symmetry

MDPI

Article

A Novel Classification Method Based on a Two-Phase Technique for Learning Imbalanced Text Data

Der-Chiang Li [1,*], Szu-Chou Chen [2], Yao-San Lin [3] and Wen-Yen Hsu [2]

[1] Department of Industrial and Information Management, National Cheng Kung University, Tainan City 70101, Taiwan
[2] Institute of Information Management, National Cheng Kung University, Tainan City 70101, Taiwan; r78081033@gs.ncku.edu.tw (S.-C.C.); r36091050@gs.ncku.edu.tw (W.-Y.H.)
[3] Singapore Centre for Chinese Language, Nanyang Technological University, Singapore 279623, Singapore; yao-san.lin@sccl.sg
* Correspondence: lidc@mail.ncku.edu.tw

Abstract: The problem of imbalanced data has a heavy impact on the performance of learning models. In the case of an imbalanced text dataset, minority class data are often classified to the majority class, resulting in a loss of minority information and low accuracy. Thus, it is a serious challenge to determine how to tackle the high imbalance ratio distribution of datasets. Here, we propose a novel classification method for learning tasks with imbalanced test data. It aims to construct a method for data preprocessing that researchers can apply to their learning tasks with imbalanced text data and save the efforts to search for more dedicated learning tools. In our proposed method, there are two core stages. In stage one, balanced datasets are generated using an asymmetric cost-sensitive support vector machine; in stage two, the balanced dataset is classified using the symmetric cost-sensitive support vector machine. In addition, the learning parameters in both stages are adjusted with a genetic algorithm to create an optimal model. A Yelp review dataset was used to validate the effectiveness of the proposed method. The experimental results showed that the proposed method led to a better performance subject to the targeted dataset, with at least 75% accuracy, and revealed that this new method significantly improved the learning approach.

Keywords: imbalanced data; sentiment analysis; text mining; support vector machine

Citation: Li, D.-C.; Chen, S.-C.; Lin, Y.-S.; Hsu, W.-Y. A Novel Classification Method Based on a Two-Phase Technique for Learning Imbalanced Text Data. *Symmetry* **2022**, *14*, 567. https://doi.org/10.3390/sym14030567

Academic Editor: László T. Kóczy

Received: 28 February 2022
Accepted: 11 March 2022
Published: 13 March 2022

1. Introduction and Background

1.1. Background of Imbalanced Data

Over the past few decades, the rapid development of machine learning and artificial intelligence technologies have transformed the business plans of enterprises as an upgrade strategy, where the quality of data is the key to success because data can provide the fundamental information necessary to create management models both efficiently and scientifically. For example, maintaining customer satisfaction, where data analysis technique plays a crucial role, is vitally important in order to evaluate and improve the development of companies and to reduce customer switching behavior [1]. In the tourism business, the lifestyle of contemporary residents has evolved, where residents often surf the internet for information, and 53% of travelers state that they would be unwilling to book a hotel without reliable reviews. There is also a phenomenon where a 10% increase in travel review ratings will increase bookings by approximately 5% [2]. Therefore, tourism companies are exerting a great deal of effort toward meeting the demands of customers by analyzing online reviews. With this strategy, companies can manage and maintain customer loyalty by meeting their specific demands.

Currently, classification has been widely used in data analysis, and most machine learning models assume that the distributions of datasets are normal among different

classes. However, most datasets tend to be skewed, which is called an imbalanced data problem and is detrimental to the effectiveness of these models.

The definition of imbalanced data is a situation where the minority class features cannot be rationally learned by models because there are more majority class samples than minority class samples. The standard machine learning procedure can result in poor classification effectiveness when dealing with imbalanced datasets. Recently, Ockham's razor theory has been considered as an effective method by which to optimize models. However, taking decision trees as an example, cutting branches may not be a useful way to optimize trees when the sample distribution is unknown. If the distribution of samples is imbalanced, the Ockham's razor method may not be effective [3]. Additionally, imbalanced data will cause several sub-problems, such as class overlapping and small disjuncts. These problems have to be solved in order to alleviate imbalanced data [4].

The imbalanced data problem has been considered a serious issue in the machine learning field for many years. In the medical and healthcare fields [5–8], the detection of cancer cells often incurs misclassification situations due to classifying normal patients as the cancer group and wasting medical resources to heal them. Even worse, misclassifying a cancer patient into the normal group can delay treatment and raise the mortality rate [9].

In the study of text datasets, one reason why these datasets are skewed such that the sentiment of the dataset may be misclassified by models is because they often consider the distribution of text datasets to be balanced [10]. Another reason for this is that some companies create fake positive reviews to establish good reputations for their products, and people often give more positive reviews than negative reviews because they are criticizing products using the same criteria. On the other hand, some reviews are sentimentally ambiguous for a variety of reasons, such as in the movie industry, where some reviews provide only negative opinions of actors and movie plots. These reviews do not carry useful information in terms of text mining.

1.2. Learning with Imbalanced Text Data

The objective of the proposed project is thus to solve the problem of imbalanced text classification. In addition to solving the problem of skewed distribution, the creation of informative word vectors for model learning is also addressed for imbalanced text classification, where texts are transformed into word vectors to allow the model to process reviews. In addition, the writing patterns in reviews can vary, and some of them are informal, which can lower the efficiency of model learning [11]. To create informative word vectors, text classifications are divided into three levels: the document level, the sentence level, and the aspect level [12]. The document level tends to the entire document or review as positive or negative; the sentence level aims at defining the polarity of each sentence; and the aspect level considers different aspects of documents based on the document level and sentence level calculations. Specifically, the main purpose of constructing informative word vectors through these levels is to enhance the representativeness of reviews. Here, we propose a novel classification method for learning tasks with imbalanced test data. It aims to construct a method for data preprocessing that researchers can apply to their learning tasks with imbalanced text data and save the efforts to search for more dedicated learning tools.

In the proposed learning strategy method, there are three aspects of the methodology that deal with imbalanced text classification: a datasets aspect, a classification model aspect, and a topic aspect. Through the analysis of the three aspects, our proposed method can tackle the problem of each aspect as: (1) Datasets aspect: information loss and overfitting can be dealt with. (2) Classification model aspect: the definition of the misclassified cost of support vector machine (SVM) can be determined to alleviate the misclassification for both the minority and majority classes. (3) Topic aspect: the relationship between words and document can be enhanced by our proposed method.

For the datasets aspect, previous studies have often used fuzzy-based techniques [13] as well as sampling techniques, including oversampling and undersampling, to solve this

problem. The core of these methods is enhancing the representativeness of the minority class. Synthetic minority oversampling Technique (SMOTE) [14] is one of the most popular oversampling methods. This technique uses a K-neighbors classification to define minority class samples as seed samples to produce more minority class samples, such that the dataset imbalance ratio can be reduced. In addition, there are also several improved SMOTE techniques, such as borderline-SMOTE [15]. Although these techniques, when used for data preprocessing, can outperform common oversampling methods, the low efficiency of SMOTE may result in a necessity for too much computational time when the datasets are extremely large because SMOTE may create too many virtual samples [16]. The other oversampling technique is an imitation and inversion strategy. This method calculates the sentiment point of each word and creates more minority class samples by transforming majority words into minority words. However, the definition of sentiment points is ambiguous, and the efficiency of this method is low.

The undersampling technique is an attempt to reduce the number of majority words to make the datasets balanced. Easy ensemble [17] is an informed undersampling system that divides datasets into several balanced subsets by resampling them and uniting every classifier into a single classified system.

In terms of oversampling and undersampling, oversampling creates exact replicates of the minority class samples, which may cause the classifier to overfit the minority class samples. On the other hand, undersampling may eliminate samples from the majority class and cause information loss [18]. In our study, the proposed method can filter informal instances to balance the distribution of the dataset.

For the classification model aspect, the SVM [19] is one of the most popular machine learning and classification models. This model is dedicated to constructing a hyperplane with the widest margin determined by the Lagrange multiplier to classify data in a high-dimensional feature space. In addition, it uses support vectors as the training data because these data can provide relevant information by which to build margins that distinguish data into different classes. It is also noteworthy that SVMs can deal with imbalanced data better than other models, such as naive Bayesian classifiers, because the number of support vectors in different classes are constrained by the Karush–Kuhn–Tucker (KKT) characteristic, which evolved from the Lagrange multiplier [20]. In other words, this condition can enhance the ability to represent minority classes. Furthermore, SVMs use kernel functions to simplify high-dimensional space calculations and increase the efficiency of models. In recent years, numerous creative extensions of SVMs have been developed to deal with several problems related to making precise predictions [9,21–24].

Regarding the applications of SVM in imbalanced datasets, a comparative study [25,26] provided supportive experimental results to show that SVM performs well in text classification through the learned decision surface. As language processing tools rapidly develop, researchers can easily leverage the dedicated text learning models and integrate them into the SVM framework, or tie in use it. This is expected to model text data from linguistics viewpoints. Note that the value of the misclassified costs of SVM is also difficult to determine. If the misclassified cost of the minority class is set too large, it may misclassify majority class samples. Therefore, our proposed method will also focus on the determination of misclassified costs to tackle the misclassified rate of majority class samples. For the topic aspect, topic modelling is crucial in imbalanced text classification. Some topic-independent classifier training features are independent of the topics themselves. The results of this kind of classifiers often perform poorly because they overlook the logic of sentences, and thus the model cannot decipher abstract reviews. Therefore, topics are considered in this study as part of the training material when it comes to imbalanced sentiment classification.

In this study, a two-phase model technique is developed as an embedded method based on the concept of the SVM and is aimed toward tackling imbalanced text classification problems. Precisely, this technique uses both LDA and word2vec to vectorize words into useful features. As for the tuning parameters, a GA is used to search for the

best combination of parameters in order to output balanced datasets and enhance the performance of the SVM. Consequently, the main contribution of this project is creating reliable word features and normal dataset distributions. Thus, the data can fit the model properly and in turn improve prediction results.

