Computing and Informatics, Vol. 41, 2022, 98–115, doi: 10.31577/cai_2022_1_98

REDUCING THE EFFECT OF IMBALANCE IN TEXT CLASSIFICATION USING SVD AND GLOVE WITH ENSEMBLE AND DEEP LEARNING

Tajbia Hossain, Humaira Zahin Mauni, Raqeebir Rab

Department of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
e-mail: tajbiahossain@gmail.com, hzm496@gmail.com

Abstract. Due to the recent escalation in the amount of text data available and used online, text classification has become a staple for data analysts when extracting relevant information. Yet, machine learning algorithms are susceptible to biases when implemented on any large-scale automated task, especially in text analysis. With the popularization of newer branches of study emerging from the field of machine learning – such as ensemble and deep learning – we must analyze the potential pitfalls in the common experimental setup centered around learning algorithms. Imbalance in text data is one such pitfall – when data is not equally distributed across all categories in a dataset, it can influence and undermine the classification of underrepresented categories. In our research, we have proposed several techniques and unique approaches to tackle this obstacle. We prepared four datasets of varying degrees of imbalance to conduct our experimentation. We proved that feature extraction techniques singular value decomposition (SVD) and GloVe are the key to reducing the effect of imbalance in text classification, especially in ensemble and deep learning. Using the result of our research, we have also proposed a modified ensemble classifier that can classify imbalanced and balanced data alike.

Keywords: Deep learning, ensemble learning, machine learning, text classification, imbalanced data, singular value decomposition, global vectors

1 INTRODUCTION

When presented with a news article, scanning a few lines of text is enough for a reader to understand what section of the newspaper they might find it in – be it politics, business, or entertainment. Yet, when presented with thousands of articles, the task becomes daunting. This is a small example of what data mining and machine learning have achieved in this day and age – we can now teach a machine learning model to classify thousands of news articles in mere seconds, eliminating the cost of time, effort, and human error.

Text classification can be defined as an automated process that learns to sort any document or segment of text into a category based on their content. Imagine handing a person a collection of news articles about to head into print and asking them to sort every article into which section of the newspaper it ought to be published under – Business or Entertainment? Politics or Sports? Now imagine millions and millions of articles and replace the newspaper with a company looking to make important decisions about product and service workflow based on this singular task of sorting data. The stakes and the magnitude of the task skyrocket, yet the resources available – the single entity assigned to said task – remains the same. This is where data mining and machine learning comes in – we can now teach a machine learning model to classify thousands of news articles in mere seconds, eliminating the cost of time, effort, and human error. Yet, just as humans, it must overcome certain weaknesses, so do the machine learning methods we employ. In order to build a classifier that can be successfully adapted for real world usage, we must be wary of all the biases and defects present in current classification approaches.

The most prevalent and challenging bias present in text classification is the severe imbalance observed in the datasets extracted from data mining. This means that the data is never equally distributed across the categories it is meant to be sorted into. When training classification models, researchers seldom think about the discrepancy in data distribution between their training/testing datasets and the real-world datasets their classification models are employed on.

Our approach to overcoming this problem of imbalance was to first obtain multiple datasets with varying data distributions – starting from highly and moderately balanced datasets (where news articles are distributed across the categories evenly) to moderately and highly imbalanced datasets (where news articles are distributed across the categories sporadically). Then, we began to formulate techniques that would be unaffected by the varying degrees of imbalance in the data it classifies. Two feature extraction methods, singular value decomposition (SVD) and global vectors (GloVe) [1] reduced the effect of imbalance in ensemble and deep learning classifiers. We employed these methods with deep learning algorithms in an attempt to create an unbiased classifier. In addition, we formulated a modified ensemble learning approach involving SVD – a modified voting classifier, which had the most promising results. To measure imbalance, we turned to the metric macro-F1 accuracy as it places equal weight on all classes when calculating the accuracy. The difference between accuracy and macro-F1 accuracy was also an indicator of the difference be-

tween *perceived* accuracy with respect to imbalance and the actual accuracy, after taking into account the dissimilar data distribution.

This multi-pronged approach to a classification problem presents a new ground for us to truly analyze the effect of imbalanced data on classifiers derived from popular branches of machine learning – such as deep learning and ensemble methods. Our experimental modified voting classifier borne from this research obtained promising results. Furthermore, using datasets of varying degrees of imbalance is a unique take on researching imbalance. From an analytical perspective, it establishes a point of comparison previously unseen and allows us to seek a classification technique that truly works when presented with the fluid, ever-changing conditions that we can expect in real-life situations.

2 BACKGROUND

Before we delve into our research, it is crucial to establish the groundwork for the task we have undertaken by providing ample background information. We do so by examining related work then formally defining the problem and finally explaining the terminologies that will frequently appear in the paper.

