




Article

Twenty Years of Machine-Learning-Based Text Classification: A Systematic Review

Ashokkumar Palanivayagam¹ , Claude Ziad El-Bayeh²  and Robertas Damaševičius^{3,*} 

¹ Sri Ramachandra Faculty of Engineering and Technology, Sri Ramachandra Institute of Higher Education and Research, Chennai 600116, India

² Department of Electrical Engineering, Bayeh Institute, Amchit 4307, Lebanon

³ Department of Software Engineering, Kaunas University of Technology, 44249 Kaunas, Lithuania

* Correspondence: robertas.damasevicius@ktu.lt

Abstract: Machine-learning-based text classification is one of the leading research areas and has a wide range of applications, which include spam detection, hate speech identification, reviews, rating summarization, sentiment analysis, and topic modelling. Widely used machine-learning-based research differs in terms of the datasets, training methods, performance evaluation, and comparison methods used. In this paper, we surveyed 224 papers published between 2003 and 2022 that employed machine learning for text classification. The Preferred Reporting Items for Systematic Reviews (PRISMA) statement is used as the guidelines for the systematic review process. The comprehensive differences in the literature are analyzed in terms of six aspects: datasets, machine learning models, best accuracy, performance evaluation metrics, training and testing splitting methods, and comparisons among machine learning models. Furthermore, we highlight the limitations and research gaps in the literature. Although the research works included in the survey perform well in terms of text classification, improvement is required in many areas. We believe that this survey paper will be useful for researchers in the field of text classification.

Keywords: machine learning; text classification; natural language processing; spam detection; sentiment analysis; rating summarization



Citation: Palanivayagam, A.; El-Bayeh, C.Z.; Damaševičius, R. Twenty Years of Machine-Learning-Based Text Classification: A Systematic Review. *Algorithms* **2023**, *16*, 236. <https://doi.org/10.3390/a16050236>

Academic Editor: Frank Werner

Received: 30 December 2022

Revised: 19 April 2023

Accepted: 27 April 2023

Published: 29 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Machine learning models provide the best alternative to traditional methods in the field of text classification. Text-classification-based research has been conducted extensively in recent years to improve the performance of machine learning models [1]. Text classification can be done manually, but it is time-consuming and has a high cost. Manual classification comes with lots of errors and is less accurate because of human error and a lack of domain knowledge understanding. There was a huge revolution in text classification when machine learning models, such as the Support Vector Machines (SVM), Naive Bayes (NB), and Random Forest (RF) started to replace manual work, because these models not only reduce the time and cost but also are highly accurate for classification [2]. Since the inception of machine learning models, numerous studies have been conducted to enhance, optimize, and refine the text classification process.

With the recent increase in demand for various Natural Language Processing (NLP) technologies, such as chatbots [3], content classification [4], Sentiment Analysis [5–7], hate speech detection [8,9], authorship recognition and attribution [10], product and service recommenders [11,12], text summarization [13,14], email spam detection [15] and phishing detection [16], intent detection [17], and search optimization [18], ML models have presented a huge advantage and have created many opportunities for researchers in the field of text classification.

Text classification is done in four stages: (a) pre-processing, (b) text representation, (c) feature selection, and finally, (d) classification [19]. The first stage is pre-processing in

which the input is cleaned and shaped according to the need for classification. The noise present in the input is removed [20]. Next, the input text is converted into a format, such as bag of words or n-gram. Feature selection is an optional step that involves identifying and picking important features. The size of the input is largely reduced at this stage [21].

Machine learning models have been applied to text classification and have produced promising results in various fields, such as finance [22], tourism [23], healthcare [24], and online news analysis [25]. This paper reviews how, when, and where these models have been successful in text classification. We studied the pros and cons of many models separately and determined how they perform in each dataset. We considered papers from 2003, and the findings of the paper will help the research community to improve the classification in the future. Furthermore, we present a few gaps noticed in the literature along with a summary of some interesting future works in the text classification domain.

Previous survey papers on text classification [2,26–31] discussed various aspects of text classification, such as feature extraction techniques, algorithms, evaluation methods, and limitations. They also provided an overview of deep-learning-based text classification models, popular datasets, future research directions, and a comparison of different methodologies. The papers also highlighted the strengths and limitations of traditional text classification methods and suggested directions for future work.

Thangaraj et al. [26] examined articles on text classification techniques in Artificial Intelligence (AI) written between 2010 and 2017 and grouped the techniques according to the algorithms involved. The results were visualized as a tree structure to show the relationship between learning procedures and algorithms, and the paper identified the strengths, limitations, and current research trends in text classification.

Mironczuk et al. [31] presented an overview of the state-of-the-art of text classification by identifying and studying key and recent studies and objectives in this discipline. The paper covered six fundamental parts of text classification and analyzed the connected works qualitatively and quantitatively.

Kowsari et al. [2] presented a brief discussion of various text feature extraction techniques, dimensionality reduction methods, existing algorithms, and evaluation methods. The limitations of each technique were also discussed, along with their applications in real-world problems.

Wu et al. [30] examined text categorization models, such as the Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Attention Mechanisms, and others. The paper summarizes the shortcomings of traditional text classification methods and introduces the deep-learning-based text classification process.

Minaee et al. [27] conducted a comprehensive review of more than 150 deep-learning-based text classification models and discussed their technical contributions, similarities, and strengths. The paper also provides a summary of more than 40 popular text classification datasets and discusses future research directions.

Bayer et al. [28] surveyed data augmentation methods for textual classification and categorized more than 100 methods into 12 different groupings based on a taxonomy. The paper provides cutting-edge references and highlights promising methods as well as providing research perspectives for future work.

Li et al. [29] covered state-of-the-art approaches in text classification from 1961 to 2021, with an emphasis on models ranging from classical to deep learning. The paper developed a text classification taxonomy and provided a comparison of different methodologies, outlining their benefits and drawbacks. It also summarized major implications, prospective research objectives, and obstacles in the studied area.

Our work is different from previously discussed studies. It focuses on machine-learning-based text classification. Our study conducted a systematic review of 224 papers published between 2003 and 2022 that employ machine learning for text classification. This paper analyzes the differences in the literature in terms of six aspects: datasets, machine learning models, best accuracy, performance evaluation metrics, training and testing splitting methods, and comparisons among machine learning models. While other

studies provide more specific perspectives on text classification techniques, such as deep learning models, data augmentation methods, and a comparison of classical to deep learning models, this article provides a broader and more comprehensive view of the field of machine-learning-based text classification.

The main contributions of this paper are to answer the following research questions:

1. What is the most frequently used dataset for machine-learning-based text classification?
2. What are the frequencies at which machine learning models are used?
3. What is the maximum accuracy for each dataset?
4. What is the most frequently used performance evaluation metric?
5. What is the most successful train–test split method?
6. How do different machine learning models compare?

The paper is organized as follows: Section 3 presents a summary of all of the surveyed papers. Section 4 discusses the solutions proposed by various researchers for some of the common problems in text classification. Section 5 contains our notable observations on text classification. Lastly, the conclusion is discussed in Section 6.

2. Survey Methodology

To conduct this review, we used the Preferred Reporting Items for Systematic Reviews (PRISMA) [32]. Systematic reviews frequently demonstrate a lack of understanding about shared principles that allow them to be replicated and scientifically sufficient. PRISMA is a standard peer-reviewed approach that employs a guideline checklist, which was closely followed in this manuscript, to contribute to the quality assurance and replicability of the revision process. A review protocol detailing the article selection criteria, search strategy, data extraction, and data analysis techniques was created.

In this review, only peer-reviewed research papers published in 2003–2022 and written in English were considered. Only applications of classification using textual data were considered. The research papers were identified through the keywords mentioned in Table 1 via sources like IEEE, Science Direct, Springer, ACM Digital Library, MDPI, and Hindawi. Initial screening was done at the title and abstract levels. Later, full text was retrieved, and works including studies related to text-based classification alone were included in this survey. Figure 1 presents the selection process for papers in this survey. The entire paper list and other data are attached in the appendix. The numbers of papers considered in this survey from 2003 to 2022 are depicted in Figure 2.

Table 1. Search keywords used for our review paper.

Search Keywords Used		
Machine-learning-based classification	text classification	text mining
text analysis	text categorization	document classification
sentiment analysis	natural language processing	Data mining

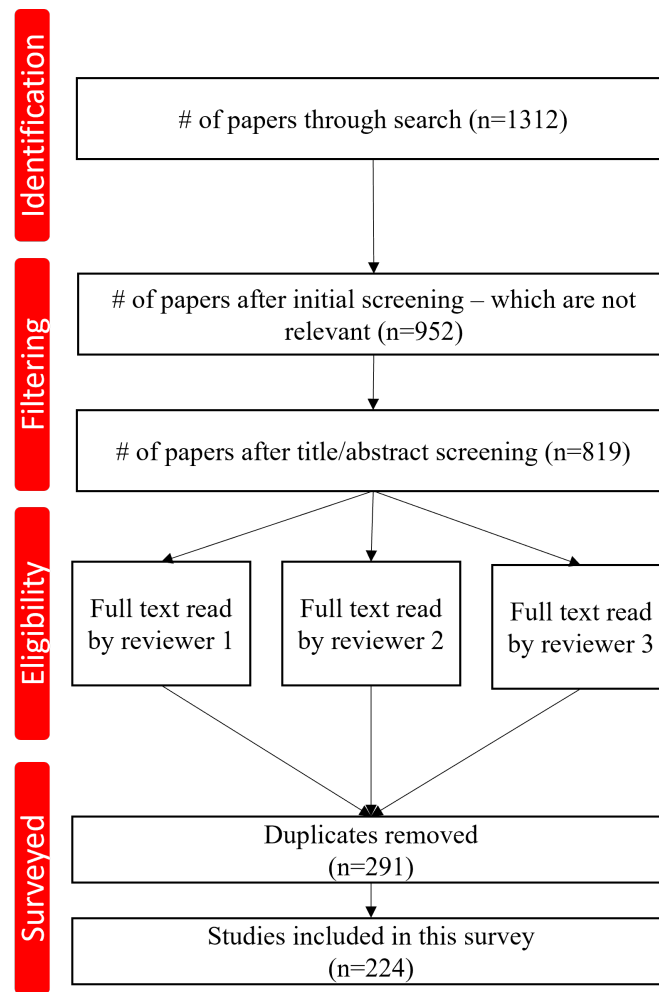


Figure 1. The criteria used for selecting papers in this survey.

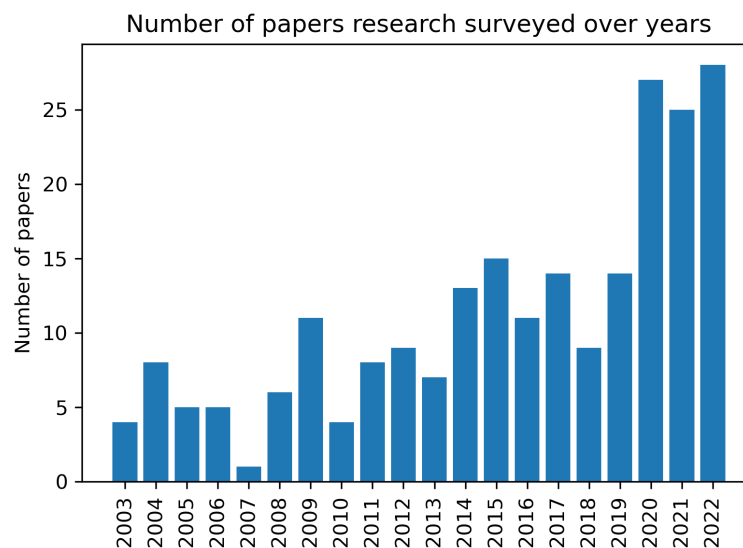


Figure 2. Number of papers surveyed in each year.

3. Overview of the Survey Results

To answer the previous 6 questions presented in Section 1, we summarize and categorize the literature considering many factors that are also presented in this section.

3.1. Study on the Dataset

In the surveyed papers, a comprehensive set of 56 distinct datasets was employed with the top three most commonly utilized being 20Newsgroup, Reuters, and Webkb, each being employed in a significant number of studies. Some datasets such as PAN-12, Tigrinya, and Emotion616 were rarely used. A total of 17.85% of the surveyed datasets used binary classification. Table 2 summarizes the frequency of the top 10 datasets used by various papers in the survey. Figure 3 displays the frequency of dataset use in the literature.

