Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105
Volume **9**, Number **4**, November **2024**

Weighted Plain Bayes-Based English Text Classification in Multilingual Interactive Environments

Wen-Li Hu*

Zhengzhou Railway Vocational and Technical College, Zhengzhou 450000, P. R. China
HU_060709@163.com

Daniel Zhang

College of Information Science and Technology
Multimedia University, Jalan Ayer Keroh Lama, 75450 Bukit Beruang, Melaka, Malaysia
vy0563@163.com

*Corresponding author: Wen-Li Hu
Received December 21, 2023, revised March 13, 2024, accepted June 14, 2024.

Abstract. *In today's world, the process of globalization is accelerating, and the communication between different countries and nationalities is becoming more and more frequent. In this background, this paper proposes a weighted plain Bayesian-based English text classification algorithm in a multilingual interactive environment (WTWNBA) to address the problems of low classification efficiency and long classification time of the current English text classification algorithm. Firstly, the orthogonal transformation method is used to eliminate the linear relationship between continuous attributes; the conditional probabilities of weighted discrete attributes and orthogonal transformed continuous attributes are differentiated and computed, so as to improve the generalization ability of the WNBA algorithm. Then, for the problem that the existing classification algorithms do not consider the influence of the location of words on the text, a weighted plain Bayesian-based English text classification algorithm is proposed for the multilingual interactive environment, which introduces interclass and intraclass discretization factors of the feature words and assigns different weights to different locations of the English words, which strengthens the ability of differentiating the information of feature words' class distributions, and realizes the accurate classification of the English text. The experimental results show that the Accuracy, Precision, Recall and F1 values of ETWNBA are more than 90% on each dataset of the experiment, and when the number of texts is 100, the classification time is 210s, which has high classification efficiency and low classification time.*
Keywords: Text categorization; Weighted plain Bayes; Orthogonal transformations; Discrete attributes; Multilingual interaction

1. **Introduction.** At present, the cooperation as well as communication of globalization is getting deeper and deeper, multilingual interaction is a difficult problem that must be paid attention to and solved, and the difference between languages is an important factor that affects the effect of information transmission between speakers of different languages [1, 2]. In the Internet, there is an imbalance in the information resources of different languages, according to the statistical report of Internet World Stats on commonly used languages on the Internet, it is pointed out that the most widely used language with the most data resources in the Internet is English, and other languages account for a

smaller percentage, but there are also a lot of speakers, which results in the difficulty of transferring information between users using different languages and the formation of a natural barrier to The formation of a natural barrier to mutual communication [3]. Therefore, how to be able to accurately and quickly select the text information that people need in the huge amount of English text information will be a problem worth exploring. In the past, people chose to classify text information manually, which is time-consuming and labor-intensive and has a low classification accuracy rate. Nowadays, it is obvious that manual classification simply relying on human beings is not enough to satisfy people's needs for classification in today's society [4, 5, 6]. Based on the idea of machine learning, the use of computers to simulate the human classification process for automatic text classification can help people to quickly and accurately obtain and identify the required text data information in the massive text data information, and has a wide range of potentials in language processing and understanding as well as information filtering.

1.1. **Related Work.** Text classification is an important development branch of machine learning algorithms, and its purpose is to organize and classify the chaotic and complicated text data in large data volumes in a regular manner, so as to facilitate information retrieval and other applications. Text classification algorithms are mainly divided into simple Bayesian algorithms [7], KNN text classification algorithms [8], decision tree text classification algorithms [9], random forest text classification algorithms [10], and support vector machine text classification algorithms [11], etc. Hinton [12] proposed back propagation algorithms, which converted the simple neural network model into a complex model, and contributed greatly to the development of subsequent deep learning. Du and Huang [13] proposed a language model based on recurrent neural network for text representation and text classification, and Enamoto et al. [14] proposed a convolutional neural network model for text categorization, which uses pre-trained text representation and convolutional operations to obtain contextual local information in the text, and then finally completes the text classification. Soni et al. [15] proposed a pooling algorithm for convolutional neural networks, which is the same as the pooling algorithm of convolutional neural networks. Xu et al. [16] improved the depth of convolutional neural network, and applied a deep convolutional neural network with 29 convolutional layers to text classification. Lyu and Liu [17] proposed a model combining recurrent neural network and convolutional neural network to obtain key information in text by maximizing the pooling layer to obtain the words that play a key role in text classification.

Subsequently, Jang et al. [18] proposed a bi-directional long and short-term memory network combined with a maximum pooling layer for text feature extraction, Singh et al. [19] proposed a hierarchical attention network combined with an attention mechanism for text classification, and Alfattni et al. [20] extracted important semantic features from text by combining an attention mechanism with a bi-directional long and short-term memory network, and achieved good results in text classification. Google team proposed a multi-attention transformer model with improved attention mechanism [21]. Keyvanpour and Imani [22] used self-training to categorize the sentiment factors expressed in multilingual texts. Namaghi et al. [23] used AdaBoost machine learning for semi-supervised text classification, but the same problem of unsatisfactory classification efficiency exists.

As a classic data mining algorithm in the field of machine learning, plain Bayesian classification is characterized by simple modeling and high execution efficiency. Therefore, Al-Salemi and Aziz [24] attempted to apply distributed plain Bayesian algorithm in text classification, and used the mutual information method to check the relevance of the feature set generated after feature selection to make up for the shortcomings of

the traditional plain Bayesian text classification, but the relevance operation takes a long time to compute. Tang et al. [25] used simple Bayesian feature weighting to classify textual sentiment data, but simple weighted simple Bayesian reduces the quality of the model, resulting in lower classification accuracy. In view of the problem that it is difficult to realize the assumption of feature independence of the traditional plain Bayesian text classification algorithm in real applications, many scholars have made relevant improvements to its research [26, 27, 28]. Most studies modify the feature weights through different probability densities and attribute weights, but they do not consider the error of self-training unlabeled samples in simple Bayesian algorithm.

1.2. **Contribution.** Aiming at the current problem of low efficiency of English text classification algorithms, a weighted plain Bayes-based English text classification algorithm in a multilingual interactive environment is proposed. Firstly, the contribution and correlation of mutual information are utilized to quantify the discrete attributes and the degree of correlation between the discrete attribute values to obtain their weights, so as to improve the classification accuracy of the WNBA algorithm. Then, to address the low performance of the existing English text classification algorithms, a weighted plain Bayesian-based English text classification algorithm is proposed for the multilingual interactive environment, which utilizes the expected cross-entropy ECE's function to compute the word frequency weights and extracts all the feature words in the English text to fuse into a feature dictionary, which strengthens the ability to differentiate the feature words' category distribution information, and realizes the accurate classification of the English text. The results show that by reasonably setting the weights, the English text can be classified accurately. The results show that the weighted plain Bayesian algorithm improves the performance of text classification by reasonably setting the number of weights and weighting multi-class attributes, and compared with the commonly used English text classification algorithms, the algorithm has higher classification accuracy and efficiency, and is suitable for text classification.