2.1 Related Work

In recent years, classification of text data has been primarily concerned with the usage of ensemble learning and deep learning algorithms on balanced text data, i.e. balanced across categories, or artificially balanced using state-of-the-art sampling techniques. Present-day research also uses cost-sensitive approaches to minimize the cost of misclassification of undermined categories. Research papers on recent findings on different feature selection, feature reduction, and classification techniques studied are summarized in brief below.

Kilimci et al. [2] focused on enhancing the overall accuracy of a text classification system by using an ensemble of base classifiers – Heterogeneous Stacking (Heter-Stck) and a Convolutional Neural Network (CNN) with two different feature extraction techniques. When applied to eight different popular Turkish and English news article datasets, including 20Newsgroups, their Heter-Stck model and CNN gave an accuracy of 94.30 % and 91.88 %, respectively. In another study of text classification based on extreme learning machine (ELM), Li et al. [3] achieved macro-F1 scores of 79.4 % for the 20Newsgroups dataset and 68.2 % for a similar variant of the Reuters-21578 dataset. Onan et al. [4] achieved an accuracy of 91.49 % on the Reuters dataset in his study to examine the predictive performance of different statistical keyword extraction methods. An ensemble bagging classifier using random forest was used in this scenario.

Other than the previously mentioned algorithms and models, deep learning models including Recurrent Neural Networks (RNN) [5] and Deep Neural Networks (DNN) [6] also provide high accuracies and F1 scores. A few researchers have also

used term weighting approaches [7], thresholding on severely imbalanced datasets [8] as well as noble loss functions for training deep networks on an imbalanced dataset to reduce the loss from smaller categories [9]. Padurariu et al. [10] proposed using different variants of feature selection and machine learning algorithms on an imbalanced text dataset to find which algorithm performed the best.

After reviewing an adequate amount of research, we began to formulate our problem to design a unique text classifier that would efficiently classify text data having *varying degrees of imbalance*, as found in real-life situations.

2.2 Problem Formulation

The problem of designing a text classifier can be defined as learning a target function ϕ that can assign a true or false value to $(d_N, c_n) \in D \times C$, where D is our dataset of N news articles and C is our set of n pre-defined categories.

$$\phi(d_j, c_n) = [0, 1], \tag{1}$$

$$D = \{d_1, d_2, \ldots, d_N\}, \tag{2}$$

$$C = \{c_1, c_2, \ldots, c_n\} \tag{3}$$

where:

- ϕ = Classifier,
- D = Dataset of news articles,
- C = Set of pre-defined categories.

Our goal is to design a classifier ϕ that will classify news articles from datasets of all distributions – both balanced and imbalanced – at an acceptable accuracy metric.

2.3 Terminology

In this section, we will put forward the important terminologies that will be frequently used in this paper defined within the context of our problem formulation.

Imbalanced Data: Imbalanced data refers to the unequal distribution of data across the categories in a dataset. In our case, imbalance is observed in a dataset when the number of news articles belonging to one or more categories is significantly higher or lower compared to the other categories [8].

Term Frequency-Inverse Document Frequency *(tf-idf)*: *Tf-idf* gives us word frequency scores that try to highlight words that are more interesting. It is a very common algorithm that transforms text into a meaningful representation of numbers which is then used to fit machine algorithms for prediction.

It is calculated using the following formula:

$$tf.idf_{t,d} = tf_{t,d} \times idf_{t,d} = tf_{t,d} \times \log\left(\frac{N}{df_t}\right) \tag{4}$$

where:

- $tf_{t,d}$ = number of occurrences of term t in document d,
- df_t = number of documents containing t terms,
- N = total number of documents.

Singular Value Decomposition (SVD): SVD is a technique derived from the field of linear algebra. It is a data preparation and dimension reduction technique that creates a projection of the sparse dataset prior to fitting a model. SVD decomposes a real matrix according to the following equation:

$$A = USV^T \tag{5}$$

where:

- A = document-term matrix of size $d_N \times t$,
- U = left singular vectors of size $d_N \times c_n$,
- S = diagonal matrix of single values of size $c_n \times c_n$,
- V^T = right singular vectors of size $c_n \times d_N$.

SVD extracts and reduces features by decomposing the matrix A into the three component matrices. It then drops columns from the matrix V^T that have no predictive value and reconstruct it to matrix B.

$$B = U_r S_r \tag{6}$$

where U_r and S_r are the reduced version of U and S. The data from this reduced matrix is mapped to the original matrix, giving us A_r.

$$A_r = B_r V^T. \tag{7}$$

Here, A_r has exactly the same dimensions as before, but has a reduced rank. This reduced matrix will be used as the input data for our classification models having dimensions according to the size of the dataset [17].

Word Embeddings: Word embeddings allow words with similar meaning to have a similar representation. This ditributed representation for text is perhaps one of the key breakthroughs for the impressive performance of deep learning methods on challenging natural language processing problems [12]. As word embeddings represent words as dense vectors, they collect more information into a smaller space, which means that they reduce the dimensionality of text data.

GloVe: GloVe is a pretrained word vector which is based on matrix factorization techniques [1]. It is used to convert our text into word embeddings which will be used as the input layer for neural networks. For our classification models, we employed two different versions of GloVe embeddings – GloVe-6B and GloVe-42B.