Percentage of top 10 datasets used in the literature

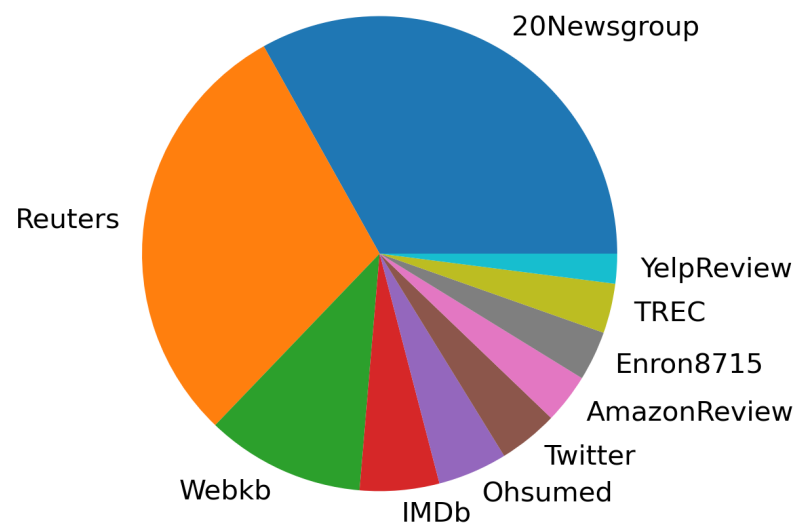


Figure 3. Frequency of dataset use in the literature.

Table 2. Top 10 datasets used in text classification studies.

Dataset	#	Model Wise Count							
		ANN	CNN	DT	kNN	LR	NB	RF	SVM
20Newsgroup	49	1	4	5	17	2	23	5	29
Reuters	44	2	4	4	20	2	17	6	27
Webkb	16	-	-	-	9	-	7	-	10
IMDb	8	1	1	-	1	1	3	1	5
Ohsumed	7	-	-	-	4	1	2	1	4
Twitter	6	1	2	3	2	2	3	3	4
AmazonReview	5	-	-	-	2	1	1	-	2
Enron8715	5	2	-	2	1	-	2	2	4
TREC	5	-	1	-	1	-	2	-	4
YelpReview	3	-	-	-	-	1	2	1	2

3.2. Study on Machine Learning Models

We found that SVM is the most frequently used machine learning model for text classification and was presented in 118 papers. NB and kNN are the next popular models

for text classification. The maximum accuracy of 98.88 was obtained by SVM in the 20Newsgroup dataset. The frequency of each machine learning model along with the maximum accuracy obtained are presented in Table 3. Figure 4 displays the best accuracy level for each dataset, and Figure 5 shows the frequency of machine learning model use in the literature.

Table 3. Frequency of model use in various papers in this area of literature across various year ranges.

Model	# of Papers	Max Accuracy (%)	Dataset	Reference
SVM	118	98.88	20Newsgroup	[33]
NB	92	97.89	Reuters	[34]
kNN	66	96.64	Reuters	[34]
RF	42	92.60	Gold	[35]
DT	34	94.50	20Newsgroup	[36]
CNN	27	98	Twitter	[37]
LR	23	98.50	PubMed	[38]
LSTM	16	96.54	Arabiya	[39]
ANN	10	86	RusPersonality	[40]
LDA	10	96.20	Reuters	[41]
Ada Boost	9	97	Twitter	[37]
RNN	6	90.65	ChineseMicroblogg	[42]
C4.5	6	74	20Newsgroup	[43]
LSA	6	96.50	Reuters	[41]
BILSTM	5	98	Twitter	[37]
GRU	4	96.76	Arabiya	[39]
XGBoost	4	98	Twitter	[37]
CNN-BiLSTM	3	81.90	THUCNews	[44]
CRF	1	98.75	PubMed	[38]

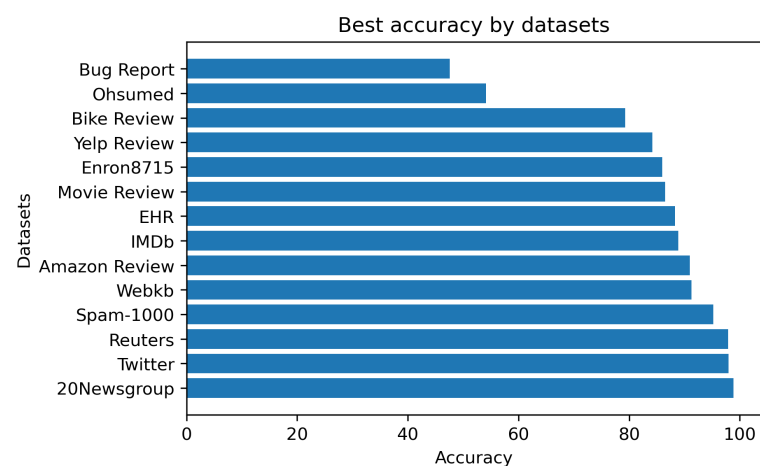


Figure 4. Best accuracy level for each dataset.

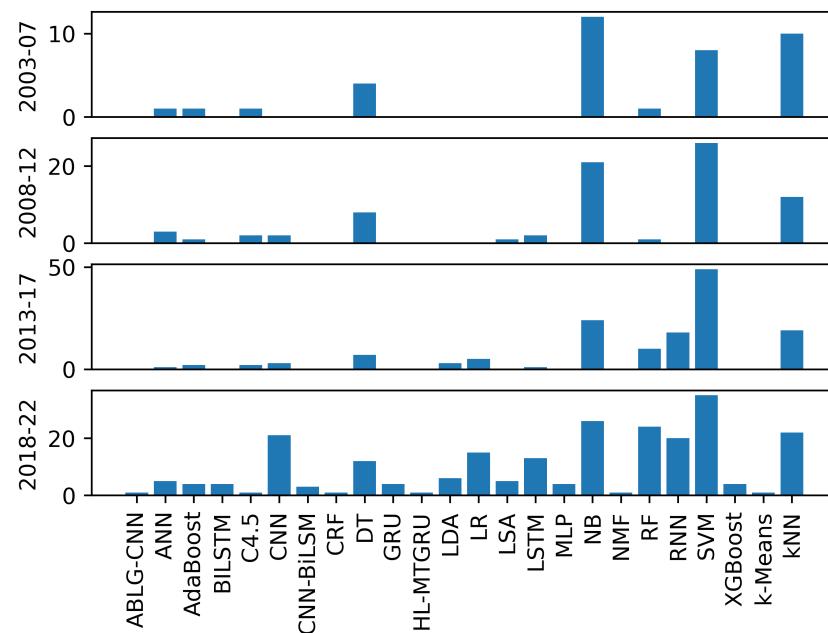


Figure 5. Frequency of machine learning model use in the literature.

3.3. Study on Accuracy

Accuracy is one of the most important factors used to evaluate the performance of a machine learning model. A highly accurate value indicates that the model has perfectly learned the relationships among the input samples, and it is ready for classifying future values. Table 4 lists a few popular datasets and their maximum accuracy levels.

Table 4. Maximum accuracy levels obtained from top datasets.

Dataset	Accuracy (%)	Algorithm	Train-Test	Reference
Enron8715	86	ANN	70–30%	[45]
Bug Report	47.60	RNN	10 Fold	[46]
EHR	88.30	RF	90–10%	[47]
Yelp Review	84.20	SVM	10 Fold	[48]
Twitter	98	CNN	10 Fold	[37]
Bike Review	79.25	RF	10 Fold	[49]
20Newsgroup	98.88	SVM	10 Fold	[33]
Amazon Review	91	LSA	60–40%	[41]
IMDb	88.87	LR	50–50%	[50]
Ohsumed	54.10	LDA	10 Fold	[51]
Movie Review	86.50	SVM	5 Fold	[52]
Spam-1000	95.20	SVM	60–40%	[53]
Reuters	97.89	NB	5 Fold	[34]
Webkb	91.30	SVM	5 Fold	[34]

3.4. Study on Performance Evaluation

Many performance evaluation metrics can be used to validate the efficiency of a machine learning model. These metrics can also be used to measure the correctness of the training process of a machine learning model. Most papers use multiple performance evaluation metrics to validate the model. Twenty-four unique metrics were used in the surveyed papers, including ROC, the Jaccard Similarity Score, and RMSE. Table 5 displays the frequencies of various performance metrics.

Accuracy is the standard measure for classification; however, if the dataset is skewed, accuracy can be misleading [54]. For an imbalanced dataset, F1 is the best metric to use [55]. If deep learning models are used and if precision seems to be dropping, macro-F1 can be used as an alternate [56].

Table 5. Various performance metrics used by the survey papers.

Parameter	Count
Accuracy	107
F1	99
Precision	64
Recall	55
Execution time	15
AUC	5
Kappa Coefficient	4
FP Rate	4
Sensitivity	4
Specificity	3
Classification error	3
Hamming Loss	3
Variance	2
TP Rate	2
FN Rate	1
MAE	1
Ranking loss	1
AULC	1
ROC	1
Jaccard Similarity Score	1
Reliability	1
ARI	1
RMSE	1

There are 42 unique combinations of metrics. Table 6 presents the top 10 combinations and their respective frequencies.

Table 6. Top 10 combinations of performance metrics used in the survey.

Validation Combination	No. of Papers
Accuracy	56
F1	37
F1, Precision, Recall	26
Accuracy, F1, Precision, Recall	12
Accuracy, Execution time	8
Accuracy, F1	8
Precision	6
Accuracy, Precision, Recall	3
Precision, Recall	3
Accuracy, Recall	2
Classification error	2

3.5. Study on Train–Test Splits

Each supervised machine learning model has two stages: training and testing. A model is first given the training set to learn all the relationships among the input samples. Once the training has been completed, a new set known as a testing set is then fed into the classifier. This time, the classifier uses the previously learned knowledge to predict and validate the input samples present in the testing set. All performance evaluation metrics that were discussed in the previous subsection were used in this stage. Figure 6 displays the frequency of the train–test splits. 10 Fold validation is the most widely used train–test split method (count = 45), followed by 5-Fold (count = 29).

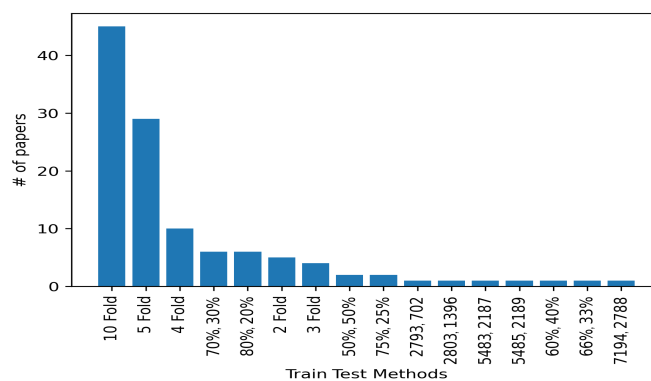


Figure 6. Frequency of Train–Test splits used by various papers.

3.6. Study on Machine Learning Algorithms

In this subsection, we present a comparison of different machine learning algorithms. **Definitions of Positive and Negative Accuracies:** In a paper, if there is a comparison between two machine learning models X and Y with accuracy levels a_1 and a_2 , if $a_1 > a_2$, then X has a positive accuracy compared to Y and Y has a negative accuracy compared to X.

3.6.1. Support Vector Machine (SVM)

SVM is the most commonly used model in the literature. SVM is both a linear and a nonlinear classifier that can perform well, especially in multilabel scenarios [57]. The core of SVM is the kernel. Choosing and optimizing the correct kernel will increase the accuracy of the classification [58]. Figure 7 shows the number of papers in which SVM performs

better than other algorithms (positive accuracies) and the number of papers in which other algorithms perform better than SVM (negative accuracies).