2. Theoretical analysis.

2.1. **Text classification algorithm.** Text categorization refers to the process of automatically determining one or several categories of unknown categories of text in a document collection according to certain rules based on predefined subject categories [28]. Macroscopically, text categorization can be regarded as a mapping process from text to categories. There exists a mapping function $f : A \times B \to \{S, G\}$, where $A = \{a_1, a_2, \cdots, a_n\}$ denotes the set of texts.

2.2. **Text Classification Process.** $B = \{b_1, b_2, \cdots, b_m\}$ denotes the set of categories, n and m denote the number of texts and categories respectively. For any data pair $\langle a_j, b_j \rangle$, if there is $f(a_j, b_j) = S$, then it means that the text a_j belongs to the category b_j; on the contrary, it means that the text a_j does not belong to the category b_j. Figure 1 shows the general process of text categorization, which mainly includes feature representation, feature extraction, classifier training, and performance evaluation.

After the feature representation and feature selection, each text is transformed into an m-dimensional feature vector, and the set C of training documents will be represented as a binary set consisting of feature vector x and category y.

$$C = \{(x^{(1)}, y_1), (x^{(2)}, y_2), \cdots, (x^{(M)}, y_M)\} \tag{1}$$

where $x^{(j)}$ is the m-dimensional feature vector and $y_j \in \{d_1, d_2, \cdots, d_l\}$ is the category labeling.

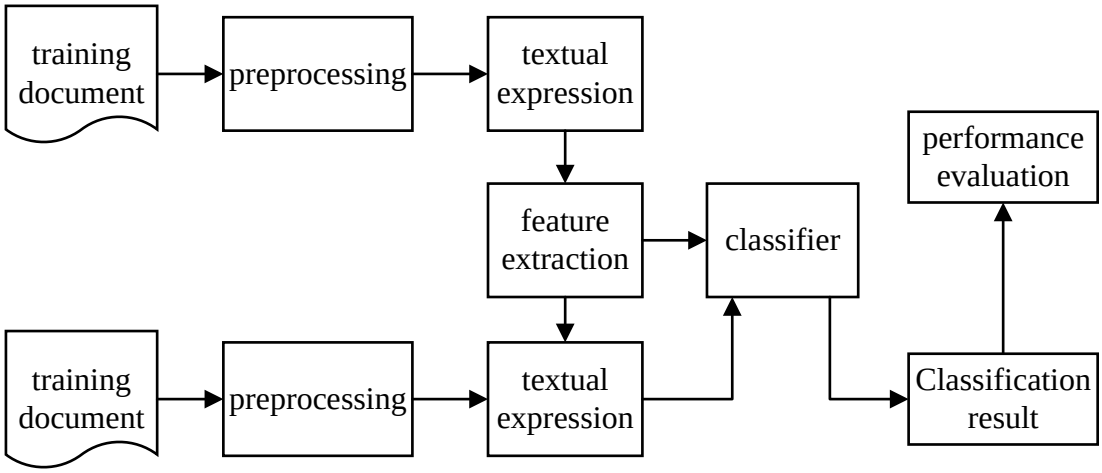

Figure 1. The general process of text categorization

The goal of text categorization is to learn the joint probability distribution of a set of training documents $P(X, Y)$, and to transform the joint probability into the product of the prior probability and the conditional probability distribution by means of Bayes' theorem.

$$P(X = x, Y = d_i) = P(Y = d_i) \cdot P(X = x|Y = d_i) \tag{2}$$

The sample size under each category was then counted according to the great likelihood estimation.

$$P(Y = d_i) = \frac{count(Y = d_i)}{M} \tag{3}$$

where $count(Y = d_i)$ refers to the amount of documents in the training document set with category d_i and M is the total amount of documents in the document set.

However, the value of the probability $P(X = x|Y = d_i)$ is difficult to estimate because the dimensionality of the feature vectors tends to be high, as can be easily seen from the following equation.

$$P(X = x|Y = d_i) = P(X_1 = x_1, X_2 = x_2, \cdots, X_m = x_m|Y = d_i) = \prod_{j=1}^{m} P(X_j = x_j|Y = d_i) \tag{4}$$

So here for the probability $P(X = x|Y = d_i)$ can again be calculated based on the great likelihood estimation.

$$P(X_j = x_j|Y = d_i) = \frac{count(X_j = x_j, y_j = d_i)}{count(y_j = d_i)} \tag{5}$$

In the prediction stage, we need to find d_i as the output y that maximizes the posterior probability $P(Y = d_i|X = x)$, which we can obtain by combining the Bayesian formula and the conditional independence assumption.

$$y = \arg\max_{d_i} P(Y = d_i) \cdot \prod_{j=0}^{m} P(X = x_j|Y = d_i) \tag{6}$$

where the probability distributions $P(Y = d_i)$ and $P(X_j = x_j|Y = d_i)$ can be calculated by Equation (3) and Equation (5).

2.3. **Weighted plain bayesian algorithm.** If $C = \{B_1, B_2, \cdots, B_n, D\}$ is the training dataset, where $\{B_1, B_2, \cdots, B_n\}$ is an n-attribute variable, and b_j is the value of the attribute B_j, a category variable with a total number of p categories. The conditional probability that a to-be-categorized instance $x_j = [b_1, b_2, \cdots, b_n]$ belongs to the class d_i, has, according to Bayes' theorem. The overall flow of the weighted plain Bayesian algorithm is shown in Figure 2.

$$P(d_i|b_1, b_2, \cdots, b_n) = \frac{P(b_1, b_2, \cdots, b_n|d_i)P(d_i)}{P(b_1, b_2, \cdots, b_n)} \tag{7}$$