3 DATA

Locating and extracting enough data – and the right kind of data – is crucial for research in any field. Reuters-21578 [11], 20Newsgroups [13], and BBC News [14] are some popular datasets mostly used in text classification. While these datasets are quite ideal, it must also be promulgated that data, in real life, is present in non-ideal conditions. Thus, we prepared our datasets to have *varying degrees of imbalance*, meaning that some were further processed to exhibit the desired degree of balance or imbalance.

3.1 Data Acquisition

For our classification models, we acquired the three different news article datasets mentioned above. These original datasets were further processed to create four datasets with four different levels of data distributions, as indicated in Table 1 and Figures 1, 2, 3 and 4.

Datasets	**News Articles**	**Categories**	**Size**	**Data Distribution**
20Newsgroups	18 846	20	Large	Highly balanced
BBC News	2 225	5	Small	Moderately balanced
BBC News (imbalanced)	1 321	5	Small	Moderately imbalanced
Reuters-21578 (modified)	10 093	25	Large	Highly Imbalanced

Table 1. Description of datasets

3.2 Data Preprocessing

Unprocessed data could be very inconsistent and give us undesirable results [15]. Data preprocessing makes data more structured and easily disposable by classification algorithms. Our datasets were first processed to achieve the desired level of balance or imbalance and then restructured via text pre-processing techniques.

The BBC News and 20 Newsgroups were naturally occurring balanced datasets that did not require further processing. BBC News (imbalanced) is a moderately imbalanced dataset obtained from our previous research [16], where we reduced the number of articles in certain categories to create an imbalanced, modified version of the original. Reuters-21578 was also further processed to exhibit a high degree

of imbalance by assigning multi labeled articles to a single class and eliminating categories having less than 50 documents.

After that, all four of our datasets went through a text cleaning function which removed punctuation marks, Html tags, extra spaces and normalized all the words. The different inflected forms of the words were then grouped together, i.e. lemmatized, to finally create four distinct, pre-processed datasets exhibiting four varying levels of imbalance. The datasets are further discussed in details in the following section with their respective data distribution shown graphically.

3.2.1 Highly Balanced Dataset – 20Newsgroup

The 20Newsgroup dataset chosen as the highly balanced dataset, comprises of 18 846 articles partitioned equally across 20 different newsgroups. Each data in the 20 different newsgroups corresponds to a different topic. Some of the newsgroups are very closely related to each other, while others are highly unrelated.

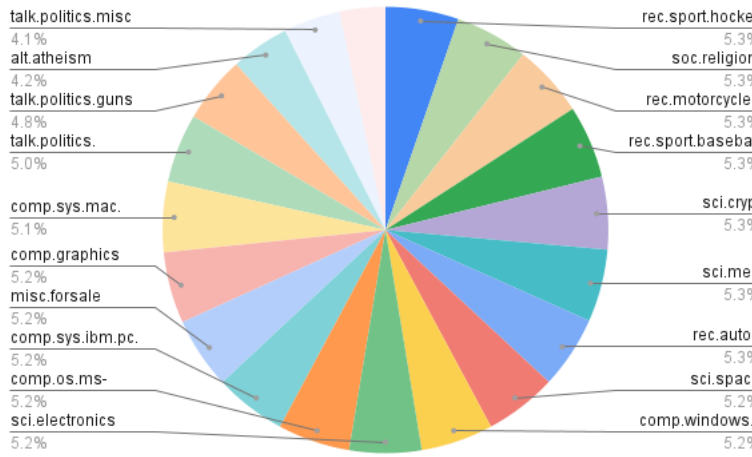

Figure 1. Data distribution of the 20Newsgroups dataset

3.2.2 Moderately Balanced Dataset – BBC News

The BBC news dataset, consisting of 2 225 news articles from the BBC website from the year 2004–2005 was chosen to be our moderately balanced dataset. The articles are distributed across 5 categories of tech, sports, politics, business, and entertainment.

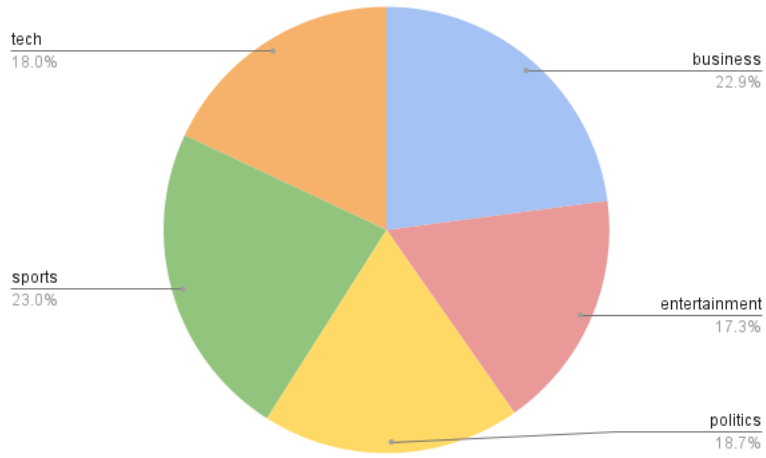

Figure 2. Data distribution of the BBC News dataset

3.2.3 Moderately Imbalanced Dataset – BBC News (Imbalanced)

The moderately imbalanced dataset used here was created manually by removing articles during our previous research. The dataset has a reduced number of articles for the tech, politics and entertainment classes, whereas the other two were kept as it was in the original dataset. All three of the reduced classes had 100 articles each.

