applied sciences

MDPI

Article

Adaptive Evolutionary Computing Ensemble Learning Model for Sentiment Analysis

Xiao-Yang Liu [1,*], Kang-Qi Zhang [1], Giacomo Fiumara [2], Pasquale De Meo [3] and Annamaria Ficara [4]

[1] Department of Computer Science and Engineering, Chongqing University of Technology, Chongqing 400054, China; zkq_1999@163.com
[2] Department of Mathematics and Computer Science, Physical Sciences and Earth Sciences, University of Messina, 98166 Messina, Italy; gfiumara@unime.it
[3] Department of Ancient and Modern Civilizations, University of Messina, 98168 Messina, Italy; pdemeo@unime.it
[4] Department of Cognitive Science, Education and Cultural Studies, University of Messina, 98121 Messina, Italy; aficara@unime.it
* Correspondence: lxy3103@cqut.edu.cn

Abstract: Standard machine learning and deep learning architectures have been widely used in the field of sentiment analysis, but their performance is unsatisfactory if the input texts are short (e.g., social media posts). Specifically, the accuracy of standard machine learning methods crucially depends on the richness and completeness of the features used to represent the texts, and in the case of short messages, it is often difficult to obtain high-quality features. Conversely, methods based on deep learning can achieve better expressiveness, but these methods are computationally demanding and often suffer from over-fitting. This paper proposes a new adaptive evolutionary computational integrated learning model (AdaECELM) to overcome the problems encountered by traditional machine learning and deep learning models in sentiment analysis for short texts. AdaECELM consists of three phases: *feature selection*, *sub classifier training*, and *global integration learning*. First, a grid search is used for feature extraction and selection of term frequency-inverse document frequency (TF-IDF). Second, cuckoo search (CS) is introduced to optimize the combined hyperparameters in the sub-classifier support vector machine (SVM). Finally, the training set is divided into different feature subsets for sub-classifier training, and then the trained sub-classifiers are integrated and learned using the AdaBoost integrated soft voting method. Extensive experiments were conducted on six real polar sentiment analysis data sets. The results show that the AdaECELM model outperforms the traditional ML comparison methods according to evaluation metrics such as accuracy, precision, recall, and F1-score in all cases, and we report an improvement in accuracy exceeding 4.5%, the second-best competitor.

Keywords: ensemble learning; evolutionary computing; support vector machine; feature selection; natural language processing; sentiment analysis

Citation: Liu, X.-Y.; Zhang, K.-Q.; Fiumara, G.; Meo, P.D.; Ficara, A. Adaptive Evolutionary Computing Ensemble Learning Model for Sentiment Analysis. *Appl. Sci.* **2024**, *14*, 6802. https://doi.org/10.3390/app14156802

Academic Editor: Alessandro Di Nuovo

Received: 19 July 2024
Revised: 30 July 2024
Accepted: 2 August 2024
Published: 4 August 2024

1. Introduction

1.1. Background

With the rapid development of artificial intelligence, big data, and the Internet of Things, online social network platforms have risen rapidly, generating a lot of social news, user comments, conversation records, and other text information [1,2]. Sentiment analysis (SA) has received considerable attention among researchers working in areas such as natural language processing, computational social science, and marketing [3]. Mainstream SA methods are mainly divided into two categories: (1) shallow learning models based on feature extraction, feature selection, and machine learning (ML) classifiers and (2) deep learning (DL) models based on collaboration between preprocessing stage and deep learning technology. However, standard ML methods for sentiment analysis of short texts have poor performance because it is hard to manually obtain high-quality features to represent short texts [4,5]. In contrast, deep learning models often incur overfitting and suffer from high computational costs [6].

We point out that a better feature extraction scheme coupled with a better machine learning model could significantly improve the performance of an SA system designed to work on short texts. On the one hand, in fact, advanced text feature extraction schemes can more accurately express semantic information in short text data [7]. On the other hand, a more refined machine learning model can obtain very accurate results by using a small amount of computing resources with the advantage of exploiting a relatively simple model structure [8,9].

Previous research work on SA for short texts applied standard machine learning models such as support vector machine (SVM), mainly due to their ability to deal with nonlinear relationships [10,11]. Luo et al. [12] found that SVM had better classification accuracy than other classifiers when feature redundancy was small. In the context of Arabic text classification, Marie-Sainte et al. [13] proposed a hybrid model combining the firefly feature selection scheme and SVM to perform text classification; the proposed system achieved an accuracy of 99.40% on OSAC data set. Li et al. [14] reviewed the development of machine learning and deep learning methods for NLP tasks in the past decade, especially in text sentiment classification models and feature extraction methods. Li et al. [14] pointed out that DL models have achieved great success in several NLP tasks, but the advantages of linear classifiers such as SVM in some text mining tasks cannot be ignored [15]. However, the classical SVM model does not perform well in most text analysis, and integrating other mechanisms, strategies, and schemes is useful. Zhang et al. [16] used a locust optimization algorithm to select the optimal values of SVM parameters and formed a classifier with strong adaptability for intelligent fault diagnosis of rotating machinery. Zhou et al. [17] adopted the same optimization idea to propose the WOA-SVM model, which achieved an accuracy of 95.65%. Inspired by the above work, we will use the cuckoo search algorithm to optimize SVM and adapt it to the complexity of semantic mining in sentiment analysis tasks.

However, when faced with large-scale complex tasks such as infrared small target monitoring [18], autonomous driving [19], and natural language understanding [20], strong classifiers are unable to obtain reasonable and reliable results during validation.

A promising strategy to improve the performance of a standard classifier is to consider a pool of classification algorithms and to properly *integrate* these algorithms to generate the final prediction. Kazmaier et al. [21] applied several heterogeneous integration strategies to explore the configuration of sentiment analysis data sets from different industries. The experimental comparison results found that the integrated model had stronger text sentiment classification ability than the optimal single strong classifier, and the median of several integrated models was 5.53% higher than that of the strong classifier. Kazmaier et al. [21] claimed that the integrated approach avoids the expensive computational consumption of a single strong classifier while integrating a weak classifier for retraining can effectively avoid the semantic mining ability of a single model. Bountakas et al. [22] proposed a new spam mailbox detection method, HELPHED, which employed Stacking and soft voting. Integrated approaches can process features in parallel, which yields significant computational advantages. Moreover, the joint application of multiple classification algorithms (known in the literature as ensemble learning) leads to better classification accuracy and improves the generalization ability of an ML approach. However, ensemble learning methods have non-negligible weaknesses: for example, the performance of some classifiers depends crucially on some hyperparameters, and therefore, effective methods are mandatory to determine the best hyperparameter configuration for each classifier before proceeding to their integration.

This paper introduces a new computing framework AdaECELM, which uses the optimal ML model as the base classifier for sentiment analysis of short texts. The model integrates three modules: TF-IDF feature selection optimization processing, CS-SVM adaptive optimization subclassifies and soft voting comprehensive learning, and finally obtains an ensemble learning model with effective performance, stable effect, and strong generalization ability.

1.2. Our Contributions

The main contributions of this paper are as follows:

1. TF-IDF was used to extract text features, and grid search was used to select the frequency and optimize the features according to the sentence size of different data sets.
2. The intelligent optimization algorithm cuckoo search (CS) is introduced to adaptively search the penalty coefficient C and kernel function radius *gamma* associated with a support vector machine classifier (SVM).
3. The training set was split into multiple subsets, and a sub-classifier was trained on each subset of the training set. The output of all sub-classifiers was combined to generate the final prediction.
4. Comparative experiments were carried out on six real data sets, including imdbs, yelp, sen_pol, amazon_cells, SST-2, and IMDBR. Experimental results showed that AdaECELM was superior to other machine learning models in all short text SA. Therefore, the proposed AdaECELM short-text sentiment analysis computational framework is superior to most existing models.