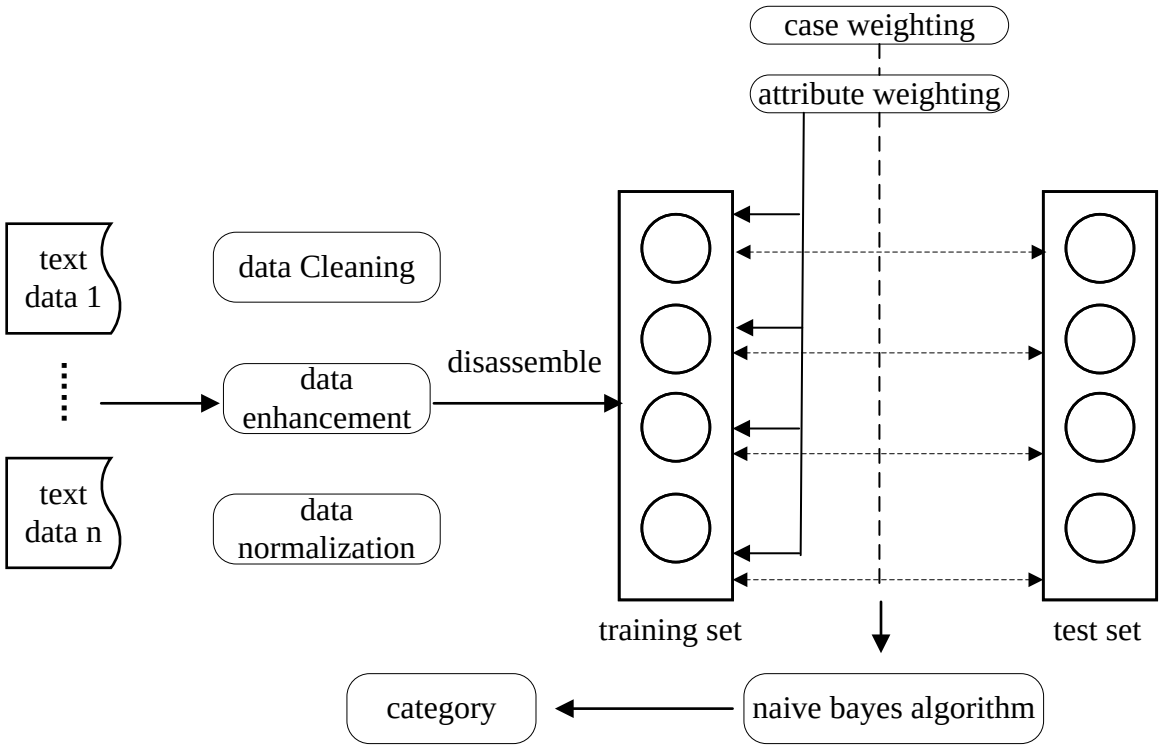

Figure 2. The whole flow of weighted naive Bayes algorithm

where $P(d_i|b_1, b_2, \cdots, b_n)$ is the posterior probability of making a categorical prediction, $P(b_1, b_2, \cdots, b_n|d_i)$ is the conditional probability of an attribute value of $[b_1, b_2, \cdots, b_n]$ given the value of the attribute under class d_i, $p(d_i)$ is the prior probability of class d_i, and $P(b_1, b_2, \cdots, b_n)$ is the prior probability of the attribute, which is the same for all classes.

Based on the precondition that the attributes are independent of each other, the conditional probability can be expressed as:

$$P(b_1, b_2, \cdots, b_n|d_i) = \prod_{j=1}^{n} P(b_j|d_i) \tag{8}$$

where $P(b_j|d_i)$ denotes the conditional probability of attribute b_j under class d_i. For the instance to be classified x_j, based on the principle of maximizing the a posteriori probability, the weight of the instance is incorporated into the prior probability and conditional

probability in the weighted plain Bayesian algorithm, and the predicted classification can be expressed as:

$$d(x) = \arg\max_{d_i \in D} P_{j=1}^{ins}(d_i) \prod_{j=1}^{n} P_{j=1}^{ins}(b_j|b_j)^{h_j^{att} \cdot h_j^{ins}} \tag{9}$$

where h_j^{att} is the weight of the j-th attribute, h_j^{ins} is the weight of the j-th training instance, the prior probability $P_{j=1}^{ins}(d_i)$ and conditional probability $P_{j=1}^{ins}(b_j|b_j)$ can be calculated as:

$$P_{j=1}^{ins}(d_i) = \frac{\sum_{i=1}^{m} h_i^{ins} g(d_j, d_i) + \frac{1}{p}}{\sum_{i=1}^{n} h_i^{ins} + 1} \tag{10}$$

$$P_{j=1}^{ins}(d_i|d_i) = \frac{\sum_{j=1}^{m} h_j^{ins} g(b_j, b_j) g(d_j, d_i) + \frac{1}{m_i}}{\sum_{j=1}^{m} h_j^{ins} g(d_j, d_i) + 1} \tag{11}$$

The computation of weights h_j^{att} and h_j^{ins} is shown in Equation (12) and Equation (13).

$$h_j^{ins} = 1 + s(x_j, y_j) \tag{12}$$

$$h_j^{att} = \frac{1}{1 + e^{-\left(1 - \frac{1}{n-1} \sum_{i=1}^{n} M(b_j, b_i)\right)}} \tag{13}$$

3. **Improved weighted plain Bayesian algorithm.** Since the existing weighted plain Bayesian algorithm ignores the correlation of multidimensional attributes of the data, which leads to great application limitations of the classification algorithm. In this regard, the improved plain Bayesian algorithm with the fusion of multiclass attribute weighting and orthogonal transformation is proposed. The contribution and correlation information are used to quantify the correlation degree between discrete attributes and discrete attribute values to obtain their weights. The orthogonal transformation method is used to eliminate the linear relationship between continuous attributes. The conditional probabilities of weighted discrete attributes and orthogonal transformed continuous attributes are differentially calculated, thus obtaining a higher classification accuracy and improving the generalization ability of the algorithm. The framework of the enhanced algorithm is shown in Figure 3.

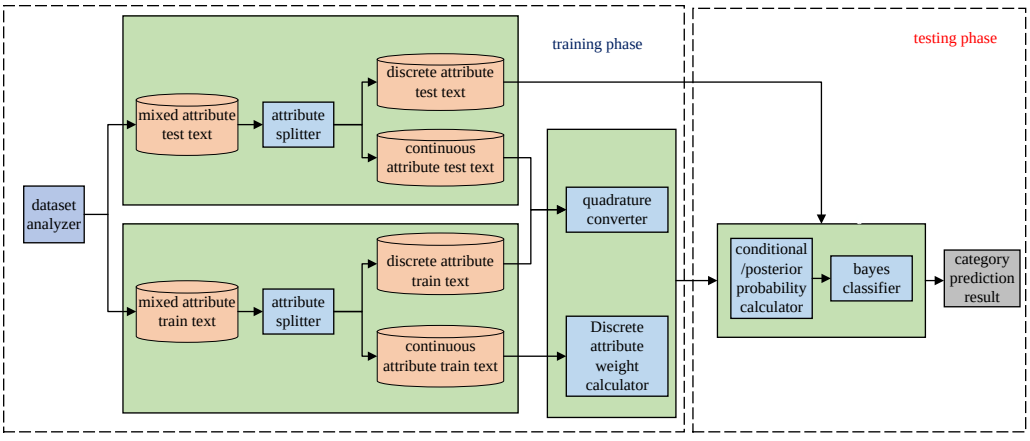

Figure 3. Enhanced weighted naive Bayes algorithm framework

The enhanced algorithm in this paper, on the one hand, adds multiple discrete attributes and discrete attribute value weights to the conditional probability calculation of "frequency count estimation" for discrete attributes, and then uses the conditional probability calculation of "probability density function" for continuous attributes after orthogonal transformations to calculate the conditional probability. The conditional probability is then calculated using the "probability density function" conditional probability for the continuous attribute after orthogonal transformation. On the other hand, the a posteriori probability formula for multidimensional mixed attributes is obtained by combining the two different conditional probability methods and adding the weights of class attributes. Firstly, Equation (14) is used to predict the class of the test sample.