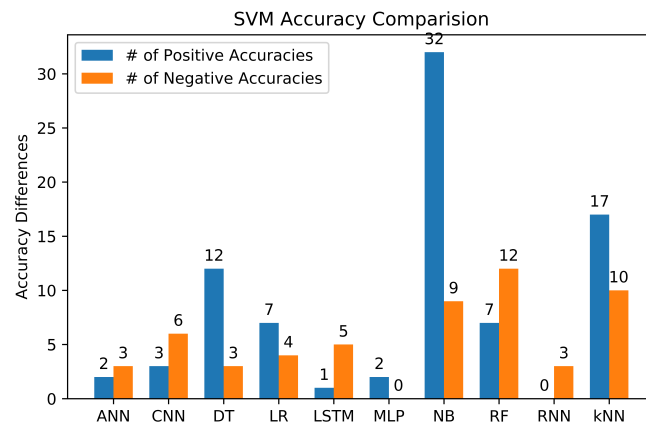


Figure 7. Accuracy differences by SVM while compared to other models.

The main advantages of SVM can be summarized as

- (a) It is more accurate than other classifiers;
- (b) It works well with nonlinear distributions;
- (c) The overfitting chances are very low.

The limitations of SVM are

- (a) Choosing the correct kernel is challenging;
- (b) There is a long training time;
- (c) It occupies more memory.

Table 7 displays the top 5 SVM accuracy levels in our survey.

Table 7. Top Performance Levels of the SVM Classifier.

Dataset	Accuracy (%)	Other Model Performance (%)	Reference
20Newsgroup	98.88	NB (95.52)	[46]
Reuters	97.60	NB (97.89), kNN (96.64)	[34]
Spam-1000	95.20	NB (92.70), ANN (85.30)	[53]
Reuters	95.10	Traditional SVM (82.76)	[59]
Reuters	93	kNN (53), NB (24)	[60]

3.6.2. Naive Bayes (NB)

NB is a probabilistic classification model based on the Bayes theorem. Ninety-two papers in our survey used NB to perform the classification. There are lots of variations of NB, such as multinomial NB, Bernoulli NB, and Gaussian NB. For text classification, multinomial NB is widely used. Figure 8 compares NB with other algorithms.

The advantages of NB are

- (a) If the training set is very small, then NB can produce a good performance;
- (b) NB can be used for multi-class classification;
- (c) It does not require much training data.

The drawbacks of this algorithm are

- (a) It is not suited for small datasets [61];
- (b) If the features are not independent, NB is not the best choice for classification.

Table 8 presents the top 5 best NB accuracy levels from our survey.

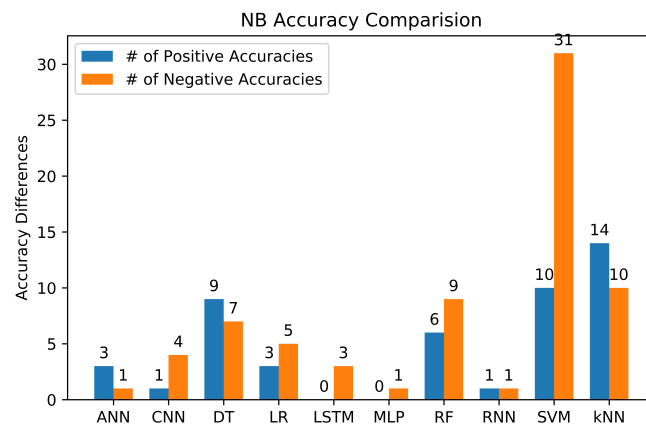


Figure 8. Accuracy differences between NB and other models.

Table 8. Top Performances of the NB Classifier.

Dataset	Accuracy (%)	Other Model Performance (%)	Reference
Reuters	97.89	SVM (97.60), kNN (96.64)	[34]
20Newsgroup	95.52	SVM (98.88)	[33]
Spam-1000	92.70	SVM (95.20), ANN (85.30)	[53]
Counter	91.40	LR (90.30), DT (89.50), kNN (91), SVM(71.70)	[62]
Chinese Microblogg	90.65	CNN (97.60), SVM (90.60), LR (90.50), DT (86.91), RF (86.75)	[42]

3.6.3. k Nearest Neighbor (kNN)

kNN is a good classifier that can predict the class of an instance based on its nearest neighbors. The value of k determines the number of nearest neighbors to consider. The value of k is chosen to be an odd number to avoid race conditions [63]. kNN can be a good model to remove all of the extreme values [64]. Figure 9 displays the comparison of kNN with other machine learning models.

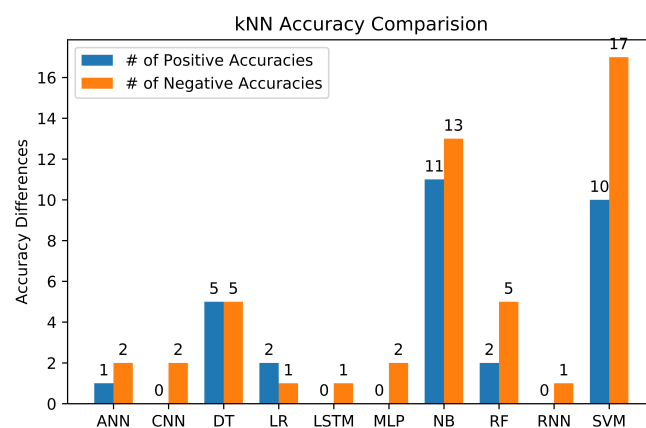


Figure 9. Accuracy differences between kNN and other models.

The main advantage of kNN is the simplicity of its implementation, and the limitations of kNN are

- (a) It needs more storage space;
- (b) It is highly sensitive to errors.

Table 9 displays the top 5 kNN accuracy levels in our survey.

Table 9. Top Performances of the kNN Classifier.

Dataset	Accuracy (%)	Other Model Performance (%)	Reference
Reuters	96.64	NB (97.89), SVM (97.60)	[34]
Reuters	92.55	SVM (81.48)	[65]
Counter	91	NB (91.40), LR (90.30), DT (89.50), SVM(71.70)	[62]
Webkb	84.07	SVM (91.30), NB (85.67)	[34]
20Newsgroup	82	SVM (84), NB (83)	[66]

3.6.4. Decision Tree (DT)

The Decision Tree classifier works by constructing a tree-like structure consisting of various branches and classifying the samples by passing through the branches. Both numerical and categorical data can be used in the DT. Figure 10 shows compares the DT with other algorithms.

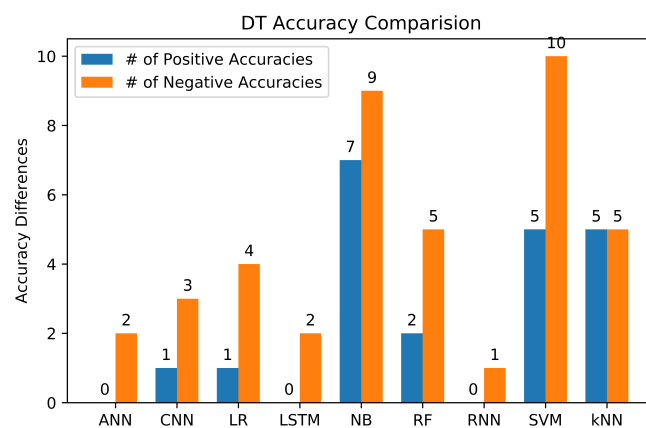


Figure 10. Accuracy differences between the DT and other models.

The advantage of DT is that it can easily process high-dimensional data. The limitation is that the DT is not a stable classifier. Table 10 shows the top 5 DT accuracy levels.

Table 10. Top Performances of the DT Classifier.

Dataset	Accuracy (%)	Other Model Performance (%)	Reference
20Newsgroup	94.50	-	[36]
Twitter	94	CNN (98), SVM (98), LR (97), NB (86)	[37]
Counter	89.50	NB (91.40), kNN (91), LR (90.30), SVM(71.70)	[62]
Chinese Microblogg	86.91	CNN (97.60), NB (90.65), SVM (90.60), LR (90.50), RF (86.75)	[42]
20Newsgroup	85.39	SVM (85.88)	[67]

3.6.5. Random Forest (RF)

RF is an ensemble of multiple DTs. The final classification result is decided by a majority vote by all DTs. RF can efficiently manage thousands of features. Figure 11 presents an accuracy comparison of RF with other models.

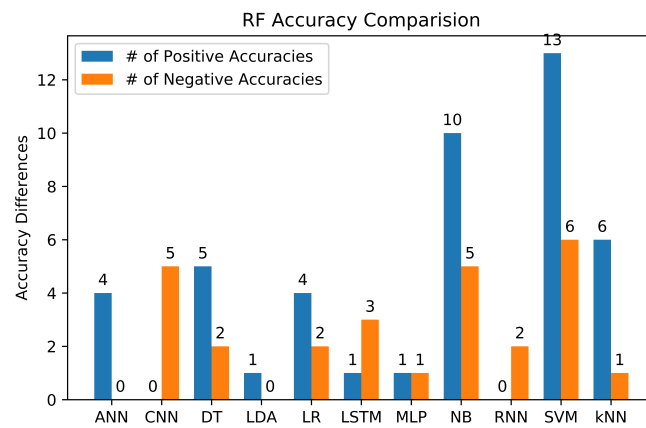


Figure 11. Accuracy differences between RF and other models.

One of the main reasons for choosing RF is that it can give good accuracy results with nonlinear data. The drawback of this model is that overfitting can easily occur. Table 11 presents the top 5 RF accuracy levels in our survey.

Table 11. Top Performances of the RF Classifier.

Dataset	Accuracy (%)	Other Model Performance (%)	Reference
Gold	92.60	LSTM (95.90), SVM (91.90), LR (70.70)	[35]
Chinese Microblogg	86.75	CNN (97.60), NB (90.65), SVM (90.60), LR (90.50), DT (86.91)	[42]
Blogger	85	DT (77)	[68]
Bike Review	79.25	NB (79.05)	[49]
IMDb	72.80	NB (77.50), ANN (72), SVM (68.30), kNN (63)	[69]

3.6.6. Logistic Regression (LR)

Logistic Regression is a statistical method for performing a binary classification. However, LR can also be extended for multiclass classifications. The requirement for LR is that the input samples should be linear. Figure 12 displays an accuracy comparison of LR with other machine learning models.

The advantage of LR is that it can perform well on smaller datasets. The drawback is that it cannot classify continuous variables. Table 12 displays the top 5 best LR accuracy levels in our survey.

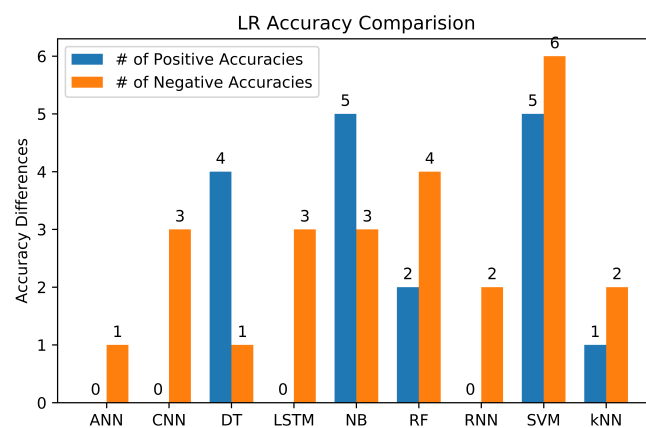


Figure 12. Accuracy differences between LR and other models.

Table 12. Top Performances of the LR Classifier.

Dataset	Accuracy (%)	Other Model Performance (%)	Reference
PAN-12	98.50	LSTM (98)	[70]
Twitter	97	CNN (98), SVM (98), DT (94), NB (86)	[37]
Chinese Microblogg	90.50	CNN (97.60), NB (90.65), SVM (90.60), DT (86.91), RF (86.75)	[42]
Counter	90.30	NB (91.40), kNN (91), DT (89.50), SVM(71.70)	[62]
Amazon Review	89.21	Others (88)	[50]

3.6.7. Summary of Machine Learning Classifiers

Finally, the performances of machine learning classifiers (DT, LR, NB, RF, SVM, kNN) on each dataset (20Newsgroup, Amazon Review, Bike Review, Blogger, Chinese Microblogg, Counter, Gold, IMDb, PAN-12, Reuters, Spam-1000, Twitter, Webkb) are summarized in Figure 13. The best and most stable performance was achieved by SVM, while RF generally had the worst and most unstable (i.e., the largest variance) performance.