In the experimental stage, we evaluated the proposed method with several well-known criteria and compared it with state-of-the-art models and ensemble learning in the machine learning field. The experimental results showed that the proposed method performed well as compared to the other methods. In addition, we discovered that the proposed method could predict minority class samples correctly without sacrificing the predictive power of majority class samples.

The remainder of this paper is organized as follows: Section 2 describes the sample preprocessing concept, which is the process in which word vectors and topic features and various SVM techniques are produced related to imbalanced training. Section 3 explains the detailed steps and theory for the proposed method. In Section 4, we use imbalanced text data to validate the proposed method. Finally, the conclusions are presented in Section 5.

2. Related Studies

2.1. Reviewing Imbalanced Dataset Issues

In the field of machine learning, most models consider dataset distributions to be balanced. However, many of the ratios between the majority class and minority class are enormous. In fact, the distributions of datasets are often skewed in the real world; thus, the features of minority class samples are not representative. Therefore, their classified effectiveness is severely jeopardized by the choice of model [27]. Some linear classifiers consider the parameters between the majority class and minority samples to be independent. This phenomenon results in the restriction of the generalization of classifiers when datasets are large [28].

There are varying definitions of imbalanced data. Previous studies often define datasets with ratios of 1:100, 1:1000, and 1:10000 as imbalanced [27]. However, these imbalance ratios seem exaggerated. Recently, Krawczyk et al. [29] used a 1:10 imbalanced dataset as a training dataset in order to put their experiment into practice and obtained good results. Therefore, in the datasets chosen for the proposed method, the imbalance ratio was allowed to be less than 1:10.

Generally speaking, there are other challenges that come with the general imbalanced data issue, such as feature space heterogeneity, class overlapping, and small disjuncts [30]:

(1) Feature space heterogeneity:

While the number of minority classes is much smaller than the majority class counterpart, the features of minority classes are also very limited, and thus are not comparable to the features of majority classes. Most models, such as the naïve Bayes classifier, aim to learn entire class features. This biased learning pattern may inevitably hamper the classification performance due to the overlooking of minority class features.

(2) Class overlapping:

Class overlapping refers to a situation where some samples in different classes may have identical or similar characteristics. In addition, these samples may overlap in instance space and make classification ambiguous. As shown in Figure 1, similar samples from two classes are difficult to classify in the instance space because they are close to each other. However, imbalanced data will make class overlapping much worse. This issue shows that limited minority samples are easily dispersed in different instance spaces.

(3) Small disjuncts:

When the minority class has multiple sub-concepts, this will increase the difficulty of learning minority class characteristics. As shown in Figure 2, each group contains different sub-concepts; thus, it is difficult for the classifier to distinguish between them. Even worse, there are relatively few of some of the sub-concepts. Therefore, the imbalanced data issue will make these sub-concepts much more difficult for models to learn.

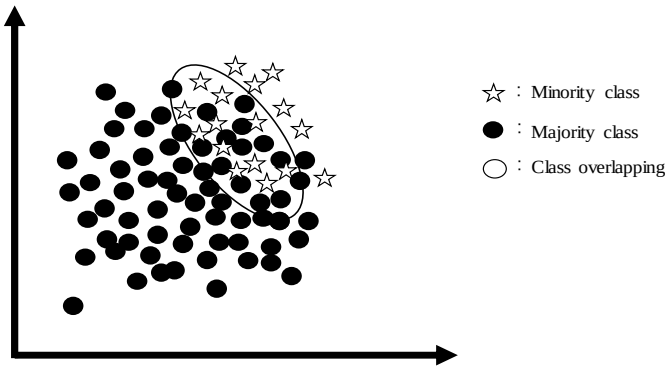


Figure 1. Class overlapping.

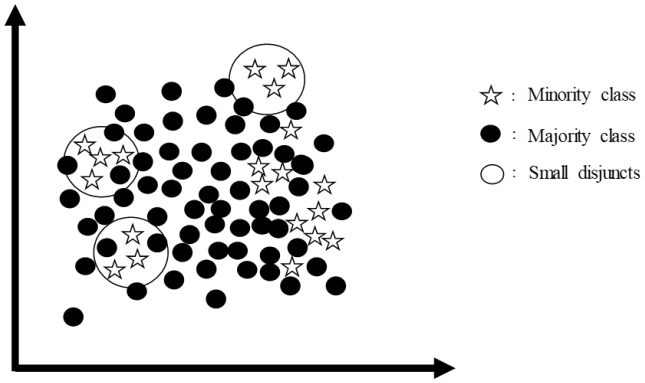


Figure 2. Small disjuncts.

The imbalanced data issue also occurs in text datasets. In addition, Li et al. [30] pointed out that the distributions of review datasets are often skewed because the reviewers give similar opinions of products. In addition, many reviewers only give reviews when they are interested in products. In other words, this will lead entire reviews of objects to become biased either positively or negatively. For example, if a restaurant is fantastic, there will be lots of positive reviews. However, there will be fewer negative reviews, and supervisors cannot make good decisions without these reviews. Therefore, the imbalance ratio of text datasets is often high.

2.2. Oversampling and Undersampling Techniques

Previous studies have often used oversampling and undersampling in order to tackle the imbalanced data issue. Oversampling aims to synthetically create more minority class samples. On the other hand, the goal of undersampling is to decrease the number of majority class samples. Both of their goals are to lower the dataset imbalance ratio. The pros and cons of their extended methods will be described in the following paragraphs.

Random oversampling is one of the traditional oversampling techniques that produces minority samples randomly. However, this method may cause overfitting, which means the accuracy will be extremely low in the testing stage and high in the training stage [27]. As for imbalanced sentiment classification, a traditional oversampling technique is not helpful for improving model performance because reviews can create thousands of word vectors, and these vectors often disperse in high-dimensional space. In addition, this method may worsen the problems of small disjuncts and data sparseness. Chawla et al. [31] proposed the synthetic minority oversampling technique (SMOTE), which uses a K-NN classification to find the relationship between minority class samples and surrounding samples to create virtual samples instead of creating synthetic samples randomly. As shown in Figure 3, first, one of the minority class samples is selected as the seed sample. Second, the K neighbors' samples of the seed samples are located, and the Euclidean distance between the seed

samples and neighbor samples is calculated. Finally, Equation (1) is used to create virtual samples and define their location in the instance space as follows:

$$m' = m + gap * dis, \quad gap \in [0,1] \tag{1}$$

where m' is the virtual samples and m is the seed samples; gap is a random value ranging from 0 to 1; and dis is the Euclidean distance. Although the performance of SMOTE is better than that of traditional oversampling methods, SMOTE will undermine the classification efficiency when datasets are large [16]. In addition, this method may worsen class overlapping and may even create more noisy samples when the seed samples and neighbor samples are too close.

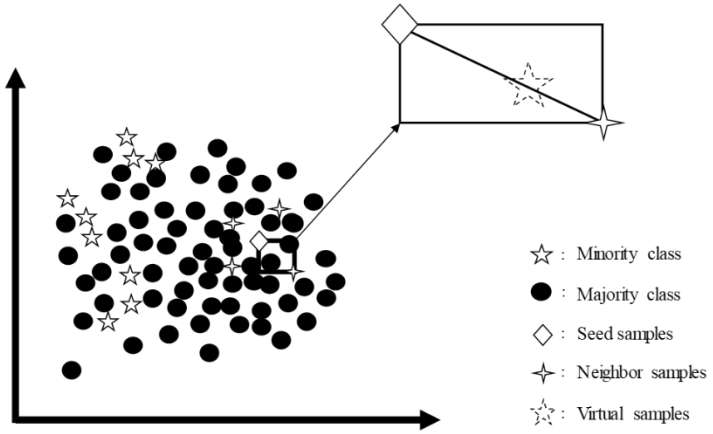


Figure 3. SMOTE procedure.

Han et al. [15] proposed two different SMOTE extensions: borderline-SMOTE (B1-SMOTE) and borderline-SMOTE2 (B2-SMOTE). B1-SMOTE uses the K-NN classification to find neighbor samples, which is same as the original SMOTE; however, B1-SMOTE only chooses border samples as seed samples. Border samples are defined as those samples where it is assumed that the neighbors of the samples are N, and the majority class samples inside the neighbors are n. If n is in $N/2 \leq n < N$, then it is defined as a border sample. The benefits of B1-SMOTE include avoiding creating noisy samples and alleviating extended processing time.

In terms of imbalanced sentiment classification, traditional oversampling techniques often create synthetic samples without considering the text sentiments. Li et al. [30] proposed inversion and imitation methods in order to replace the traditional oversampling techniques. They mentioned that texts in different domains may not reflect the original meaning of the text. For example, "big size" is positive in terms of computer memory, but in the mobile phone market "big size" may become negative in terms of the size of a smartphone.

Initially, inversion and imitation methods calculate the sentiment value, which ranges from 0 to1 for each text in the dataset. The inversion stage is aimed toward finding the majority class samples and exchanging the sentiment text with opposite words. For example, "beautiful" has 0.6 points in a sentence exchange with the text "ugly," for which the sentiment value is 0.4. This can reduce the quantity of the majority class. On the other hand, the goal of the imitation stage is to boost sentiment texts with better text in the minority class. For instance, "sad" is a 0.3 of a point exchange with the word "pathetic," for which the sentiment value is 0.32. This can improve the representativeness of the minority class. Therefore, this method can improve the relationship between text features and sentiment and improve model performance. Nevertheless, exchanging words may cause incorrect grammar and lead to erroneous information.