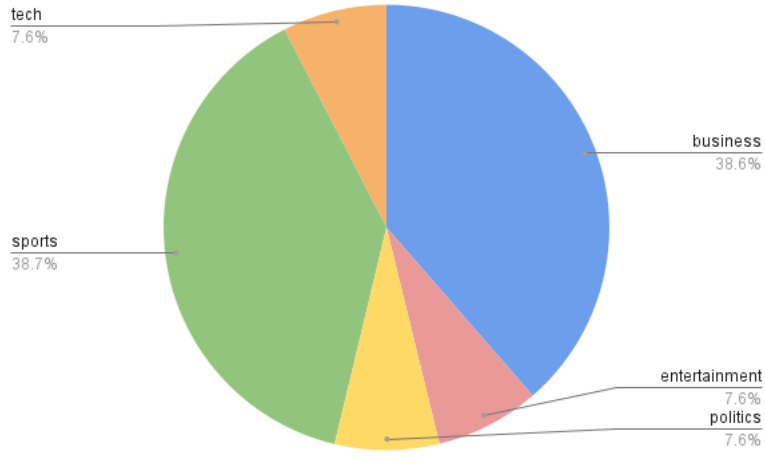

Figure 3. Data distribution of the BBC News (imbalanced) dataset

3.2.4 Highly Imbalanced Dataset – Reuters-21578

The Reuters-21578 dataset contains structured information about newswire articles that are assigned to several classes, making it a multi-label problem. It has a highly skewed distribution of documents over 25 categories, where a large proportion of documents belong to few topics.

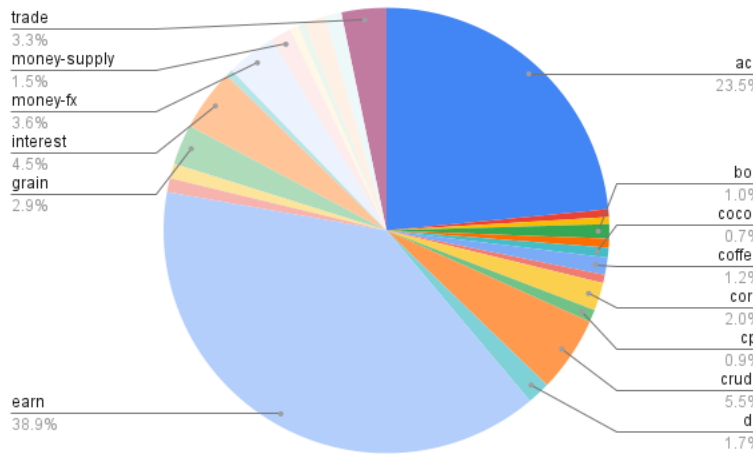

Figure 4. Data distribution of the Reuters-21578 dataset

4 CLASSIFICATION VIA DEEP LEARNING

Once our four datasets were processed, we began formulating various classifiers that aimed to reduce the effect of imbalance in text classification. For this, we chose to focus on deep learning algorithms, as they are proven to work well with imbalanced classification [9]. Deep learning uses interconnected nodes and layers to predict class C from a dataset D by learning a set of weights, $W = \{w_1, \ldots, w_x\}$, and biases, $B = \{b_1, \ldots, b_y\}$, using activation and loss functions. The following proposed techniques here consist of five deep learning models chosen from a pool of two different feature extraction methods and three neural network architectures. Feature extraction maps such text data to real-valued vectors for our classification model. Extracting a set of features using effective algorithms will not only reduce the dimensions of the feature space, but it will also delete redundant features for our model [16].

4.1 Proposed Technique I – Deep Neural Network Using Singular Value Decomposition

Our first DNN model was formulated with SVD as its feature extraction method. At first, the processed news articles were vectorized into an array of real numbers using *tf-idf*. This conversion from text to numbers usually creates a sparse matrix where most of the elements are zeros, thus irrelevant [17]. SVD decomposed this sparse matrix according to the following equation:

$$A = USV^T. \tag{8}$$

The use of SVD removed the columns from the matrix V^T with the most irrelevant features in such a way that the richness of the data was preserved. This gave us a reduced matrix B, having 5 000 columns reduced from 75 000, but still maintaining the same number of rows as A. The matrix B can be represented as:

$$B = U_r S_r. \tag{9}$$

The smallest singular values from the matrix S were set to zero and the best axis to project our data was calculated which would give the minimum reconstruction error. After removing the columns the data matrix is reconstructed into matrix A_r, which can be represented as:

$$A_r = BV^T. \tag{10}$$

A_r had the same dimensions as the original matrix A but a reduced rank. This data matrix was our input layer for our DNN model used for classification.