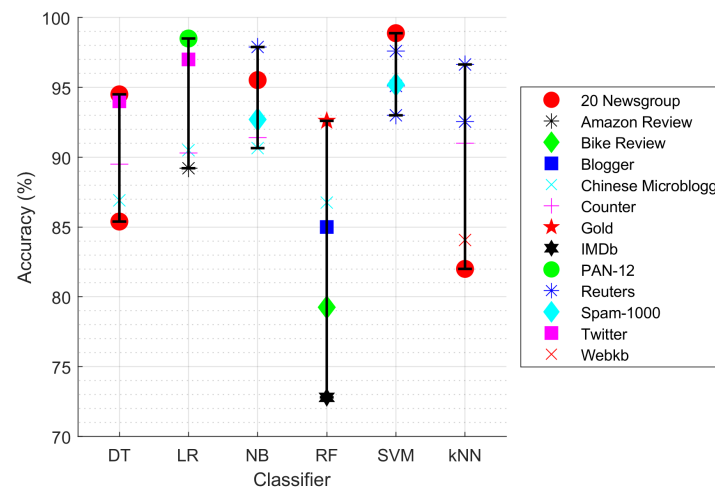


Figure 13. Summary of machine learning classifier performances achieved on various datasets.

3.7. Deep-Learning-Based Models for Text Classification

Most of the machine learning-based models on text classifications rely on bag of words or term frequencies. However, there are many problems with these approaches, such as similarity (in terms of semantics), scalability, and ambiguity. Similarity is caused when two words with the same meaning are represented in two different ways, e.g., the words ‘biscuit’ and ‘cookie’ often have the same meaning, but the representation is different if the bag of words model is used. Similarly, the bag of words model represents each unique word by one vector; this causes exponential growth of unique vectors. Thus, it is not scalable. The final and most important problem is ambiguity. This is caused by the order of words, e.g., “The food is ready” and “Is the food ready” are represented the same way because of the unique words employed. The above-mentioned problems can be solved using deep-learning-based text representation methods, such as word embedding. Word embedding considers the semantics of the words, so different words with the same meaning are represented as the same vector or a similar vector. In word embedding, each word is mapped to an N-dimensional vector. Table 13 shows some examples of the most popular text representations across the years.

Recently, deep learning models have been used to achieve very good results on various text classification problems [27,30]. Deep learning models, such as ANN, CNN, RNN, and LSTM seem to have better accuracy levels than machine learning algorithms for solving multiple NLP subproblems, such as part-of-speech (POS) tagging [71]. There are many

advantages of using deep learning over machine learning, such as handling noisy data, high accuracy, and better identification of the relationship between the input and output features. The few drawbacks of deep learning-based classification are overfitting and time consumption. Figures 14–17 show an accuracy comparison of ANN, CNN, RNN, and LSTM with other models. Word embedding is the most fundamental step in deep-learning-based text classification.

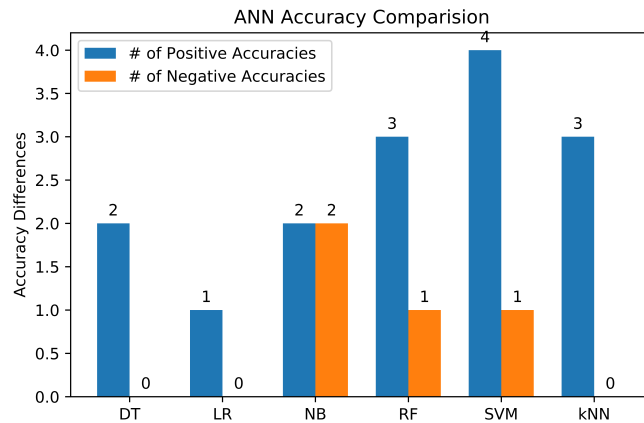


Figure 14. Accuracy differences between ANN and other models.

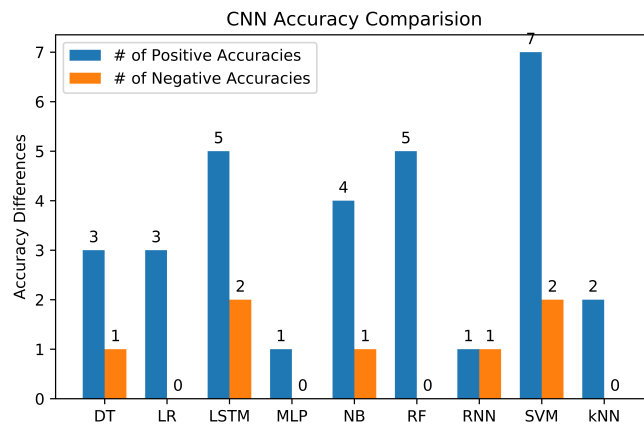


Figure 15. Accuracy differences between CNN and other models.

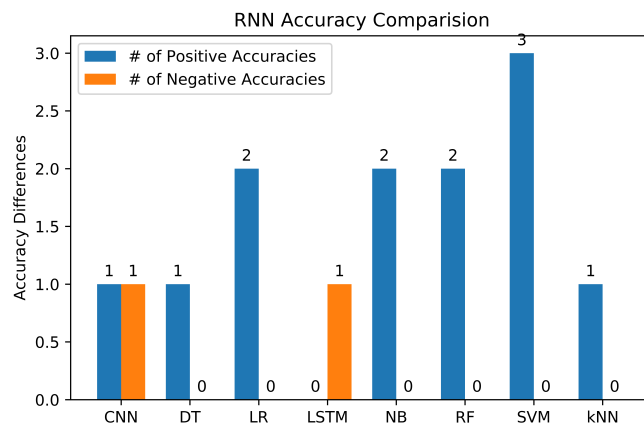


Figure 16. Accuracy differences between RNN and other models.

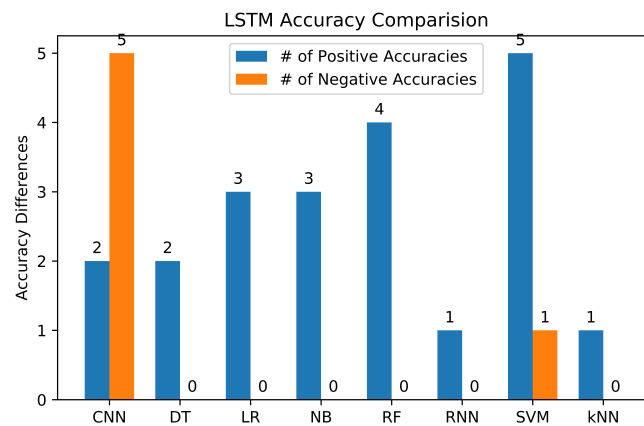


Figure 17. Accuracy differences between LSTM and other models.

Table 13. Popular Text Representation Methods.

Year	Popular Text Representation Method
2013	Word2Vec
2014	Doc2Vec
2015	Character Embedding
2016	Subword Embedding
2017	FastText
2018	Transformer
2019	BERT
2020	ALBERT
2021	GPT
2022	GPT

3.7.1. Word2Vec

Word2vec [72] is one of the most popular text representing methods that predicts the probability of word distribution based on neighboring words. It consists of two deep learning architectures known as the continuous bag of words (CBOW) and the skip-gram model. CBOW predicts the probability of neighboring words based on the center word. Skip-gram predicts the probability of the center word based on the neighboring words. Figure 18 shows the architecture of the word2vec model. Equation (1) displays how the probability estimation of a word is done based on its neighboring words. Equation (2) shows the representation of Word2Vec.

$$f(x) = \frac{1}{T} \sum_{t=k}^{T-k} \log p(w_t | w_{t-k}, \dots, w_{t+k}) \quad (1)$$

$$y = Uh(w_{t-k}, \dots, w_{t+k}; W) + b \quad (2)$$

3.7.2. Doc2Vec

Doc2vec [73] is an extension of word2vec in which the semantic relationship can be expressed across a large number of words (paragraphs, documents). Each document is represented as a unique vector. Equation (3) shows the representation of Word2Vec. Figure 19 displays the architecture of the Doc2Vec model.

$$y = Uh(w_{t-k}, \dots, w_{t+k}; WD) + b \quad (3)$$

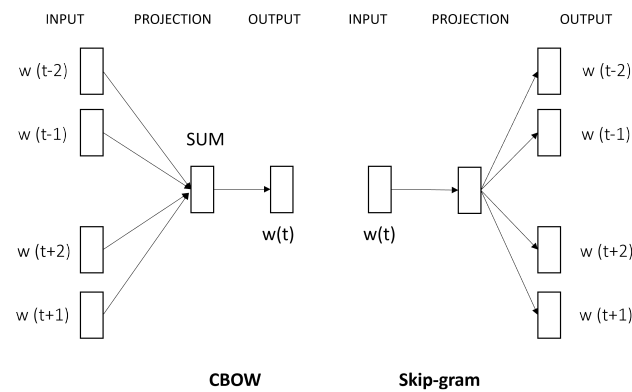


Figure 18. The Word2Vec Model.

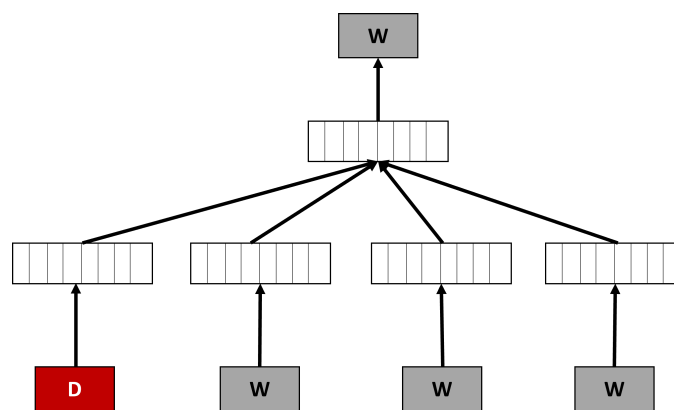


Figure 19. The Doc2Vec Model.

3.7.3. FastText

FastText [74] improves the performance of the word2vec model by considering subwords. The final vector of a word w is the sum of all the vectors of subwords in w . FastText can create a new vector when the target word is not present in the training set. The architecture of FastText is shown in Figure 20.

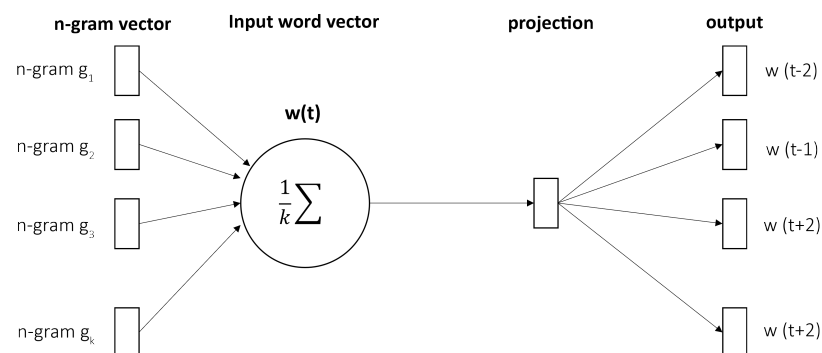


Figure 20. The FastText Model.

3.7.4. Transformers

The RNN model outperforms other machine learning models in extracting relationships across sequential text; however, the computational cost of running an RNN is very huge. This problem can be overcome by using parallel processing. Transformers use pre-trained models to run each word in parallel and thus reduce the computational complexity. Deep learning methods provide various ways to overcome the problems in machine learning models. For example, in [75], the inverse document frequency was used to prevent the

semantic problem. Creating multiple vectors, one for each unique meaning of a word, is also one solution, as proposed by [76]. The bidirectional language model is used to adapt pre-trained models and their knowledge to classify multi-meaning input texts as proposed by [77]. Online texts have been tagged using fuzzy logic in [78]. Deep-learning-based models involve the development of complex nested and deep architectures, which naturally increases the computation time. In [79], novel work was proposed to decrease the execution time. Deep learning is also used in other problems, such as hierarchical classification, big data classification [80], and malware analysis [81].

4. Problems in Text Classification

This section explains some of the common problems faced by researchers in the field of text classification along with a few solutions proposed by the papers in the survey. Table 14 presents the most frequently used considerations to improve the accuracy of text clarifications using machine learning algorithms.

Table 14. Considerations for improving the accuracy of text classification using traditional machine learning.

