 dataset. The data in this dataset was cleaned and pre-processed through tokenization, label encoding, document and text cleaning, as well as TF- IDF vectorization. After pre-processing, the data was used to train and test the 4 machine-learning algorithms based on the commonly used performance evaluation metrics; the XGBoost algorithm performed the best with 91% accuracy. In comparison with other models from other studies, the proposed models show their capabilities by achieving similar or superior results. To enable more accurate and context-aware text analysis, future research in machine learning algorithms for text document categorization may concentrate on improving model interpretability and accuracy through the integration of sophisticated methods like transformers and deep learning architectures. Furthermore, investigating the use of transfer learning and unsupervised learning techniques might enhance the capacity of classification models to adjust to a variety of dynamic datasets, enabling more reliable and scalable text classification solutions in a range of fields.

REFERENCES

[1] Allam, H., Makubvure, L., Gyamfi, B., Graham, K.N., Akinwolere, K. (2024). Text classification: How machine learning is revolutionizing text categorization. https://doi.org/10.20944/preprints202412.1304.v1

[2] Guha, A., Samanta, D. (2020). Real-time application of document classification based on machine learning. In Proceedings of the First International Conference on Innovative Computing and Cutting-edge Technologies (ICICCT 2019), Istanbul, Turkey, pp. 366-379 https://doi.org/10.1007/978-3-030-38501-9_37

[3] Basarkar, A. (2017). Document classification using machine learning. Master's Project, 531. https://doi.org/10.31979/etd.6jmu-9xdt

[4] Kastrati, Z., Imran, A.S., Yayilgan, S.Y. (2019). The impact of deep learning on document classification using semantically rich representations. Information Processing & Management, 56(5): 1618-1632. https://doi.org/10.1016/j.ipm.2019.05.003

[5] Deng, X., Li, Y., Weng, J., Zhang, J. (2019). Feature selection for text classification: A review. Multimedia Tools and Applications, 78(3): 3797-3816. https://doi.org/10.1007/s11042-018-6083-5

[6] Bu, C., Zhang, Z. (2020). Research on overfitting problem and correction in machine learning. Journal of Physics: Conference Series, 1693(1): 012100. https://doi.org/10.1088/1742-6596/1693/1/012100

[7] Hachiya, H., Yoshida, H., Shimada, U., Ueda, N. (2024). Multi-class AUC maximization for imbalanced ordinal

multi-stage tropical cyclone intensity change forecast. Machine Learning with Applications, 17: 100569. https://doi.org/10.1016/j.mlwa.2024.100569

[8] Liu, Y., Loh, H. T., Sun, A. (2009). Imbalanced text classification: A term weighting approach. Expert Systems with Applications, 36(1): 690-701. https://doi.org/10.1016/j.eswa.2007.10.042

[9] Audebert, N., Herold, C., Slimani, K., Vidal, C. (2020). Multimodal deep networks for text and image-based document classification. In: Cellier, P., Driessens, K. (eds) Machine Learning and Knowledge Discovery in Databases, Springer, Cham. https://doi.org/10.1007/978-3-030-43823-4_35

[10] Ali, S.I.M., Nihad, M., Sharaf, H.M., Farouk, H. (2024). Machine learning for text document classification-efficient classification approach. IAES International Journal of Artificial Intelligence (IJ-AI), 13(1): 703-710. https://doi.org/10.11591/ijai.v13.i1.pp703-710

[11] Muaad, A.Y., Kumar, G.H., Hanumanthappa, J., Benifa, J.B., Mourya, M.N., Chola, C., Pramodha, M., Bhairava, R. (2022). An effective approach for Arabic document classification using machine learning. Global Transitions Proceedings, 3(1): 267-271. https://doi.org/10.1016/j.gltp.2022.03.003

[12] Schnawa, S.A., Rafie, M., Taha, M.S. (2024). DAE-DBN: An effective lung cancer detection model based on hybrid deep learning approaches. In Intelligent Systems, Intelligent Health Informatics, Intelligent Big Data Analytics and Smart Computing, Johor Bahru, Malaysia, pp. 108-118. https://doi.org/10.1007/978-3-031-59711-4_10

[13] Li, Y., Shawe-Taylor, J. (2006). Using KCCA for Japanese–English cross-language information retrieval and document classification. Journal of Intelligent Information Systems, 27: 117-133. https://doi.org/10.1007/s10844-006-1627-y

[14] Yang, S., Wei, R., Guo, J., Tan, H. (2020). Chinese semantic document classification based on strategies of semantic similarity computation and correlation analysis. Journal of Web Semantics, 63: 100578. https://doi.org/10.1016/j.websem.2020.100578

[15] Zhou, X., Gururajan, R., Li, Y., Venkataraman, R., Tao, X., Bargshady, G., Barua, P.D., Kondalsamy-Chennakesavan, S. (2020). A survey on text classification and its applications. Web Intelligence, 18(3): 205-216. https://doi.org/10.3233/WEB-200442

[16] Muhammad, B.A., Iqbal, R., James, A., Nkantah, D., Hla, N.N., Aung, T.M. (2020). Comparative performance of machine learning methods for text classification. In 2020 International Conference on Computing and Information Technology (ICCIT-1441), Tabuk, Saudi Arabia, pp. 1-5. https://doi.org/10.1109/ICCIT-144147971.2020.9213788

[17] Maw, M., Balakrishnan, V., Rana, O., Ravana, S.D. (2020). Trends and patterns of text classification techniques: A systematic mapping study. Malaysian Journal of Computer Science, 33(2): 102-117. https://doi.org/10.22452/mjcs.vol33no2.2

[18] Kim, D. (2022). Research on text classification based on deep neural network. International Journal of Communication Networks and Information Security, 14(1s): 100-113.