Undersampling is one of the most popular techniques used to alleviate the imbalanced data issue. In addition, this method has been shown to improve the sensitivity of classifiers [21].

Xu et al. [31] pointed out that reducing the number of majority class samples can undermine the classification results because these samples may contain important information.

Liu et al. [17] proposed informed undersampling in the form of the unsupervised learning techniques easy ensemble and balance cascade, which are aimed toward alleviating the problem of information loss. The concept of easy ensemble is similar to that of a random forest, where a problem is divided into subproblems and the dataset is separated into several balanced datasets by the random sampling of majority class samples and their individual combination with minority class samples. Additionally, these balanced datasets will be trained by several different classifiers, which are given a higher weight if the performance of the classifiers is good. In the end, these classifiers can be considered as an ensembled classified system that uses classifiers as features and will not reduce the number of majority class samples.

Balance cascade, similar to a cascade classifier, arranges individual classifiers in order from weak to strong, where, initially, the same number of minority classes are sampled from the majority class samples after which the number of samples is reduced and classified as majority class. These samples are noisy samples. Therefore, the results of balance cascade can provide precise minority classifications.

2.3. Word Vectorization and Feature Selection

Word vectorization, which is also called word embedding, can transform words into a numeric value to allow the model to interpret and learn the dataset pattern. Before word vectorization was available, previous studies often utilized one-hot encoding to transform words into a numeric value; however, this method cannot represent the relationships between words or the sentiment and semantics of words [32].

One popular word embedding technique, word2vec, is an artificial neural network. Wang et al. [33] considered word2vec to be composed of three stages: an input layer, a projection layer, and an output layer. There are two types of word2vec systems—continuous bag of words (CBOW) and skip-gram—assuming the text dataset is $W = \{w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}\}$:

1. CBOW:

CBOW selects a word as the target word and then executes a weighted average on the words around the target word in order to calculate the vector of the target word. Shown in Figure 4, CBOW will encode words into 0 and 1 in the input layer. In the projection layer, skip-window is used to fetch the encoded value to calculate the weighted average. Finally, a vector is assigned to the target word in the output layer.

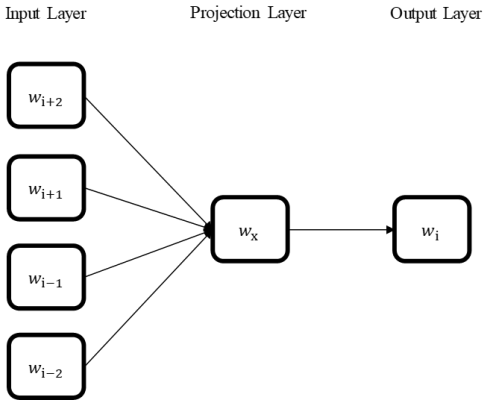


Figure 4. The CBOW procedure.

2. Skip-gram:

In skip-gram, a word is selected as the target word, and then the vectors of the surrounding words are calculated. As shown in Figure 5, skip-gram will encode words into 0 and 1 in the input layer. In the projection layer, the size of the skip-window will be

determined by how many surrounding words should be given vectors. Finally, a vector is assigned to the surrounding words in the output layer.

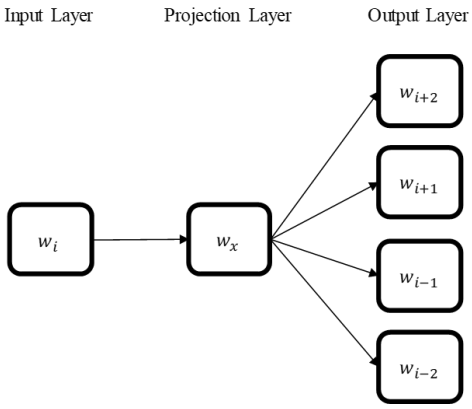


Figure 5. The skip-gram procedure.

Practically, imbalanced data problems are frequently caused by high-dimensional data [9], where latent Dirichlet allocation (LDA) is used as an unsupervised learning method that can utilize Gibb's sampling to extract topics as important features of reviews. Furthermore, it can be used to express the relationships among reviews based on topics and words. The LDA concept is shown in Figure 6:

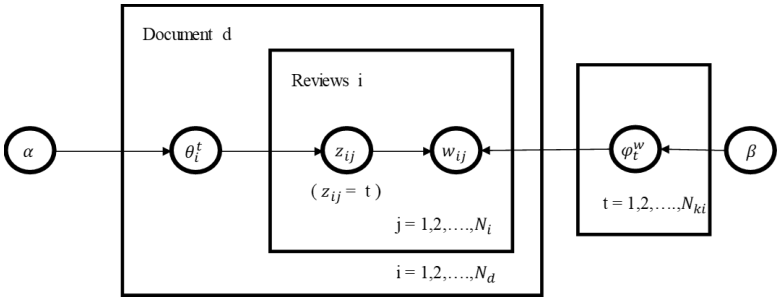


Figure 6. The LDA concept.

where N_i is the amount of letters used in an individual review; N_d is the number of reviews in an individual document; N_{ki} is the number of topics in an individual review; and θ_i^t is the probability of topic t occurring in review i. Topic z_{ij} is usually represented by a word w_{ij}, and φ_t^w is the probability of a word occurring in topic t. Both θ_i^t and φ_t^w have prior probabilities, α and β, respectively. The relationship among these parameters can be shown as:

$$\theta_i^t \sim Dirichlet(\alpha) z_{ij} \sim Multinomial\left(\theta_i^t\right) \quad (2)$$

$$\varphi_t^w \sim Dirichlet(\beta)\, w_{ij} \sim Multinomial(\varphi_t^w) \quad (3)$$

LDA uses a Dirichlet distribution as the prior probabilities because the conjugate distribution of a multinomial distribution can be easily derived using a Dirichlet distribution. With the LDA concept, a project can output a probability sparse matrix that contains the topics from every review.

In a sentiment analysis, LDA is often used for the feature selection and ensemble learning because it enhances the representativeness of a topic. Based on the LDA concept, the biterm topic model focuses on short text modeling by improving the relationship between words [34]. Guo et al. [35] improved customer satisfaction by using LDA in online reviews so that the services customers complained about could be amended. Based on these studies, LDA has proven useful in boosting classification performance.

2.4. Dealing Imbalanced Datasets with Support Vector Machine

Cortes et al. [19] proposed the support vector machine (SVM), which is a classification technique aimed toward classifying samples by finding a hyperplane. In addition, maximizing the hyperplane margin can optimize classification performance. A hyperplane can be shown as shown in Equation (4), where W is a set of weights; X is a set of samples; and b is an offset value. We assume that there are S_A, S_B classes and select one of the samples to fit Equation (4). If $g(x) > 0$, then the samples are designated as S_A classes, whereas $g(x) < 0$ samples are classified as S_B. Additionally, the distance of the samples from the hyperplane can be shown as Equation (5), where y_i is a classification value and N is a quantity of class samples.

$$g(X) = WX + b \tag{4}$$

$$d_i(X) = y_i(WX + b)\ ,\ i = 1, \ldots N \tag{5}$$

Support vectors are a vital part of learning samples for SVMs; however, support vectors are challenging to classify and obtaining quality information is difficult. Thus, SVMs still only consider support vectors when learning the dataset pattern. In addition, the location of support vectors is on the margin, whereas $d_i = 1$ in Equation (5). This condition derives Equation (6), which is the margin formula. In order to maximize the margin of the hyperplane, Equation (7) must minimize $||W||$.

$$\text{Margin} = \frac{1}{||W||} \tag{6}$$

$$\text{Min}\frac{1}{2}||W||^2 \tag{7}$$

The Lagrange multiplier is the core SVM method used to search for the maximized hyperplane margin. This mathematical technique can find extremum under specific restrictions. Therefore, SVMs apply this method to search for the minimum $||W||$ by providing simple restrictions in high-dimensional calculations. Additionally, we can derive Equations (5)–(8), where y_i is a classification value; X_i is the ith set of the training dataset; X is the test dataset; and λ_i is the multiplier.

$$d_i(X) = \sum_{i=1}^{N} y_i \lambda_i X_i X\ ,\ i = 1, \ldots N \tag{8}$$

SVMs perform better than other classifiers when it comes to non-linear problems and high-dimensional issues. This chaos is often observed in the real world. As shown in Figure 7, SVMs can project samples to high-dimensional space in order to use a hyperplane to separate them. However, separating samples in a high-dimensional space will lead to numerous dot support vector calculations. Thus, kernel functions can help SVMs calculate in primal space. This strategy can reduce execution time dramatically and derive Equations (8) and (9), where $K(X_i, X)$ is the kernel function.

Akbani et al. [20] suggested that there are many more majority class support vectors than minority class support vectors in imbalanced data situations. However, Karush–Kunh–Tucker (KKT), which is one of an SVM's characteristics, can balance support vectors from both classes by constructing conditions. Therefore, the weights of support vectors from the minority class are larger than those for their majority class counterparts. This phenomenon can let the hyperplane move toward the minority class, which can result in a decreased probability of misclassifying the minority class.

There are many variations and ensemble learning methods in SVMs in the machine learning field. Several ensemble methods are based on SVMs. Ertekin et al. [21] combined active learning with a SVM to make iterative learning. This learning technique will classify a dataset in order to find high-information class samples that will become training sets. The benefit of iterative learning is the ability to control the number of effective training samples.

In an imbalanced data scenario, this can alleviate misclassified minority classes and boost execution time. In addition, early stopping can also be applied to optimize the number of training samples. Early stopping is applied to avoid a drop in performance after a specific number of training samples is obtained. Wu et al. [23] combined a random forest with an SVM. Random forests have been confirmed to generate good classification performance in datasets in different fields. The authors replaced the nodes from a random forest with an SVM model so the dataset could be transformed into a balanced status. They constructed SVM in every node in the random forest. The performance using this technique was better than the random forest and the SVM alone.