2. Related Works

The models presented in this paper are closely related to the fields of machine learning, ensemble learning, and intelligent computing, and the literature in these fields will be reviewed in this section.

2.1. Machine Learning

In recent years, machine learning methods, such as decision tree (DT), random forest (RF), support vector machine (SVM), multinomial naive Bayes (MNB), and multilayer perceptron (MLP), have been applied to text sentiment analysis tasks. Many researchers believe that the optimization algorithm uses the NARM big data analysis method and can be extended to the data mining process of artificial intelligence [23–27].

Hartmann et al. [23] propose a new pretrained sentiment analysis model, called SiEBERT, and demonstrate the interpretative complexity of machine learning methods through quantitative experiments and empirical frameworks, while the transfer learning-based method has better sentiment analysis ability in this task. However, this approach does not do much to validate and analyze the interpretability and performance improvements that exist in machine learning. Li et al. [24] use RF to analyze the attitude of listed companies toward environmental protection based on the text information on environmental protection issue; experimental results suggest that policies and different industries hold quite different opinions on carbon emissions. Han et al. [25] proposed a new sum function fisher and applied it to SVM to analyze the sentiment orientation from an X (formerly known as Twitter) data set. The proposed method significantly improves vocabulary and semantic information mining, but it only works for specific application scenarios and performance improvement, and it does not draw a research conclusion of portability and universal applicability. Aiming at the interpretability of sentiment analysis, Chen et al. [26] propose a computational framework for logical knowledge learning (called CLK) for SA tasks. Experimental results at different granularity show that the interpretation generation and logical reasoning of CLK are highly consistent with the decision results of implicit models. Cam et al. [27] combine lexical representation and machine learning models to propose a new ML classifier and verify the effectiveness of the classifier through extensive experiments.

2.2. Ensemble Learning

Ensemble learning has been widely applied in the field of NLP. For example, Xu et al. [28] propose an adaptive integrated text classification method, integrating enhanced Text CNN into local and global stages, which has a higher classification effect on multiple data sets. Zhou et al. [29] describe a new domain adaptive ensemble learning (DAEL) framework that utilizes interactive information and domain adaptability to achieve a higher level of recognition on multi-source UDA unlabeled targets and DG data sets. Alam et al. [30] illustrate an integrated dynamic bible network integration algorithm (DEL) and conducted extensive experiments on many real data such as cancer and diabetes; experimental results prove that the integrated algorithm has a wide range of applications and good generalization ability. Lee et al. [31] propose a simple and unified integration method called SUNRISE, which effectively combines weighted Bellman backup based on uncertainty and an inference method based on efficient exploration. Kaushik et al. [32] address the challenges of association rule mining tasks in the real world and evaluate the usability of a new discretization technique to overcome the difficulties and challenges in this field. Through bootstrap random initialization, the diversity among different agents is strengthened. Experiments show that SUNRISE can significantly improve the performance of existing off-strategy RL algorithms in both continuous and discrete control tasks and perform well in both low and high-dimensional environments.

2.3. Intelligent Computing

Since the rapid development of the computer field, researchers have proposed a variety of intelligent optimization methods. Examples include cuckoo search (CS) [33], salp swarm algorithm (SSA) [34], and golden jackal optimization (GJO) [35].

Compared with new algorithms such as SSA and GJO, CS has been applied to many scientific and engineering optimization problems, and it displays very good optimization performance. For example, Cao et al. [36] use the SARSA algorithm to build a reinforcement learning cuckoo search algorithm (KCSA) and successfully apply it to flexible job shop scheduling problems. She et al. [37] introduce gradient information to enhance local optimization ability and adaptive strategies to increase population diversity, and the proposed HAGCS algorithm has good effects in terms of convergence speed and calculation accuracy. Lin et al. [38] propose a cuckoo search optimization algorithm based on autoencoder, factorization, and reinforcement learning, which achieves excellent performance.

Today, computational intelligence has evolved into a new phase of data-driven artificial intelligence. In the task of sentiment analysis, advanced technologies such as BERT [39] and GAT [40] have been applied to the fusion of computational intelligence and the processing of NLP tasks [41,42].

2.4. Fairness in Sentiment Analysis

Sentiment analysis is a powerful analysis tool for extracting high-value information from huge and rapidly time-varying sources. Like all approaches based on machine learning and artificial intelligence, sentiment analysis algorithms must offer an adequate level of *fairness* [43–46], defined as the property that ensures that a model does not generate outputs that may discriminate (or harm) individuals or groups of individuals because of sensitive information such as race, gender, religious conventions, or disability. A system that tends to discriminate against some groups over others is defined as *biased.*

Building sentiment analysis models that ensure fairness is a moral obligation for the researcher; in addition, many countries around the world are actively working to adapt their legislation to penalize software systems that violate fairness. Beyond moral and legislative obligations, ensuring fairness has deep consequences on the success of an ML model and contributes to better acceptance of the system by its users.

Unfairness and bias can have many causes, and among them, we mention *data imbalance*; that is, some segments of society may be underrepresented compared to others. Several empirical studies assessing the fairness of certain sentiment analysis approaches have been proposed in the literature. A further source of bias arises from *data preprocessing*: at this stage, in fact, human choices (e.g., to include or exclude certain samples from the training set, to include/avoid some features, etc.) could exacerbate the bias in the source data. Natural language processing techniques could be biased since they could favor certain languages and dialects over others [47]; finally, any errors on the labels attributed to the sentences to be analyzed could affect the behavior of the sentiment analysis algorithm [48].

In general, measuring the fairness of a sentiment analysis model is very difficult since the evaluation depends on the application context, and certain features used in the training of the model could be sensitive in a specific application domain and non-sensitive in other domains.

The process of measuring the level of fairness of a system and improving its fairness must be considered an iterative process. To this end, it is necessary to ensure that the data collection process is *rigorous*, and it must eliminate noisy and missing data. To cope with unbalanced data, one must think of ad hoc strategies such as random undersampling or oversampling or, alternatively, data augmentation strategies that improve the quality of the training set (see [49] for a detailed review).

Some authors (see [50]) suggest combining the output of multiple commercial sentiment detection APIs to avoid gender bias. A further interesting option is proposed in [51]; here, the authors carry out adversarial pretraining on several models, and they recommend retraining a sentiment analysis model if unfairness is detected.

Despite the considerable scientific efforts that have been made, we believe that the problem of fairness still needs to be studied.

Another issue with sentiment analysis is the potential for privacy breaches. In fact, the content of even a short text message can reveal information such as an individual's gender, political orientation, and mood. Several methods have been studied to anonymize data for sentiment analysis purposes, including differential privacy, bloom filters, or a combination of both [49,52–54].

This paper mainly studies the combination of intelligent optimization algorithms to improve the performance of machine learning models and ensemble learning. Based on the adaptive search advantages of intelligent optimization algorithms, a CS-SVM classifier (which can effectively mine text features and adaptively find an optimal hyperplane partition) is proposed.

3. Proposed Method

3.1. Problem Definition

This paper is about determining the polarity of short texts. The research questions that inspired the proposed approach are as follows:

a. How do we design a text analysis system that can leverage multiple classifiers and combine the answers provided by each of them to obtain a more precise estimate of the polarity of a sentence?
b. How do we achieve an accurate representation of sentences using numerical features that are free from issues such as redundancy and feature sparsity?
c. Since available classifiers often depend on multiple hyperparameters, is it possible to implement a fast and automated mechanism to choose the best hyperparameter configuration (i.e., the configuration that maximizes the accuracy of our system)?

In what follows, the AdaECELM model, which was designed to answer the three questions above in a positive manner, is discussed in detail.