$$\hat{d}(x_j) = \arg\max_{d_c \in D} \hat{P}(d_c|x_j; C_c, T_c, H) \infty = \arg\max_{d_c \in D} \hat{P}(d_c) P_{C_c}(x_{jC}|d_c) \hat{P}_{T_c}(x_{jT}|d_c) \tag{14}$$

where x_j is the j-th test sample in the test dataset, $\hat{d}(x_j)$ is the class prediction function of the test sample x_j, D is the set of classes, d_c is the d-th class in the set of classes D, C_c is the discrete attribute identifier of the class d_c, T_c is the continuous attribute identifier of the class d_c. H is the set of weights, and $W = \{W_{ilc}, W_{i,b}, W_{il}\}$ in the set W_{ilc} is the joint discrete attribute value weight of the i-th discrete attribute of class d_c that takes the value of b_i on the l-th discrete attribute that takes the value of b_j, $W_{i,b}$ is the weight of the discrete attribute value b_j, and W_{il} is the joint discrete attribute weight of the i-th discrete attribute and the l-th discrete attribute in the sample. x_{jC} is the discrete attribute vector and x_{jT} is the continuous attribute vector. The prior probability is represented by Equation (15).

$$\hat{P}(d_c) = \left(\frac{M_c + 1}{M + C}\right)^{H_c} \tag{15}$$

where M_c is the number of samples in the class d_c of the training dataset, M is the number of samples in the training dataset, C is the number of classes in the training dataset, and H_c is the weight of the class d_c.

Then constructing class-specific joint discrete attribute value weights via Equation (16).

$$H_{ilc} = \frac{m_{ic} + \frac{Count(b_i, b_j, d_c) + m_{ic}/m_{ilc}}{Count(b_j, d_c) + 1/m_{ilc}}}{n_{jd}} \tag{16}$$

where m_{ic} and m_{ilc} are the number of values of the i-th discrete attribute and the l-th discrete attribute in class d_c respectively, and $1 \le i \le m_c$, $1 \le l \le m_c$, m_c are the number of sample discrete attributes. $Count()$ represents the number of samples of discrete attributes in the class, and H_{ilc} denotes the joint discrete attribute value weight of the j-th discrete attribute with value b_j on the kth discrete attribute with value b_i in class d_c. The values H_{ilc}, $Count(b_i, b_j, d_c)$, $Count(b_j, d_c)$, m_{ic}, and m_{ilc} are jointly determined.

The weight of a single discrete attribute value is obtained from the product of the correlation between the discrete attribute value and the class as well as between the discrete attribute values, which is calculated as follows:

Step 1: The correlation between discrete attribute values and class label c, and the correlation between discrete attribute values b_i and b_l are measured using the correlation mutual information as shown below.

$$I(b_i, d) = \sum_{c=1}^{c} P(b_i, d_c) \log \frac{P(b_i, d_c)}{P(b_i) P(d_c)} \tag{17}$$

$$I(b_i, b_j) = P(b_i, b_j) \log \frac{P(b_i, b_j)}{P(b_i)P(b_j)} \tag{18}$$

where $I(b_i, d)$ is the correlation information between the discrete attribute value b_i and the class label d, $P(b_i, d_c)$ is the joint probability between the discrete attribute value b_i and the class attribute value d_c. $P(b_i)$ and $P(d_c)$ are the a priori probabilities of the discrete attribute value b_i and the class attribute value d_c, respectively. $I(b_i, b_j)$ is the correlation information between the discrete attribute values b_i and b_j, $P(b_i, b_j)$ is the joint probability of the discrete attribute values b_i and b_j, $P(b_i)$ and $P(b_j)$ are the prior probabilities of the discrete attribute values b_i and b_j, respectively.

Step 2: Use Equation (19) and Equation (20) to normalize $I(b_i, d)$ and $I(b_i, b_j)$ respectively.

$$R(b_i, d) = \frac{I(b_i, d)}{\sum_{i=1}^{m_c} I(b_i, d)/m_c} \tag{19}$$

$$R(b_i, b_j) = \frac{m_c \cdot m_c \cdot I(b_i, b_j)}{\sum_{i=1}^{m_c} \sum_{j=1, j\neq i}^{m_c} I(b_i, b_j)/[m_c(m_c - 1)]} \tag{20}$$

where m_c is the number of discrete attributes, $R(b_i, d)$ and $R(b_i, b_j)$ are the correlation information between the normalized discrete attribute values and classes, and between discrete attribute values, respectively, which can be used to calculate the weights of individual discrete attribute values.

Step 3: Individual discrete attribute value weights are defined by Equation (21).

$$H_{i,b} = R(b_i, d) \times \sum_{j=1, j\neq i}^{m_c} R(b_i, b_j) \tag{21}$$

where $H_{i,b}$ is the weight of the discrete attribute value b_i, which is used to quantify the extent to which the discrete attribute value b_i contributes to the classification of the samples and will not be changed after its value is determined with a fixed training data set.

The conditional mutual information of discrete attributes b_i and b_j to obtain joint discrete attribute weights is defined by Equation (22).

$$H_{i,j} = \frac{I(b_i, b_j|d)}{\sum_{i=1, i\neq j}^{m_c} I(b_i, b_j|d)/m_c} \tag{22}$$

4. **Weighted Plain Bayes-Based English Text Classification in Multilingual Interactive Environments.**

4.1. **English Text Feature Selection and Feature Dictionary Building.**

Existing English text classification algorithms only evaluate the importance of words in the text according to the frequency of their occurrence in the text, without considering the influence of the location of the words on the text, so the accuracy of text classification is limited to a certain extent. In order to solve this problem, a weighted plain Bayesian-based English text classification method is proposed in a multilingual interactive environment, which utilizes the expected cross-entropy (ECE) and ECE'-valued functions to calculate the word frequency weights and extract all the feature words in the English text to form a feature dictionary. This method introduces inter- and intra-class discretization factors of the feature words and assigns different weights to different locations of the English text to reflect the effect of the location of the English words on the text.