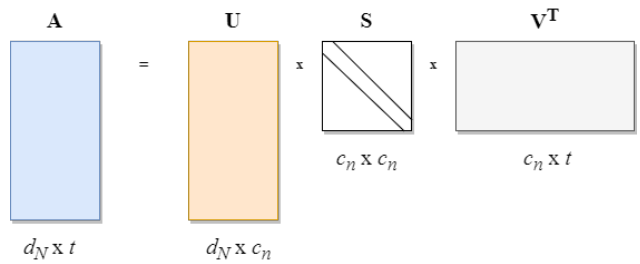

Figure 5. Feature extraction using SVD

The DNN model we are proposing consisted of one input layer, four hidden layers, and an output layer. Every dense layer was separated by a dropout layer – this reduced overfitting by creating 'dead neurons' that do not learn the training set too perfectly. The activation function used in the dense layers was 'relu' – which is used because the dense layer has many neurons that require activation, and it is a computationally efficient and effective function. The output layer uses 'softmax' activation, which is the norm for multiclass classification. The model was

trained using 'sparse categorical cross-entropy' loss, 'adam' optimizer, and 'accuracy' metrics.

4.2 Proposed Technique II – Recurrent Neural Network and Convolutional Neural Network Using GloVe-6B and GloVe-42B

The second technique we are proposing uses GloVe as the input embedding layer of two different neural network architectures – Long Short-Term Memory (LSTM) Recurrent Neural Network and Convolutional Neural Network. GloVe, standing for "*Global Vectors*", is a pretrained word vector which is based on matrix factorization techniques [1].

$$WC = WF \times FC. \tag{11}$$

The matrix factorization depicted in Figure 6 and Equation (11) summarizes how GloVe uses the pre-trained word-context co-occurrence matrix WC to obtain the word-feature matrix WF that can then serve as our input word embedding.

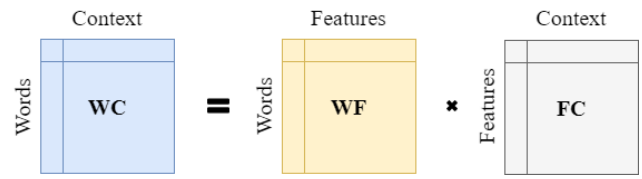

Figure 6. Feature extraction using GloVe

This process converted our sparse text into dense word embeddings. Dense vectors collect more information into a smaller space, which means that they reduce the dimension of text data. The word embeddings produced were chosen as the embedding layer for the mentioned neural network models as, unlike traditional algorithms, GloVe does not rely just on local context information of words, but incorporates global word co-occurrence to obtain word vectors. The first version of the GloVe pre-trained vector – GloVe-6B, was trained on 6 billion words and had an embedding dimension of 50. The second version, GloVe-42B was trained on 42 billion words with an embedding dimension of size 300.

Our LSTM RNN model with GloVe as the input embedding layer and four Gated Recurrent Units (GRU) served as one approach. An RNN is a deep learning model that preserves the information passed in each neuron by creating loops within them. This preserves the sequence of text data, much like word embeddings. Our CNN model consisted of an input word embedding layer, three convolutional layers, and three max-pooling layers. The final layers are made from fully connected four dense layers. The same activation functions and training parameters used by our DNN model were applied for both the RNN and CNN.

5 CLASSIFICATION VIA MODIFIED ENSEMBLE LEARNING

After conducting our research on deep learning algorithms, we then attempted to create a unique voting ensemble classifier suited to our task. When a collection of algorithms work together to classify data instead of a singular classifier, it is known as an ensemble. The ensemble voting classifier we have used applies k different classifiers $\phi_1, \ldots, \phi_k$ to the task of deciding whether a news article d_j belongs in class c_i (if $d_j \in c_i$), and then combines their outcomes appropriately. Our proposed method included SVD and *tf-idf* as the feature extraction techniques paired with the modified ensemble classifier.

5.1 Proposed Technique III – Modified Voting Classifier Using tf-idf and SVD

For this unique classifier, we began by extracting features using Term Frequency – Inverse Document Frequency (*tf-idf*). *Tf-idf* gave us word frequency scores that try to highlight words that are more interesting. It is calculated for each term in a document using the following formula:

$$tf.idf_{t,d} = tf_{t,d} \times idf_{t,d} = tf_{t,d} \times \log\left(\frac{N}{df_t}\right). \tag{12}$$

The word frequency scores were then converted into a vectorized matrix. A maximum of 75 000 features was extracted which were then reduced to 5 000 features using SVD, just as how it was done with our DNN model. Once the processed and reduced input data was ready, it was then classified using a modified voting classifier configured, as shown in Figure 7.