3.2. AdaECELM Model

This section introduces the AdaECELM model for sentiment analysis of short texts. Figure 1 shows the overall framework of the model. The AdaECELM model mainly preprocesses text from TF-IDF feature extraction, feature selection, and feature compression standardization. Then, the text features are divided into multiple training sets and test set (we considered three training subsets in this paper). A sub-classifier is constructed to train the classifier over each available training subset and optimize the combination parameters. Finally, AdaBoost integrates the three trained sub-classifiers to perform sentiment analysis for short texts.

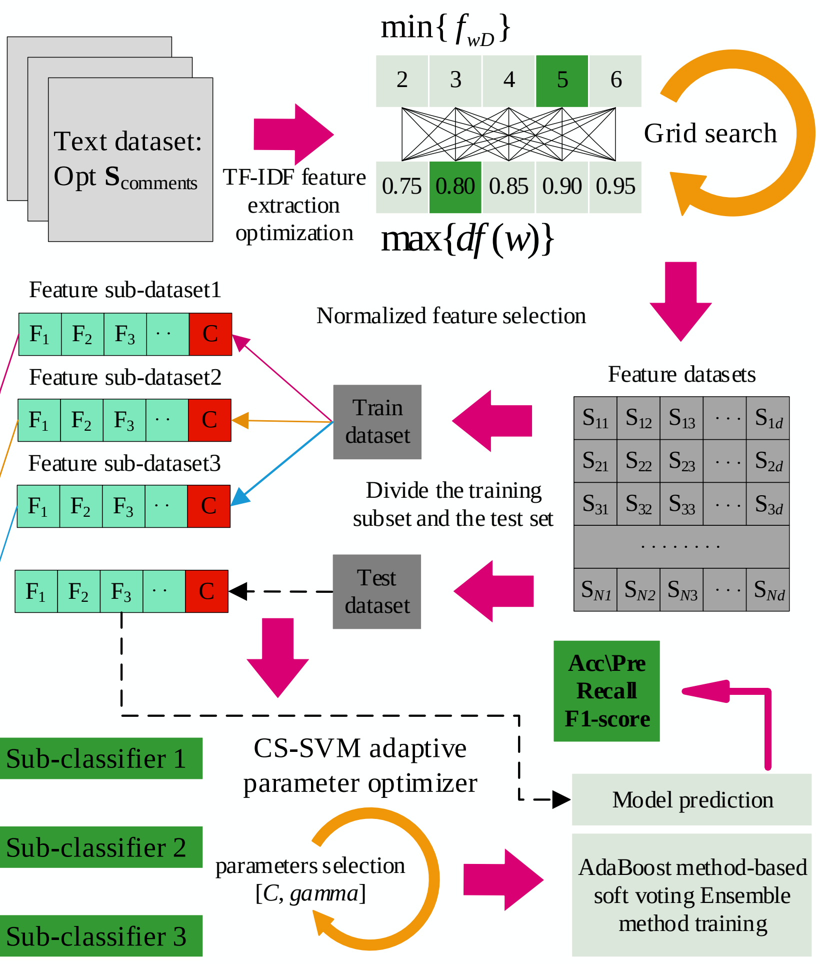


Figure 1. Architecture of AdaECELM for sentiment analysis.

In the AdaECELM model, word segmentation and feature extraction are carried out, and, more specifically, TF-IDF is used to extract word vector features. The TF-IDF model is shown in Equations (1) and (2).

$$T_f(w, D) = \frac{f_{wD}}{N_D} \tag{1}$$

$$I_{df}(w, D) = 1 + \log \frac{N}{1 + df(w)} \tag{2}$$

where T_f represents the frequency of w in text D, and f_{wD} and N_D represent the number and total number of words w in text D, respectively. I_{df} represents the inverse document frequency of w, N is the total number of data, and df represents the total number of w words in N.

Due to the difference in the size of different data sets, it is necessary to select the maximum document word frequency and minimum inverse document frequency of TF-IDF. The AdaECELM uses grid search to extract the features of different data sets.

The extracted features are highly sparse, which is not conducive to further machine learning training and prediction. To save computing resources, accurately express text features, and reduce the redundancy of features, we carry out intensive transformation of this sparse matrix. The conversion process is shown in Figure 2.

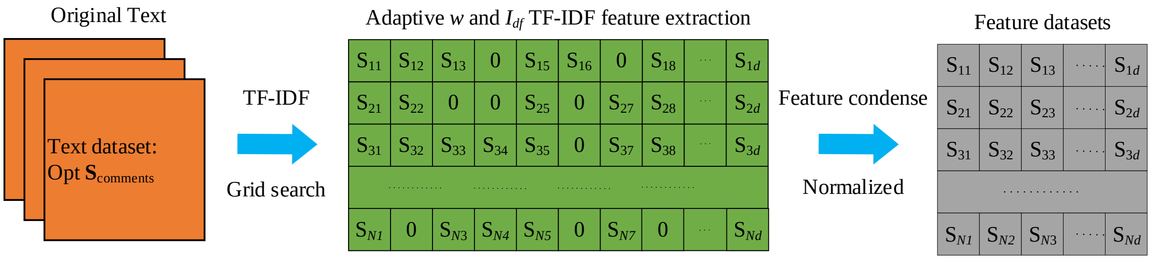

Original Text
Text dataset: Opt S_comments
TF-IDF
Grid search
Adaptive w and I_df TF-IDF feature extraction
S11 S12 S13 0 S15 S16 0 S18 ··· S1d
S21 S22 0 0 S25 0 S27 S28 ··· S2d
S31 S32 S33 S34 S35 0 S37 S38 ··· S3d
SN1 0 SN3 SN4 SN5 0 SN7 0 ··· SNd
Feature condense
Normalized
Feature datasets
S11 S12 S13 ····· S1d
S21 S22 S23 ····· S2d
S31 S32 S33 ····· S3d
SN1 SN2 SN3 ····· SNd

Figure 2. Feature extraction and sparse matrix normalization.

The processed features are divided into three training subsets and a test set. In the sampling process, text features are first divided into a training set and a test set according to a certain ratio. The training set is then randomly divided into three different training sets. To ensure that the text distribution is better divided into different training subsets to increase the diversity of the training set, we randomly constructed each training subsets by sampling data from the original subset; to this end, data from the training set were sampled through a normal distribution with mean equal to zero and standard deviation equal to one A sub-classifier is trained for each training subset.

At this time, the cuckoo search (CS) algorithm was used to optimize the hyperparameters of SVM. Cuckoo search draws inspiration from the parasitic breeding behavior of cuckoo [33]. Cuckoos lay their eggs in other nests and use other nests to hatch, but there is a risk of detection by the host. To cope with this risk, cuckoo learns the songs of other young birds, drawing on this feature, and designs a CS algorithm.

The proposed algorithm is based on three rules: one egg is laid each time, nests are randomly selected for incubation, and optimal nests are retained until the next iteration, with a fixed number of nests and a probability of discovery. The algorithm simulates the local search of the host discovery behavior through the global search of Levi's flight mechanism, evaluates the merits of the function fitness, and ensures the optimal solution set of the surviving individuals.

