computers

MDPI

Article

Topic Classification of Online News Articles Using Optimized Machine Learning Models

Shahzada Daud [1], Muti Ullah [1], Amjad Rehman [2], Tanzila Saba [2], Robertas Damaševičius [3,*] and Abdul Sattar [1]

[1] Department of Computer Science, Khwaja Fareed University of Engineering & Information Technology, Rahim Yar Khan 64200, Pakistan
[2] Artificial Intelligence & Data Analytics Lab (AIDA), CCIS Prince Sultan University, Riyadh 11586, Saudi Arabia
[3] Department of Applied Informatics, Vytautas Magnus University, 44404 Kaunas, Lithuania
* Correspondence: robertas.damasevicius@vdu.lt

Abstract: Much news is available online, and not all is categorized. A few researchers have carried out work on news classification in the past, and most of the work focused on fake news identification. Most of the work performed on news categorization is carried out on a benchmark dataset. The problem with the benchmark dataset is that model trained with it is not applicable in the real world as the data are pre-organized. This study used machine learning (ML) techniques to categorize online news articles as these techniques are cheaper in terms of computational needs and are less complex. This study proposed the hyperparameter-optimized support vector machines (SVM) to categorize news articles according to their respective category. Additionally, five other ML techniques, Stochastic Gradient Descent (SGD), Random Forest (RF), Logistic Regression (LR), K-Nearest Neighbor (KNN), and Naïve Bayes (NB), were optimized for comparison for the news categorization task. The results showed that the optimized SVM model performed better than other models, while without optimization, its performance was worse than other ML models.

Keywords: topic categorization; model parameter tuning; hyperparameter optimization; grid search; SVM; NLP; TF-IDF; human rights; fair societies

Citation: Daud, S.; Ullah, M.; Rehman, A.; Saba, T.; Damaševičius, R.; Sattar, A. Topic Classification of Online News Articles Using Optimized Machine Learning Models. *Computers* **2023**, *12*, 16. https://doi.org/10.3390/computers12010016

Academic Editors: Phivos Mylonas, Katia Lida Kermanidis, Manolis Maragoudakis and Paolo Bellavista

Received: 8 December 2022
Revised: 31 December 2022
Accepted: 6 January 2023
Published: 9 January 2023

1. Introduction

News has been around people for a long time, and it is almost impossible to assume its starting period. News is a set of information about recent events. These events can be of any kind [1]. Web news sources provide huge amounts of news, so it has become convenient for everyone to have news reports at a click away. News and local information coverage is increasingly facilitated through online social networks [2]. Additionally, an enormous amount of information is being posted on different sources. Some common sources are social networking websites, news agencies' websites, and blogs. Even though these technologies have the potential to provide people with various viewpoints [3,4], they can also limit people from more attitude-challenging information [5], which is related to the adoption of thrilling attitudes with time [6] and misunderstanding of facts about current events [7].

The main characteristics of the news text in online publications are (1) the prevalence of conversational style; (2) the brevity of the information provided, in which, of course, an information note is a genre of small volume, but the material requires not only the fact of the news that happened but also a detailed description of what is happening and even the author's assessment; And (3) the absence of complicated lexical and grammatical forms. The text performs the functions of informing the audience and influencing it with the help of expressive means. Online news achieves this thanks to simple sentences that do not require the reader to delve into the essence of what is written for a long time. The topics of

online news are thematically homogeneous sets of documents. In this case, the set of topics is set a priori by experts, when processing documents in the information-analytical system. An automatic classifier functions is a program that determines the subject of documents and carries out their assignment to topics.

As there are many online sources for the news and information, the need to categorize is realized, and for this purpose, text mining is the most common approach that has served greatly in the text-classification domain. Text mining is the domain that provides the machine with intelligent algorithms that can analyze text and find patterns and make appropriate decisions [8]. There are several ways of text mining, including steps from information extraction, information retrieval, natural language processing, and ML or deep learning [9–12]. The applications of text mining include sentiment analysis [13] and emotion recognition [14], authorship attribution [15] and author gender recognition [16], recommendation systems [17,18], fake information recognition [19], text summarization [20], part-of-speech tagging [21], and medical diagnostics [22]. Text topic modeling and classification have also been used in various domains for various tasks, including electronic health record (EHR) management [23], tourism [24], etc.

ML is a general term for an algorithm that allows machines to learn from historical data to make predictions on future data. Despite recent successes of deep learning methods in various domains, machine learning methods remain relevant and competitive in the NLP research field [25]. Other methods, such as fuzzy logic [26] and heuristic optimization algorithms [27] have also been used. This study used supervised ML in which historical data contain predefined class labels. In our case, the class labels were the news categories. Before ML, natural language processing was used to clean the data from noise in the data [28].

Feature extraction is known to transform the features from text, video, or audio to less dimensional data. There is a need to transform data into a mathematical form [29]. These techniques must be selected carefully as some techniques perform better on specific data while others do not. The techniques used for feature extraction include N-Grams [30], Term Frequency—Inverse Document Frequency (TF-IDF) [31], Bag-Of-Words (BOW) [32], Word2vec [33], and Doc2Vec [34].

Categorizing news articles, several techniques are available that can perform documents the task of categorization [35]. These techniques include heuristic methods, ML, deep learning, and many more. However, in this research, the focus is on ML as it is less complex and more efficient in the case of resources [36]. Many types of ML exist, but the focus is on the more mature classification type ML like supervised ML. Supervised ML works on classifying data according to the given categories or classes. The classifiers that were used are RF, SVM, SGD, KNN, LR, and NB [37].

The rest of the paper is organized as follows. Section 2 describes the literature review where relevant studies are discussed. Section 3 discusses the adopted methodology of the experiment. Section 4 evaluates the results gathered from the experiment. Finally, Section 5 presents the conclusions observed in the research.

2. Literature Review

The authors in [38] categorized news based on important news topics for a specific region. The study used a dataset that scraped news from Twitter and performed news classification. In their study, they used articles that were related to Sri Lanka. They also identified that using many features will increase the dimension; therefore, to make the model more efficient, they reduced features. They discovered that the more frequent words in the dataset carry much less information for text classification.

Furthermore, they extended this by removing the rare words, suggesting that rarer words also do not have much importance as they mostly act as noise in the data. They evaluated their models by finding the recall and precision for each class. The evaluation was carried out for each class independently of how accurately each group was classified.

In [39] the authors also analyzed news articles to summarize them according to their topic in which topic detection and tracking are performed to group news headlines in similar groups. Their study used TF*PDF (Term Frequency * Proportional Document Frequency), which works on a similar principle of F.T.F.—IDF. In [40], the authors classified news articles by their authenticity in which they classified news articles as real or fake. Their research mainly aimed to create a feature from floating languages to carry out fake news classification. They created a dataset by taking news articles from several different publishers and labelled the dataset by examining articles as fake or real. The articles they selected were in the Slovak language, a floating language. They analyzed news articles by using morphological methods. They presented an efficient approach by which floating languages can be classified by only using Part-of-Speech (POS) with reasonable accuracy from a basic classifier.