[19] Desai, M. (2023). An exploration of the effectiveness of machine learning algorithms for text classification. In 2023 2nd International Conference on Futuristic Technologies (INCOFT), Belagavi, Karnataka, India, pp. 1-6. https://doi.org/10.1109/INCOFT60753.2023.10425568

[20] Agarwal, A., Aravindan, M.K., Shrivastava, M., Sathya, S., Verma, D.A., Sohal, J. (2023). An exploration of the effectiveness of machine learning algorithms for text classification. In 2023 IEEE International Conference on Paradigm Shift in Information Technologies with Innovative Applications in Global Scenario (ICPSITIAGS), Indore, India, pp. 35-41. https://doi.org/10.1109/ICPSITIAGS59213.2023.10527545

[21] Köksal, Ö., Akgül, Ö. (2022). A comparative text classification study with deep learning-based algorithms. In 2022 9th International Conference on Electrical and Electronics Engineering (ICEEE), Alanya, Turkey, pp. 387-391. https://doi.org/10.1109/ICEEE55327.2022.9772587

[22] He, B., Yang, Y., Wang, L., Zhou, J. (2024). The text classification method based on BiLSTM and multi-scale CNN. Computer Life, 12(2): 43-49.

[23] Reusens, M., Stevens, A., Tonglet, J., De Smedt, J., Verbeke, W., Vanden Broucke, S., Baesens, B. (2024). Evaluating text classification: A benchmark study. Expert Systems with Applications, 254: 124302. https://doi.org/10.1016/j.eswa.2024.124302

[24] Gao, J., Liu, G., Zhu, B., Zhou, S., Zheng, H., Liao, X. (2025). Multi-level attention and contrastive learning for enhanced text classification with an optimized transformer. arXiv preprint arXiv:2501.13467. https://doi.org/10.48550/arXiv.2501.13467

[25] Rodríguez, J.M., Merlino, H.D., Pesado, P., García-Martínez, R. (2018). Evaluation of open information extraction methods using Reuters-21578 database. In Proceedings of the 2nd International Conference on Machine Learning and Soft Computing, pp. 87-92. https://doi.org/10.1145/3184066.3184099

[26] Khan, S.U.R. (2019). Identification of temporal specificity and focus time estimation in news documents. Doctoral dissertation, Capital University.

[27] Ben-Haim, Y., Tom-Tov, E. (2010). A streaming parallel decision tree algorithm. Journal of Machine Learning Research, 11(2): 849-872.

[28] Ramraj, S., Uzir, N., Sunil, R., Banerjee, S. (2016). Experimenting XGBoost algorithm for prediction and classification of different datasets. International Journal of Control Theory and Applications, 9(40): 651-662.

[29] [Jaiswal, J.K., Samikannu, R. (2017). Application of random forest algorithm on feature subset selection and classification and regression. In 2017 world congress on computing and communication technologies (WCCCT), Tiruchirappalli, India, pp. 65-68. https://doi.org/10.1109/WCCCT.2016.25

[30] García, V., Mollineda, R.A., Sánchez, J.S. (2008). On the k-NN performance in a challenging scenario of imbalance and overlapping. Pattern Analysis and Applications, 11: 269-280. https://doi.org/10.1007/s10044-007-0087-5

[31] Luo, X. (2021). Efficient English text classification using selected machine learning techniques. Alexandria Engineering Journal, 60(3): 3401-3409. https://doi.org/10.1016/j.aej.2021.02.009

[32] Hanoon, F.S., Çevik, M., Taha, M.S. (2024). A fault

classification for defective solar cells in electroluminescence imagery based on deep learning approach. AIP Conference Proceedings, 3097(1): 050006. https://doi.org/10.1063/5.0209371

[33] Ali, N.S., Alsafo, A.F., Ali, H.D., Taha, M.S. (2024). An effective face detection and recognition model based on improved YOLO v3 and VGG 16 networks. International Journal of Computational Methods and Experimental Measurements, 12(2): 107-119. https://doi.org/10.18280/ijcmem.120201

[34] Naser, Z.S., Khalid, H.N., Ahmed, A.S., Taha, M.S., Hashim, M.M. (2023). Artificial neural network-based fingerprint classification and recognition. Revue d'Intelligence Artificielle, 37(1): 129-137. https://doi.org/10.18280/ria.370116

[35] Zhao, D., He, J., Liu, J. (2014). An improved LDA algorithm for text classification. In 2014 International Conference on Information Science, Electronics and Electrical Engineering, Sapporo, Japan, pp. 217-221. https://doi.org/10.1109/InfoSEEE.2014.6948100

[36] Adhikari, A., Ram, A., Tang, R., Lin, J. (2019). Rethinking complex neural network architectures for document classification. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4046-4051. https://doi.org/10.18653/v1/N19-1408