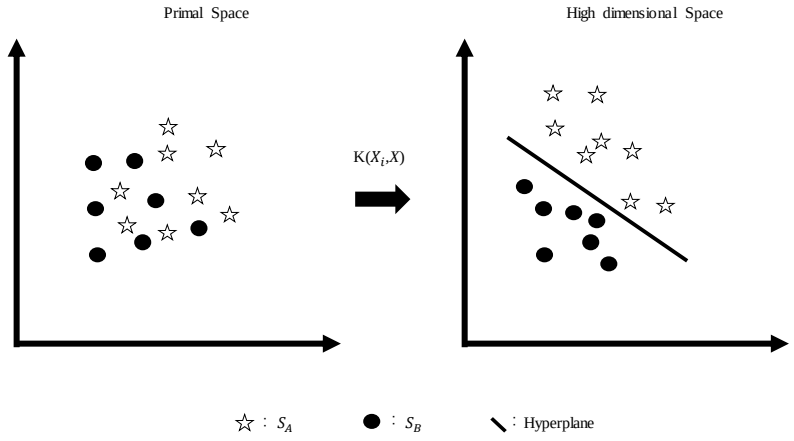


Figure 7. Tackling non-linear problems with SVM.

The choices of misclassified costs C and Γ in a kernel function will affect the performance and efficiency of models. Thus, previous studies have often used different algorithms to find the best combination of parameters. Genetic algorithms have been shown to produce stable results. Many studies have also applied GAs as an optimization technique. Chunhong et al. [36] used real coded genetic algorithms that utilized real codes instead of binary codes in the chromosome stage to find combinations of parameters for the SVM. Huang et al. [37] created a fitness function with an optimizing feature selection function and SVM parameters. This improved classification performance because the coordinated features and parameters were adjusted properly.

3. Proposed Methodology

The configuration of the proposed two-phased classification encompasses following processes: data preprocessing; outputting the topic in each review; vectorizing the reviews and topics to produce features; producing a robust, balanced dataset; and testing model effectiveness by tuning parameters, as explained in the following subsections. The major objective of this paper is to improve model effectiveness by developing a data preprocess that can be used to create a data scenario by which the SVM can classify a dataset that is balanced and full of reliable features.

The preprocessing and vectorizing method is shown in Algorithm 1. Text datasets usually contain many redundant words that are detrimental to the classification results. Therefore, the aim of the text data preprocessing procedure here is to remove such words and improve the quality of the datasets. The text data preprocessing procedure is outlined in the steps below.

Tokenization: tokenization is used to divide every sentence into individual words in order for the model to understand the meanings of the text, since words can be treated as features during the classification stage. For example, the sentence "I love you." can be divided into "I," "love," "you," and "." and then can be recognized as word features.

Step 1. Normalization: In some cases, text datasets contain multiple languages, which may cause errors during classification because languages have different logic or speaking

rules. Therefore, many projects typically use English as the dataset language. Most studies suggest that capital letters in reviews should be converted into lower case letters; however, Li et al. [12] suggested that reviewers use capital letters to enhance word sentiments. In this project, capital letters are retained.

Step 2. Stop words and punctuation filtering: There are many stop word genres, such as conjunctions and particles, which will not provide sentiment to the model but will further increase the model processing time. Therefore, part-of-speech tagging is used to find and delete stop words in this project.

Step 3. Lemmatization: Words usually have different parts of speech that often contain the same meanings. Therefore, lemmatization is used in this project to transform words into the same part of speech. For instance, "civilization" and "civilized" are transformed into "civilize."

Algorithm 1 Preprocessing and Vectorizing Method

Input: Experimental dataset = {review (r), class (c)}
Output: Vectorized dataset = {vectors (v), class (c)}
Step 1: For each r do
Step 2: Tokenize, normalize, filtering stop words and punctuation, and lemmatize r
Step 3: Generate topics by LDA model
Step 4: For each word (w) in r do
Step 5: Vectorize w by word2vec model
Step 6: Calculate weight of w from $\omega_{n_{ik}} = \frac{p_{n_{ik}}}{\sum_{r=1}^{5} p_{n_{ir}}}$
Step 7: end for
Step 8: For each topic (t) do
Step 9: Construct topic vectors from $v(t_{n_i}) = \sum_{r=1}^{5} \omega_{n_{ir}} v(w'_{n_{ir}})$
Step 10: end for
Step 11: Construct review vectors from $v(d_n) = \frac{\sum_{k=1}^{K} v(w_{n_k})}{K}$
Step 12: Calculating Euclidean $(v(d_n), v(t_{n_i}))$= $|v(d_n) - v(t_{n_i})|$ as new vectors (v)
Step 13: end for

Vectorizing word features is vitally important in this project because the model is designed to only recognize vectorized data. Therefore, the LDA and word2vec technique are combined in this work to produce reliable features since word2vec can focus on the relationships among words in sentences [33]. Word2vec is an artificial neural network that can transform words into vectors. Using this technique, words are converted through word2vec to create review vectors and topic vectors. The process of vectorizing reviews and topics is shown in Figure 8.

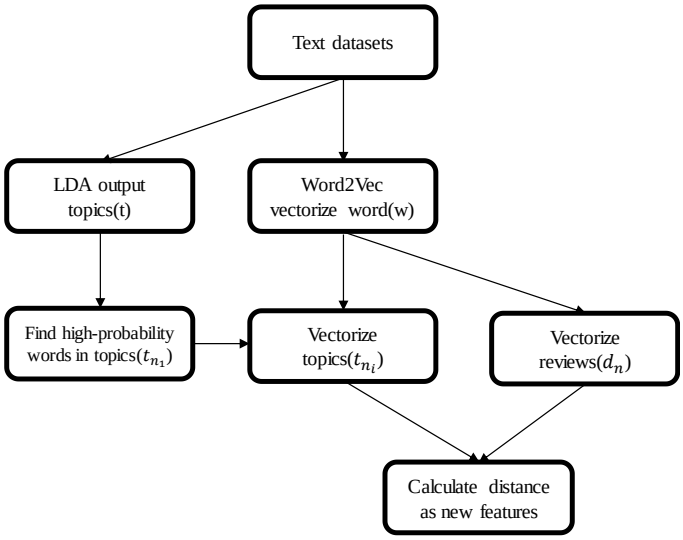


Figure 8. The process used to vectorize reviews and topics.

To explain this process, it is assumed that document D has N reviews as $D = \{d_1, d_2, d_3, \ldots d_n\}$, $n = 1..N$. Every review d_n has K words, shown as $d_n = \{w_{n_1}, w_{n_2}, w_{n_3}, \ldots w_{n_k}\}$, $k = 1..K$, and every review d_n contains I probability of topics, shown as $d_n = \{t_{n_1}, t_{n_2}, t_{n_3}, \ldots t_{n_i}\}$, $i = 1..I$, where every topic t_i has the probability of having words, shown as $t_{n_i} = \left\{p_{n_{i_1}}, p_{n_{i_2}}, p_{n_{i_3}}, \ldots p_{n_{i_k}}\right\}$.

Initially, topics are generated using LDA and the words are converted into vectors $\{v(w_{n_1}), v(w_{n_2}), v(w_{n_3}), \ldots v(w_{n_k})\}$ using word2vec. Secondly, the goal at this stage is to generate vectorized topics. Therefore, five high probability words w for topics t_i are selected in order to calculate the individual weight of the selected words using the following formula:

$$\omega_{n_{i_k}} = \frac{p_{n_{i_k}}}{\sum_{r=1}^{5} p_{n_{i_r}}} \tag{9}$$

where the probability of the selected word is p_{i_r} and the specific probability of a word is p_{i_k}. Both parameters are used to calculate the weights of specific words ω_{i_k}. The method used to calculate the topic vector involves calculating the weighted average of selected word vectors using the following formula:

$$v(t_{n_i}) = \sum_{r=1}^{5} \omega_{n_{i_r}} v\left(w'_{n_{i_r}}\right) \tag{10}$$

For the review vectors, the weighted average of all word vectors is calculated because it expresses the meaning of the reviews. The formula is shown below:

$$v(d_n) = \frac{\sum_{k=1}^{K} v\left(w_{n_k}\right)}{K} \tag{11}$$

After going through the above processes, three different kinds of vectors are obtained: word vectors, topic vectors, and review vectors. All of these vectors are set in the same vector space to allow different ways of expressing the relationships between words. With respect to the problem of a sparse feature matrix, the Euclidean distance between the topics and reviews is calculated. If the distance is closer, this means that the topic is related to the review to a certain degree. The formula is shown as:

$$\text{Euclidean}(v(d_n), v(t_{n_i})) = |v(d_n) - v(t_{n_i})| \tag{12}$$

Following this procedure, the relationships between the reviews and topics and the relationships between the words and sentences become dense. In addition, the problem of a sparse matrix will no longer exist.

The core processes of the two-phase classification method comprise two main stages. The first involves outputting a balanced dataset, where the majority and minority samples are almost in the same ratio; thus, a cost-sensitive support vector machine (CS-SVM) can be used to implement model training without imbalance issues. The second includes the testing of model effectiveness, which is aimed toward improving the CS-SVM model by tuning the parameters using a GA method, is shown in Figure 9.