Considerations for Improving the Accuracy of Text Clarifications	References
Frequency of features	
Initial letter	
Paragraph	[82]
Question mark	
Full stop	
Term weighting	[57]
Term embedding	[83]
Optimization	[84]
NLP	[54]
Feature fusion	[85]
Ensemble approach	[33,85,86]
Feedback systems	[87]
Feedback from experts	[88]
Selection of sub-features	[89]

4.1. Increased Accuracy

Traditional machine learning algorithms can be used for text classification. Despite their better accuracy, there is lots of room for improvement. In this subsection, we present a few research works on improving accuracy.

In [82], a few improvements in the pre-processing stage are identified by considering the frequency of features, the initial letter, paragraphs, question marks, and full stops. The frequency of features can be assessed using term weighting or embedding. Reference [57] shows that term weighting can iterate over each word and has an upper hand over embedding techniques. However, few deep learning models, such as that presented in ref. [83], improve the accuracy of classification by using embedding. Some optimizations can be done to avoid the consideration of all features. Instead, the classifier can jump from one location to another [84].

Many works concentrate on integrating deep learning and machine learning models to increase the accuracy of text classification. Article [48] explains the use of NLP to increase the accuracy. In [54], resampling is performed along with NLP to increase the accuracy. When the number of features is large, feature fusion can be done to combine multiple features to obtain one or more prominent features. Feature fusion can outperform the classification model performance, as stated by [85]. As each machine learning model has

its advantages and disadvantages, an ensemble approach can be implemented to increase the accuracy of a classification task [33,85,86]. Feedback systems can also improve the performance [87]. Sometimes, the accuracy of the classifier can be increased in external ways, for example, domain experts can be involved in providing feedback during the training process [88]. A few studies have shown [89] that the selection of subfeatures can also have a positive impact on the classification.

4.2. Feature Selection

The statistical process of reducing the number of input features in the classification is known as feature selection. Many research papers have shown that feature selection can increase the performance of a classification [90]. Many criteria [70] are used to determine whether a feature should be selected for training purposes or not, such as, measuring its significance towards the class, finding the overlaps between classes, and determining unwanted features. An interesting paper [91] focused on selecting highly discriminative features (those which are present only in one class) in the training set. Moreover, the integration of multiple feature selection techniques can output decent results [92].

Many research works have concentrated on the addition of new features by, for example, considering missing features [60], side information [93,94], assigning weights [95], using semantic relationships [96,97], and utilizing the structure of documents [67,98]. Performing subsampling [99] can also benefit the classification performance.

One of the problems with feature selection is redundancy, that is, the same feature may occur more than once in different forms. Although stemming can eliminate a few redundant features, it is difficult to remove all redundant features, because there may be duplicates in the form of synonyms. Topic modeling is one of the methods proposed by [51] to remove duplicates. Stemming in the English language may be very easy; however, other languages, such as Arabic, require custom stemming algorithms to [80]. To improve the results, heuristic optimization methods have been applied to improve feature selection [100]. Other methods, such as redundant feature mapping [101] and word co-occurrences [102], can also help with the classification process to improve the performance.

4.3. Feature Drift

For some kinds of classification, like spam and reviews, there is no constant list of features that can determine the class of a sample. Features change over time, for example, a feature that is responsible for determining the class of a sample will be no longer responsible for the same in the near future. Thus, to obtain a good performance over the longer term, feature drift should be considered. Incremental learning, as discussed in [103], is one of the methods that can solve feature drift. The information value over time should be determined for each feature so that it is easy to find out which feature is important over a period of time [104].

4.4. Representation of Features

There are lots of ways in which features can be represented, such as, the bag of words and semantic models. Choosing the best representation according to the problem needs to be addressed before performing the classification process. While performing the bag of words model, the order of words is lost. Few research works have been done to optimize the bag of words model using methods such as including semantic information [105] and integrating fuzzy concepts [41]. Representing features in the form of word2vec can allow more relationships to be detected among them [106]. Different weighting schemes can be embedded to get good results, such as merging TF-IDF and word2vec [107]. When the number of features is greater, few works [108] show good performance when representing the relationship among features instead of considering all features independently.

4.5. Overfitting

Overfitting occurs when a model is trained too well. A high level of accuracy during training and a low level of accuracy during testing mean that the model is overfitted. An optimal number of samples should be selected for training and testing purposes. Reference [109] introduced a new metric known as the Rate of Overfitting (RO), which is used to determine the numbers of samples in the training and testing sets to avoid overfitting.

4.6. Short Text

A short text is one of the major difficulties in text classification, because it leads to limited features. Since the number of features is small, the classifier will struggle to learn all the relationships among classes. Reference [110] used an end-to-end learning hybrid network with multiple timescales. Other methods, such as character encoding [44], feature expansion [111], and rich feature generation [112] can also improve the performance of a classifier.

Sometimes, adding extra features from external sources (such as Wikipedia [113]) or additional datasets can increase the vocabulary size. To add extra features, topic modeling can be used to determine the classes of the extra features [66]. Transfer learning is one of the proven methods for handling short text. A model that has already been trained on one dataset can be reused for short datasets. Reference [114] showed an improvement in short text classification by using transfer learning methodology.

4.7. Imbalanced Data

When there is an unequal distribution of data samples among the classes, the classifier learns very little about the minority class. Imbalanced data are one of the challenges for a machine learning model. In a previous study [115], the authors used sampling techniques, such as SMOTE-ENN to overcome the imbalanced data scenario. The authors of [56] talked about multi-task learning, which can solve the imbalanced data problem. Assigning weights to members of the minority class [116] can also give a good classification performance. Resampling and instance weighting are other methods proposed by [117] to handle imbalanced datasets.

4.8. Misclassification

Misclassification means that a classifier wrongly classifies a sample. A classifier with a high false positive or false negative rate means that the classifier is misclassifying the samples. Few optimization techniques are used on the evaluation side [118], such as finding the product of precision and using F1 to find the misclassification. In [119], deep learning models and machine learning models were mixed. This fusion reduces the rate of misclassification. Making use of a virtual category can also prevent a classifier from performing misclassification, as described in [120]. A blocking mechanism was implemented in the research done by [121], whereby the classification is done iteratively and at each iteration, the misclassified samples are prevented from propagating to the next iteration.

Through the use of clustering and classification, most of the misclassified samples can be avoided by grouping positive samples with positive classes and negative samples with negative classes [122].

Identifying patterns in the pre-processing step can significantly prevent the samples from being wrongly classified. Regular expressions are used to extract the patterns and help the classifier to reduce misclassification [123].

4.9. Lack of Labeled Data

Labeled data are the best way to train a classifier. However, it is difficult to find a perfectly labeled dataset. Manual labeling is done by a domain expert and this requires significant time and costs. Moreover, machine learning has been used to replace

all of the manual work. Many research targets automatically label data by using various methodologies, such as active learning [124], detailed pre-processing [125], and interactive visualization [126].

While labeled data are not available, supervised classification becomes difficult. Ref. [55] shows how a semi-supervised classification can be done with good accuracy. In most cases, an unsupervised algorithm, such as clustering [127] can be combined with a classifier to achieve greater accuracy [59].

Topic modeling is an efficient method to tackle the labeling issue. Topic modeling algorithms, such as LDA, are efficient for labeling the unlabeled dataset [128].

4.10. High Dimensional Data

High dimensional data not only reduce the speed of the classifier but also degrade the performance badly. Few optimization techniques, such as the light-weighting protocol [129], semantic concept extraction [130], and ensembling features [131] are done to reduce the number of dimensions of input. Feature selection is done most of the time to reduce the number of dimensions. Reference [132] used a filter-based feature selection method to reduce the number of features.

Clustering can also be used before the classification begins, as mentioned in [133]. A good weighting scheme can be added to the clustering or classification step to increase the performance future [134].

4.11. Long Text

The classification of long text has many difficulties, such as redundancy and a mix of unwanted content. To solve this problem, in [135], the authors used feature fusing and identified the most important or related information from the text corpus. All previously mentioned problems and solutions are summarized in Figure 21.

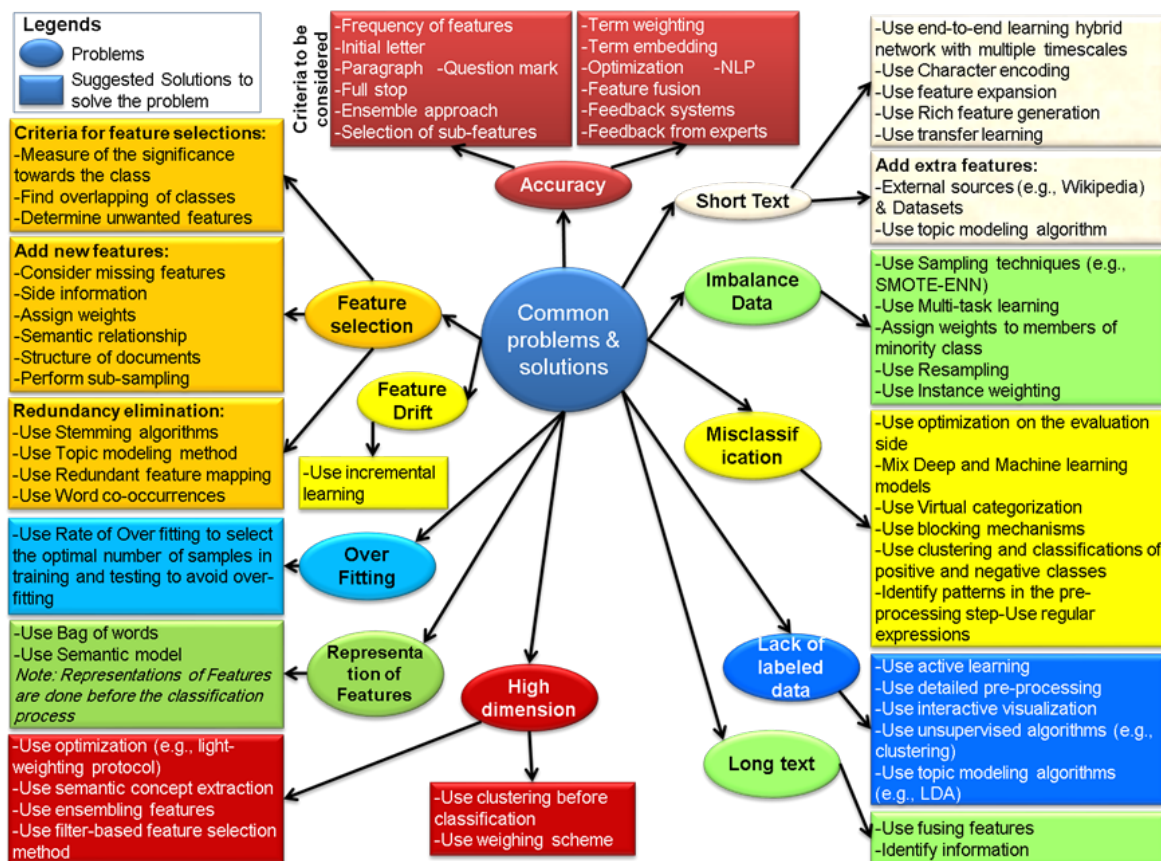


Figure 21. Summary of the main problems and solutions considered for improving text classification.

5. Discussion

In this section, we discuss our thoughts on the future directions in the field of text classification. We list our observations as follows:

5.1. Notable Observations

- A good correlation factor can be found between a pre-classified dataset and a classified dataset. This can enable a transfer learning approach that can easily classify an unmapped instance;
- Efficient feature extraction by incorporating textual algorithms (such as sentiment analysis, NLP) can focus on finding important terms (e.g., Smiley in social text classification);
- Implementing the GAN model for generating dummy text can convert a short-text input to a normal-sized input.

5.2. Research Gaps

We have identified a few gaps in our review. These are listed below.

- Many research studies have focused on self-generated datasets. However, many datasets exist for a given domain. All of these datasets differ in terms of the format and structure of data. Thus, a multi-model classification should be developed to address this issue.
- A publicly available database that contains the federated-based classifiers of the top datasets should be created. This will significantly help future researchers to develop high-quality and fast outcomes. This step will also enable researchers to compare their local results with community results.
- The use of active learning can improve the performance of classification by using only a few inputs. The majority of the papers skip the use of active learning. Thus, future research works can focus on including active learning in the classification.
- Text representation still requires improvement. Many research works should focus on labeling or segmenting the features. For example, there may be a pronoun that represents the noun from the previous sentence. Thus, a good labeling scheme should be developed.
- The majority of research included in this survey focused on ranking features based on their frequencies. Highly frequent features are ranked the highest. However, this may not be a generalized case. This limitation can be overcome with the help of domain experts who rank the least frequent words based on their importance.