According to the authors of [41], deep learning methods tend to achieve more accuracy in text classification of Chinese language text. They implemented four models that are CNN (Convolutional Neural Network), LSTM (Long Short-Term Memory), B-LSTM (bi-directional LSTM), and BLSTM-C (bi-directional LSTM CNN). BLSTM-C is a hybrid model formed by using layers of both BLSTM and CNN to enhance classification performance. They used three benchmarked datasets to evaluate their proposed BLSTM-C. The highest accuracy they achieved is 90.8% on THUCNews.

Furthermore, their proposed model also outperformed other researchers on similar datasets. Although their proposed model was formed with more than 29 layers and achieved better results, there is always a tradeoff between performance and complexity. Deep learning models tend to be more complex, and the training time of these models is higher than that of ML.

The authors of [42] classified Chinese comments by applying sentiment analysis. The authors introduced aspect-based sentiment analysis (ABSA), for which they proposed a hybrid attention-based aspect-level recurrent convolutional neural network (AARCNN) model. Their proposed model was designed to focus on the attention-based parts of the sentences, using important information from the sentences. The model uses the B-LSTM layer that behaves as a memory that stores important information from the sentences, and then the CNN layer was added to extract stored information and analyze it.

Many researchers have tried to show the SVM model's ability to classify text documents with great accuracy successfully, and the authors of [43] also used SVM as a base classifier to classify news articles according to their category. The categorization they performed has two types: generalized and personalized classification. The authors used the Reuters-21578 dataset to train the model in generalized classification. Their focus is to classify financial news articles into their respective categories. The authors trained SVM using general categories given in the dataset, and then the categorizer trained on positive and negative documents, positive documents are the documents in the category that is being classified. An equal number of documents are taken from other categories. Then the categorizer is trained for the given category using the imported news articles. The user can retrain the categorizer. In the retraining part, the user then creates a personalized category that can check if the trained model is retrieving relevant articles with the given category. The user mentions irrelevant documents then the classifier retrains by discarding the mentioned documents, so the classifier refines itself.

In [44], TextNetTopics is an innovative technique to feature selection that considers Bag-of-topics (BoT) rather than the usual Bag-of-words approach (BoW). As a result, rather than selecting words, the technique selects topics. TextNetTopics outperforms different feature-selection algorithms while also outperforming when used to CAMDA validation data. In addition, we tested our system on several textual datasets.

Recently, deep learning methods have started to be applied. Shao et al. [45] proposed W-LDA as a neural network topic model based on the Wasserstein Auto-Encoder that offers enhanced Wasserstein-Latent Dirichlet Allocation (W-LDA) (WAE). The preprocessed, Bag of Words (BOW), and Term Frequency-Inverse Document Frequency (TF-IDF) features are

used as input in the W-LDA. then, the the sparsemax layer is introduced after the hidden layer inferred by the encoder network to generate a sparse document–topic distribution.

Researchers in recent years have been more focused on the classification of fake news, while few studies present the categorization of news articles according to their category. The authors of [46] also applied fake detection on news articles to classify whether a news article is fake or real. In their research, they used three datasets from various sources. The first dataset contained news about the United States (U.S.) presidential elections collected from Buzzfeed in 2016. The second dataset contains random news articles about politics that the authors collected from Zimdar's for fake news and Business Insiders for real news. The last dataset contained real news articles from Reuters.com and fake news from websites such as Politifact and Wikipedia.

As the literature review suggests, not much work about news categorization has been observed from the recent studies. Moreover, most studies used benchmarked datasets. The benchmark dataset is great for learning and discover news categorization, but it is not applicable in the real world as the text in most real-world news articles are not structured as well as in the benchmarked datasets. Another observation made from studying literature is that many researchers have categorized news articles as fake or authentic news, but few researchers focused on news categorization.

3. Methods and Experimentation

3.1. Mathematical Model of Topic Classification Problem

In this section, the model of an automatic text classifier is formalized in the form of an algebraic system, within which the classification methods studied in this paper were subsequently formulated and described. We represent the general model of the text classifier as the following system:

$$R = \left\langle T\ ,\ C\ ,\ F\ ,\ R^F,\ f \right\rangle \tag{1}$$

Here R is a model for an automatic natural language text classifier, T is a collection of texts, C is a non-empty set of topics, F is a non-empty set of topic descriptions (each description contains data necessary for classification, such as lists of keywords and their importance), $R^F : C \times F$ is a relation correlating topics and descriptions corresponding to them, $f : T \rightarrow 2C$ is a classification operation such that $f(t) = \sigma$, where t is a text from T, and σ is an element of the set of all subsets of C, i.e., a set of topics from C. Thus, the mapping f allows each document of the set T to associate to some topic from C.

We will also assume that:

1. RCF relation is functional. The following property holds: $\forall\, c \in C\ \exists!\ , \phi \in F : (c, \phi) \in R$, i.e., each topic corresponds to a unique description.
2. Each description of the topic contains a set of features used by the classification operation and their values.
3. For each topic c, we define $D(c)$, which is the set of documents associated with it (a priori or with the help of the classification operation), i.e., $D(c) = \{t \in T | c \in f(t)\}$. Topic $c_i \subset c_j$ by definition, if $D(c_i) \subset D(c_j)$.
4. It can be shown that C is an upper semilattice, i.e., there is a unique element $\sigma_{top} \in C$, $\sigma \in \sigma_{top}$. Here σ_{top} is the root topic containing all other classes (topics).
5. The result of determining the topic of the document (text), i.e., the classification of the document (text) $t \in T$—is σ the set of topics to which the document corresponds.

In problems of hierarchical classification, on the set of topics C, a relation is specified that determines the a priori nesting of topics into each other. In this paper, we restricted ourselves to the consideration of a flat classification problem, in which all topics are nested in a single root topic, and there is no other nesting. The construction of a classifier implies the partial or complete formation of C , F , R^F, f based on some a priori data. In practice, this means that the expert forms a hierarchy of topics. Descriptions of topics are formed manually (for example, in the form of rules for assigning documents to topics based on

some features) or automatically using machine learning methods. The training set is a set of documents previously correlated with the topics based on expert assessments. For further consideration of texts as objects of classification and identification of significant features of classification, we used the concept of a lexical descriptor. Lexical descriptor of a natural language, means individual lexemes as sets of paradigmatic forms (word forms) of one word. Different forms of the same word do not differ; and phrases in canonical forms (the main word is reduced to the dictionary form. orm of dependent words is subject to the control of the main word) regardless of the different forms.

Let D be the set of documents (collection), W the set of all tokens used in the collection (collection dictionary). A collection document $d \in D$ is a sequence of tokens $(w_1, \ldots, w_{n_d}) \in W$, and each token w is associated with the number n_{dw} of its occurrences in the document d. The collection is represented as follows:

$$F = (f_{wd})_{W \times D} \tag{2}$$

$$f_{wd} = \frac{n_{dw}}{n_d} \tag{3}$$

We assume that several topics are distributed among the entire collection of documents. Then the collection is a set (w_i, d_i, t_i), $i = 1 \ldots n$, generated by a discrete distribution $p(w, d, t)$, defined on a finite probability space $W \times D \times T$.