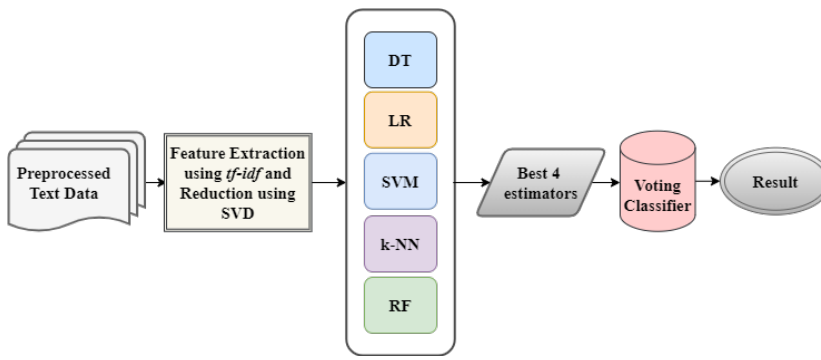

Figure 7. Proposed modified voting classifier using *tf-idf* and SVD

This model uses the best four out of the five estimators – Decision Tree (DT), Logistic Regression (LR), Support Vector Machine (SVM), k-Nearest Neighbor (k-NN), and Random Forest (RF). First, we individually trained these five low-cost,

efficient algorithms on each dataset, and then assigned the best four performers as the ensemble estimators instead of all five. This added step of assigning the best performers (based on the dataset the algorithms are trained on) as estimators tailors the voting classifier to the incoming imbalanced or balanced dataset. The final result of classification is obtained via *hard voting*, where the category receiving the highest number of votes among all estimators is chosen.

	Highly Balanced Dataset		**Moderately Balanced Dataset**		**Moderately Imbalanced Dataset**		**Highly Imbalanced Dataset**	
	20News-groups		**BBC News (original)**		**BBC News (imbalanced)**		**Reuters-21578**	
	Acc. (%)	**M-F1 Avg. (%)**	**Acc. (%)**	**M-F1 Avg. (%)**	**Acc. (%)**	**M-F1 Avg. (%)**	**Acc. (%)**	**M-F1 Avg. (%)**
Neural Network Classifier								
DNN (SVD)	87.36	87.41	95.96	95.88	96.6	94.6	91.13	80.22
RNN (GloVe-6B)	77.11	77.15	94.61	94.6	92.83	87.26	86.97	66.69
RNN (GloVe-42B)	84.75	84.49	94.61	94.51	87.92	80.6	88.41	70.59
CNN (GloVe-6B)	82.28	82.34	96.18	96.03	95.85	92.13	90.74	78.43
CNN (GloVe-42B)	87.58	87.36	96.18	96.03	97.36	95.12	91.18	80.68
Modified Ensemble Classifier								
Voting (tf-idf)	91.52	91.42	97.98	97.94	97.98	97.94	92.22	85.03
Voting (tf-idf, SVD)	90.81	90.78	97.98	97.94	94.34	91.29	92.32	86.1

Table 2. Accuracy and macro-F1 average of the different models applied to the four different datasets

6 RESULTS

After implementing our different proposed techniques on the four datasets in Table 1, the results of our experimentation are disclosed in Table 2. The metric used to measure imbalanced classification was macro-F1 average (M-F1). The more common metric for accuracy was also included to provide a basis for comparison.

$$\textit{F1-score} = \frac{2 \cdot \textit{Precision} \cdot \textit{Recall}}{\textit{Precision} + \textit{Recall}}. \tag{13}$$

Here, *Precision* is the ratio between correctly predicted positive classifications to the total predicted positive classifications, and *Recall* is the ratio between correctly predicted positive classifications to all positive and negative in the category.

$$\text{M-F1} = \frac{1}{n} \cdot \sum_{i=0}^{n} \textit{F1-score}_i \tag{14}$$

Here, i is the class index and n the number of classes. M-F1 gives us the average of each category's F1 scores, without taking into consideration the number of samples in each class. When classifying imbalanced datasets, we hope to achieve similar accuracy and macro-F1 score, so that the accuracy reflects the classification for all classes.

6.1 Performance of Neural Network Classifiers

As observed in Table 2, we can see that CNN using GloVe-42B gave us the predominantly best results across all the differently balanced datasets. DNN using SVD was a close second, only surpassing CNN (GloVe-42B) by achieving an 87.41 % M-F1 average over the former's 87.36 % in highly balanced classification. A visual comparison between the two models is also shown in Figure 8. These two algorithms were also successful in bridging the gap between accuracy and M-F1 average as the datasets grew more imbalanced. This shows that DNN using SVD and CNN using GloVe-42B can classify imbalanced data in a way where the common metric, accuracy, is not misleading and does not differ drastically from the M-F1 average.