The inter-class and intra-class dispersion factors are introduced to assign different weights to the different positions of English words, so as to reflect the different effects of the positions of English words on the text, highlight the importance of key positions, strengthen the ability to distinguish the information of the distribution of feature word classes, and realize the accurate classification of English text. The flow of the algorithm is shown in Figure 4.

The English text is calculated by the expected cross entropy (ECE) to get the text distribution probability and feature words, and the specificity of the feature words itself is enhanced with the increase of the expected cross entropy. The expected cross-entropy expression of the text h_j is as follows:

$$ECE(h_j) = P(h_j) \sum_i P(D_i|h_j) \log P(D_i|h_j)/P(D_i) \tag{23}$$

where $P(h_j)$ indicates the number of times that the text h_j appears in all the texts, $P(D_i)$ indicates the number of English text feature words in the text h_j per D_i count, $P(D_i|h_j)$ indicates the number of randomly sampled feature correlations per $D_i|h_j$ count.

The core of constructing a lexicon of English text features is to select feature words that can represent certain subordinate features and have the significance of summarizing and guiding. The selection of feature words is based on the outline content of the English text and utilizes the concise word frequency to summarize the overall information of the English text to the greatest extent. The feature word weights of the text h_j are expressed as follows.

$$ECE(h_j, D_i) = P(h_j)P(D_i|h_j)P(D_i|h_j) \log \left(\frac{P(D_i|h_j)}{P(D_i)} \right) \tag{24}$$

where the text h_j and the category D_i are negatively correlated.

When the relative value of $P(h_j|D_i)$ is small and the relative value of $P(D_i|h_j)$ is large, the effect of the text h_j on the category D_i is negligible, and the weight of the text h_j in the randomized classification $ECE(h_j, D_i)$ can be found out by $P(D_i)$.

The ECE' evaluation function for the text data in the category D_i is given in the following equation:

$$ECE'(h_j, D_i) = -\sum_{w \neq i} ECE(h_j, D_i) + ECE(h_j, D_i) \tag{25}$$

where $ECE(h_j, D_i)$ denotes the weight value of the category D_i, and $-\sum_{w \neq i} ECE(h_j, D_i)$ denotes the end of weighting all the data of h_j when $w \neq i$, and w denotes the categorized category.

Based on the ECE' value of the text data, the feature words of any category D_i can be calculated with the following expression:

$$h_{i,j} = ECE'(h_j, D_i)sk \tag{26}$$

where k denotes English text coefficients, and s denotes feature word parameters.

4.2. **English text classification based on weighted plain bayes.** In this paper, we introduce the interclass discretization factor DJ_{bD} and intraclass discretization factor DJ_{jD} for English text feature words. The standard deviation of the distribution of word frequency of a feature word in different categories of document sets DJ_{bD} is used to describe the inter-class distribution information of the feature word. DJ_{jD} describes the intra-class distribution information of the feature word by the difference between the word frequency of the feature word in the category D_i and the word frequency of the documents

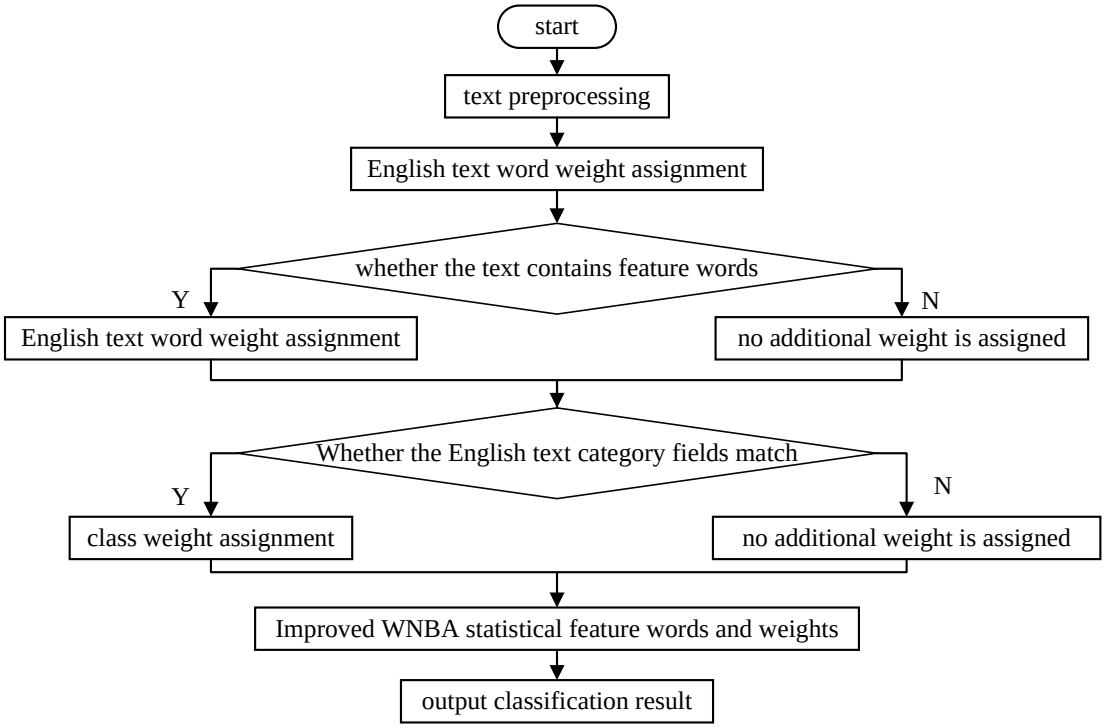

Figure 4. English Text Classification Algorithm Flow

that actually contain the feature word in the category D_i. With the introduction of class information, the improved algorithm enhances the ability to distinguish feature word class distribution information. The methods of measuring the inter-class dispersion DJ_{bD} and intra-class dispersion DJ_{jD} are given below.

$$DJ_{bD}(s_i) = 2\arctan(T(s_i))/\pi \tag{27}$$

$$DJ_{jD}(s_i, D_i) = 2\arctan(t(s_i, D_i))/\pi \tag{28}$$

where $T(s_i)$ refers to the standard deviation of the distribution of the word frequency of the feature word s_i among the categories, $t(s_i, D_i)$ refers to the difference between the word frequency of the feature word s_i in the category D_i and the word frequency of the document that actually contains the feature word in the category D_i, calculated as follows.

$$T(s_i) = \sqrt{\left[\sum_{l=1}^{|D|}\left(TF\left(s_i, D_i\right) - \overline{TF\left(s_i\right)}\right)^2\right] / \left(|D-1|\right)} \tag{29}$$

$$t(s_i, D_i) = TF(s_i, D_i) \cdot \frac{M(D_i)}{m(s_i, D_i)} - TF(s_i, D_i) \tag{30}$$

where $TF(s_i, d_i)$ indicates the frequency of occurrence of the feature word s_i in the category D_i, $TF(s_i)$ indicates the average frequency of occurrence of the feature word s_i in each category, $M(D_i)$ indicates the number of documents in the category D_i, $m(s_i, D_i)$ indicates the number of documents containing the feature word s_i in the category D_i, and D is the total number of categories in the document set.