The text of the document t is characterized by the set of lexical descriptors $\varphi_t = \{ w_i \}_{i=1}^{S_t}$ contained in the text, here S_t is the total number of lexical descriptors in the text. For the LD document's text, numerical characteristics (i.e., features) are defined (for example, frequency of occurrence, significance, weight in the text, etc.).

The document is characterized by a feature value vector, where ϕ is the feature space, and specific numerical values are the coordinates. The solution to the classification problem consists in choosing a suitable topic for the document t or assigning it to the root topic in the case of an open classification problem.

3.2. Dataset Description

In different datasets such as Ag_news, Reuters, and British Broadcasting Corporation (BBC), the dataset was categorized into two: breaking news and international news. Apart from these two categories, some other categories also exist, but varied from dataset to dataset. Other commonly used categories included political news, domestic news, sports news, and others. From this knowledge and the available datasets, a real-world dataset that included news articles from reuters.com was selected. These news articles are scraped from reuters.com and categorized into four categories: Domestic News, Political News, Top News, and World News.

In this research, the Reuters news dataset was used. This dataset contains 40,063 English news articles of four different categories of news articles from Reuters (www.reuters.com, accessed on 1 December 2022) covering from 2016 to 2018 years. The dataset is a real-world dataset, not a benchmarked one designed and structured for study purposes. The purpose of using this dataset is to show results on real-world data, so a system can be designed that can be implemented on real-world news articles.

The class labels of these news articles are specified in the German, namely Inlandsnachrichten, Politik, Weltnachrichten, and TopNachrichten. These categories translate to Domestic, Politics, International, and Top news in English. The dataset contains two news articles with the BreakingViews Label; these articles were discarded as there were only 2 observations, which would negatively affect the performance of the trained model. The features of a dataset are shown and described in Table 1.

Table 1. Dataset description.

Column Name	Description
Id	The numbered ID of the news article
Headline	The headline of the news article
Body	The main text of the news article
Kat	Category/Label of the news article
Date	The date news article was published

3.3. Workflow of Methodology

The research experiment's methodology includes several steps in a structured flow. The flow of the methodology shown in Figure 1 is discussed in the below sections. The methodology uses a benchmark dataset as a corpus. It consists of the data preprocessing, feature selection, cross-validation (dataset splitting), model training, prediction, and accuracy evaluation.

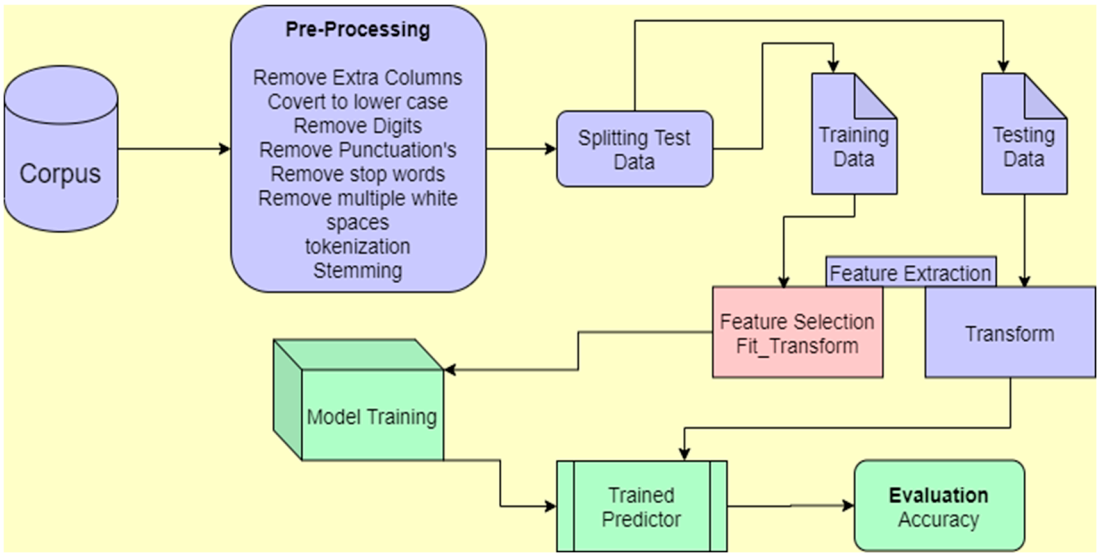


Figure 1. Dataflow diagram of the methodology used in this paper.

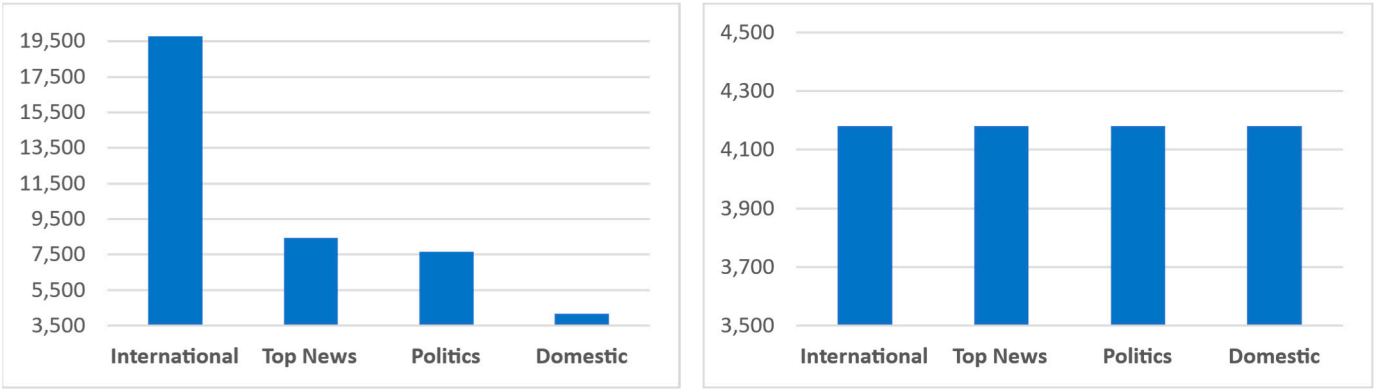
3.4. Preprocessing

Several preprocessing steps were performed on the dataset to remove the noise and structure the dataset for classification purposes. The preprocessing steps performed are discussed below.

3.4.1. Dataset Structuring and Balancing

First, headline and body of the news articles were concatenated and stored in a single column called news. A column named kat was renamed to category. After that headline, body and date columns were removed. Next, the dataset was balanced. The first two observations from the category BreakingViews were removed as this category only contains two observations. The extra articles from each category were removed.