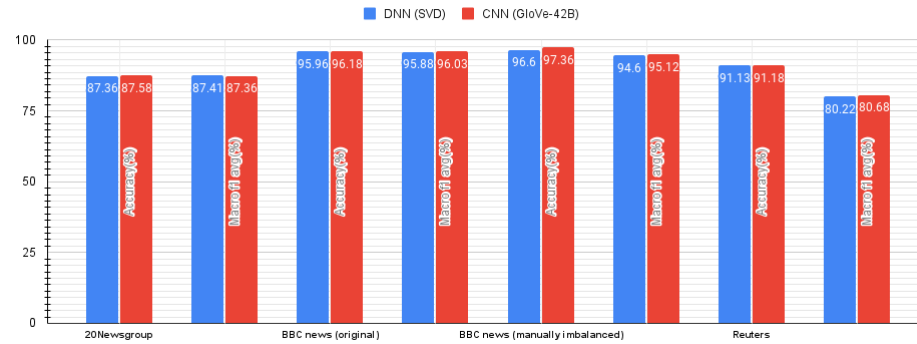

Figure 8. Comparison of accuracy and macro-F1 scores of DNN (SVD) vs. CNN (GloVe-42B)

6.2 Performance of Modified Voting Classifier

The voting classifier using *tf-idf* and SVD combined achieved the highest results across all data distributions except for the moderately imbalanced dataset. As the voting classifier using only *tf-idf* achieved much higher M-F1 and accuracy, as observed in both Table 2 and Figure 9, we can presume that the feature reduction resulting from SVD performed poorly on the much smaller moderately imbalanced dataset (1 321 news articles) compared to the highly imbalanced dataset with 10 093 news articles, resulting in underfitting. Nevertheless, this classifier had the lowest

gap between accuracy and M-F1 out of all the proposed methods. Thus, we can conclude that SVD and *tf-idf* combined with our modified voting classifier significantly boost the performance of text classification of imbalanced data.

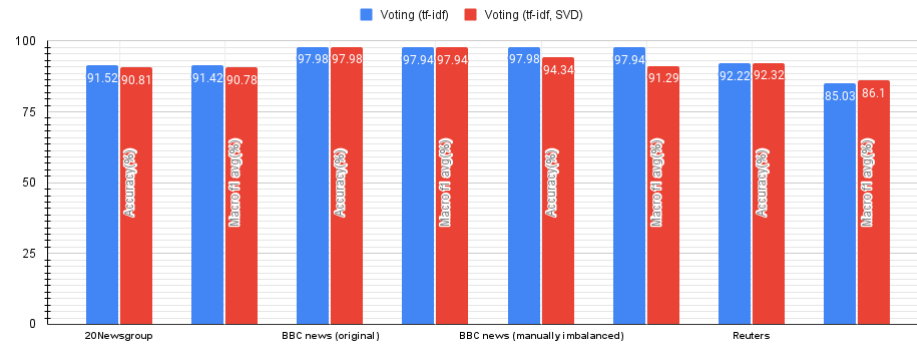

Figure 9. Comparison of accuracy and macro-F1 scores of voting (tf-idf) vs. voting (tf-idf, SVD)

7 CONCLUSION

In order to classify imbalanced data, we obtained datasets of various degrees of imbalance and formulated different techniques to carry out this task. The result of our research indicated that both SVD and GloVe were key to boosting the performance of neural network and ensemble classifiers. In addition, we proposed a novel approach involving SVD and a modified ensemble classifier that outperformed the extreme learning machine (ELM) proposed by Li et al. [3], achieving macro-F1 scores of 90.78 % as opposed to their 79.4 % for the 20Newsgroups dataset, and 86.1 % against their 68.2 % for a less imbalanced variant of the Reuters-21578 dataset. All our deep learning approaches similarly surpassed these values. Our modified voting classifier and two best performing deep learning classifiers – DNN (SVD) and CNN (GloVe-42B) – outmatched the ensemble classifier proposed by Onan et al. [4], which achieved an accuracy of 91.49 % on the Reuters dataset. Thus, we can conclude that while researchers have begun to address the biases and defects present in current machine learning practices, it is important to systematically analyze these issues to figure out solutions that are best applicable to real-life conditions.

REFERENCES

[1] Pennington, J.—Socher, R.—Manning, C. D.: Glove: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543, doi: 10.3115/v1/D14-1162.

[2] Kilimci, Z. H.—Akyokus, S.: Deep Learning – and Word Embedding-Based Heterogeneous Classifier Ensembles for Text Classification. Complexity, Vol. 2018, 2018, Art. No. 7130146, doi: 10.1155/2018/7130146.

[3] Li, M.—Xiao, P.—Zhang, J.: Text Classification Based on Ensemble Extreme Learning Machine. 2018, arXiv: 1805.06525.

[4] Onan, A.—Korukoğlu, S.—Bulut, H.: Ensemble of Keyword Extraction Methods and Classifiers in Text Classification. Expert Systems with Applications, Vol. 57, 2016, pp. 232–247, doi: 10.1016/j.eswa.2016.03.045.