The strategy in the first stage, called phase one, is to use a CS-SVM to generate a low imbalance ratio dataset to avoid an imbalance problem. The initialization process is shown in Algorithm 2. The CS-SVM is a variant of the SVM that has an added slack variable ξ_i and individual costs for the majority and minority samples, where ξ_i refers to the distance between a misclassified sample and the hyperplane margin. CS-SVM is formulated as:

$$y_i(WX + b) \geq 1 - \xi_i, \; i = 1, \ldots N, \; \xi_i \geq 0 \tag{13}$$

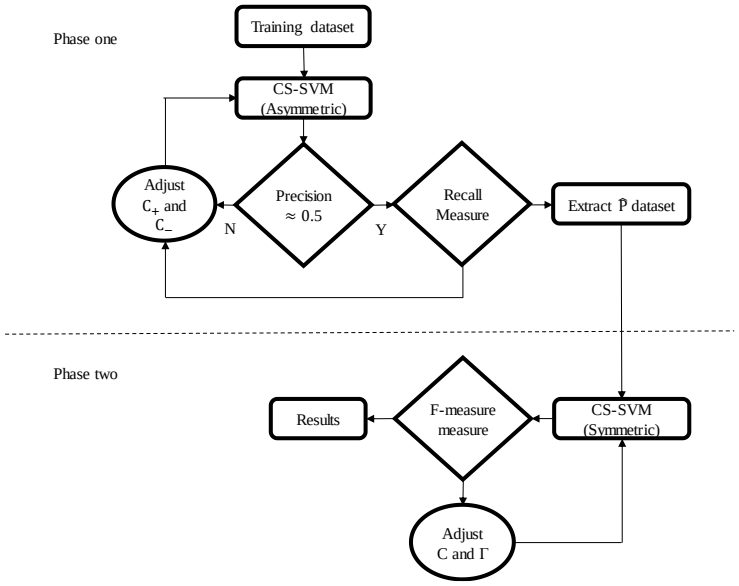


Figure 9. The two-phase model procedure.

Algorithm 2 Two-Phased Classification—Phase One

Input:	Vectorized dataset, chromosome size (*D_size*), population size (*P_size*), range of C^+ in decimal (C^+_r), range of C^- in decimal (C^-_r), require amount of parents (μ), probability of crossover (ν), probability of mutation (φ) and tournament rounds (ω)
Output:	Balance dataset
Step 1:	For each generation do
Step 2:	If generation = 1 then
Step 3:	Construct chromosomes (*D_size*, *P_size*) for C^+ in C^+_r.
Step 4:	Construct chromosomes (*D_size*, *P_size*) for C^- in C^-_r.
Step 5:	end if
Step 6:	For each chromosome do
Step 7:	Calculate fitness of chromosome (C^+, C^-).
Step 8:	end for
Step 9:	Filter fitness in the range of [0.45,0.55].
Step 10:	Implement tournament selection to find μ parents (*p*) with ω rounds
Step 11:	For each *p* do
Step 12:	Crossover with ν probability and produce 2 children in each stage
Step 13:	Mutate children with φ probability
Step 14:	end for
Step 15:	Gather children for next generation
Step 16:	Find the best set of C^+ and C^- to implement LinearSVC algorithm
Step 17:	Collect data classified as minority samples which distribution is balanced

The CS-SVM has a soft margin, which means that it can tolerate misclassified samples to maximize the hyperplane margin. In addition, asymmetric cost is used here for both classes when the CS- SVM is focused on decreasing the misclassified minority rate. The objective function of the CS-SVM is formulated as follows:

$$\text{Min}\frac{1}{2}||W||^2 + C_+ \sum_{i:y=+1} \xi_i + C_- \sum_{j:y=-1} \xi_j \quad (14)$$

where W refers to the weight matrix of the individual sample; C^+ is the cost of the misclassified minority samples; and C^- is the cost of the misclassified majority samples. Most previous researchers have preferred to increase C^+ to solve the imbalance problem. However, the classified minority samples may contain a great deal of misclassified majority data. Therefore, the focus in this project is on the classified minority class intended for use

to form the entire balanced dataset. Additionally, this approach attempts to match two conditions: a value for model precision of approximately 0.5 where the set range is between [0.40 and 0.60]. This means that the ratio between the misclassified majority samples and the correctly classified minority samples is close to 1. In other words, the minority class is seen as having a low imbalance ratio. The model precision is formulated as follows:

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

where true positive (TP) refers to correctly classified minority samples and false positive (FP) comprises the misclassified majority samples. The other condition considers that recall metrics can be the portions of correctly classified minority samples. If the number of correctly classified minority samples is higher under the first condition, this means that to a certain degree, the minority information is not lost. The model recall is formulated as below:

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

where true positive (TP) refers to correctly classified minority samples, and false negative (FN) comprises misclassified minority samples. The concept for this stage is shown in Figure 10.

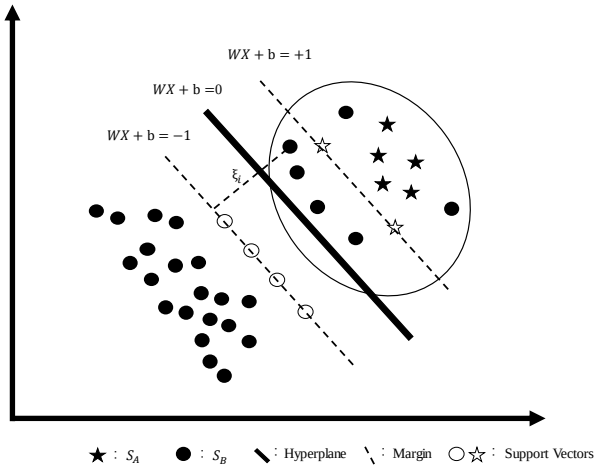


Figure 10. The concept for producing a balanced dataset.

The aim of the second stage, phase two, was to let the CS-SVM with symmetric cost C learn data without an imbalance issue. The initialization is shown in Algorithm 3. In addition, this scenario follows previous studies which have claimed that most models are assumed to learn data with a normal distribution. The concept for the classification of the CS-SVM is shown in Figure 11.

While the model is learning information from the dataset, the parameters at both core stages will be adjusted to obtain better results. In addition, there are different parameters between the stages. Firstly, the stage during which the balanced dataset is produced required the adjustment of C^+, C^-, and Γ. Secondly, the stage at which model effectiveness is tested required the adjustment C and Γ.

Γ is a part of a kernel function called a radial basis function (RBF). This kernel function has been widely implemented in several CS-SVM studies. This technique makes every calculation remain in primal space to reduce execution time and resthisces. The RBF is formulated as below:

$$\text{RBF} = \exp\left(-\Gamma\left|\left|X - X_j\right|\right|\right) \tag{17}$$

$$\Gamma = \frac{1}{2\sigma^2} \tag{18}$$

where Γ is seen as a distribution of samples in a high-dimensional space. Further, the RBF will control the number of support vectors and the distribution of samples in a high-dimensional space. These traits are highly related to the construction of a hyperplane in a high-dimensional space.

Algorithm 3 Two-Phased Classification-Phase Two

Input:	Balance dataset, chromosome size (*D_size*), population size (*P_size*), range of Γ in decimal (Γ*_r*), range of cost matrix in decimal (*C_r*), require amount of parents (μ), probability of crossover (ν), probability of mutation (φ) and tournament rounds (ω).
Output:	Results of Model
Step 1:	For each generation do
Step 2:	If generation = 1 then
Step 3:	Construct chromosomes (*D_size*, *P_size*) for Γ in Γ*_r*.
Step 4:	Construct chromosomes (*D_size*, *P_size*) for *C* in *C_r*
Step 5:	end if
Step 6:	For each chromosome do
Step 7:	Calculate fitness of chromosome (Γ, *C*).
Step 8:	end for
Step 9:	Implement tournament selection to find μ parents (*p*) with ω rounds.
Step 10:	For each *p* do
Step 11:	Crossover with ν probability and produce 2 children in each stage.
Step 12:	Mutate children with φ probability
Step 13:	end for
Step 14:	Gather children for next generation
Step 15:	Find the best set of Γ and *C* to implement RBF-SVM algorithm.

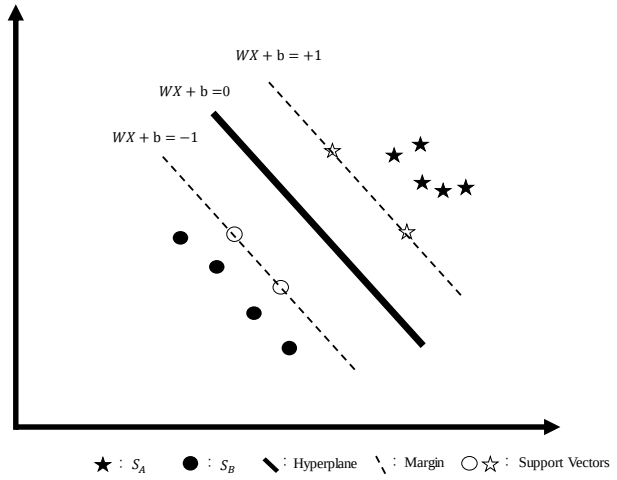


Figure 11. The classification concept for the CS-SVM.

A GA is an advanced method that has been approved as a useful technique to search for a combination of optimal parameters. Additionally, GAs have many variants, such as real coded genetic algorithms [36] and customized fitness functions [37]. This method comprises the following main steps:

Step 1. Chromosome design: Generating N chromosomes composed of random binary numbers, where these chromosomes are considered as representatives of parameters. Then, instead of trying out every parameter, this study uses the representatives of such parameters to boost efficiency.

Step 2. Fitness function: As mentioned previously, the objectives of the two core stages are where phase one requires a precision of around [0.45,0.55] and a high recall, and phase two requires only a high F-measure. These conditions are used to create a fitness function for chromosomes in order to determine the quality of individual chromosomes.

Step 3. Crossover: After evaluating the fitness of every chromosome, we use tournament selection to choose representatives in crossover stages. This selection technique gives every chromosome a specific probability of being chosen as calculated based on its fitness value. This may preserve chromosomes with low levels of fitness into the crossover stage because these chromosomes may contain crucial information. Consequently, the chosen chromosomes can be obtained to join the next generation of the chromosome selection process.