5.3. Recommendations

We present recommendations for improving the performance of text classification methods, as follows:

- **Increase Accuracy:** Embedding methods can be improved by incorporating graph-based embedding approaches [83]. The SVM classifier can significantly increase the accuracy by focusing on improvements in kernels [136];
- **Feature Selection:** A future study has been proposed by [90] to analyze the feature space of the non-English corpus. The extraction of features can also be done by word basic or sentence basic methods [99];
- **Misclassification:** To reduce the rate of misclassification, fusion models [119] can be improved by assigning different weights to each model and by using some recent fusion models, such as hierarchical deep genetic networks and transfer-learning-based deep models. Creating a hybrid classification model by mixing both instance selection and feature selection could be done in the future [137];
- **Feature Drift:** In the future, work will determine whether a feature has significant importance in the upcoming period or not [103], thus removing very old features to improve the classification.

- **Long Text:** To improve the speed of classification on long-text information, the authors of [135] proposed the use of parallel computing.
- **Lack of labeled data:** Implementing a machine learning model for labeling multi-class samples can be considered a good future direction [124]. The pre-processing stage, which is done before the use of any machine learning model, can also be optimized to extract hidden labels [125].

5.4. Strengths and Weaknesses

We employed the PRISMA approach in this review and attempted to find as many suitable studies as possible. Through active conversations, we widened the search terms and databases and resolved any conflicts. Despite our intention to provide an international component to our analysis, we elected to limit our search to two databases recognized for their quality and commitment to research in order to ensure the rigor and quality of the papers included in our assessment. We prioritized the quality of the articles chosen above the scope of the study, although this resulted in a selection of just 224 research publications. Given this result, one wonders if, by integrating additional databases, more research from a larger range of nations prevented from being included.

6. Conclusions

Text classification is a foundation for many popular research areas such as sentiment analysis, web searching and summarizing, and spam detection. This paper comprehensively reviews articles on text classification. We selected papers from six publishers produced between 2003 and 2022 and presented an analysis on six aspects: dataset frequency, machine learning model frequency, best performance on each dataset, evaluation metric frequency, train–test splitting frequency, and a comparison among machine learning models. In this survey, we investigated 224 papers and conducted a comprehensive comparison. We found that SVM (59%), NB (46%), and kNN (33%) are the most commonly used machine learning models in the field of text classification. Additionally, 10-Fold validations is the most commonly used metric for validating the learning process of a classifier (22.50%). Accuracy is the most frequently used metric to measure the performance of a machine learning model (28%). SVM seems to perform better in many scenarios, while DT gives the worst results most of the time. Furthermore, we presented a summary of how machine learning is used to tackle various problems in the domain and also provided possible future directions for text classification. This systematic review serves as groundwork for researchers in the field of machine-learning-based text classification to further extend and optimize the models.

Author Contributions: Conceptualization, A.P. and C.Z.E.-B.; methodology, A.P. and C.Z.E.-B.; validation, A.P., C.Z.E.-B. and R.D.; formal analysis, A.P., C.Z.E.-B. and R.D.; investigation, A.P., C.Z.E.-B. and R.D.; data curation, A.P. and C.Z.E.-B.; writing—original draft preparation, A.P. and C.Z.E.-B.; writing—review and editing, R.D.; visualization, A.P., C.Z.E.-B. and R.D.; supervision, A.P.; funding acquisition, R.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets generated and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sebastiani, F. Machine Learning in Automated Text Categorization. *ACM Comput. Surv.* **2002**, *34*, 1–47. [\[CrossRef\]](#)
2. Kowsari, K.; Jafari Meimandi, K.; Heidarysafa, M.; Mendu, S.; Barnes, L.; Brown, D. Text Classification Algorithms: A Survey. *Information* **2019**, *10*, 150. [\[CrossRef\]](#)
3. Kapočiute-Dzikiene, J. A domain-specific generative chatbot trained from little data. *Appl. Sci.* **2020**, *10*, 2221. [\[CrossRef\]](#)

4. Rogers, D.; Preece, A.; Innes, M.; Spasić, I. Real-Time Text Classification of User-Generated Content on Social Media: Systematic Review. *IEEE Trans. Comput. Soc. Syst.* **2022**, *9*, 1154–1166. [[CrossRef](#)]
5. Karayigit, H.; Akdagli, A.; Aci, Ç.İ. BERT-based Transfer Learning Model for COVID-19 Sentiment Analysis on Turkish Instagram Comments. *Inf. Technol. Control* **2022**, *51*, 409–428. [[CrossRef](#)]
6. Kapočiūtė-Dzikienė, J.; Damaševičius, R.; Woźniak, M. Sentiment analysis of Lithuanian texts using traditional and deep learning approaches. *Computers* **2019**, *8*, 4. [[CrossRef](#)]
7. Tesfagergish, S.G.; Kapočiūtė-Dzikienė, J.; Damaševičius, R. Zero-Shot Emotion Detection for Semi-Supervised Sentiment Analysis Using Sentence Transformers and Ensemble Learning. *Appl. Sci.* **2022**, *12*, 8662. [[CrossRef](#)]
8. Karayigit, H.; Akdagli, A.; Aci, Ç.İ. Homophobic and Hate Speech Detection Using Multilingual-BERT Model on Turkish Social Media. *Inf. Technol. Control* **2022**, *51*, 356–375. [[CrossRef](#)]
9. Aldjanabi, W.; Dahou, A.; Al-Qaness, M.A.A.; Elaziz, M.A.; Helmi, A.M.; Damaševičius, R. Arabic offensive and hate speech detection using a cross-corpora multi-task learning model. *Informatics* **2021**, *8*, 69. [[CrossRef](#)]
10. Kapociute-Dzikiene, J.; Venckauskas, A.; Damasevicius, R. A comparison of authorship attribution approaches applied on the Lithuanian language. In Proceedings of the 2017 Federated Conference on Computer Science and Information Systems, FedCSIS 2017, Prague, Czech Republic, 3–6 September 2017; pp. 347–351.
11. Mathews, A.; Sejal, N.; Venugopal, K.R. Text Based and Image Based Recommender Systems: Fundamental Concepts, Comprehensive Review and Future Directions. *Int. J. Eng. Trends Technol.* **2022**, *70*, 124–143. [[CrossRef](#)]
12. Ji, Z.; Pi, H.; Wei, W.; Xiong, B.; Wozniak, M.; Damasevicius, R. Recommendation Based on Review Texts and Social Communities: A Hybrid Model. *IEEE Access* **2019**, *7*, 40416–40427. [[CrossRef](#)]
13. Sun, G.; Wang, Z.; Zhao, J. Automatic text summarization using deep reinforcement learning and beyond. *Inf. Technol. Control* **2021**, *50*, 458–469. [[CrossRef](#)]
14. Jiang, M.; Zou, Y.; Xu, J.; Zhang, M. GATSum: Graph-Based Topic-Aware Abstract Text Summarization. *Inf. Technol. Control* **2022**, *51*, 345–355. [[CrossRef](#)]
15. Shrivasa, A.K.; Dewangan, A.K.; Ghosh, S.M.; Singh, D. Development of proposed ensemble model for spam e-mail classification. *Inf. Technol. Control* **2021**, *50*, 411–423. [[CrossRef](#)]
16. Salloum, S.; Gaber, T.; Vadera, S.; Shaalan, K. A Systematic Literature Review on Phishing Email Detection Using Natural Language Processing Techniques. *IEEE Access* **2022**, *10*, 65703–65727. [[CrossRef](#)]
17. Kapočiūtė-Dzikienė, J.; Balodis, K.; Skadiņš, R. Intent detection problem solving via automatic DNN hyperparameter optimization. *Appl. Sci.* **2020**, *10*, 7426. [[CrossRef](#)]
18. Iqbal, W.; Malik, W.I.; Bukhari, F.; Almustafa, K.M.; Nawaz, Z. Big data full-text search index minimization using text summarization. *Inf. Technol. Control* **2021**, *50*, 375–389. [[CrossRef](#)]
19. Dogra, V.; Verma, S.; Kavita; Chatterjee, P.; Shafi, J.; Choi, J.; Ijaz, M.F. A Complete Process of Text Classification System Using State-of-the-Art NLP Models. *Comput. Intell. Neurosci.* **2022**, *2022*, 1883698. [[CrossRef](#)] [[PubMed](#)]
20. Ashokkumar, P.; Arunkumar, N.; Don, S. Intelligent optimal route recommendation among heterogeneous objects with keywords. *Comput. Electr. Eng.* **2018**, *68*, 526–535. [[CrossRef](#)]
21. Haque, R.; Islam, N.; Tasneem, M.; Das, A.K. Multi-class sentiment classification on Bengali social media comments using machine learning. *Int. J. Cogn. Comput. Eng.* **2023**, *4*, 21–35. [[CrossRef](#)]
22. Gupta, A.; Dengre, V.; Kheruwala, H.A.; Shah, M. Comprehensive review of text-mining applications in finance. *Financ. Innov.* **2020**, *6*, 39. [[CrossRef](#)]
23. Li, Q.; Li, S.; Zhang, S.; Hu, J.; Hu, J. A review of text corpus-based tourism big data mining. *Appl. Sci.* **2019**, *9*, 3300. [[CrossRef](#)]
24. Omoregbe, N.A.I.; Ndaman, I.O.; Misra, S.; Abayomi-Alli, O.O.; Damaševičius, R. Text messaging-based medical diagnosis using natural language processing and fuzzy logic. *J. Healthc. Eng.* **2020**, *2020*, 8839524. [[CrossRef](#)]
25. Tesfagergish, S.G.; Damaševičius, R.; Kapočiūtė-Dzikienė, J. *Deep Fake Recognition in Tweets Using Text Augmentation, Word Embeddings and Deep Learning*; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Berlin/Heidelberg, Germany, 2021; Volume 12954, pp. 523–538.
26. Thangaraj, M.; Sivakami, M. Text Classification Techniques: A Literature Review. *Interdiscip. J. Inf. Knowl. Manag.* **2018**, *13*, 117–135. [[CrossRef](#)]
27. Minaee, S.; Kalchbrenner, N.; Cambria, E.; Nikzad, N.; Chenaghlu, M.; Gao, J. Deep Learning-based Text Classification. *ACM Comput. Surv.* **2021**, *54*, 1–40. [[CrossRef](#)]
28. Bayer, M.; Kaufhold, M.A.; Reuter, C. A Survey on Data Augmentation for Text Classification. *ACM Comput. Surv.* **2022**, *55*, 3544558. [[CrossRef](#)]
29. Li, Q.; Peng, H.; Li, J.; Xia, C.; Yang, R.; Sun, L.; Yu, P.S.; He, L. A Survey on Text Classification: From Traditional to Deep Learning. *ACM Trans. Intell. Syst. Technol.* **2022**, *13*, 1–41. [[CrossRef](#)]
30. Wu, H.; Liu, Y.; Wang, J. Review of text classification methods on deep learning. *Comput. Mater. Contin.* **2020**, *63*, 1309–1321. [[CrossRef](#)]
31. Mirończuk, M.M.; Protasiewicz, J. A recent overview of the state-of-the-art elements of text classification. *Expert Syst. Appl.* **2018**, *106*, 36–54. [[CrossRef](#)]
32. Moher, D.; Shamseer, L.; Clarke, M.; Ghersi, D.; Liberati, A.; Petticrew, M.; Shekelle, P.; Stewart, L.A. Preferred reporting items for systematic review and meta-analysis protocols (PRISMA-P) 2015 statement. *Syst. Rev.* **2015**, *4*, 1. [[CrossRef](#)]