By integrating these elements, the cuckoo search algorithm can efficiently find the optimal solution. The model using the Levy Flights mechanism is as follows:

$$x_i^{t+1} = x_i^{t} + \alpha \oplus Levy(\beta) \tag{3}$$

According to Equation (3), at the t iteration, the global search update formula is obtained by the i-th individual by multiplying Levy's random search path Levy (β) by a step point a (with step size ranging from 0 to 1). Levy (β) and time t obey the Levy distribution, and its probability density function is as specified in Equation (4).

$$Levy(\beta) \sim u = t^{-\lambda} \tag{4}$$

where, λ ϵ (1,3), when the ideal Levy distribution cannot take a fixed form, a more programmable Mantegna rule is used to compute Levy (β), as shown in Equation (5).

$$Levy(\beta) \sim \frac{\phi \times \mu}{|v|^{1/\beta}} \tag{5}$$

The parameters μ and v obey the standard normally distributed pseudo range numbers; the value range of β is (0,2), and the calculation formula of ϕ is as follows:

$$\phi = \left(\frac{\Gamma(1+\beta) \times \sin(\pi \times \frac{\beta}{2})}{\Gamma(\frac{1+\beta}{2} \times \beta \times 2^{\frac{\beta-1}{2}})} \right)^{1/\beta} \tag{6}$$

In Equation (6), Γ is the gamma function, and the computational model simulated by the above formula is used as the global search mechanism of cuckoo search, and

Equation (7) is the local search mechanism of the CS algorithm, where *i* and *j*, respectively, represent two random cuckoo solutions in the population when the population evolves to the *t* generation, that is, the random walk strategy of cuckoo.

$$x_i{}^{t+1} = x_i{}^t + rand \times (x_j{}^t - x_i{}^t) \tag{7}$$

The steps for optimizing CS algorithm are detailed below:

Step 1: Set parameters, initialize the population, and calculate the fitness value of everyone. At the same time, record the current optimal cuckoo nest and its fitness.
Step 2: Continuously update the cuckoo's location within a preset number of cycles, combined with the cuckoo's local and global search strategy. Once a better solution set is found, the corresponding nest is updated immediately, and the global fitness and optimal solution are retained.
Step 3: At each generation, the current nests may be discarded with a specific probability, and new nests will be randomly generated to keep the total population unchanged.
Step 4: Check whether the termination conditions are met. If not, repeat the loop consisting of Step 2 and Step 3. If yes, the loop is stopped, and the global optimal fitness value and corresponding solution are output.

The CS algorithm is applied to select hyperparameters [*C*, *gamma*] of the SVM algorithm and use it as a sub-classifier of the AdaECELM model. The SVM algorithm has good generalization ability; it is widely used in a variety of classification problems due to its advantages of simple principle, good classification effect, and multiple hybrid improvement strategies. The SVM algorithm uses the structural risk minimization principle to find out the best hyperplane separating positive classes from negative ones. In the case of non-linearly separable data sets, suitable procedures (e.g., kernels) are employed to deal with non-linearities in input data. Figure 3 shows the schematic diagram of plane division in two-dimensional space.

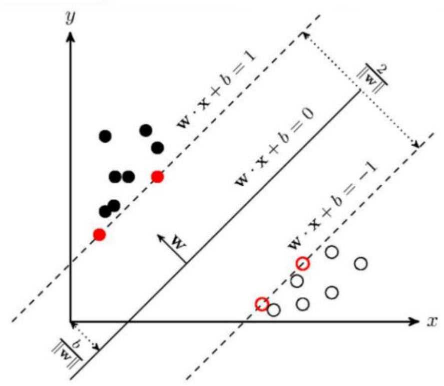


Figure 3. Hyperplane diagram. The two dotted lines and dots on either side represent the decision boundaries and different classes of data samples, respectively. The solid line in the middle represents the final partition boundary (hyperplane in higher dimensional space).

In the processing of a nonlinear data set, the VC dimension theory is used to map the multi-attribute sample points in the low-dimensional space to the high-dimensional space. After processing in the high-dimensional space, another linearly separable hyperplane is obtained: $w^T x + b = 0$. The kernel function is introduced to transform the inner product of the space mapping into the function in the original dimension, which can better ensure the accuracy of SVM nonlinear classification. The constructed nonlinear discriminant function is transformed into a dual optimization problem.

$$\max_a Q(a) = \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i,j=1}^{n} a_i a_j y_i y_j K(x_i, x_j) \tag{8}$$

$$s.t.\sum_{i=1}^{n} a_i y_i = 0(0 \leq a_i \leq C, i = 1, 2, \ldots, n.) \tag{9}$$

The nonlinear decision function obtained from the above equation is shown in Equation (9).

$$f(x) = \operatorname{sgn}(\sum_{i=1}^{n} a_i{}^* y_i K(x_i, x) + b^*) \tag{10}$$

The kernel function K in Equation (8) is the radial basis kernel function (Gaussian kernel function, RBF), and the expression is

$$f(x) = \operatorname{sgn}(\sum_{i=1}^{n} a_i{}^* y_i K(x_i, x) + b^*) \tag{11}$$

The SVM classification process includes data set partitioning, data preprocessing, SVM model construction, combination hyperparameters selection for training, model performance evaluation, and classification accuracy output. SVM will use CS to optimize it to obtain an adaptive optimal hyperplane partition hybrid classifier. This greatly improves the performance of the model and gives it stronger adaptive abilities than traditional manual parameter tuning and random parameter optimization strategies. The CS+SVM execution pseudocode is shown below.

Algorithm1: CS+SVM parameters optimization

```
Input: Range of penalty coefficient C and RBF nuclear coefficient gamma to be searched
Output: Result of adaptive hyperparameter combination
1 CS Parameters:
2 //Maximum number of iterations, population size, discovery probability, step size adjustment
3 Initialization:
4 //Total number of nests, number of iterations, discovery probability, dimension, step size
5 for t in Max_iterations do //t is the current iteration of the CS
6 for i in Np do //Np is the population size of the CS
7 Levy Flight Updates: $Levy(\beta) \sim \frac{\phi \times \mu}{|v|^{1/\beta}}$
8 Regeneration of next generation population
9 Discovery probability: Reset abandoned nest
10 Population renewal: $x_i{}^{t+1} = x_i{}^t + rand \times (x_j{}^t - x_i{}^t)$
11 Update hyperparameter combination [C, gamma]
12 return the hyperparameter combination with the highest score
```

The corresponding sub-classifier can be trained by CS-SVM. Finally, the AdaBoost method is applied to combine the predictions generated by each sub-classifier. The category prediction confidence of feature F_i can be expressed as Equation (12).

$$C_z(F_i) = \prod_{t=1}^{Max\_iterations} H_{zt}(F_i) \tag{12}$$

The class z of feature F_i is C_z. H_{zt} (F_i) is the result predicted by the t-th sub-classifier for class z. Thus, the F_i prediction results integrated with AdaBoost can be obtained, as shown in Equation (13).

$$y_i = (\operatorname{argmax}_{z \in \{0,1\}} C_z(F_i)) \tag{13}$$

Algorithm 1 summarizes the CS-SVM sub-classifier procedure of the AdaECELM model. In the time complexity analysis of the AdaECELM model, the time complexity we need to consider is divided into the sum of the complexity of three modules: feature extraction, model training, and classification prediction. Let a data set have N samples, and then the time complexity of the feature extraction process is O(N), and the resulting compressed feature vector dimension is D. When SVM classification is adopted, M support vectors are required, and the required computation amount is M^2. In this case, the training time complexity is O($N_{train}M^2$), and the test time complexity is O($N_{test}D$). Therefore, the time complexity of the text data set using SVM is O ($N + N_{train}M^2 + N_{test}D$). For the training stage of a sub-classifier in AdaECELM, the sample is $N/3$, the support vector is M, and the RBF hyperparameter combination needs to be optimized by the k-order CS optimizer. In this case, the time complexity of the training phase is O($k(N_{train}/3)\ M^2$), then the final time complexity is O ($N + k(N_{train}/3)\ M^2 + N_{test}D$).

From the theoretical stage, compared with the traditional SVM model, when the number of CS iterations in the AdaECELM model is 3, the computational efficiency of the two models is exactly the same. It should be noted that the number of CS iterations is a fixed parameter. When we deal with large-scale data sets, when $N >> k$, M will increase with the increase of N, and the time complexity of processing multiple sub-data sets using multiple sub-classifiers is better than the traditional SVM model. In addition, the subsequent experimental results and analysis show that AdaECELM has higher and more stable text classification advantages under the same or slightly lower time complexity than the traditional ML model.