The lowest frequency in the news articles was 4180 with the category "Inlandsnachrichten". Therefore, 4180 articles were kept from each category, and the rest were discarded. After this step, the total number of articles was 16,720. As the classes were imbalanced, this step was important as this study has an adequate number of articles to evaluate the model. The frequency plots of the dataset according to the categories are shown in Figure 2. Figure 2a shows class distribution before balancing the data, and Figure 2b shows the distribution plot of the dataset after class balancing.



(**a**) (**b**)

Figure 2. Dataset class distribution: (**a**) before dataset balancing and (**b**) after balancing.

3.4.2. Label Encoding

In this step, the class labels' categories were encoded into integers as some models do not accept class labels in text form. The categories of the articles were encoded into numeric values. In Table 2, the class labels are shown with respective encoded values.

Table 2. Label Encoding.

Category	Encoded Label
Weltnachrichten	0
TopNachrichten	1
Politik	2
Inlandsnachrichten	3

3.4.3. Article Preprocessing

In this step, several steps were performed to structure the news articles and remove unwanted terms that behave as noise in training the models. All characters were transformed into lowercase letters to keep the semantics independent of letter cases to the model. The output after case lowering can be seen in Table 3.

Table 3. Lowercase output.

Input	Output
U.S. weekly jobless claims rebound from near 45 year lows	u.s. weekly jobless claims rebound from near 45 year lows
'Thomson Reuters says CEO Jim Smith to make full recovery after an arrhythmia incident	thomson reuters says ceo jim smith to make full recovery after an arrhythmia incident
'Trump says FBI missed signs on Florida shooting due to Russia probe, draws criticism	trump says fbi missed signs on florida shooting due to russia probe, draws criticism
'Moscow says no evidence behind U.S. indictment of Russians for alleged election meddling	moscow says no evidence behind u.s. indictment of russians for alleged election meddling
'Do you fear me?': Venezuela's Maduro vows to gatecrash regional summit	'do you fear me?': venezuela's maduro vows to gatecrash regional summit

In the next stage of preprocessing, all the special characters and numeric values were removed from the articles as they do not contribute to the classification of news articles; rather, they serve as noise in the data. Special characters include punctuation marks, ad other characters such "$", "#", "@", and others. The output after removing these characters can be seen in Table 4.

Table 4. Output after removal of digits and special characters.

Input	Output
u.s. weekly jobless claims rebound from near 45 year lows	u s weekly jobless claims rebound from near year lows
thomson reuters says ceo jim smith to make full recovery after arrhythmia incident	thomson reuters says ceo jim smith to make full recovery after arrhythmia incident
trump says fbi missed signs on florida shooting due to russia probe, draws criticism	trump says fbi missed signs on florida shooting due to russia probe draws criticism
moscow says no evidence behind u.s. indictment of russians for alleged election meddling	moscow says no evidence behind u s indictment of russians for alleged election meddling
'do you fear me?': venezuela's maduro vows to gatecrash regional summit	do you fear me venezuela s maduro vows to gatecrash regional summit

Removing special and numeric entities leaves unwanted blank spaces. These extra whitespaces were also removed and replaced with a single white space. Subsequently, stopwords were also removed using a predefined list of stopwords in Python. After removing stopwords and multiple spaces, the output is shown in Table 5.

Table 5. Output after removing stopwords and extra white spaces.

Input	Output
u s weekly jobless claims rebound from near year lows	u weekly jobless claims rebound near year lows
thomson reuters says ceo jim smith to make full recovery after arrhythmia incident	Thomson reuters says ceo jim smith make full recovery arrhythmia incident
trump says fbi missed signs on florida shooting due to russia probe draws criticism	Trump says fbi missed signs florida shooting due Russia probe draws criticism
moscow says no evidence behind u s indictment of russians for alleged election meddling	moscow says evidence behind u indictment Russians alleged election meddling
do you fear me venezuela s maduro vows to gatecrash regional summit	Fear Venezuela maduro vows gatecrash regional summit

Then the news articles were tokenized and stemmed. Tokenization is the process where each document is divided into separate words, while stemming is the process of transforming a word into its root form. As for a human being, it is easy to understand that two words "go" and "going" have the same meaning and are used in different contexts, but this is not the case with the ML algorithm. With the help of stemming, the model can get to the root. For stemming, the Porter Stemmer algorithm was used, and the output of stemming and tokenization is shown in Table 6.

Table 6. Output after tokenizing and stemming.

Input	Output
u weekly jobless claims rebound near year lows	['u', 'weekli', 'jobless', 'claim', 'rebound', 'near', 'year', 'low']
Thomson reuters says ceo jim smith make full recovery arrhythmia incident	['thomson', 'reuter', 'say', 'ceo', 'jim', 'smith', 'make', 'full', 'recoveri', 'arrhythmia', 'incid']
Trump says fbi missed signs florida shooting due Russia probe draws criticism	['trump', 'say', 'fbi', 'miss', 'sign', 'florida', 'shoot', 'due', 'russia', 'probe', 'draw', 'critic']
moscow says evidence behind u indictment Russians alleged election meddling	['moscow', 'say', 'evid', 'behind', 'u', 'indict', 'russian', 'alleg', 'elect', 'meddl']
Fear Venezuela maduro vows gatecrash regional summit	['fear', 'venezuela', 'maduro', 'vow', 'gatecrash', 'region', 'summit']

3.5. Feature Extraction

In this step, document features were converted into vector form; as the ML algorithms work in the mathematical form, these models cannot understand the text form of the feature. To convert features into vectors, the TFIDF technique was used as it is an efficient and reliable technique as in categorizing news articles; word order is not much important, so more complex techniques such as Word2Vec and Doc2Vec are not focused.

In TFIDF, two different terms are used: TF and IDF. These two terms define two different types of frequencies for each feature. After combining these two frequencies, each feature is formed. TF defines the Term Frequency, which is the frequency of a word in a document [47], which calculates how many times a single term has occurred in a single document from all the terms present in a document.

Let tf be the word frequency in the given document and N be the total number of documents in the corpus. IDF is the document frequency of a term, such as in how many documents a term has appeared. IDF of a term is independent of the frequency of a term in the entire corpus.

$$idf(t, D) = \log \frac{|D|}{|(d_i \supset t_i|} \tag{4}$$

where $|D|$—is the number of documents in the corpus; and $|(d_i \supset t_i|$ is the number of documents in which the term t_i Is present.

The effect of IDF can be noted as the frequency of a term increases in the corpus, the IDF will also increase. When both the TF and IDF combine, this results in a feature that is not biased to the frequency of a term. As it takes both the document and corpus frequency the resultant feature will be an averaged vector that does not favor the terms with higher frequency in a document. This behavior of TFIDF has helped researchers to achieve higher accuracies in solving problems where the order of words does not affect results. With the help of TFIDF, it can be easily determined which words are more common to a group of documents [48]. Terms that appear more often in a group of documents will have a higher score. In this way, the more common words can be filtered, and rarer words can also be filtered by using a threshold for a lower score. In this way, rarer words can be less important for a classification model as they will have a lower weight.