33. Isa, D.; Lee, L.H.; Kallimani, V.P.; RajKumar, R. Text Document Preprocessing with the Bayes Formula for Classification Using the Support Vector Machine. *IEEE Trans. Knowl. Data Eng.* **2008**, *20*, 1264–1272. [[CrossRef](#)]
34. Han, H.; Ko, Y.; Seo, J. Using the revised EM algorithm to remove noisy data for improving the one-against-the-rest method in binary text classification. *Inf. Process. Manag.* **2007**, *43*, 1281–1293. [[CrossRef](#)]
35. Haneczok, J.; Piskorski, J. Shallow and deep learning for event relatedness classification. *Inf. Process. Manag.* **2020**, *57*, 102371. [[CrossRef](#)]
36. Wang, T.Y.; Chiang, H.M. Fuzzy support vector machine for multi-class text categorization. *Inf. Process. Manag.* **2007**, *43*, 914–929. [[CrossRef](#)]
37. Devaraj, A.; Murthy, D.; Dontula, A. Machine-learning methods for identifying social media-based requests for urgent help during hurricanes. *Int. J. Disaster Risk Reduct.* **2020**, *51*, 101757. [[CrossRef](#)]
38. Chukwuocha, C.; Mathu, T.; Raimond, K. Design of an Interactive Biomedical Text Mining Framework to Recognize Real-Time Drug Entities Using Machine Learning Algorithms. *Procedia Comput. Sci.* **2018**, *143*, 181–188. [[CrossRef](#)]
39. Elnagar, A.; Al-Debsi, R.; Einea, O. Arabic text classification using deep learning models. *Inf. Process. Manag.* **2020**, *57*, 102121. [[CrossRef](#)]
40. Sboev, A.; Litvinova, T.; Gudovskikh, D.; Rybka, R.; Moloshnikov, I. Machine Learning Models of Text Categorization by Author Gender Using Topic-independent Features. *Procedia Comput. Sci.* **2016**, *101*, 135–142. [[CrossRef](#)]
41. Zhao, R.; Mao, K. Fuzzy Bag-of-Words Model for Document Representation. *IEEE Trans. Fuzzy Syst.* **2018**, *26*, 794–804. [[CrossRef](#)]
42. Xu, D.; Tian, Z.; Lai, R.; Kong, X.; Tan, Z.; Shi, W. Deep learning based emotion analysis of microblog texts. *Inf. Fusion* **2020**, *64*, 1–11. [[CrossRef](#)]
43. Baker, L.D.; McCallum, A.K. Distributional Clustering of Words for Text Classification. In Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '98, Melbourne, Australia, 24–28 August 1998; Association for Computing Machinery: New York, NY, USA, 1998; pp. 96–103. [[CrossRef](#)]
44. Zhu, Y.; Li, Y.; Yue, Y.; Qiang, J.; Yuan, Y. A Hybrid Classification Method via Character Embedding in Chinese Short Text With Few Words. *IEEE Access* **2020**, *8*, 92120–92128. [[CrossRef](#)]
45. Halim, Z.; Waqar, M.; Tahir, M. A machine learning-based investigation utilizing the in-text features for the identification of dominant emotion in an email. *Knowl.-Based Syst.* **2020**, *208*, 106443. [[CrossRef](#)]
46. Lopes, F.; Agnelo, J.; Teixeira, C.A.; Laranjeiro, N.; Bernardino, J. Automating orthogonal defect classification using machine learning algorithms. *Future Gener. Comput. Syst.* **2020**, *102*, 932–947. [[CrossRef](#)]
47. Goodrum, H.; Roberts, K.; Bernstam, E.V. Automatic classification of scanned electronic health record documents. *Int. J. Med. Inform.* **2020**, *144*, 104302. [[CrossRef](#)]
48. Vijayakumar, B.; Fuad, M.M.M. A New Method to Identify Short-Text Authors Using Combinations of Machine Learning and Natural Language Processing Techniques. *Procedia Comput. Sci.* **2019**, *159*, 428–436. [[CrossRef](#)]
49. Singh, A.; Tucker, C.S. A machine learning approach to product review disambiguation based on function, form and behavior classification. *Decis. Support Syst.* **2017**, *97*, 81–91. [[CrossRef](#)]
50. Park, E.L.; Cho, S.; Kang, P. Supervised Paragraph Vector: Distributed Representations of Words, Documents and Class Labels. *IEEE Access* **2019**, *7*, 29051–29064. [[CrossRef](#)]
51. Rashid, J.; Adnan Shah, S.M.; Irtaza, A.; Mahmood, T.; Nisar, M.W.; Shafiq, M.; Gardezi, A. Topic Modeling Technique for Text Mining Over Biomedical Text Corpora Through Hybrid Inverse Documents Frequency and Fuzzy K-Means Clustering. *IEEE Access* **2019**, *7*, 146070–146080. [[CrossRef](#)]
52. Liu, C.; Hsaio, W.; Lee, C.; Lu, G.; Jou, E. Movie Rating and Review Summarization in Mobile Environment. *IEEE Trans. Syst. Man Cybern. Part Appl. Rev.* **2012**, *42*, 397–407. [[CrossRef](#)]
53. Yu, B.; Xu, Z.B. A comparative study for content-based dynamic spam classification using four machine learning algorithms. *Knowl.-Based Syst.* **2008**, *21*, 355–362. [[CrossRef](#)]
54. Espejo-Garcia, B.; Martinez-Guanter, J.; Pérez-Ruiz, M.; Lopez-Pellicer, F.J.; Javier Zarazaga-Soria, F. Machine learning for automatic rule classification of agricultural regulations: A case study in Spain. *Comput. Electron. Agric.* **2018**, *150*, 343–352. [[CrossRef](#)]
55. Ligthart, A.; Catal, C.; Tekinerdogan, B. Analyzing the effectiveness of semi-supervised learning approaches for opinion spam classification. *Appl. Soft Comput.* **2021**, *101*, 107023. [[CrossRef](#)]
56. Song, D.; Vold, A.; Madan, K.; Schilder, F. Multi-label legal document classification: A deep learning-based approach with label-attention and domain-specific pre-training. *Inf. Syst.* **2021**, *106*, 101718. [[CrossRef](#)]
57. Rostam, N.A.P.; Malim, N.H.A.H. Text categorisation in Quran and Hadith: Overcoming the interrelation challenges using machine learning and term weighting. *J. King Saud Univ.-Comput. Inf. Sci.* **2019**, *33*, 658–667. [[CrossRef](#)]
58. Altinel, B.; Can Ganiz, M.; Diri, B. A corpus-based semantic kernel for text classification by using meaning values of terms. *Eng. Appl. Artif. Intell.* **2015**, *43*, 54–66. [[CrossRef](#)]
59. Shafiabady, N.; Lee, L.; Rajkumar, R.; Kallimani, V.; Akram, N.A.; Isa, D. Using unsupervised clustering approach to train the Support Vector Machine for text classification. *Neurocomputing* **2016**, *211*, 4–10. [[CrossRef](#)]
60. Sabbah, T.; Selamat, A.; Selamat, M.H.; Al-Anzi, F.S.; Viedma, E.H.; Krejcar, O.; Fujita, H. Modified frequency-based term weighting schemes for text classification. *Appl. Soft Comput.* **2017**, *58*, 193–206. [[CrossRef](#)]

61. Milosevic, N.; Dehghantanha, A.; Choo, K.K.R. Machine learning aided Android malware classification. *Comput. Electr. Eng.* **2017**, *61*, 266–274. [[CrossRef](#)]
62. Akhter, M.P.; Jiangbin, Z.; Naqvi, I.R.; Abdelmajeed, M.; Mehmood, A.; Sadiq, M.T. Document-Level Text Classification Using Single-Layer Multisize Filters Convolutional Neural Network. *IEEE Access* **2020**, *8*, 42689–42707. [[CrossRef](#)]
63. Huang, L.; Song, T.; Jiang, T. Linear regression combined KNN algorithm to identify latent defects for imbalance data of ICs. *Microelectron. J.* **2023**, *131*, 105641. [[CrossRef](#)]
64. Li, W.; Miao, D.; Wang, W. Two-level hierarchical combination method for text classification. *Expert Syst. Appl.* **2011**, *38*, 2030–2039. [[CrossRef](#)]
65. Wan, C.H.; Lee, L.H.; Rajkumar, R.; Isa, D. A hybrid text classification approach with low dependency on parameter by integrating K-nearest neighbor and support vector machine. *Expert Syst. Appl.* **2012**, *39*, 11880–11888. [[CrossRef](#)]
66. Vo, D.T.; Ock, C.Y. Learning to classify short text from scientific documents using topic models with various types of knowledge. *Expert Syst. Appl.* **2015**, *42*, 1684–1698. [[CrossRef](#)]
67. Khabbaz, M.; Kianmehr, K.; Alhaji, R. Employing Structural and Textual Feature Extraction for Semistructured Document Classification. *IEEE Trans. Syst. Man Cybern. Part Appl. Rev.* **2012**, *42*, 1566–1578. [[CrossRef](#)]
68. Asim, Y.; Shahid, A.R.; Malik, A.K.; Raza, B. Significance of machine learning algorithms in professional blogger’s classification. *Comput. Electr. Eng.* **2018**, *65*, 461–473. [[CrossRef](#)]
69. Hartmann, J.; Huppertz, J.; Schamp, C.; Heitmann, M. Comparing automated text classification methods. *Int. J. Res. Mark.* **2019**, *36*, 20–38. [[CrossRef](#)]
70. Ngejane, C.; Eloff, J.; Sefara, T.; Marivate, V. Digital forensics supported by machine learning for the detection of online sexual predatory chats. *Forensic Sci. Int. Digit. Investig.* **2021**, *36*, 301109. [[CrossRef](#)]
71. Tesfagergish, S.G.; Kapočičūtė-Dzikiėnė, J. Part-of-speech tagging via deep neural networks for northern-Ethiopic languages. *Inf. Technol. Control* **2020**, *49*, 482–494.
72. Mikolov, T.; Chen, K.; Corrado, G.S.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arXiv:1301.3781.
73. Le, Q.V.; Mikolov, T. Distributed Representations of Sentences and Documents. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014.
74. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching Word Vectors with Subword Information. *Trans. Assoc. Comput. Linguist.* **2016**, *5*, 135–146. [[CrossRef](#)]
75. Choi, J.; Lee, S.W. Improving FastText with inverse document frequency of subwords. *Pattern Recognit. Lett.* **2020**, *133*, 165–172. [[CrossRef](#)]
76. Athiwaratkun, B.; Wilson, A.G.; Anandkumar, A. Probabilistic FastText for Multi-Sense Word Embeddings. In Proceedings of the ACL, Melbourne, Australia, 15–20 July 2018.
77. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep Contextualized Word Representations. In Proceedings of the NAACL, New Orleans, LA, USA, 1–6 June 2018.
78. Damasevicius, R.; Valys, R.; Wozniak, M. Intelligent tagging of online texts using fuzzy logic. In Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence, SSCI, Athens, Greece, 6–9 December 2016.
79. Khasanah, I.N. Sentiment Classification Using fastText Embedding and Deep Learning Model. *Procedia Comput. Sci.* **2021**, *189*, 343–350. [[CrossRef](#)]
80. Ait Hammou, B.; Ait Lahcen, A.; Mouline, S. Towards a real-time processing framework based on improved distributed recurrent neural network variants with fastText for social big data analytics. *Inf. Process. Manag.* **2020**, *57*, 102122. [[CrossRef](#)]
81. Fang, Y.; Huang, C.; Su, Y.; Qiu, Y. Detecting malicious JavaScript code based on semantic analysis. *Comput. Secur.* **2020**, *93*, 101764. [[CrossRef](#)]
82. Luo, X. Efficient English text classification using selected Machine Learning Techniques. *Alex. Eng. J.* **2021**, *60*, 3401–3409. [[CrossRef](#)]
83. Ibrahim, M.A.; Ghani Khan, M.U.; Mehmood, F.; Asim, M.N.; Mahmood, W. GHS-NET a generic hybridized shallow neural network for multi-label biomedical text classification. *J. Biomed. Inform.* **2021**, *116*, 103699. [[CrossRef](#)] [[PubMed](#)]
84. Liu, X.; Mou, L.; Cui, H.; Lu, Z.; Song, S. Finding decision jumps in text classification. *Neurocomputing* **2020**, *371*, 177–187. [[CrossRef](#)]
85. Ye, X.; Dai, H.; Dong, L.A.; Wang, X. Multi-view ensemble learning method for microblog sentiment classification. *Expert Syst. Appl.* **2021**, *166*, 113987. [[CrossRef](#)]
86. Fragos, K.; Belsis, P.; Skourlas, C. Combining Probabilistic Classifiers for Text Classification. *Procedia-Soc. Behav. Sci.* **2014**, *147*, 307–312. [[CrossRef](#)]
87. Shang, C.; Li, M.; Feng, S.; Jiang, Q.; Fan, J. Feature selection via maximizing global information gain for text classification. *Knowl.-Based Syst.* **2013**, *54*, 298–309. [[CrossRef](#)]
88. Matošević, G.; Dobša, J.; Mladenčić, D. Using Machine Learning for Web Page Classification in Search Engine Optimization. *Future Internet* **2021**, *13*, 9. [[CrossRef](#)]
89. Mesleh, A.M. Feature sub-set selection metrics for Arabic text classification. *Pattern Recognit. Lett.* **2011**, *32*, 1922–1929. [[CrossRef](#)]
90. Santucci, V.; Santarelli, F.; Forti, L.; Spina, S. Automatic Classification of Text Complexity. *Appl. Sci.* **2020**, *10*, 7285. [[CrossRef](#)]