The category to which the text to be categorized belongs is also the one with the largest a posteriori probability value, and the equation can be expressed as follows.

$$P(d_i) = \frac{\sum_{j=1}^{J} I(y_j = d_i)}{m(s_i, D_i)} \tag{31}$$

Let x denote the input random text feature word vector and y denote the output random feature word vector. Let the set of values of the e-th feature be $x_e = \{b_1, b_2, \ldots, b_{ek}\}$, then the conditional probability is calculated as follows.

$$P(s_l \mid d_i, H_j^{s_l}) = \frac{\sum_{j=1}^{n} H_j^{s_l} I(s_l = d_i) + 1}{\sum_{j=1}^{n} H_j^{s_l} I(s_j = d_i) + d_i + n} \tag{32}$$

where $I(\cdot)$ is the indicator function, the output is 1 when the input is true and 0 when the input is false, and n is the number of feature words.

For the purpose of improving the classification accuracy, an enhanced weighted plain Bayesian algorithm is used to compare the weight values of all English texts with the following functional expression.

$$h(T_j) = \sum_{j=1}^{m} dis[d_j^T, mw(d_j^T)] + h_{i,j} - \sum_{j=1}^{m} dis[d_j^T, mw(d_j^T)] \tag{33}$$

where $mw(d_j^T)$ denotes the difference in weights between text i and text j.

The final classification result D_{map} is shown below.

$$D_{map} = \arg\max_{d_i \in D} P(d_i \mid C_m) h(T_j) = \arg\max_{d_i \in D} P(d_i) \prod_{j=1}^{n} P(s_l \mid d_i, H_j^{s_l}) h(T_j) \tag{34}$$

5. **Experimentation and Analysis.**

5.1. **Optimized performance of WNBA.** To estimate the performance of weighted plain Bayesian-based English text classification (ETWNBA) in a multilingual interactive environment, firstly, the WNBA algorithm and the weighted optimization WNBA algorithm with multi-class attributes designed in this paper are simulated to verify the optimization performance of the algorithms; secondly, the common English text classification algorithms and the ETWNBA algorithm proposed in this paper are simulated to verify the performance of different classification algorithms.

The classification performance indexes are Accuracy, Precision, Recall, and the reconciled mean of Precision and Recall F1. The comparison algorithms are trained in Python v3.7 environment. For ease of description, the literature [15] is denoted as TCNET, the literature [17] as BICNN, the literature [29] as NASVM, and the algorithm in this paper as ETWNBA.

The data source of text simulation in multilingual interactive environment is Twitter, and two kinds of English text data are categorized. The number of samples and the number of categories to be categorized are shown in Table 1.

Table 1. Text Sets in Multilingual Interactive Environments

Dataset number	**Text name**	**Number of texts**	**Number of categories**
1	sneakers	39247	11
2	handsets	16294	8

To verify the Accuracy (Acc), Precision (Prec), Recall (Rec) and F1 value of the multi-class attribute weighting designed in this paper on the plain Bayesian English text classification in a multilingual interactive environment, the plain Bayesian algorithm (NBA), WNBA, and ETWNBA are used to simulate the 2 datasets in Table 1, and the results are shown in Table 2.

Table 2. Text Sets in Multilingual Interactive Environments

Dataset number	Algorithm	Acc	Prec	Rec	F1
1	NBA	0.8164	0.8053	0.8174	0.8113
	WNBA	0.8351	0.8162	0.8295	0.8228
	ETWNBA	0.9247	0.9083	0.9143	0.9113
2	NBA	0.8276	0.8392	0.8135	0.8262
	WNBA	0.8491	0.8264	0.8376	0.8320
	ETWNBA	0.9361	0.9176	0.9228	0.9202

As can be seen from Table 2, in the English text classification of the two datasets, ETWNBA shows better performance, with all the four indexes exceeding 0.9, while the values of the four indexes of NBA classification are maintained at around 0.8. The maximum classification accuracy of ETWNBA is 93.61%, while that of NBA is 82.76%, which is a big difference between the two, and the classification performance of ETWNBA is not ideal under the multilingual interactive environment, but after optimization by weighting multiple attributes, the classification performance improves significantly, mainly because more accurate attribute weights are obtained after weight optimization. The text classification effect of WNBA in the multilingual interactive environment is not ideal, but after optimizing the weighting of multi-class attributes, the classification performance improves obviously, mainly because more accurate attribute weights are obtained after weight optimization. In the following, we will continue to compare the classification efficiency of the two algorithms.

The classification time performance of different algorithms is shown in Figure 5. ETWNBA takes the shortest time to classify the text, and the difference between NBA and WNBA is very small, which is due to the fact that NBA does not have a weighting parameter solving process, so it is more time-saving, and WNBA and ETWNBA both need to solve for the weights, but the experiments found that the optimization of multi-class attribute weighting does not increase the time consumption, because the time to solve for the optimal attribute weights becomes shorter after the optimization. However, it is found that the optimization of multi-class attribute weighting does not increase the time consumption, because the time to solve the optimal attribute weights is shorter after the optimization.