4. Experimental Results and Analysis

This section reports the numerical tests to evaluate the competitiveness of AdaECELM against some state-of-the-art baselines. Publicly available data sets were used to facilitate the reproducibility of experiments. The aim of experimental tests is to prove that AdaECELM can correctly identify the polarity of a sentence, assuming that it is trained on short texts. Widely known metrics such as accuracy, precision, recall, and F-score have been used in experimental analysis.

4.1. Experimental Description

In AdaECELM, to ensure that all the text messages of the training set are used for learning knowledge, the model adopts a combination of random sampling and integer multiple sampling to select training samples. Specifically, we use *model_selection.train_test_split()* from the *Python* programming language *sklearn* library to implement random sampling of the training set and test machine, *training*:*test* = 0.8:0.2. At the same time, to meet the requirement that the training sets of different subclassifications remain independent and non-repetitive, the randomly divided training set samples are divided into three parts (the number of subclassifies is 3). One-third of each sub-training set is extracted according to the integer multiple sampling principle. The data set selected in this way can capture more affective semantic information through the training of different classifiers, which provides the diversity of semantic knowledge and the effectiveness of emotion recognition for the AdaECELM model.

The data sets used in this paper are imdbs, yelp, sen_pol, amazon_cells, SST-2, and IMDBR, whose features are reported in Table 1 [55].

Table 1. Summary of experiment data sets (accessed on 20 April 2024).

Dataset	Task	#Cat	#Docs	Source
imdbs	Sentiment	2	748	https://bit.ly/3mGPBx4
yelp	Sentiment	2	1000	https://bit.ly/3mGPBx4
sen_pol	Sentiment	2	10,662	https://bit.ly/3Lbh3xf
amazon_cells	Sentiment	2	1000	https://bit.ly/3mGPBx4
SST-2	Sentiment	2	9602	https://www.kaggle.com/datasets/jkhanbk1/sst2-dataset
IMDBR	Sentiment	2	50,000	https://bit.ly/3ZUUUYd

For the partitioning of each data set, 80% of the training set is divided into three sub-training sets, and the remaining 20% is used as the test set. When building sub-classifier training, the CS optimizer forms the SVM sub-classifier by iterating 100 times to obtain the relatively optimal [C, *gamma*] setting. The step size in CS is adjusted to 0.01, the probability of finding bad nests is 0.25, parameter optimization is carried out in the environment where the population size is 10, and the dimension setting is the number of hyperparameters that need to be tuned. AdaECELM and other machine learning models use four classical evaluation indicators—accuracy, precision, recall, and F1-score—to conduct results statistics and performance analysis. The corresponding calculation formula is shown as Equations (14)–(17).

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{14}$$

$$Precision = \frac{TP}{TP + FP} \tag{15}$$

$$Recall = \frac{TP}{TP + FN} \tag{16}$$

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{17}$$

4.2. Experimental Results and Performance Analysis

First, to make the training and prediction of the model independent of each other, the data is divided into 80% training sets and 20% training sets. The feature extraction method TF-IDF of the AdaECELM model was adopted, and the minimum word frequency T_f and maximum inverse document frequency I_{df} were set to 5 and 0.8, respectively. Then, using the advantages of CS algorithm search, adaptive parameter search is carried out for the combined control hyperparameters of SVM [C, $gamma$]. The maximum number of CS *Max_iterations* is set to 100. The hyperparameter combination of the output sub-classifier is used for AdaECELM sentiment analysis.

We compared eight classical machine learning models and compared the average results of 30 independent runs to ensure the comprehensive classification performance of each model [56]. The experimental results are shown in Table 2. In Table 2, the results with the highest accuracy are shown in bold, and the results with the second highest accuracy are shown in underline. In the imdbs data set, the accuracy of the eight comparison ML models is the highest at 73.50% of MNB, while that of the model in this paper is 76.83%, an increase of 3.33%. At the same time, compared with the second-ranked method on the other three data sets, the improvement was 1.9%, 0.54%, and 0.59%, respectively.

Table 2. Comparison of experimental results of different classification models (Accuracy).

Methods	Imdbs	Yelp	Sen_Pol	Amazon_Cells
LR	0.7205	0.7782	0.7595	0.7895
DT	0.6277	0.7072	0.6002	0.7327
RF	0.6958	0.7618	0.6956	0.7785
SVM	0.7253	<u>0.7820</u>	0.7655	<u>0.7958</u>
KNN	0.6518	0.6453	0.5094	0.6502
MNB	<u>0.7350</u>	0.7675	<u>0.7678</u>	0.7873
MLP	0.7023	0.7513	0.7082	0.7683
GBT	0.7102	0.7610	0.6575	0.7832
AdaECELM	**0.7683**	**0.8010**	**0.7732**	**0.8017**

In general, SVM and MNB are generally better. Compared with all models, AdaECELM has the highest accuracy, has certain effectiveness and robustness, and is a new baseline method worthy of consideration. Second, we also visualized precision, recall, and F1-score, as shown in Figures 4 and 5, respectively.

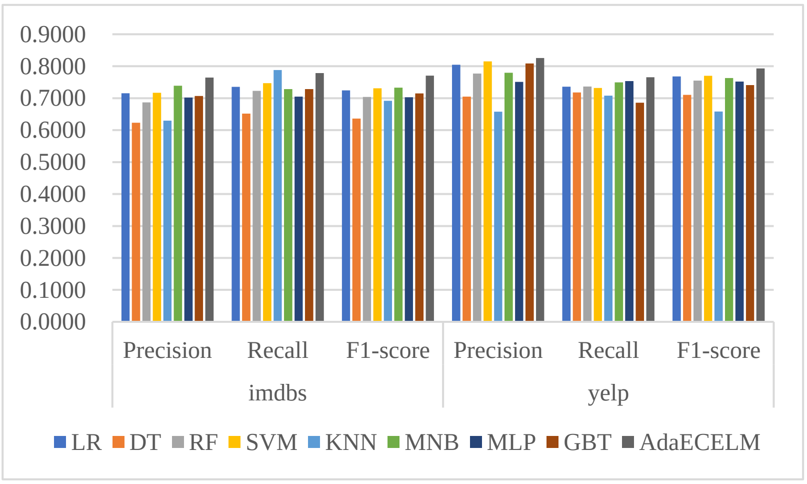


Figure 4. Precision\Recall\F1-score data comparison of imdbs and yelp.