Step 4. Mutation: As for the new chromosomes generated through crossover, this stage is aimed toward breaking through the local optimum among the chromosomes by mutating them. In addition, the rate of mutation is set in order to apply the NOT function to part of the chromosomes. By doing so, overfitting can be avoided, and the information becomes more general.

4. Experimental Study

In this section, we briefly demonstrate the implementation of the proposed method, and then introduce the dataset description, evaluation criteria, and experimental results, which show a comparison between well-known algorithms and the proposed method.

4.1. Experimental Design

An experimental environment was established for the purpose of demonstrating the proposed method, called two-phased classification (TPC). The classification tool used in this experiment was the cost-sensitive support vector machine (CS-SVM), and scikit-learn was used as the modeling tool for the CS-SVM. Additionally, there were two different CS-SVMs used in the toolkit: LibLinear [38] and LibSVM [39]. LibLinear replaces the kernel function with a simple linear formula. This modification can effectively perform as a linear kernel function in a large dataset. Therefore, LibLinear was implemented as LinearSVC in phase one of the TPC. LibSVM is the traditional CS-SVM tool and uses RBF as the kernel function. Thus, this toolkit was used as a symmetric CS-SVM in phase two of the TPC.

The CS-SVM parameter selection was performed using a genetic algorithm, as outlined in Section 3. The DNA size was set at 20; the population size was set at 100; the target chromosome was set at 5; the crossover rate was set at 0.9; the mutation rate was set at 0.5; and the generation was set at 20 in the first phase and 50 in the second phase. In the first phase, the asymmetric cost matrix was set at (0,150) for the minority class and at (0,10) for the majority class. In the second phase, the symmetric cost matrix was set at (0,50), and the gamma was set at (0,0.01).

In the comparison among the robust methods: (1) RBF was used in the chosen SVM as the kernel function (RBF-SVM); (2) the CS-SVM cost matrix was set to have the same imbalance ratio as that of the dataset in order to make correct predictions; (3) the random forest (RF), which is popular in the machine learning field, was used (it is often combined with other methods to tackle imbalance problems); (4) SVM was combined with the SMOTE technique (SMOTE-SVM) to produce virtual samples to balance datasets; (5) similarly, SVM was integrated with the B1-SMOTE technique (B1-SMOTE-SVM) to produce virtual samples only for the border samples; and (6) the easy ensemble classifier (EEC) was used, which is an informed undersampling technique and one of the variants of AdaBoost classifiers.

To conduct the experiments, we adopted the Python integrated development environment in Anaconda, including the libraries pandas, NumPy, scikit-learn, and TensorFlow. All the tools mentioned above are open-sourced and available from the internet. When setting up the experimental environment, the hardware specification we selected included an Intel Core i7 processor, 16 GB of RAM, and a NVIDIA GeForce GTX 1650 graphics chip. Table 1 summarizes the environment components for conducting the experiments, including hardware and software. For the hardware adopted, all the experiments were conducted on computers with identical specifications. The software environmental setup adopted the Python-Anaconda platform for executing all the learning tools.

Table 1. Summary of hardware and software adopted in the experiments.

Item	Experimental Environments
Hardware	Intel Core i7 processor, 16 GB of RAM, and NVIDIA GeForce GTX 1650 graphics chip.
Software	Python in Anaconda, including the libraries pandas, NumPy, scikit-learn, and TensorFlow.

4.2. Dataset Description

Since the reviews and comments of various service platforms are well known to be unevenly distributed amid different kinds of ranking and classes, we considered these kinds of datasets to be qualified for our desired imbalanced text datasets. We formed a very large imbalanced text data pool by collecting customers' feedback from the internet, including reviews and comments of business services and company platforms all over the world. Yelp is a famous brand that operates as the above business, providing a service where users can post their opinions about businesses, services, and products through free-form comments in the text and a numeric rating. That is, Yelp collects imbalanced text data from global customers. In addition, it held text mining competitions with its database, through the well-known Kaggle platform [40]. For these reasons, we adopted Yelp datasets as the experimental subject. We used a Yelp review dataset to address the imbalance problem. The Yelp review dataset was composed of several reviews and ratings that were ranked from one 1-star to 5-stars, as marked on the Yelp website. Additionally, we created three different kinds of imbalanced text datasets by randomly resampling the Yelp review dataset. The details of these datasets are shown in Table 2.

Table 2. The datasets.

Dataset	Total Instances	Imbalanced ratio	No. of Instances in Minority	No. of Instances in Majority
Yelp_α	14,927	1:10	1300	13,627
Yelp_β	16,227	1:5	2600	13,627
Yelp_γ	18,169	1:3	4542	13,627

We removed 3-star reviews because these reviews did not represent any sentiment. In addition, 1-star and 2-star reviews were considered to express a negative sentiment, while 4-star and 5-star reviews were considered to express a positive sentiment. Following the above procedure, we randomly sampled instances from the negative reviews that were considered to be minority class and combined those instances with positive reviews, which were considered to be majority class, in order to create three datasets with different imbalance ratios: Yelp_α, Yelp_β, and Yelp_γ.

4.3. Evaluation Criteria

Four criteria were used to evaluate and compare the performance of the proposed method and the other well-known algorithms: accuracy, F-measure, adjusted G-mean, and AUC. These metrics are constructed from the confusion matrix, which can show the relationship between correctly classified and misclassified samples [12]. In the confusion matrix, Table 3, TP (true positive) represents the number of minority classes that are correctly predicted, whereas FN (false negative) is the number of misclassified minority class samples. Similarly, TN (true negative) represents the number of majority class samples that are correctly predicted, and FP (false positive) is the number of misclassified majority class samples.

Table 3. The confusion matrix.

		Hypothesis Output	
		Minority Class	**Majority Class**
True class	Minority class	TP (True positive)	FN (False negative)
	Majority class	FP (False positive)	TN (True negative)

By convention, accuracy is used as the main criteria by which to evaluate the performance of a classification; however, this metric may not evaluate the prediction of minority classes in an imbalanced dataset because accuracy is calculated as the ratio of the number of correctly classified samples to the number of samples of both classes. Therefore, the accuracy goal is to evaluate the general performance of the classifier. The formula for accuracy is shown as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{19}$$

The F-measure is one of the most effective criteria by which to evaluate the classifier performance of an imbalanced dataset [27]. In addition, this metric is a harmonic mean of recall and precision; thus, it can show the ratio of correctly classified minority class samples to misclassified majority class samples caused by the high misclassified cost of minority class samples. This phenomenon can be used to control the number of misclassified majority class samples while improving the classifier recall. The formula for the F-measure is as follows:

$$\text{F} - \text{measure} = \frac{\left(1 + \beta^2\right) * Recall * Precision}{Recall + Precision} \tag{20}$$

where β is the coefficient used to adjust the relative importance between the recall and precision. If β is larger than 1, the importance of recall is larger than that of precision. On the contrary, if β is smaller than 1, the importance of recall is smaller than that of precision. Thus, we decided to set β as 1 in order to make the F-measure a harmonic mean, which treats recall and precision equally.

Batuwita et al. [41] proposed the adjusted G-mean (AGM) metric, which modifies the original G-mean to increase the sensitivity of the specificity metric. The formula for the specificity and original G-mean are shown below in Equations (21) and (22), respectively:

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{21}$$

$$\text{G} - \text{mean} = \sqrt{Recall * Specificity} \tag{22}$$

This modification can enhance the importance of correctly classified majority class samples because these samples may be misclassified easily with increases in the misclassified cost of minority class samples. Therefore, the AGM can show the correctly predicted samples for both classes. The formula for the AGM is shown below:

$$\text{AGM} = \frac{GM + specificity * N_n}{1 + N_n} \tag{23}$$

where N_n is the portion of the majority class. The difference between the F-measure and the AGM is that the former metric focuses on the majority class based on the misclassified rate, which can dramatically influence the value of the F-measure, and the latter metric focuses on the majority class based on the correctly classified rate, which slightly impacts the value of the AGM. Thus, the AGM is aimed toward evaluating the minority class samples more than the majority class counterparts, and the F-measure is aimed toward evaluating the classification performance of both classes.

Airola et al. [42] suggested that the AUC is a popular evaluation metric in the field of machine learning. This metric utilizes the TP and FP rates to construct a curve and plot the

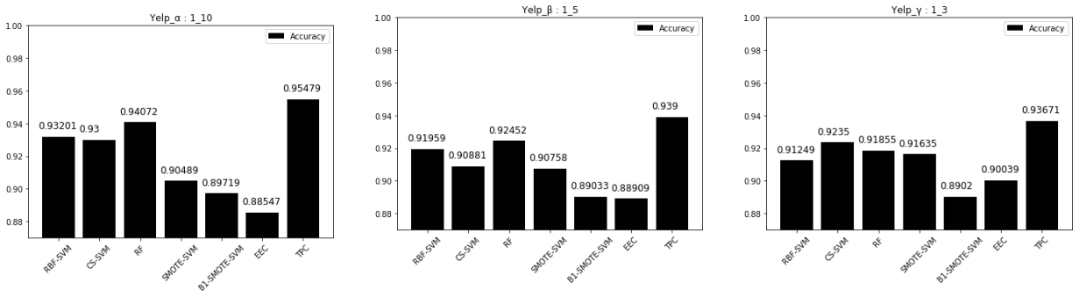
area. The value of the AUC can be interpreted as a probability classifier, which can distinguish randomly chosen minority class samples from randomly chosen majority class samples. The formulae for the TP and FP rates are shown below in Equations (24) and (25), respectively.

$$\text{TP rate} = \frac{TP}{TP + FN} \tag{24}$$

$$\text{FP rate} = \frac{FP}{TP + FP} \tag{25}$$

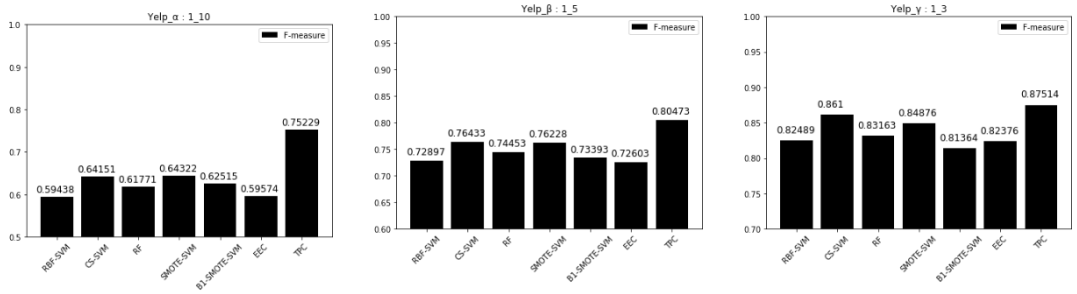
4.4. Experimental Results

The results of experiment are sorted by the evaluation criteria, where we compare TPC with several methods: RBF-SVM, CS-SVM, RF, SMOTE-SVM, B1-SMOTE-SVM, and EEC in Figures 12–15.