5.2. **Experimental comparison and analysis.** In order to further verify the performance of different algorithms in English text categorization, three commonly used English text categorization algorithms are used to simulate the two datasets in Table 1. Due to space limitation, only the classification performance of dataset 1 is captured, as shown in Figure 6. The classification accuracy of the four algorithms is directly proportional to the classification time, and when the classification time is 400 s, the accuracy of ETWNBA is 91%, the accuracy of NASVM is 83%, the accuracy of BICNN is 72%, and the accuracy of TCNET is 69%. ETWNBA has the highest accuracy in text classification, which is more than 0.9 when it is stable, and the worst accuracy is less than 0.9 when it is stable, which

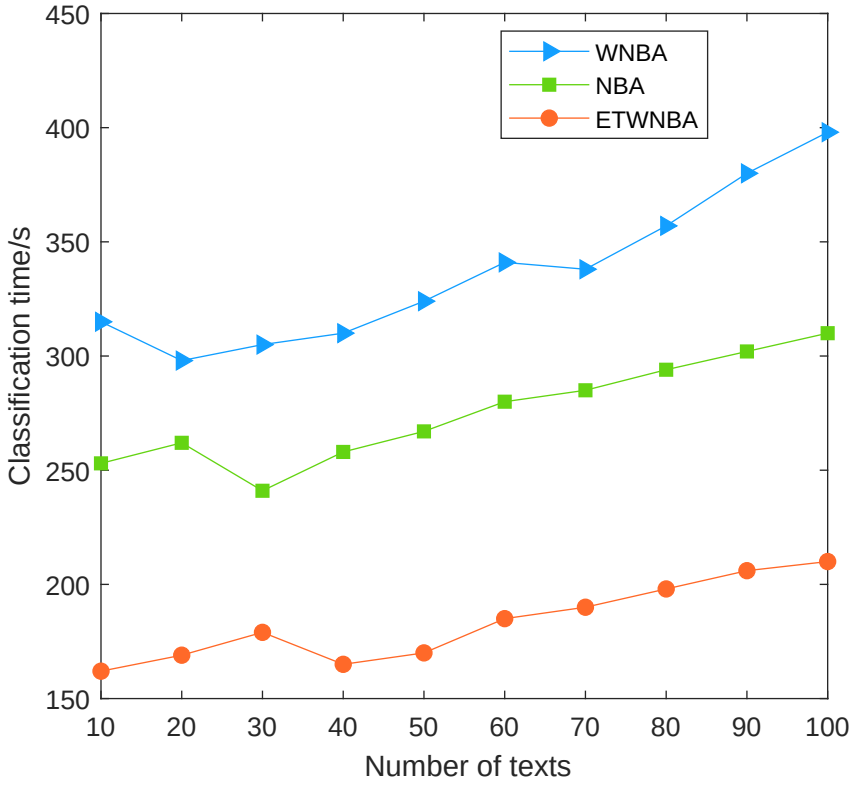

Figure 5. Classification time performance of different algorithms

is less than 0.5 when it is stable. The classification accuracy of ETWNBA algorithm is the highest, which is more than 0.9 when stable, and the classification accuracy of TCNET is the worst, which is less than 0.8. Therefore, in the case of the same accuracy, the proposed algorithm in this paper has an obvious advantage in the classification time performance over other algorithms.

Simulation of the classification stability of the four algorithms in English text is carried out to verify the performance of the four algorithms in terms of the root mean square error (RMSE) of the accuracy, and the results are shown in Table 3, from which it can be seen that, for the two datasets, the RMSE of the classification accuracy is the best in the case of ETWNBA, and the worst in the case of TCNET. When categorizing English text in a multilingual interactive environment, too many categories cause the classification accuracy to fluctuate a lot in multiple classifications, which also indicates that the classification accuracy RMSE value is sensitive to the number of categories, and when categorizing multiple categories, it is necessary to control the fluctuation of classification accuracy by combining with the attributes of the English text itself, in order to ensure the stability of the categorization of English text.

Table 3. RMSE of accuracy for different algorithms

Dataset number	**Accuracy of RMSE**			
	TCNET	**BICNN**	**NASVM**	**ETWNBA**
1	0.0864	0.0762	0.0549	0.0329
2	0.0817	0.0694	0.0618	0.0351

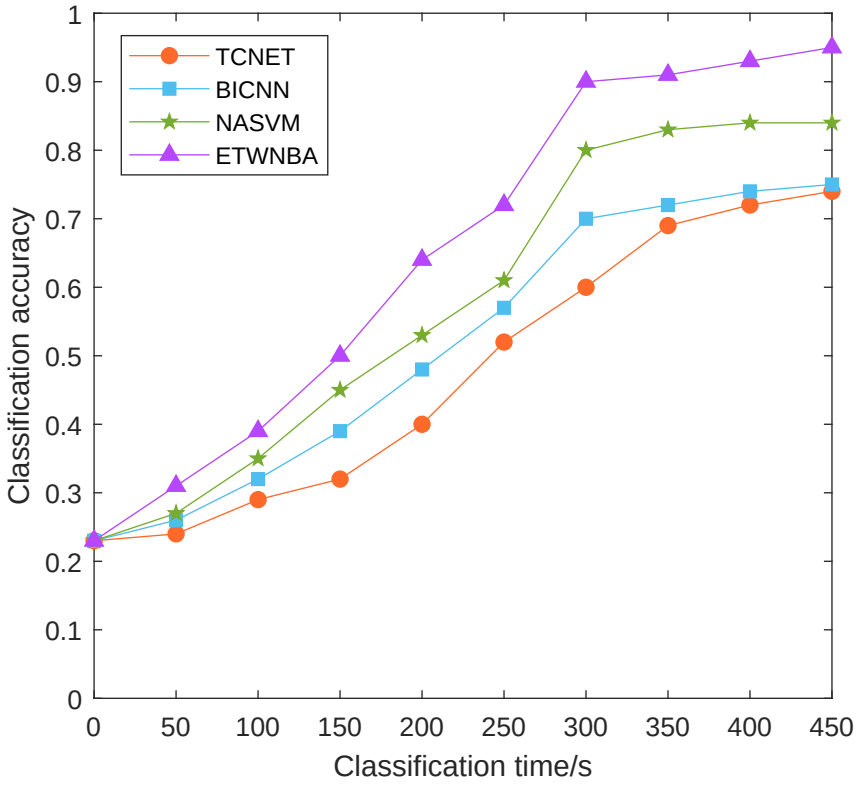

Figure 6. Comparison of classification efficiency

6. **Conclusion.** Aiming at the issue that the accuracy of the current English text classification algorithm is not high, a weighted plain Bayesian-based English text classification algorithm is proposed for the multilingual interactive environment. Firstly, the weights of multi-class discrete attributes and discrete attribute values are added into the conditional probability calculation of frequency count estimation for discrete attributes, and then the conditional probability of density function is used to differentiate the orthogonal transformed continuous attributes, in order to optimize the WNAB. Then for the problem of low classification effect of existing English text classification algorithms, a weighted plain Bayesian-based English text classification method in multilingual interactive environment is developed, in which the distribution probability of text and feature words are obtained through the expected cross-entropy ECE, and the inter-class distribution information of feature words is described by the standard deviation of the distribution of the word frequency of the feature words in the different categories of the document set, which strengthens the ability of distinguishing the class distribution information of feature words by the introduction of the over-class information. The introduction of over-class information strengthens the ability of distinguishing feature word class distribution information. The experimental results show that the method designed in this paper has high accuracy, precision, recall and F1 values as well as short classification time, which exhibits a good classification performance.