The mathematical working of TFIDF can be understood by Equation (5), where D is the corpus of documents and d is the document for which the score is being calculated. w_d is the TFIDF score for a word w in a document d. This study calculated a score for an individual word for every document by using Equation (5).

$$w_d = f_{w,d} * \log\left(\frac{|D|}{f_{w,D}}\right) \tag{5}$$

where $f_{w,d}$ the frequency of a word is in a document. $f_{w,D}$ is the frequency of a word in the corpus or a set of documents D. The resultant w_d gives insight into the terms and whether they are more important to the given set of documents in the form of vectors.

$$TFIDF\ (\ w,\ c\ ) = TF\ (\ w,\ c\ )\ \cdot\ IDF\ (\ w,\ D\ (\ c))$$

Here the vector form achieves two goals for training a model: (a) vector space is greatly reduced by comparing to the alphabetic form, and (b) it is in an understandable form for the ML algorithms.

First, the dataset was divided into two sets to convert features into words: a training set and a testing set. The training set was used to train the model, and the testing set was used to evaluate the model after the training was completed. For splitting the dataset, the 70/30 ratio was used [49], where 70% of the articles were used for training the model, and 30% were used for testing.

In feature extraction, two methods were used; one is used to create feature space and subsequently transform it into vector space for training and testing data separately. In the

transformation process number of features is restricted, and not all the features are used. Only features are considered that lie in document frequency ranging between 0.02% and 70%. This is also known as dimensionality reduction. This left 23,713 feature instances. However, the number of features varied when the whole process was carried out; at the start, while balancing the dataset, random news articles were discarded.

3.6. Model Training

In this step, both the articles have now been converted into feature vectors, and their categories, which are the class labels, are fed into five different ML models: LR, NB, SGD, KNN, RF, and hyperparameter-optimized SVM; then, the models were trained using this training data. This step was performed repetitively to optimize these models by their tuning parameters. This step is called hyperparameter tuning.

Each model has a different set of parameters which can be changed to make the model more efficient to train with given features. SVM has many different parameters. The important parameters that may affect the model's performance are C, kernel, and decision_function_shape. From which, the only kernel that affected the performance of the model. SVM has four different types of kernels, which include linear, RBF, poly, and sigmoid. The kernel function determines how the hyperplane of SVM will be drawn. The default value of the given parameters are as follows: $C = 1$, $kernel = rbf$ and $decision\_function\_shape = ovr$. C parameter is a regularization function to minimize the error rate of training and testing of the model. The kernel of SVM selects how the hyperplane will be selected to distinguish between classes. A summary of hyper-parameters used in each algorithm is given in Table 7.

Table 7. Optimized hyper-parameters used for the machine learning models.

Model	Hyper-Parameters
RF	n_estimators = 3000, max_depth = 100
NB	Alfa is reduced to 0.01
SGD	max_depth = 10, average = True
KNN	Default setting K = 9
SVM	Kernel = rbf, C = 1.0
LR	Solver=saga, C = 2.8

3.7. Model Evaluation

The evaluation of trained models is an important task to evaluate the model's performance [49]. By evaluating models, one can compare different classifiers and see if the model has achieved an acceptable amount of accuracy in prediction. There are many ways a model can be evaluated: accuracy, precision, n-fold cross-validation, recall, and f1-score [50]. The more focused technique used in this study is accuracy [51], because the amount of data is sufficiently large, and accuracy is mostly more suitable for the dataset with hundreds of observations with evenly distributed class labels [52]. Accuracy is a ratio of correctly classified predictions from all the predictions. Accuracy is simple to understand and implement. It does not consider the class imbalance, which means it evaluates all the observations as the same class. Accuracy is meant to be more important and more useful than other performance measures when the amount of data is large and has an even class distribution [53].

For performance measures, only accuracy is used as the articles were class balanced in the preprocessing stage so the need for other performance measures is unnecessary. Furthermore, other performance measures were implemented on the models at the start, but all of the measures calculated the same results as the classes were balanced in the training data as well as testing data. In this stage, to evaluate the trained model, first, the model was used to predict categories for the test data, and then the accuracy measure was used to evaluate the models. Accuracy is the performance measure that determines

how often an observation is classified correctly. Accuracy can be calculated by using the following equation using parameters from the confusion matrix.

$$acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

Here accuracy (*acc*) is the ratio of truly predicted from all the examples. *TP* is the example that is positives and was classified as positives. *TN* are the examples that are negative and were classified as negative. *FN* are examples that were positive and were classified as negative. *FP* are examples that were negative and were classified as positive.

4. Results

In this section, the results obtained after the evaluation of each model are discussed. Each of the six classifiers, including hyperparameter-optimized SVM, performed well in the purpose of classifying news articles according to their respective category and achieved very good accuracies.

However, hyperparameter tuning optimizes each classifier to the highest possible accuracy. The focused classifier only required one parameter to be tuned, the kernel. The kernel of SVM is set to "linear". Other models are also optimized by hyperparameter tuning to obtain justified results. The "C" parameter of LR is tuned with the value of "2.8" which is the inverse of the regularization strength of the L.R. N.B. was also optimized, and alpha was reduced to "0.01", which is reduced to reduce the smoothing function. SGD classifier is optimized by changing its "average" to "True", which allows SGD to calculate the weighted average of the observations. In KNN, "n_neighbors" was selected to be 9, in which it classifies observations using nine relevant observations, and the "weights" was tuned to "Distance" so that more relevant observations will have more weight. Lastly, RF was optimized, and "n_estimators" were tuned to "3000", which means "3000" DTs will be trained in RF, moreover "max_features" were tuned to "100", so that at each split 100 features are considered. After the final models are trained, the accuracies that are achieved can be seen in Table 8 and Figure 3.

Table 8. Model performance without and with optimization.

Classifier	Accuracy (w/o Optimization)	Accuracy (With Optimization)
SVM	0.6435	0.8516
SGD	0.8480	0.8476
LR	0.8437	0.8470
RF	0.7587	0.8110
NB	0.8106	0.8183
KNN	0.8104	0.8135

By taking a close look, it can be easily understood that SVM outperformed all the other classifiers in the classification of real-world news articles. However, SGD, KNN, and L.R. also competed for the best-performing classifier, and their results were not much lower than SVM. RF is a great model in text classification, but in this scenario, R.F. failed in competing with other classifiers and provided the worst performance. The evaluation of models without parameter tuning was also carried out to see which algorithm performs better in classifying news articles, and the results are shown in Table 8 and as a bar plot in Figure 3.

After assessing the results shown in the above table, it can be observed that the model that provided the best accuracy was SGD and LR was also not far behind. RF, KNN, and NB also provided average results. In models that are not hyperparameter, tuned SVM was the worst-predicting classifier.

The discussed models were also tested on the benchmarked dataset ag_news. The results from these datasets showed that these models can achieve higher accuracies, but

benchmarked datasets are highly structured datasets containing as little noise as possible. The results from a benchmarked dataset are shown in Table 9. Results among tuned models and models that were not tuned by hyperparameter tuning are compared in Figure 3.