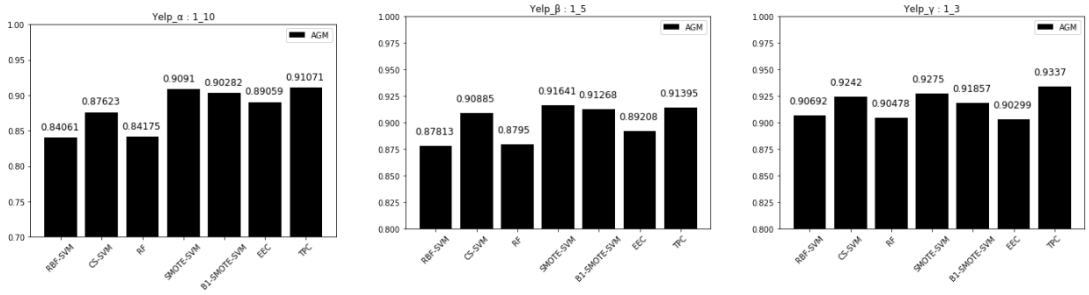
(**a**) (**b**) (**c**)

Figure 12. (**a**) Accuracy of methods in Yelp_α; (**b**) accuracy of methods in Yelp_β; and (**c**) accuracy of methods in Yelp_γ obtained by the experiment.



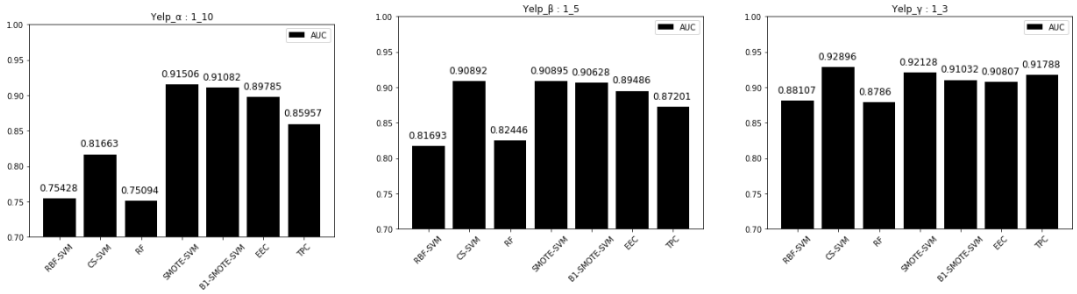
(**a**) (**b**) (**c**)

Figure 13. (**a**) F-measure of methods in Yelp_α; (**b**) F-measure of methods in Yelp_β; and (**c**) F-measure of methods in Yelp_γ obtained in the experiment.



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

Figure 14. (**a**) AGM of methods in Yelp_α; (**b**) AGM of methods in Yelp_β; and (**c**) AGM of methods in Yelp_γ obtained in the experiment.



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

Figure 15. (**a**) AUC of methods in Yelp_α; (**b**) AUC of methods in Yelp_β; and (**c**) AUC of methods in Yelp_γ obtained in the experiment.

4.4.1. Accuracy

Our proposed method, TPC, exhibited the best performance among the three datasets. Most of the method accuracies were higher than 90%. The accuracy of the oversampling techniques, SMOTE-SVM and B1-SMOTE-SVM, were comparatively low because oversampling produces many virtual samples and can enhance the issue of class overlapping and small disjuncts. As for EEC, this undersampling technique causes information loss from the majority class; thus, its accuracy was the lowest.

4.4.2. F-Measure

Our proposed method, TPC, showed a significantly robust performance among the methods when the imbalance ratio was higher. This phenomenon shows that TPC can improve the overall classifier performance instead of only enhancing the rate of the correctly predicted minority class. Furthermore, TPC adopts a genetic algorithm to find the best set of parameters to support the classifier learning in a balanced dataset.

4.4.3. AGM

Our proposed method, TPC, performed the best in the Yelp_α and Yelp_γ datasets. In the Yelp_β dataset, SMOTE-SVM was slightly higher than TPC. In addition, the oversampling techniques, SMOTE-SVM and B1-SMOTE-SVM, had a high AGM because this metric does not consider the misclassified majority class samples and mainly focuses on minority class samples. This is why the oversampling technique performed well in the AGM with the exception of the F-measure and accuracy; however, TPC can perform well among these metrics.

4.4.4. AUC

The proposed method (TPC), SMOTE-SVM, B1-SMOTE-SVM, and EEC performed well in AUC—higher than 85%—and can be considered as having excellent discrimination. Thus, it can be inferred that these discrimination methods can predict minority class samples with high probability. In other words, the TP rate of these methods is higher than the FP rate to a certain degree.

5. Conclusions

In studies of imbalanced sentiment problems, the number of majority class samples is much larger than that of minority class samples. This phenomenon results in an undermining of the performance of classifiers by misclassifying several minority class samples. In this study, our proposed method, a two-phase classification method, could establish a balanced dataset in order to improve the learning of the classifier.

Our proposed method can deal with the imbalanced sentiment problem without information loss and overfitting. In previous studies, sampling techniques and misclassified cost setups were often applied to the methods; however, these approaches may lead to

several side effects, such as information loss and overfitting. This phenomenon may improve the correct prediction of the minority class by sacrificing the correct prediction of the majority class.

To confirm the time cost of our proposed method, including Algorithms 1–3, we referred to related articles and inferred the complexity. For Algorithm 1, the word2vec step is responsible for the largest part of the time cost, and the complexity of running word2vec would be $O(\log(V))$ [43], where V denotes the size of text input. For both Algorithms 2 and 3, the greatest time cost is spent in the SVM step. The complexity of SVM has been well explored and confirmed. Referring to several pieces of research [44,45], we concluded that the complexity of linear SVM would be $O(d)$ and that of RBF-SVM would be $O(d^2)$, where d denotes the dimension of input.

To summarize, this experiment demonstrated that a two-phase classification can predict minority class samples correctly and will not misclassify many majority class samples as a trade-off. In the experiments, this method was proven to optimize both the dataset and the classifier parameters. Further validation of the proposed method using higher imbalanced ratio datasets is worthy of study in the future.

For future studies, a metric to measure how imbalanced the dataset is distributed is necessary, and it is worth exploring its relationship with the accuracy of learning tools. For developing the imbalanced metric or measurement, a survey of more datasets with unevenly distributed class variables will help elucidate the key factors that determine the reasons for the low performance of learning tools, and these factors can be adopted to establish a mathematical formula to model how the dataset is distributed. When future work is carried out, we will make the proposed two-phase method more complete by integrating the metric mentioned above. Another related topic for future work is to expand the scope of our method's application since the study focused only on imbalanced text data. We believe it would be worthwhile to devote more effort to common imbalanced data, and not just text data. That is, the present limitations will be the next research tasks to overcome.

Author Contributions: Conceptualization, D.-C.L. and Y.-S.L.; methodology, S.-C.C.; software, W.-Y.H.; validation, S.-C.C., Y.-S.L. and W.-Y.H.; writing—original draft preparation, S.-C.C. and W.-Y.H.; writing—review and editing, Y.-S.L.; supervision, D.-C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Science and Technology, Taiwan, grant number MOST-110-2221-E-006-194.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here: https://www.kaggle.com/yelp-dataset/yelp-dataset (accessed on 30 June 2021).