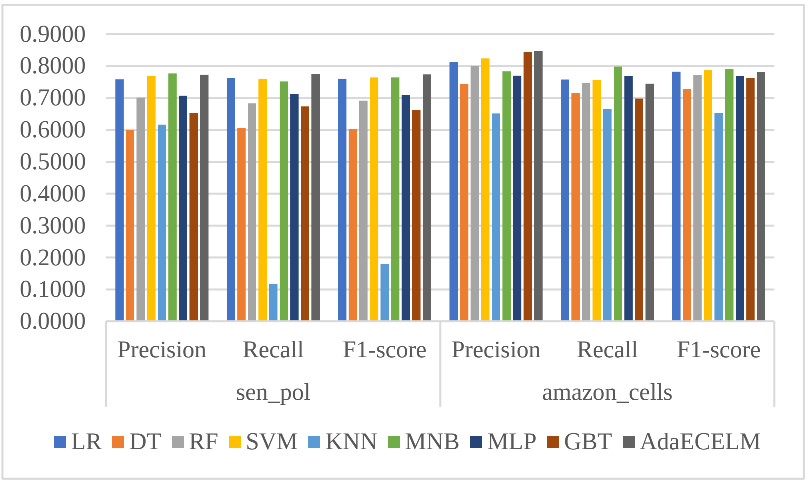
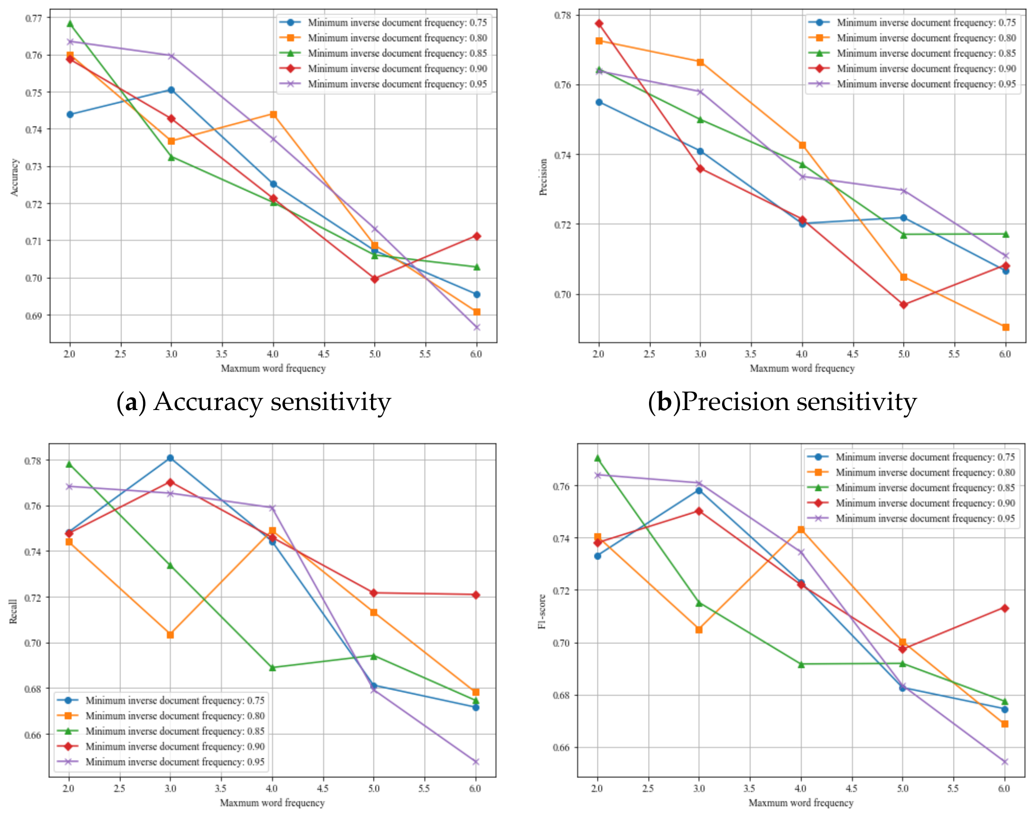


Figure 5. Precision\Recall\F1-score data comparison of sen_pol and amazon_cells.

As shown in Figure 4, among the precision, recall, and F1-score indicators on two relatively small data sets of imdbs and yelp, the models with poor overall emotion classification performance include DT and KNN. LR, RF, SVM, MNB, MLP, and GBT are all on the same level as our model. However, from the comprehensive level, the classification correctness of the AdaECELM model and all F1 scores have been improved to varying degrees.

In the amazon_cells data set shown in Figure 5, these metrics are consistent with the performance gains described in Figure 4 due to similarities in data size and problem-solving. However, when processing sen_pol, a relatively large data set, KNN showed a very serious classification fatigue, and DT, RF, MLP, GBT, and the other four methods were no longer at the same level as SVM and MNB. Compared with SVM and MNB, the two ML models have the best performance stability, but AdaECELM does not have a degree of index improvement.

Second, because the text of imdbs data set is short, the wrong feature extraction of a certain word will have a great impact on the final fitting result of the model [56]. To this end, sensitivity analysis was carried out on the feature extraction process of the two parameters T_f and I_{df} of the word vector representation method, wherein T_f selection field was [2–6] and I_{df} selection field was [0.75, 0.80, 0.85, 0.90, 0.95]. The sensitivity analysis curve of adaptive search for different indicators of the AdaECELM model is shown in Figure 6.



(**a**) Accuracy sensitivity (**b**)Precision sensitivity

(**c**) Recall sensitivity (**d**) F1-score sensitivity

Figure 6. Sensitivity analysis of imdbs data set feature optimization.

As can be clearly seen from Figure 6, in terms of the accuracy and precision of the AdaECELM sentiment analysis model, as the maximum document word frequency increases, the extracted features are negatively correlated with the recognition effect of the model. The maximum inverse document frequency set at 0.80 on the other two metrics is not relevant, but the other settings are also consistent with the first two metrics. From the comprehensive sensitivity trend, our model is more effective when choosing a smaller maximum document word frequency. On the one hand, it shows the adaptability of our model; on the other hand, it also shows that the learning ability of the model and the length of the sentence will have a certain impact on semantic mining in short text sentiment analysis.

4.3. Ablation Experiment

To further clarify the robustness and scalability of our proposed model, two larger data sets, such as SST-2 and IMDBR, are used in this section to conduct ablation experiments on the AdaECELM model. The comparison models are SVM, an ensemble learning model (ELM) introduced with an ensemble learning scheme, and an ensemble learning model (ECELM) introduced with a CS-SVM optimizer. The models are compared with AdaECELM, and the statistical data are shown in Table 3.

Table 3. Comparison of ablation results.

Models	**SST-2**				**IMDBR**			
	Acc	**Pre**	**Rec**	**F1**	**Acc**	**Pre**	**Rec**	**F1**
SVM	0.6220	0.6367	0.6214	0.6290	0.7731	0.7655	0.8114	0.7878
ELM	0.6420	0.6605	0.6698	0.6651	0.7760	0.7737	0.7694	0.7715
ECELM	0.6570	0.6498	0.7168	0.6817	0.7938	0.7931	0.7950	0.7940
AdaECELM	**0.6880**	**0.6949**	**0.9379**	**0.7983**	**0.8180**	**0.8591**	**0.7979**	**0.8274**

As shown in Table 3, the AdaECELM model still has a strong effect on emotion classification in larger data sets and higher dimensional feature spaces. In addition to the fact that the recall of the IMDBR data set is slightly lower than SVM and ranks second, the two data sets are superior to SVM, ELM, and ECELM classification methods that reduce different modules in other indicators. At the same time, it is clear from the comparison model that (1) the effect of soft voting with ensemble learning is better than SVM in most cases; (2) after CS algorithm is introduced into the sub-classifier SVM in ELM to optimize the combined hyperparameters, the results of the ECELM scheme are better than ELM, and rank second among the comparison algorithms; and (3) the AdaECELM method based on the CS adaptive hyperparameter search and grid search strategy of text feature representation shows strong emotion classification performance in comparison.

To sum up, we compared the accuracy results, analyzed the three indexes, including the F1-score, and carried out sensitivity analysis on the adaptive parameter adjustment extracted from the model's special detection. With extensive experimental results and data analysis, AdaECELM has an excellent performance in short-text sentiment analysis and can be used as a baseline method superior to most ML sentiment classification models. By contrast, in the test with a machine learning model, parameter sensitivity analysis of feature extraction and ablation experiment on two large-scale data sets, AdaECELM shows a very exciting effect when facing the emotion classification task of most classical data sets and has good model robustness and scalability.

5. Discussion

The experiments presented in the previous section demonstrate that AdaECELM displays excellent performance in the classification of short texts and is, therefore, superior to many existing classification models. The results obtained, as already highlighted, are particularly encouraging if we wish to classify emotions, which is known to be a complex task.

A potential limitation of our study (and a possible research topic to be developed in the future) concerns the variety and numerosity of the data sets used for validation. We plan to increase the number of data sets used in the experimental evaluation to obtain a more robust confirmation of the effectiveness of our method. Moreover, other types of data sets should also be considered; for example, it is interesting to verify whether our system works well on texts extracted from social platforms or, more generally, from Web platforms that support interactions between their members through the exchange of messages (think of forums, for example).