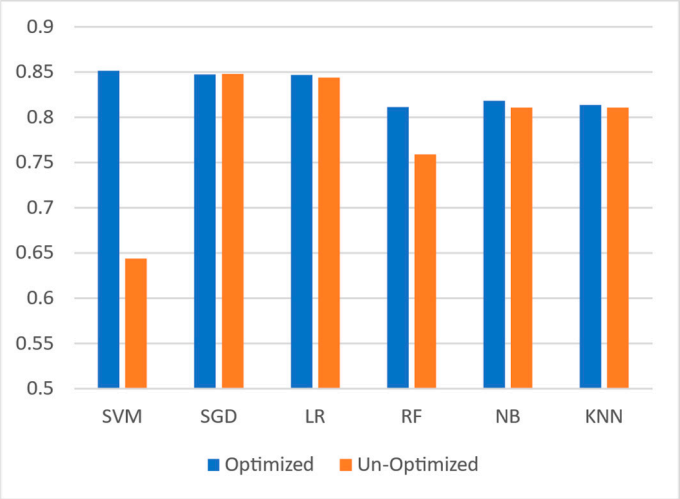


Figure 3. Comparison of optimized and unoptimized models.

Table 9. Accuracy of models on the benchmark dataset.

Classifier	Accuracy
SVM	0.9130
SGD	0.9186
LR	0.9126
RF	0.9118
NB	0.8995

By comparing the results among tuned models and models that are not tuned, SGD, KNN, and LR had quite similar results, which means the performance of these models was not much improved even with hyperparameter tuning for the news classification. SVM and RF showed significant improvement in performance after tuning, where RF has increased by 5.23% accuracy and SVM improved after tuning with a 20.81% improvement in accuracy. SVM, the worst-performing classifier without hyperparameter tuning, became the best-performing model after hyperparameter tuning.

Figure 4 shows the comparison between different kernels of SVM. These results show that when kernels are changed in the SVM classifier, the difference in accuracy is quite noticeable. The results also imply that linear classifier works better in text classification most importantly when the main objective is to classify text according to the genre.

In Figure 5a, the confusion matrix shows the classification for each class when the kernel was set to linear. SVM achieved the best accuracy as the diagonal of the confusion matrix showed the highest amount of prediction, and as the matrix showed the accurate prediction in the diagonal, the confusion matrix clearly showed that most of the predictions were made correctly. Figure 5b represents a confusion matrix with the sigmoid kernel; Figure 5c shows the confusion matrix with the polynomial kernel, and Figure 5d shows the confusion matrix with the RBF kernel. These figures help us access how SVM reacts to different kernels with this dataset. From this confusion matrix, it is observed that the hyperplane for the domestic class was not being calculated correctly, and it was being identified on the feature space of other classes. Confusion matrixes below showed that when the kernel was not linear, most predictions were incorrectly classified as domestic. Figure 5b shows that many predictions were made incorrectly, but almost all the predictions from the domestic class were made correctly. This shows the kernel's behavior; it makes

the model biased toward the domestic class. In Figure 5c, the confusion matrix showed that almost every observation from classes other than domestic was made incorrectly, and the algorithm failed to classify news articles. This behavior happens because the domestic class's hyperplane covers almost all the feature space.

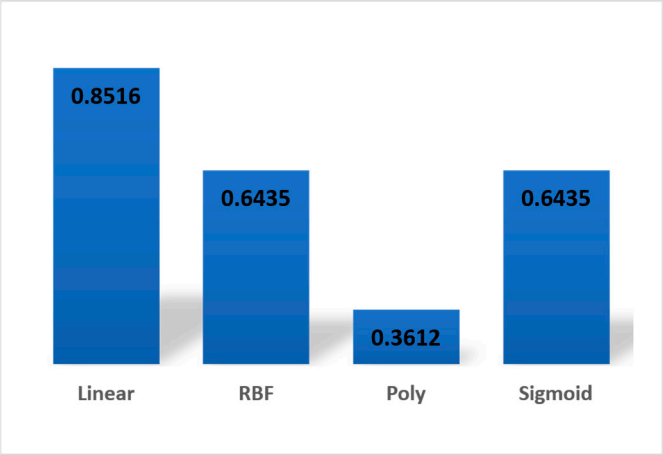


Figure 4. Comparison of SVM kernel performance.

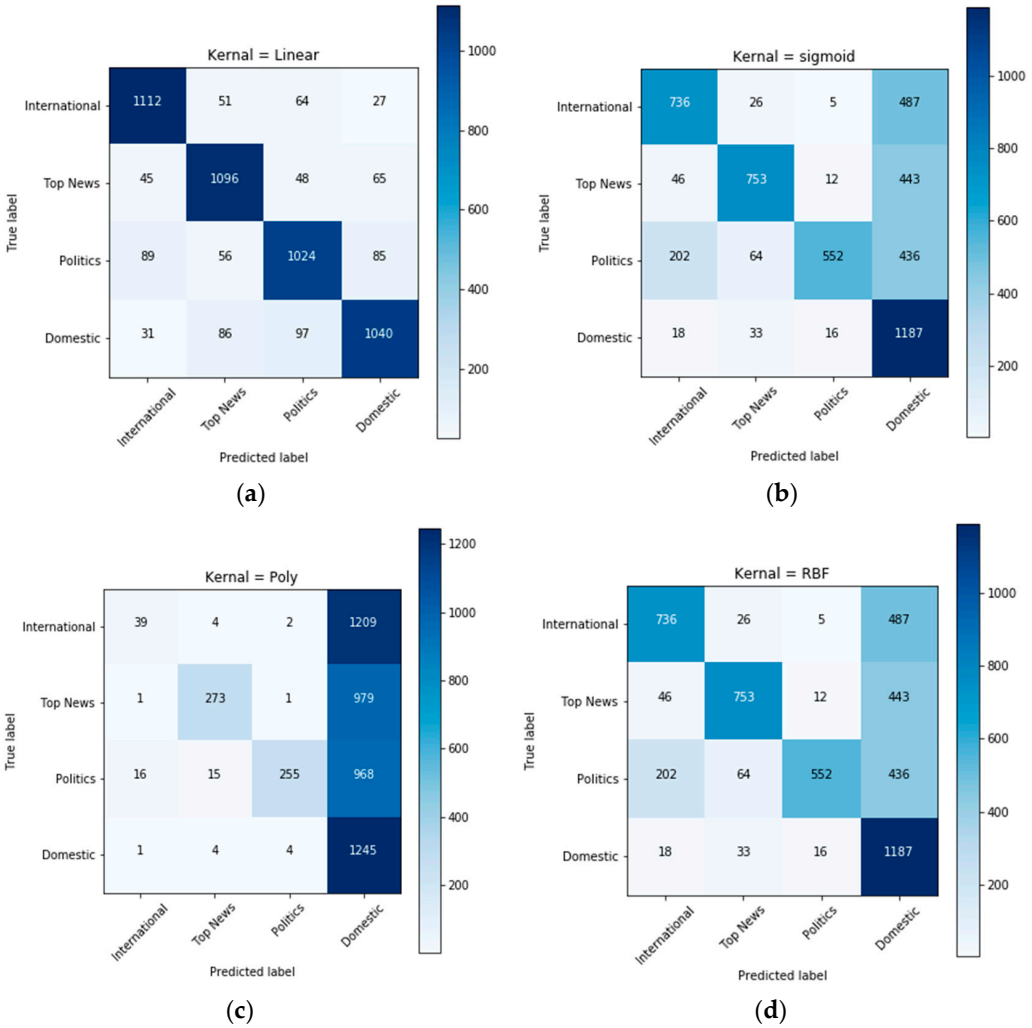


Figure 5. Comparison of SVM kernels: (**a**) linear, (**b**) sigmoid, (**c**) polynomial, (**d**) RBF.