REFERENCES

[1] O. Garcia, "The multiplicities of multilingual interaction," *International Journal of Bilingual Education and Bilingualism*, vol. 21, no. 7, pp. 881-891, 2018.

[2] M. H. Almeida, and S. Melo-Pfeifer, "Introduction: Multilingual interaction–Dynamics and achievements," *International Journal of Bilingual Education and Bilingualism*, vol. 21, no. 7, pp. 781-787, 2018.

[3] D. Gorter, "Multilingual interaction and minority languages: Proficiency and language practices in education and society," *Language Teaching*, vol. 48, no. 1, pp. 82-98, 2015.
[4] T.-Y. Wu, J. Lin, Y. Zhang, and C.-H. Chen, "A Grid-Based Swarm Intelligence Algorithm for Privacy-Preserving Data Mining," *Applied Sciences*, vol. 9, no. 4, 774, 2019.
[5] T.-Y. Wu, J. C.-W. Lin, U. Yun, C.-H. Chen, G. Srivastava, and X. Lv, "An efficient algorithm for fuzzy frequent itemset mining," *Journal of Intelligent & Fuzzy Systems*, vol. 38, no. 5, pp. 5787-5797, 2020.
[6] A. Dhar, H. Mukherjee, N. S. Dash, and K. Roy, "Text categorization: past and present," *Artificial Intelligence Review*, vol. 54, pp. 3007-3054, 2021.
[7] W. J. vander Linden, and H. Ren, "A fast and simple algorithm for Bayesian adaptive testing," *Journal of Educational and Behavioral Statistics*, vol. 45, no. 1, pp. 58-85, 2020.
[8] Z. Yong, L. Youwen, and X. Shixiong, "An improved KNN text classification algorithm based on clustering," *Journal of Computers*, vol. 4, no. 3, pp. 230-237, 2009.
[9] B. Charbuty, and A. Abdulazeez, "Classification based on decision tree algorithm for machine learning," *Journal of Applied Science and Technology Trends*, vol. 2, no. 01, pp. 20-28, 2021.
[10] K. Shah, H. Patel, D. Sanghvi, and M. Shah, "A comparative analysis of logistic regression, random forest and KNN models for the text classification," *Augmented Human Research*, vol. 5, pp. 1-16, 2020.
[11] V. Mitra, C.-J. Wang, and S. Banerjee, "Text classification: A least square support vector machine approach," *Applied Soft Computing*, vol. 7, no. 3, pp. 908-914, 2007.
[12] G. E. Hinton, "Deep belief networks," *Scholarpedia*, vol. 4, no. 5, pp. 5947, 2009.
[13] C. Du, and L. Huang, "Text classification research with attention-based recurrent neural networks," *International Journal of Computers Communications & Control*, vol. 13, no. 1, pp. 50-61, 2018.
[14] L. Enamoto, L. Weigang, and G. P. R. Filho, "Generic framework for multilingual short text categorization using convolutional neural network," *Multimedia Tools and Applications*, vol. 80, pp. 13475-13490, 2021.
[15] S. Soni, S. S. Chouhan, and S. S. Rathore, "TextConvoNet: A convolutional neural network based architecture for text classification," *Applied Intelligence*, vol. 53, no. 11, pp. 14249-14268, 2023.
[16] J. Xu, Y. Cai, X. Wu, X. Lei, Q. Huang, H.-f. Leung, and Q. Li, "Incorporating context-relevant concepts into convolutional neural networks for short text classification," *Neurocomputing*, vol. 386, pp. 42-53, 2020.
[17] S. Lyu, and J. Liu, "Convolutional recurrent neural networks for text classification," *Journal of Database Management*, vol. 32, no. 4, pp. 65-82, 2021.
[18] B. Jang, M. Kim, G. Harerimana, S.-u. Kang, and J. W. Kim, "Bi-LSTM model to increase accuracy in text classification: Combining Word2vec CNN and attention mechanism," *Applied Sciences*, vol. 10, no. 17, pp. 5841, 2020.
[19] G. Singh, A. Nagpal, and V. Singh, "Optimal feature selection and invasive weed tunicate swarm algorithm-based hierarchical attention network for text classification," *Connection Science*, vol. 35, no. 1, pp. 2231171, 2023.
[20] G. Alfattni, N. Peek, and G. Nenadic, "Attention-based bidirectional long short-term memory networks for extracting temporal relationships from clinical discharge summaries," *Journal of Biomedical Informatics*, vol. 123, pp. 103915, 2021.
[21] G. Liu, and J. Guo, "Bidirectional LSTM with attention mechanism and convolutional layer for text classification," *Neurocomputing*, vol. 337, pp. 325-338, 2019.
[22] M. R. Keyvanpour, and M. B. Imani, "Semi-supervised text categorization: Exploiting unlabeled data using ensemble learning algorithms," *Intelligent Data Analysis*, vol. 17, no. 3, pp. 367-385, 2013.
[23] H. Hasan Nezhad Namaghi, H. Mashayekhi, and M. Zahedi, "Detecting Concept Drift in Data Stream Using Semi-Supervised Classification," *Signal and Data Processing*, vol. 18, no. 4, pp. 153-164, 2022.
[24] B. Al-Salemi, and M. J. A. Aziz, "Statistical bayesian learning for automatic arabic text categorization," *Journal of Computer Science*, vol. 7, no. 1, pp. 39, 2011.
[25] B. Tang, S. Kay, and H. He, "Toward optimal feature selection in naive Bayes for text categorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 9, pp. 2508-2521, 2016.
[26] M. Mendoza, "A new term-weighting scheme for naïve Bayes text categorization," *International Journal of Web Information Systems*, vol. 8, no. 1, pp. 55-72, 2012.
[27] K. Deshmukh, S. Raut, J. Bhargaw, and A. Saurkar, "An overview on implementation using hybrid naïve Bayes algorithm for text categorization," *International Journal on Future Revolution in Computer Science & Communication Engineering*, vol. 4, no. 3, pp. 142-146, 2018.

[28] W. Zhang, X. Tang, and T. Yoshida, "TESC: An approach to TExt classification using Semi-supervised Clustering," *Knowledge-Based Systems*, vol. 75, pp. 152-160, 2015.

[29] C.-M. Tan, Y.-F. Wang, and C.-D. Lee, "The use of bigrams to enhance text categorization," *Information Processing & Management*, vol. 38, no. 4, pp. 529-546, 2002.

[30] M. Goudjil, M. Koudil, M. Bedda, and N. Ghoggali, "A novel active learning method using SVM for text classification," *International Journal of Automation and Computing*, vol. 15, pp. 290-298, 2018.