A further point of discussion is related to the topic of a short text. In some cases, a text might contain an objective description of a good/service (e.g., a product review); on the other hand, we might have a debate (with potentially conflicting points of view) on topics of any nature (e.g., topics related to the risks of new technologies in our society or discussions concerning the social and political life of a community). It is interesting to understand whether the topic of a short text affects the performance of our approach to some extent and, in detail, whether it continues to maintain its effectiveness compared to other competitors.

Finally, in our application scenario, we assumed that the possible sentiment categories were *only two* (positive and negative), which is also the case of biggest interest in the sentiment analysis literature. It would be interesting to consider a *multi-class scenario* in which more than two categories are possible (e.g., think of categories like strongly positive, positive, neutral, negative, and strongly negative); it would be interesting to test whether our approach continues to provide adequate performance in terms of accuracy, precision, recall, and F-measure even in the multi-class scenario.

6. Conclusions

To solve the problem of poor performance of machine learning in sentiment analysis of short texts, an adaptive evolutionary computational ensemble learning (AdaECELM) sentiment analysis model is proposed in this paper. In the feature extraction stage, the model can obtain features that are more consistent with the features of the data set itself. At the same time, in the training stage, the diversity of semantic mining is increased by repartitioning the training set. Three CS+SVM sub-classifiers are used to train each feature subset to form a weak classifier. Finally, the soft voting mechanism of the AdaBoost method is used for the final sentiment analysis of ensemble learning. The experimental results show that our model significantly improves the accuracy of sentiment classification on 4 data sets and outperforms all comparative ML sentiment analysis methods. This shows that AdaECELM can be used as a new baseline method that is superior to most machine learning methods.

7. Future Work

In the next research work, we will explore how to make better use of the computational framework to expand the deep learning model and apply it to more complex sentiment analysis tasks to verify the correctness and generalization ability of the research in this paper. In future research, we will explore how to use this computational idea to further extend and enhance the semantic recognition capability of deep learning models and strive to apply these models to more complex and refined sentiment analysis tasks. At the same time, strengthening cooperation with the platform of user comments and applying the model to real life are also important topics for future research.

Author Contributions: X.-Y.L. and K.-Q.Z. contributed to the conception of the study and contributed significantly to analysis and manuscript preparation; G.F. and P.D.M. performed the wrote the manuscript and English language refinement; A.F. helped perform the data analysis. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported in part by Key Project of Chongqing Municipal Education Commission (23SKGH247), Graduate Innovation Fund of Chongqing (CYS240677).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data and code presented in this study are available upon request from the corresponding author.

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

References

1. Zhu, L.; Zhang, Z. A nearly-linear time algorithm for minimizing risk of conflict in social networks. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022; pp. 2648–2656.
2. Liu, X.; Ye, S.; Fiumara, G.; De Meo, P. Influence Nodes Identifying Method via Community-based Backward Generating Network Framework. *IEEE Trans. Netw. Sci. Eng.* **2024**, *11*, 236–253. [CrossRef]
3. Hupkes, D.; Giulianelli, M.; Dankers, V.; Artetxe, M.; Elazar, Y.; Pimentel, T.; Christodoulopoulos, C.; Lasri, K.; Saphra, N.; Sinclair, A.; et al. A taxonomy and review of generalization research in NLP. *Nat. Mach. Intell.* **2023**, *5*, 1161–1174. [CrossRef]
4. Wang, S.I.; Manning, C.D. Baselines and bigrams: Simple, good sentiment and topic classification. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL), Jeju, Republic of Korea, 8–14 July 2012; Volume 2, pp. 90–94.
5. Malla, S.J.; Alphonse, P.J.A. COVID-19 outbreak: An ensemble pre-trained deep learning model for detecting informative tweets. *Appl. Soft Comput.* **2021**, *107*, 107495. [CrossRef] [PubMed]
6. Liu, J.; Chen, L.; Luo, R.; Zhu, J. A combination model based on multi-angle feature extraction and sentiment analysis: Application to EVS sales forecasting. *Expert Syst. Appl.* **2023**, *224*, 119986. [CrossRef]
7. Liu, X.; Ye, S.; Fiumara, G.; De Meo, P. Information Propagation Prediction Based on Spatial–Temporal Attention and Heterogeneous Graph Convolutional Networks. *IEEE Trans. Comput. Soc. Syst.* **2024**, *11*, 945–958. [CrossRef]
8. Tang, D.; Wei, F.; Qin, B.; Yang, N.; Liu, T.; Zhou, M. Sentiment embeddings with applications to sentiment analysis. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 496–509. [CrossRef]
9. Zhu, L.; Li, W.; Shi, Y.; Guo, K. SentiVec: Learning sentiment-context vector via kernel optimization function for sentiment analysis. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 2561–2572. [CrossRef] [PubMed]