Our results outperformed the results of a recent study [54] using deep learning (LSTM with heuristc hyena optimization algorithma), which achieved only 63.81% accuracy on the Reuters dataset.

5. Conclusions

We performed an in-depth study on text classification using ML, more specifically news article classification in which news articles were classified according to their respective categories. The focus of this research was to study and learn the significance of SVM for news articles classification when it is optimized using hyperparameter tuning. In this study, other popular classifiers in the domain of text classification were optimized and compared with SVM. The purpose of the study was successfully achieved by creating the SVM model and optimizing it by hyperparameter tuning, achieving the best accuracy in classifying news articles. This study also showed the significance of the hyperparameter tuning and its effect on the model's performance as it achieved a 20.81% accuracy improvement for the SVM. Using a real-world dataset, a model was trained that can be used in real-world applications to classify news articles with acceptable accuracy.

Author Contributions: Conceptualization, T.S. and A.S.; methodology, T.S. and A.S.; software, S.D., M.U. and A.R.; validation, S.D., M.U., A.R., T.S., R.D. and A.S.; formal analysis, S.D., M.U., A.R., T.S., R.D. and A.S.; investigation, S.D., M.U., A.R., T.S., R.D. and A.S.; data curation, S.D., M.U. and A.R.; writing—original draft preparation, S.D., M.U., A.R. and A.S.; writing—review and editing, T.S. and R.D.; visualization, S.D., M.U. and A.R.; supervision, T.S.; funding acquisition, R.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: In this study a publicly available Reuters news dataset was used.

Acknowledgments: This research is technically supported by Artificial Intelligence & Data Analytics Lab (AIDA) CCIS Prince Sultan University, Riyadh, Saudi Arabia.

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

References

1. Strömbäck, J.; Karlsson, M.; Hopmann, D.N. Determinants of News Content. *J. Stud.* **2012**, *13*, 718–728. [CrossRef]
2. Mitchell, A.; Rosenstiel, T. Navigating News Online: Where People Go, How They Get There and What Lures Them Away. PEW Research Center's Project for Excellence in Journalism. 2011. Available online: http://www.journalism.org/2011/05/09/navigatingnewsonline/ (accessed on 8 January 2022).
3. Harouni, M.; Rahim, M.S.M.; Al-Rodhaan, M.; Saba, T.; Rehman, A.; Al-Dhelaan, A. Online Persian/Arabic script classification without contextual information. *Imaging Sci. J.* **2014**, *62*, 437–448. [CrossRef]
4. Bakshy, E.; Rosenn, I.; Marlow, C.; Adamic, L. The Role of Social Networks in Information Diffusion. In Proceedings of the WWW 2012: 21st World Wide Web Conference, Lyon, France, 16–20 April 2012; pp. 519–528. [CrossRef]
5. Bennett, W.L.; Iyengar, S. A New Era of Minimal Effects? The Changing Foundations of Political Communication. *J. Commun.* **2008**, *58*, 707–731. [CrossRef]
6. Rehman, A.; Saba, T. Off-line cursive script recognition: Current advances, comparisons and remaining problems. *Artif. Intell. Rev.* **2012**, *37*, 261–288. [CrossRef]
7. Kull, S.; Ramsay, C.; Lewis, E. Media, Misperceptions, and the Iraq War. *Polit. Sci. Q.* **2003**, *118*, 569–598. [CrossRef]
8. Chen, Z.Q.; Zhang, G.X. Survey of text mining, Pattern Recognit. *Artif. Intell.* **2005**, *18*, 65–74. [CrossRef]
9. Schutze, H.; Manning, C.D.; Raghavan, P. *Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008.
10. Javed, R.; Rahim, M.S.M.; Saba, T.; Rehman, A. A comparative study of features selection for skin lesion detection from dermoscopic images. *Netw. Model. Anal. Health Inform. Bioinform.* **2020**, *9*, 1–13. [CrossRef]
11. Larabi-Marie-Sainte, S.; Aburahmah, L.; Almohaini, R.; Saba, T. Current Techniques for Diabetes Prediction: Review and Case Study. *Appl. Sci.* **2019**, *9*, 4604. [CrossRef]
12. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural Language Processing (Almost) from Scratch. *J. Mach. Learn. Res.* **2011**, *12*, 2493–2537. [CrossRef]