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

Abbreviations

AGM	Adjusted G-mean
AUC	Area under the ROC curve
B1-SMOTE	Borderline-SMOTE1
B2-SMOTE	Borderline-SMOTE2
CS-SVM	Cost-sensitive SVM
FN	False negative
FP	False positive
GA	Genetic algorithm
KKT	Karush–Kunh–Tucker conditions
K-NN	K-nearest neighbors' algorithm
LDA	Latent Dirichlet allocation

RBF	Radial basis function
SMOTE	Synthetic minority oversampling Technique
SVM	Support vector machine
TP	True positive
TPC	Two-phased classification

References

1. Assaf, A.G.; Magnini, V. Accounting for customer satisfaction in measuring hotel efficiency: Evidence from the US hotel industry. *Int. J. Hosp. Manag.* **2012**, *31*, 642–647. [CrossRef]
2. Tao, X.; Li, Q.; Guo, W.; Ren, C.; Li, C.; Liu, R.; Zou, J. Self-adaptive cost weights-based support vector machine cost-sensitive ensemble for imbalanced data classification. *Inf. Sci.* **2019**, *487*, 31–56. [CrossRef]
3. Lane, P.C.R.; Clarke, D.; Hender, P. On developing robust models for favourability analysis: Model choice, feature sets and imbalanced data. *Decis. Support Syst.* **2012**, *53*, 712–718. [CrossRef]
4. Batista, G.E.; Prati, R.C.; Monard, M.C. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explor. Newsl.* **2004**, *6*, 20–29. [CrossRef]
5. Elakkiya, R.; Subramaniyaswamy, V.; Vijayakumar, V.; Mahanti, A. Cervical Cancer Diagnostics Healthcare System Using Hybrid Object Detection Adversarial Networks. *IEEE J. Biomed. Health Inform.* **2021**. [CrossRef]
6. Chegini, H.; Beltran, F.; Mahanti, A. Fuzzy Logic Based Pasture Assessment Using Weed and Bare Patch Detection. In Proceedings of the International Conference on Smart and Sustainable Agriculture, Virtual Conference, 21–22 June 2021; Springer: Cham, Switzerland, 2021; pp. 1–18.
7. Elakkiya, R.; Jain, D.K.; Kotecha, K.; Pandya, S.; Reddy, S.S.; Rajalakshmi, E.; Varadarajan, V.; Mahanti, A.; Subramaniyaswamy, V. Hybrid Deep Neural Network for Handling Data Imbalance in Precursor Mi-croRNA. *Front. Public Health* **2021**, *9*, 821410. [CrossRef]
8. Jain, D.K.; Mahanti, A.; Shamsolmoali, P.; Manikandan, R. Deep neural learning techniques with long short-term memory for gesture recognition. *Neural Comput. Appl.* **2020**, *32*, 16073–16089. [CrossRef]
9. Longadge, R.; Dongre, S. Class imbalance problem in data mining review. *arxiv* **2013**, arXiv:1305.1707.
10. Li, S.; Zhou, G.; Wang, Z.; Lee, S.Y.M.; Wang, R. Imbalanced sentiment classification. In Proceedings of the 20th ACM International Conference on Information and Knowledge Management, Glasgow, UK, 24–28 October 2011; ACM: New York, NY, USA, 2010; pp. 2469–2472.
11. Tirunillai, S.; Tellis, G.J. Mining Marketing Meaning from Online Chatter: Strategic Brand Analysis of Big Data Using Latent Dirichlet Allocation. *J. Mark. Res.* **2014**, *51*, 463–479. [CrossRef]
12. Tripathy, A.; Agrawal, A.; Rath, S.K. Classification of Sentimental Reviews Using Machine Learning Techniques. *Procedia Comput. Sci.* **2015**, *57*, 821–829. [CrossRef]
13. Li, D.-C.; Chen, H.-Y.; Shi, Q.-S. Learning from small datasets containing nominal attributes. *Neurocomputing* **2018**, *291*, 226–236. [CrossRef]
14. Li, D.-C.; Shi, Q.-S.; Li, M.-D. Using an attribute conversion approach for sample generation to learn small data with highly uncertain features. *Int. J. Prod. Res.* **2018**, *56*, 4954–4967. [CrossRef]
15. Han, H.; Wang, W.-Y.; Mao, B.-H. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In Proceedings of the International Conference on Intelligent Computing, Hefei, China, 23–26 August 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 878–887.
16. Li, D.-C.; Liu, C.-W. Extending attribute information for small data set classification. *IEEE Trans. Knowl. Data Eng.* **2010**, *24*, 452–464. [CrossRef]
17. Liu, X.-Y.; Wu, J.; Zhou, Z.-H. Exploratory undersampling for class-imbalance learning. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **2008**, *39*, 539–550.
18. Ditzler, G.; Polikar, R. Incremental learning of concept drift from streaming imbalanced data. *IEEE Trans. Knowl. Data Eng.* **2012**, *25*, 2283–2301. [CrossRef]
19. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
20. Akbani, R.; Kwek, S.; Japkowicz, N. Applying support vector machines to imbalanced datasets. In Proceedings of the European Conference on Machine Learning, Pisa, Italy, 20–24 September 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 39–50.
21. Ertekin, S.; Huang, J.; Bottou, L.; Giles, L. Learning on the border: Active learning in imbalanced data classification. In Proceedings of the sixteenth ACM Conference on Information and Knowledge Management, Lisbon, Portugal, 6–10 November 2007; ACM: New York, NY, USA, 2007; pp. 127–136.
22. Wang, X.Y.; Yang, H.-Y.; Zhang, Y.; Fu, Z.-K. Image denoising using SVM classification in nonsubsampled contourlet transform domain. *Inf. Sci.* **2013**, *246*, 155–176. [CrossRef]
23. Wu, Q.; Ye, Y.; Zhang, H.; Ng, M.K.; Ho, S.-S. ForesTexter: An efficient random forest algorithm for imbalanced text categorization. *Knowl. -Based Syst.* **2014**, *67*, 105–116. [CrossRef]
24. Yang, L.; Bi, J.W.; Fan, Z.P. A method for multi-class sentiment classification based on an improved one-vs-one (OVO) strategy and the support vector machine (SVM) algorithm. *Inf. Sci.* **2017**, *394*, 38–52.

25. Liu, Y.; Loh, H.T.; Sun, A. Imbalanced text classification: A term weighting approach. *Expert Syst. Appl.* **2009**, *36*, 690–701. [CrossRef]
26. Sun, A.; Lim, E.P.; Liu, Y. On strategies for imbalanced text classification using SVM: A comparative study. *Decis. Support Syst.* **2009**, *48*, 191–201. [CrossRef]
27. He, H.; Garcia, E.A. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **2008**, *21*, 1263–1284.
28. Joulin, A.; Grave, E.; Bojanowski, P.; Mikolov, T. Bag of tricks for efficient text classification. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, 3 April 2016; Volume 2, pp. 427–431.
29. Krawczyk, B.; Woźniak, M.; Schaefer, G. Cost-sensitive decision tree ensembles for effective imbalanced classification. *Appl. Soft Comput.* **2014**, *14*, 554–562. [CrossRef]
30. Li, Y.; Guo, H.; Zhang, Q.; Gu, M.; Yang, J. Imbalanced text sentiment classification using universal and domain-specific knowledge. *Knowl.-Based Syst.* **2018**, *160*, 1–15. [CrossRef]
31. Xu, R.; Chen, T.; Xia, Y.; Lu, Q.; Liu, B.; Wang, X. Word embedding composition for data imbalances in sentiment and emotion classification. *Cogn. Comput.* **2015**, *7*, 226–240. [CrossRef]
32. Jiang, Z.; Li, L.; Huang, D.; Jin, L. Training word embeddings for deep learning in biomedical text mining tasks. In Proceedings of the 2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Washington, DC, USA, 9–12 November 2015; IEEE: Piscataway Township, NJ, USA, 2015; pp. 625–628.
33. Wang, Z.; Ma, L.; Zhang, Y. A hybrid document feature extraction method using latent Dirichlet allocation and word2vec. In Proceedings of the 2016 IEEE First International Conference on Data Science in Cyberspace (DSC), Changsha, China, 13–16 June 2016; IEEE: Piscataway Township, NJ, USA, 2016; pp. 98–103.
34. Cheng, X.; Yan, X.; Lan, Y.; Guo, J. BTM: Topic Modeling over Short Texts. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 2928–2941. [CrossRef]
35. Guo, Y.; Barnes, S.J.; Jia, Q. Mining meaning from online ratings and reviews: Tourist satisfaction analysis using latent dirichlet allocation. *Tour. Manag.* **2017**, *59*, 467–483. [CrossRef]
36. Chunhong, Z.; Licheng, J. Automatic parameters selection for SVM based on GA. In Proceedings of the Fifth World Congress on Intelligent Control and Automation (IEEE Cat. No. 04EX788), Hangzhou, China, 15–19 June 2004; IEEE: Piscataway Township, NJ, USA, 2004; pp. 1869–1872.
37. Huang, C.-L.; Wang, C.-J. A GA-based feature selection and parameters optimizationfor support vector machines. *Expert Syst. Appl.* **2006**, *31*, 231–240. [CrossRef]
38. Fan, R.-E.; Chang, K.; Hsieh, C.; Wang, X.; Lin, C. LIBLINEAR: A library for large linear classification. *J. Mach. Learn. Res.* **2008**, *9*, 1871–1874.
39. Chang, C.-C.; Lin, C.-J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 27. [CrossRef]
40. Batuwita, R.; Palade, V. Adjusted geometric-mean: A novel performance measure for imbalanced bioinformatics datasets learning. *J. Bioinform. Comput. Biol.* **2012**, *10*, 1250003. [CrossRef]
41. Yelp Dataset. Available online: https://www.kaggle.com/yelp-dataset/yelp-dataset (accessed on 30 June 2021).
42. Airola, A.; Pahikkala, T.; Waegeman, W.; De Baets, B.; Salakoski, T. An experimental comparison of cross-validation techniques for estimating the area under the ROC curve. *Comput. Stat. Data Anal.* **2011**, *55*, 1828–1844. [CrossRef]
43. Li, J.; Zhang, H.; Wei, Z. The weighted word2vec paragraph vectors for anomaly detection over HTTP traffic. *IEEE Access* **2020**, *8*, 141787–141798. [CrossRef]
44. Forti, L.; Alfredo, M.; Luisa, P.; Santarelli, F.; Santucci, V.; Spina, S. Measuring text complexity for Italian as a second language learning purposes. In Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications, Florence, Italy, 2 August 2019; Association for Computational Linguistics: Florence, Italy, 2019; pp. 360–368.
45. Ray, S. An analysis of computational complexity and accuracy of two supervised machine learning algorithms—K-nearest neighbor and support vector machine. In *Data Management, Analytics and Innovation*; Springer: Singapore, 2021; pp. 335–347.