10. Forman, G. BNS feature scaling: An improved representation over tf-idf for svm text classification. In Proceedings of the 17th ACM conference on Information and knowledge management, Napa Valley, CA, USA, 26–30 October 2008; pp. 263–270.
11. Hu, C.; Zhang, C.; Lei, D.; Wu, T.; Liu, X.; Zhu, L. Achieving privacy-preserving and verifiable support vector machine training in the cloud. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 3476–3491. [CrossRef]
12. Luo, X. Efficient English text classification using selected machine learning techniques. *Alex. Eng. J.* **2021**, *60*, 3401–3409. [CrossRef]
13. Marie-Sainte, S.L.; Alalyani, N. Firefly algorithm based feature selection for Arabic text classification. *J. King Saud Univ. Comput. Inf. Sci.* **2020**, *32*, 320–328. [CrossRef]
14. 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]
15. Lin, Y.; Chen, S.; Liu, J.; Lin, C. Linear Classifier: An Often-Forgotten Baseline for Text Classification. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Toronto, ON, Canada, 9–14 July 2023; pp. 1876–1888.
16. Zhang, J.; Zhang, Q.; Qin, X.; Sun, Y. A two-stage fault diagnosis methodology for rotating machinery combining optimized support vector data description and optimized support vector machine. *Measurement* **2022**, *200*, 111651. [CrossRef]
17. Zhou, J.; Zhu, S.; Qiu, Y.; Armaghani, D.J.; Zhou, A.; Yong, W. Predicting tunnel squeezing using support vector machine optimized by whale optimization algorithm. *Acta Geotech.* **2022**, *17*, 1343–1366. [CrossRef]
18. Ying, X.; Liu, L.; Wang, Y.; Li, R.; Chen, N.; Lin, Z.; Sheng, W.; Zhou, S. Mapping degeneration meets label evolution: Learning infrared small target detection with single point supervision. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 15528–15538.
19. Wang, H.; Xu, Y.; Wang, Z.; Cai, Y.; Chen, L.; Li, Y. Centernet-auto: A multi-object visual detection algorithm for autonomous driving scenes based on improved centernet. *IEEE Trans. Emerg. Top. Comput. Intell.* **2023**, *7*, 742–752. [CrossRef]
20. Du, M.; He, F.; Zou, N.; Tao, D.; Hu, X. Shortcut learning of large language models in natural language understanding. *Commun. ACM* **2023**, *67*, 110–120. [CrossRef]
21. Kazmaier, J.; Van Vuuren, J.H. The power of ensemble learning in sentiment analysis. *Expert Syst. Appl.* **2022**, *187*, 115819. [CrossRef]
22. Bountakas, P.; Xenakis, C. Helphed: Hybrid ensemble learning phishing email detection. *J. Netw. Comput. Appl.* **2023**, *210*, 103545. [CrossRef]
23. Hartmann, J.; Heitmann, M.; Siebert, C.; Schamp, C. More than a feeling: Accuracy and application of sentiment analysis. *Int. J. Res. Mark.* **2023**, *40*, 75–87. [CrossRef]
24. Li, C.; Li, L.; Zheng, J.; Wang, J.; Yuan, Y.; Lv, Z.; Wei, Y.; Han, Q.; Gao, J.; Liu, W. China's public firms' attitudes towards environmental protection based on sentiment analysis and random forest models. *Sustainability* **2022**, *14*, 5046. [CrossRef]
25. Han, K.X.; Chien, W.; Chiu, C.C.; Cheng, Y.T. Application of support vector machine (SVM) in the sentiment analysis of twitter dataset. *Appl. Sci.* **2020**, *10*, 1125. [CrossRef]
26. Chen, Y.; Yuan, B.; Liao, B.; Gabbay, D.M. A self-explanatory contrastive logical knowledge learning method for sentiment analysis. *Knowl. Based Syst.* **2023**, *278*, 110863. [CrossRef]
27. Cam, H.; Cam, A.V.; Demirel, U.; Ahmed, S. Sentiment analysis of financial Twitter posts on Twitter with the machine learning classifiers. *Heliyon* **2024**, *10*, e23784. [CrossRef] [PubMed]
28. Xu, Y.; Yu, Z.; Cao, W.; Chen, C.L.P. Adaptive dense ensemble model for text classification. *IEEE Trans. Cybern.* **2022**, *52*, 7513–7526. [CrossRef] [PubMed]
29. Zhou, K.; Yang, Y.; Qiao, Y.; Xiang, T. Domain adaptive ensemble learning. *IEEE Trans. Image Process.* **2021**, *30*, 8008–8018. [CrossRef] [PubMed]
30. Alam, K.M.R.; Siddique, N.; Adeli, H. A dynamic ensemble learning algorithm for neural networks. *Neural Comput. Appl.* **2020**, *32*, 8675–8690. [CrossRef]
31. Lee, K.; Laskin, M.; Srinivas, A.; Abbeel, P. Sunrise: A simple unified framework for ensemble learning in deep reinforcement learning. Proceedings of the International Conference on Machine Learning. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 6131–6141.
32. Kaushik, M.; Sharma, R.; Peious, S.A.; Shahin, M.; Yahia, S.B.; Draheim, D. A systematic assessment of numerical association rule mining methods. *SN Comput. Sci.* **2021**, *2*, 348. [CrossRef]
33. Yang, X.S.; Deb, S. Cuckoo search via Lévy flights. In Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), IEEE, Coimbatore, India, 9–11 December 2009; pp. 210–214.
34. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [CrossRef]
35. Chopra, N.; Ansari, M.M. Golden jackal optimization: A novel nature-inspired optimizer for engineering applications. *Expert Syst. Appl.* **2022**, *198*, 116924. [CrossRef]
36. Cao, Z.C.; Lin, C.R.; Zhou, M.C. A knowledge-based cuckoo search algorithm to schedule a flexible job shop with sequencing flexibility. *IEEE Trans. Autom. Sci. Eng.* **2021**, *18*, 56–69. [CrossRef]
37. She, B.; Fournier, A.; Yao, M.; Wang, Y.; Hu, G. A self-adaptive and gradient-based cuckoo search algorithm for global optimization. *Appl. Soft Comput.* **2022**, *122*, 108774. [CrossRef]

38. Lin, C.R.; Cao, Z.C.; Zhou, M.C. Learning-based cuckoo search algorithm to schedule a flexible job shop with sequencing flexibility. *IEEE Trans. Cybern.* **2022**, *53*, 6663–6675. [CrossRef] [PubMed]
39. Bello, A.; Ng, S.C.; Leung, M.F. A BERT framework to sentiment analysis of tweets. *Sensors* **2023**, *23*, 506. [CrossRef] [PubMed]
40. Im, S.K.; Chan, K.H. Neural Machine Translation with CARU-Embedding Layer and CARU-Gated Attention Layer. *Mathematics* **2024**, *12*, 997. [CrossRef]
41. Chan, K.H.; Ke, W.; Im, S.K. CARU: A content-adaptive recurrent unit for the transition of hidden state in NLP. In *Neural Information Processing: 27th International Conference, ICONIP 2020, Bangkok, Thailand, 23–27 November 2020*; Proceedings, Part I 27; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 693–703.
42. Darwish, A.; Hassanien, A.E.; Das, S. A survey of swarm and evolutionary computing approaches for deep learning. *Artif. Intell. Rev.* **2020**, *53*, 1767–1812. [CrossRef]
43. Kiritchenko, S.; Mohammad, S. Examining Gender and Race Bias in Two Hundred Sentiment Analysis Systems. *arXiv* **2018**, arXiv:1805.04508.
44. Liu, H.; Dacon, J.; Fan, W.; Liu, H.; Liu, Z.; Tang, J. Does gender matter? Towards fairness in dialogue systems. *arXiv* **2019**, arXiv:1910.10486.
45. Raza, S.; Reji, D.J.; Ding, C. Dbias: Detecting biases and ensuring fairness in news articles. *Int. J. Data Sci. Anal.* **2024**, *17*, 39–59. [CrossRef] [PubMed]
46. Mehrabi, N.; Morstatter, F.; Saxena, N.; Lerman, K.; Galstyan, A. A survey on bias and fairness in machine learning. *ACM Comput. Surv.* **2021**, *54*, 1–35. [CrossRef]
47. Chen, Y.; Mahoney, C.; Grasso, I.; Wali, E.; Matthews, A.; Middleton, T.; Njie, M.; Matthews, J. Gender bias and under-representation in natural language processing across human languages. In Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society, Virtual Event, 19–21 May 2021; pp. 24–34.
48. Liu, X.; Gao, L.; Fiumara, G.; Meo, D.P. Key Node Identification Method Integrating Information Transmission Probability and Path Diversity in Complex Network. *Comput. J.* **2024**, *67*, 127–141. [CrossRef]
49. Liu, X.; Zhang, M.; Liu, Y.; Liu, C.; Li, C.; Wang Wei Zhang, X.; Bouyer, A. Semi-supervised Community Detection Method Based on Generative Adversarial Networks. *J. King Saud Univ. Comput. Inf. Sci.* **2024**, *36*, 102008. [CrossRef]
50. Almuzaini, A.A.; Singh, V.K. Balancing fairness and accuracy in sentiment detection using multiple black box models. In Proceedings of the 2nd International Workshop on Fairness, Accountability, Transparency and Ethics in Multimedia, Seattle, WA, USA, 20 October 2020; pp. 13–19.
51. Khoo, L.S.; Bay, J.Q.; Yap, M.L.K.; Lim, M.K.; Chong, C.Y.; Yang, Z.; Lo, D. Exploring and repairing gender fairness violations in word embedding-based sentiment analysis model through adversarial patches. In Proceedings of the 2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), IEEE, Taipa, Macao, 21–24 March 2023; pp. 651–662.
52. Pastaltzidis, I.; Dimitriou, N.; Quezada-Tavarez, K.; Aidinlis, S.; Marquenie, T.; Gurzawska, A.; Tzovaras, D. Data augmentation for fairness-aware machine learning: Preventing algorithmic bias in law enforcement systems. In Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, Seoul, Republic of Korea, 21–24 June 2022; pp. 2302–2314.
53. Dwork, C. Differential Privacy. In *Automata, Languages and Programming*; Springer: Berlin/Heidelberg, Germany, 2006.
54. Liu, X.; Xiao, W.; Liu, C.; Wang, W.; Li, C. Meta Graph Network Recommendation Based on Multi-Behavior Encoding. *J. King Saud Univ. Comput. Inf. Sci.* **2024**, *36*, 102050. [CrossRef]
55. Ramírez, J.J.E.; Gomez, J.C. Evolutionary learning of selection hyper-heuristics for text classification. *Appl. Soft Comput.* **2023**, *147*, 110721. [CrossRef]
56. Bryman, A. *Social Research Methods*; Oxford University Press: Oxford, UK, 2016.