91. Ganiz, M.C.; Lytkin, N.I.; Pottenger, W.M. Leveraging Higher Order Dependencies between Features for Text Classification. *Mach. Learn. Knowl. Discov. Databases Lect. Notes Comput. Sci.* **2009**, *5781*, 375–390. [[CrossRef](#)]
92. Sabbah, T.; Selamat, A.; Selamat, M.H.; Ibrahim, R.; Fujita, H. Hybridized term-weighting method for Dark Web classification. *Neurocomputing* **2016**, *173*, 1908–1926. [[CrossRef](#)]
93. Aggarwal, C.C.; Zhao, Y.; Yu, P.S. On the Use of Side Information for Mining Text Data. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 1415–1429. [[CrossRef](#)]
94. Ojewumi, T.; Ogunleye, G.; Oguntunde, B.; Folorunsho, O.; Fashoto, S.; Ogbu, N. Performance evaluation of machine learning tools for detection of phishing attacks on web pages. *Sci. Afr.* **2022**, *16*, e01165. [[CrossRef](#)]
95. Moreo, A.; Esuli, A.; Sebastiani, F. Learning to Weight for Text Classification. *IEEE Trans. Knowl. Data Eng.* **2020**, *32*, 302–316. [[CrossRef](#)]
96. Hasan, M.; Kotov, A.; Idalski Carcone, A.; Dong, M.; Naar, S.; Brogan Hartlieb, K. A study of the effectiveness of machine learning methods for classification of clinical interview fragments into a large number of categories. *J. Biomed. Inform.* **2016**, *62*, 21–31. [[CrossRef](#)]
97. Galitsky, B. Machine learning of syntactic parse trees for search and classification of text. *Eng. Appl. Artif. Intell.* **2013**, *26*, 1072–1091. [[CrossRef](#)]
98. Liang, J.; Zhou, X.; Liu, P.; Guo, L.; Bai, S. An EMM-based Approach for Text Classification. *Procedia Comput. Sci.* **2013**, *17*, 506–513. [[CrossRef](#)]
99. He, J.; Wang, L.; Liu, L.; Feng, J.; Wu, H. Long Document Classification From Local Word Glimpses via Recurrent Attention Learning. *IEEE Access* **2019**, *7*, 40707–40718. [[CrossRef](#)]
100. Alhaj, Y.A.; Dahou, A.; Al-Qaness, M.A.A.; Abualigah, L.; Abbasi, A.A.; Almaweri, N.A.O.; Elaziz, M.A.; Damaševičius, R. A Novel Text Classification Technique Using Improved Particle Swarm Optimization: A Case Study of Arabic Language. *Future Internet* **2022**, *14*, 194. [[CrossRef](#)]
101. Lin, Y.; Jiang, J.; Lee, S. A Similarity Measure for Text Classification and Clustering. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 1575–1590. [[CrossRef](#)]
102. Figueiredo, F.; Rocha, L.; Couto, T.; Salles, T.; Gonçalves, M.A.; Meira, W., Jr. Word co-occurrence features for text classification. *Inf. Syst.* **2011**, *36*, 843–858. [[CrossRef](#)]
103. Chen, C.; Wang, Y.; Zhang, J.; Xiang, Y.; Zhou, W.; Min, G. Statistical Features-Based Real-Time Detection of Drifted Twitter Spam. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 914–925. [[CrossRef](#)]
104. Babapour, S.M.; Roostae, M. Web pages classification: An effective approach based on text mining techniques. In Proceedings of the 2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI), Tehran, Iran, 22 December 2017; pp. 0320–0323. [[CrossRef](#)]
105. Kim, H.J.; Kim, J.; Kim, J.; Lim, P. Towards perfect text classification with Wikipedia-based semantic Naïve Bayes learning. *Neurocomputing* **2018**, *315*, 128–134. [[CrossRef](#)]
106. Fesseha, A.; Xiong, S.; Emiru, E.D.; Diallo, M.; Dahou, A. Text Classification Based on Convolutional Neural Networks and Word Embedding for Low-Resource Languages: Tigrinya. *Information* **2021**, *12*, 52. [[CrossRef](#)]
107. Lilleberg, J.; Zhu, Y.; Zhang, Y. Support vector machines and Word2vec for text classification with semantic features. In Proceedings of the 2015 IEEE 14th International Conference on Cognitive Informatics Cognitive Computing (ICCI*CC), Beijing, China, 6–8 July 2015; pp. 136–140. [[CrossRef](#)]
108. Ganiz, M.C.; George, C.; Pottenger, W.M. Higher Order Naive Bayes: A Novel Non-IID Approach to Text Classification. *IEEE Trans. Knowl. Data Eng.* **2011**, *23*, 1022–1034. [[CrossRef](#)]
109. Feng, X.; Liang, Y.; Shi, X.; Xu, D.; Wang, X.; Guan, R. Overfitting Reduction of Text Classification Based on AdaBELM. *Entropy* **2017**, *19*, 330. [[CrossRef](#)]
110. Moirangthem, D.S.; Lee, M. Hierarchical and lateral multiple timescales gated recurrent units with pre-trained encoder for long text classification. *Expert Syst. Appl.* **2021**, *165*, 113898. [[CrossRef](#)]
111. Wang, P.; Xu, B.; Xu, J.; Tian, G.; Liu, C.L.; Hao, H. Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification. *Neurocomputing* **2016**, *174*, 806–814. [[CrossRef](#)]
112. Li, J.; Rao, Y.; Jin, F.; Chen, H.; Xiang, X. Multi-label maximum entropy model for social emotion classification over short text. *Neurocomputing* **2016**, *210*, 247–256. [[CrossRef](#)]
113. Wang, X.; Chen, R.; Jia, Y.; Zhou, B. Short Text Classification Using Wikipedia Concept Based Document Representation. In Proceedings of the 2013 International Conference on Information Technology and Applications, Chengdu, China, 16–17 November 2013; pp. 471–474. [[CrossRef](#)]
114. Xu, J.; Du, Q. Learning transferable features in meta-learning for few-shot text classification. *Pattern Recognit. Lett.* **2020**, *135*, 271–278. [[CrossRef](#)]
115. Kim, N.; Hong, S. Automatic classification of citizen requests for transportation using deep learning: Case study from Boston city. *Inf. Process. Manag.* **2021**, *58*, 102410. [[CrossRef](#)]
116. Liu, Y.; Loh, H.T.; Sun, A. Imbalanced text classification: A term weighting approach. *Expert Syst. Appl.* **2009**, *36*, 690–701. [[CrossRef](#)]
117. Sun, A.; Lim, E.P.; Liu, Y. On strategies for imbalanced text classification using SVM: A comparative study. *Decis. Support Syst.* **2009**, *48*, 191–201. [[CrossRef](#)]

118. Triantafyllou, I.; Drivas, I.C.; Giannakopoulos, G. How to Utilize My App Reviews? A Novel Topics Extraction Machine Learning Schema for Strategic Business Purposes. *Entropy* **2020**, *22*, 1310. [[CrossRef](#)]
119. Basiri, M.E.; Abdar, M.; Cifci, M.A.; Nemati, S.; Acharya, U.R. A novel method for sentiment classification of drug reviews using fusion of deep and machine learning techniques. *Knowl.-Based Syst.* **2020**, *198*, 105949. [[CrossRef](#)]
120. Stein, R.A.; Jaques, P.A.; Valiati, J.F. An analysis of hierarchical text classification using word embeddings. *Inf. Sci.* **2019**, *471*, 216–232. [[CrossRef](#)]
121. Sun, A.; Lim, E.; Ng, W.; Srivastava, J. Blocking reduction strategies in hierarchical text classification. *IEEE Trans. Knowl. Data Eng.* **2004**, *16*, 1305–1308. [[CrossRef](#)]
122. Alsmadi, I.; Alhami, I. Clustering and classification of email contents. *J. King Saud Univ.-Comput. Inf. Sci.* **2015**, *27*, 46–57. [[CrossRef](#)]
123. Galgani, F.; Compton, P.; Hoffmann, A. LEXA: Building knowledge bases for automatic legal citation classification. *Expert Syst. Appl.* **2015**, *42*, 6391–6407. [[CrossRef](#)]
124. Hu, R.; Mac Namee, B.; Delany, S.J. Active learning for text classification with reusability. *Expert Syst. Appl.* **2016**, *45*, 438–449. [[CrossRef](#)]
125. Jung, N.; Lee, G. Automated classification of building information modeling (BIM) case studies by BIM use based on natural language processing (NLP) and unsupervised learning. *Adv. Eng. Inform.* **2019**, *41*, 100917. [[CrossRef](#)]
126. Heimerl, F.; Koch, S.; Bosch, H.; Ertl, T. Visual Classifier Training for Text Document Retrieval. *IEEE Trans. Vis. Comput. Graph.* **2012**, *18*, 2839–2848. [[CrossRef](#)]
127. Palanivinaayagam, A.; Nagarajan, S. An optimized iterative clustering framework for recognizing speech. *Int. J. Speech Technol.* **2020**, *23*, 767–777. [[CrossRef](#)]
128. Pavlinek, M.; Podgorelec, V. Text classification method based on self-training and LDA topic models. *Expert Syst. Appl.* **2017**, *80*, 83–93. [[CrossRef](#)]
129. Silva, R.M.; Almeida, T.A.; Yamakami, A. MDLText: An efficient and lightweight text classifier. *Knowl.-Based Syst.* **2017**, *118*, 152–164. [[CrossRef](#)]
130. Hoai Nam, L.N.; Quoc, H.B. Integrating Low-rank Approximation and Word Embedding for Feature Transformation in the High-dimensional Text Classification. *Procedia Comput. Sci.* **2017**, *112*, 437–446. [[CrossRef](#)]
131. Onan, A.; Korukoğlu, S.; Bulut, H. Ensemble of keyword extraction methods and classifiers in text classification. *Expert Syst. Appl.* **2016**, *57*, 232–247. [[CrossRef](#)]
132. Uysal, A.K.; Gunal, S. A novel probabilistic feature selection method for text classification. *Knowl.-Based Syst.* **2012**, *36*, 226–235. [[CrossRef](#)]
133. Seara Vieira, A.; Borrajo, L.; Iglesias, E. Improving the text classification using clustering and a novel HMM to reduce the dimensionality. *Comput. Methods Programs Biomed.* **2016**, *136*, 119–130. [[CrossRef](#)]
134. Selamat, A.; Omatu, S. Web page feature selection and classification using neural networks. *Inf. Sci.* **2004**, *158*, 69–88. [[CrossRef](#)]
135. Deng, J.; Cheng, L.; Wang, Z. Attention-based BiLSTM fused CNN with gating mechanism model for Chinese long text classification. *Comput. Speech Lang.* **2021**, *68*, 101182. [[CrossRef](#)]
136. Liu, Y.; Bi, J.W.; Fan, Z.P. Multi-class sentiment classification: The experimental comparisons of feature selection and machine learning algorithms. *Expert Syst. Appl.* **2017**, *80*, 323–339. [[CrossRef](#)]
137. Tsai, C.F.; Chen, Z.Y.; Ke, S.W. Evolutionary instance selection for text classification. *J. Syst. Softw.* **2014**, *90*, 104–113. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.