[5] Liu, P.—Qiu, X.—Huang, X.: Recurrent Neural Network for Text Classification with Multi-Task Learning. 2016, arXiv: 1605.05101.

[6] Gargiulo, F.—Silvestri, S.—Ciampi, M.—De Pietro, G.: Deep Neural Network for Hierarchical Extreme Multi-Label Text Classification. Applied Soft Computing, Vol. 79, 2019, pp. 125–138, doi: 10.1016/j.asoc.2019.03.041.

[7] Liu, Y.—Loh, H. T.—Sun, A.: Imbalanced Text Classification: A Term Weighting Approach. Expert Systems with Applications, Vol. 36, 2009, No. 1, pp. 690–701, doi: 10.1016/j.eswa.2007.10.042.

[8] Johnson, J. M.—Khoshgoftaar, T. M.: Deep Learning and Thresholding with Class-Imbalanced Big Data. 2019 18th IEEE International Conference on Machine Learning and Applications (ICMLA), Boca Raton, FL, USA, 2019, pp. 755–762, doi: 10.1109/ICMLA.2019.00134.

[9] Wang, S.—Liu, W.—Wu, J.—Cao, L.—Meng, Q.—Kennedy, P. J.: Training Deep Neural Networks on Imbalanced Data Sets. 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, 2016, pp. 4368–4374, doi: 10.1109/IJCNN.2016.7727770.

[10] Padurariu, C.—Breaban, M. E.: Dealing with Data Imbalance in Text Classification. Procedia Computer Science, Vol. 159, 2019, pp. 736–745, doi: 10.1016/j.procs.2019.09.229.

[11] UCI Machine Learning Repository: Reuters-21578 Text Categorization Collection Data Set. `https://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection`, Last accessed: 26 May 2021.

[12] Machine Learning Mastery: What Are Word Embeddings for Text? `https://machinelearningmastery.com/what-are-word-embeddings/`. Last accessed: 31 October 2021.

[13] Lang, K.: NewsWeeder: Learning to Filter Netnews. Proceedings of the 12th International Machine Learning Conference (ML95), 1995, pp. 331–339, doi: 10.1016/B978-1-55860-377-6.50048-7.

[14] Greene, D.—Cunningham, P.: Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering. Proceedings of the 23rd International Conference on Machine Learning (ICML '06), 2006, pp. 377–384, doi: 10.1145/1143844.1143892.

[15] Kalra, V.—Aggarwal, R.: Importance of Text Data Preprocessing and Implementation in RapidMiner. In: Jaiswal, A. et al. (Eds.): Proceedings of the 2017 International Conference on Information Technology and Knowledge Management.

Annals of Computer Science and Information Systems, Vol. 14, 2017, pp. 71–75, doi: 10.15439/2017KM46.

[16] Mauni, H. Z.—Hossain, T.—Rab, R.: Classification of Underrepresented Text Data in an Imbalanced Dataset Using Deep Neural Network. 2020 IEEE Region 10 Symposium (TENSYMP), Dhaka, Bangladesh, 2020, pp. 997–1000, doi: 10.1109/TENSYMP50017.2020.9231021.

[17] Wall, M. E.—Rechtsteiner, A.—Rocha, L. M.: Singular Value Decomposition and Principal Component Analysis. In: Berrar, D. P., Dubitzky, W., Granzow, M. (Eds.): A Practical Approach to Microarray Data Analysis. Springer, Boston, MA, 2003, pp. 91–109, doi: 10.1007/0-306-47815-3_5.

Tajbia Hossain is currently working as Software Engineer at Samsung Research and Development Institute Bangladesh (SRBD). She received her B.Sc. in computer science and engineering from the Ahsanullah University of Science and Technology, Bangladesh, in 2021. She has presented her past conference paper on the classification of imbalanced data using deep neural networks at the 2020 IEEE Region 10 Symposium. Her research interests include but are not limited to artificial intelligence, deep learning, natural language processing (NLP), software testing automation and software development.

Humaira Zahin Mauni has graduated with a B.Sc. degree in computer science and engineering and is currently enrolled in the M.Sc. in computer science program at the Florida State University. Her research interests include – but are not limited to – machine learning, natural language processing, deep learning, development and integration of AI systems, data mining, and HCI. She presented her work on deep neural networks and imbalanced data at the 2020 IEEE Region 10 symposium. She is currently the co-author of two publications under both international and national peer-reviewed scientific conferences and plans to further expand her research experience in the years to come.

Raqeebir Rab received her B.Sc. (Hons.) degree in science with a major in computer science from the Augustana Faculty, University of Alberta, Canada in 2004. She completed her M.Sc. in computer science from the Concordia University, Montreal, Canada in 2012. She is currently working as Assistant Professor at the Department of Computer Science and Engineering (CSE), Ahsanullah University of Science and Technology (AUST), Dhaka, Bangladesh. Her research interests include wireless multihop networks (ad hoc and sensor networks) with an added emphasis on mathematical modelling, performance analysis and protocol design and data science.