13. Rehman, A.; Saba, T. Performance analysis of character segmentation approach for cursive script recognition on benchmark database. *Digit. Signal Process.* **2011**, *21*, 486–490. [CrossRef]
14. 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]
15. Saba, T.; Rehman, A.; Altameem, A.; Uddin, M. Annotated comparisons of proposed preprocessing techniques for script recognition. *Neural Comput. Appl.* **2014**, *25*, 1337–1347. [CrossRef]
16. Dalyan, T.; Ayral, H.; Özdemir, Ö. A Comprehensive Study of Learning Approaches for Author Gender Identification. *Inf. Technol. Control* **2022**, *51*, 429–445. [CrossRef]
17. Shambour, Q.Y.; Abu-Shareha, A.A.; Abualhaj, M.M. A Hotel Recommender System Based on Multi-Criteria Collaborative Filtering. *Inf. Technol. Control* **2020**, *51*, 390–402. [CrossRef]
18. Wei, W.; Wang, Z.; Fu, C.; Damaševičius, R.; Scherer, R.; Woźniak, M. Intelligent recommendation of related items based on naive bayes and collaborative filtering combination model. *J. Phys. Conf. Ser.* **2020**, *1682*, 012043. [CrossRef]
19. Tesfagergish, S.G.; Damaševičius, R.; Kapočiūtė-Dzikienė, J. Deep fake recognition in tweets using text augmentation, word embeddings and deep learning. In *Computational Science and Its Applications, ICCSA 2021*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2021; Volume 12954, pp. 523–538. [CrossRef]
20. Jiang, M.; Zou, Y.; Xu, J.; Zhang, M. GATSum: Graph-Based Topic-Aware Abstract Text Summarization. *Inf. Technol. Control* **2022**, *51*, 345–355. [CrossRef]
21. Kapočiūtė-Dzikienė, J.; Tesfagergish, S.G. Part-of-Speech Tagging via Deep Neural Networks for Northern-Ethiopic Languages. *Inf. Technol. Control* **2020**, *49*, 482–494. [CrossRef]
22. 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. Health Eng.* **2020**, *2020*, 8839524. [CrossRef]
23. Rijcken, E.; Kaymak, U.; Scheepers, F.; Mosteiro, P.; Zervanou, K.; Spruit, M. Topic Modeling for Interpretable Text Classification from EHRs. *Front. Big Data* **2022**, *5*, 846930. [CrossRef]
24. Chang, I.-C.; Horng, J.-S.; Liu, C.-H.; Chou, S.-F.; Yu, T.-Y. Exploration of Topic Classification in the Tourism Field with Text Mining Technology—A Case Study of the Academic Journal Papers. *Sustainability* **2022**, *14*, 4053. [CrossRef]
25. Kapočiūtė-Dzikienė, J.; Damaševičius, R.; Woźniak, M. Sentiment analysis of lithuanian texts using deep learning methods. In *Information and Software Technologies. ICIST 2018*; Communications in Computer and Information Science; Springer: Berlin/Heidelberg, Germany, 2018; Volume 920, pp. 521–532. [CrossRef]
26. 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 2016, Athens, Greece, 6–9 December 2016. [CrossRef]
27. 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. *Futur. Internet* **2022**, *14*, 194. [CrossRef]
28. Zhang, X.; LeCun, Y. Text Understanding from Scratch. *arXiv* **2015**, arXiv:1502.01710.
29. Jadooki, S.; Mohamad, D.; Saba, T.; Almazyad, A.S.; Rehman, A. Fused features mining for depth-based hand gesture recognition to classify blind human communication. *Neural Comput. Appl.* **2017**, *28*, 3285–3294. [CrossRef]
30. Sidorov, G.; Velasquez, F.; Stamatatos, E.; Gelbukh, A.; Chanona-Hernández, L. Syntactic N-grams as machine learning features for natural language processing. *Expert Syst. Appl.* **2014**, *41*, 853–860. [CrossRef]
31. Ramos, J. Using tf-idf to determine word relevance in document queries. *Proc. First Instr. Conf. Mach. Learn.* **2003**, *242*, 29–48.
32. Wallach, H.M. Topic Modeling: Beyond Bag-of-Words. In Proceedings of the ICML '06: 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006; pp. 977–984. [CrossRef]
33. 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]
34. Shuai, Q.; Huang, Y.; Jin, L.; Pang, L. Sentiment Analysis on Chinese Hotel Reviews with Doc2Vec and Classifiers. In Proceedings of the 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 12–14 October 2018; pp. 1171–1174. [CrossRef]
35. Umakanth, N.; Santhi, S. Classification and ranking of trending topics in twitter using tweets text. *J. Crit. Rev.* **2020**, *7*, 895–899. [CrossRef]
36. Domingos, P. A Few Useful Things to Know about Machine Learning. *Commun. ACM* **2012**, *55*, 79–88. [CrossRef]
37. Yar, H.; Hussain, T.; Khan, Z.A.; Koundal, D.; Lee, M.Y.; Baik, S.W. Vision Sensor-Based Real-Time Fire Detection in Resource-Constrained IoT Environments. *Comput. Intell. Neurosci.* **2021**, *2021*, 5195508. [CrossRef]
38. Dilrukshi, I.; De Zoysa, K. Twitter news classification: Theoretical and practical comparison of SVM against Naive Bayes algorithms. In Proceedings of the 2013 International Conference on Advances in ICT for Emerging Regions (ICTer), Colombo, Sri Lanka, 11–15 December 2013. [CrossRef]
39. Bun, K.K.; Ishizuka, M. Topic extraction from news archive using TF*PDF algorithm. In Proceedings of the Third International Conference on Web Information Systems Engineering, 2002. WISE 2002, Singapore, 14 December 2002. [CrossRef]
40. Kapusta, J.; Obonya, J. Improvement of Misleading and Fake News Classification for Flective Languages by Morphological Group Analysis. *Informatics* **2020**, *7*, 4. [CrossRef]

41. Li, Y.; Wang, X.; Xu, P. Chinese Text Classification Model Based on Deep Learning. *Futur. Internet* **2018**, *10*, 113. [CrossRef]
42. Zhu, Y.; Gao, X.; Zhang, W.; Liu, S.; Zhang, Y. A Bi-Directional LSTM-CNN Model with Attention for Aspect-Level Text Classification. *Futur. Internet* **2018**, *10*, 116. [CrossRef]
43. Debole, F.; Sebastiani, F. Supervised Term Weighting for Automated Text Categorization. In *Text Mining and its Applications: Studies in Fuzziness and Soft Computing*; Sirmakessis, S., Ed.; Association for Computing Machinery: New York, NY, USA, 2004; Volume 138, pp. 81–97. [CrossRef]
44. Yousef, M.; Voskergian, D. TextNetTopics: Text Classification Based Word Grouping as Topics and Topics' Scoring. *Front. Genet.* **2022**, *13*, 893378. [CrossRef] [PubMed]
45. Shao, D.; Li, C.; Huang, C.; An, Q.; Xiang, Y.; Guo, J.; He, J. The short texts classification based on neural network topic model. *J. Intell. Fuzzy Syst.* **2022**, *42*, 2143–2155. [CrossRef]
46. Ozbay, F.A.; Alatas, B. Fake news detection within online social media using supervised artificial intelligence algorithms. *Phys. A Stat. Mech. Its Appl.* **2019**, *540*, 123174. [CrossRef]
47. Zhang, W.; Yoshida, T.; Tang, X. A comparative study of TF*IDF, LSI and multi-words for text classification. *Expert Syst. Appl.* **2011**, *38*, 2758–2765. [CrossRef]
48. Hiemstra, D. A probabilistic justification for using tf $\times$ idf term weighting in information retrieval. *Int. J. Digit. Libr.* **2000**, *3*, 131–139. [CrossRef]
49. Gholamy, A.; Kreinovich, V.; Kosheleva, O. *Why 70/30 or 80/20 Relation Between Training and Testing Sets: A Pedagogical Explanation*; Departmental Technical Reports (C.S.): El Paso, TX, USA, 2018; pp. 1–7.
50. Goutte, C.; Gaussier, E. A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In *Advances in Information Retrieval*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2005; pp. 345–359.
51. Sokolova, M.; Lapalme, G. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* **2009**, *45*, 427–437. [CrossRef]
52. Rehman, A. Neural computing for online Arabic handwriting recognition using hard stroke features mining. *Int. J. Innov. Comput. Inf. Control* **2021**, *17*, 171–191.
53. Meethongjan, K.; Dzulkifli, M.; Rehman, A.; Altameem, A.; Saba, T. An Intelligent Fused Approach for Face Recognition. *J. Intell. Syst.* **2013**, *22*, 197–212. [CrossRef]
54. Maragheh, H.K.; Gharehchopogh, F.S.; Majidzadeh, K.; Sangar, A.B. A New Hybrid Based on Long Short-Term Memory Network with Spotted Hyena Optimization Algorithm for Multi-Label Text Classification. *Mathematics* **2022**, *10*, 488. [CrossRef]