Breaking the Curse of Class Imbalance: Bangla Text Classification

MD. RAFI-UR-RASHID*, MAHIM MAHBUB*, and MUHAMMAD ABDULLAH ADNAN, Bangladesh University of Engineering & Technology (BUET), Bangladesh and United International University, Bangladesh

This paper addresses the class imbalance issue in a low-resource language called Bengali. As a use-case, we choose one of the most fundamental NLP tasks, i.e., text classification, where we utilize three benchmark text corpus- fake news dataset, sentiment analysis dataset, and song lyrics dataset. Each of them contains a critical class imbalance. We attempt to tackle the problem by applying several strategies that include data augmentation with synthetic samples via text and embedding generation in order to augment the proportion of the minority samples. Moreover, we apply ensembling of deep learning models by subsetting the majority samples. Additionally, we enforce the focal loss function for class imbalanced data classification. We also apply the outlier detection technique, data resampling, and hidden feature extraction to improve the minority-f1 score. All of our experimentations are entirely focused on textual content analysis, which results in more than **90%** minority-f1 score for each of the three tasks. It is an excellent outcome on such highly class-imbalanced datasets.

CCS Concepts: • **Computing methodologies** → *Machine learning*.

Additional Key Words and Phrases: class imbalance, fake news, sentiment analysis, song lyrics, text classification, data augmentation, ensembling, resampling, hidden feature extraction, neural networks

1 INTRODUCTION

Class imbalance usually reflects an unequal distribution of classes within a dataset. Running classification on such a dataset often derives a large scale of accuracy, which provides a false estimation of the overall performance because of the exclusive dominance of the majority class. In some particular applications [Aggarwal and Yu 2001; Cieslak et al. 2006; Li et al. 2010] where the main purpose is to segregate the defective samples from the regular data, it is essential to ensure that the accuracy is not solely contributed by the majority class samples, thereby nullifying the intended purpose of such a classification task. The issue of class imbalance has been addressed in several works [Japkowicz and Stephen 2002; Johnson and Khoshgoftaar 2019]. Although a notable amount of work [Huang et al. 2016; Wang et al. 2016] has been done to deal with class imbalance problems in English, there is hardly any literature on handling this issue for low resource languages like Bengali. Two conspicuous challenges in working with Bengali include its linguistic diversity and scarcity of quality data [Islam 2009; Karim et al. 2012]. Some of the dominant causes that make Natural Language Processing tasks (e.g. text classification, information retrieval, hidden feature extraction) in Bangla much challenging include superfluous polymorphism of sentence and word structure, a large number of unique characters, conjugate alphabets, and difficulty in lemmatization. As a result, all of the existing techniques that usually work well to deal with the class imbalance problem in high resource languages (e.g. English) may not necessarily show good performance in the case of some low resource language like Bengali. Hence, we explored several strategies to reduce the performance bottlenecks in the text classification

*They are the primary authors of this research.

Authors' address: Md. Rafi-Ur-Rashid, rafiurrashid150@gmail.com; Mahim Mahbub, mahim.mahbub.97@gmail.com; Muhammad Abdullah Adnan, abdullah.adnan@gmail.com, Bangladesh University of Engineering & Technology (BUET), Dhaka, Bangladesh, 1205 and United International University, Dhaka, Bangladesh, 1212.

2375-4699/2022/8-ART111 $15.00
https://doi.org/10.1145/3511601

task due to class imbalance. Later we applied them on three class imbalanced Bangla text corpus, and eventually came to a conclusion on which of these techniques show promising performance and which ones do not.

The experimentation we performed to mitigate the class imbalance problem includes a couple of data augmentation techniques, **synthetic text**, and **synthetic embedding generation**. To synthesize new text samples, we utilized three strategies: next word prediction, next character prediction, and replacement with similar words. Subsequently, we extend the synthesis task on text embeddings. Three deep learning generative models: Generative Adversarial Network, Variational Autoencoder, and LSTM Autoencoder were used in order to synthesize new text embeddings. Secondly, we utilized an **ensembling** strategy by integrating several deep learning models using suitable subsets of majority samples and all minority samples. Next, we applied a distinctive loss function called **focal loss** which was introduced by FacebookâĂŹs AI research lab, intended to address the issue of the class imbalance problem. Apart from that, we performed an **outlier detection** technique considering the minority class samples as the outliers in this case. We also experimented with a few **resampling** strategies, including random undersampling of the majority class and oversampling of the minority class. Finally, we extracted features from the feature space in the hidden layers of a deep learning model (i.e., Bidirectional-LSTM) to obtain a more compressed representation of the text embeddings and use them in training linear and tree-based classifier models.

We trained several tree-based, linear, and deep learning models to evaluate the performance of the aforementioned techniques in reducing the class imbalance problem. The linear and tree-based state-of-the-art models include extreme Gradient Boosting, Random Forest, BernoulliâĂŹs Naive Bayes, Support Vector Classifier of Support Vector Machines, and Light Gradient Boosting Machine. On the other hand, the deep learning models we use in this work include Long Short Term Memory (LSTM), Convolutional Neural Network (CNN), Bidirectional LSTM (BiLSTM), and Bidirectional Gated Recurrent Unit (BiGRU).

We applied the techniques mentioned above on three benchmark datasets in this work. The first one is *Bangla sentiment analysis classification benchmark dataset* prepared by Sazzed [2020]. The original dataset has some class imbalance (72% - 28%). However, we increased the imbalance (85%- 15%) by removing 1.5K samples randomly from the minority class. The second dataset we used in this work is an extended version of *Bangla fake news dataset* developed by Hossain et al. [2020]. This dataset has **96.73%** of authentic class samples along with just **3.27%** of fake samples. So, the imbalance is quite acute in this corpus. Finally, The third dataset is *Bangla Song Lyrics* from Kaggle where the contribution ratio of the two classes is 86%-14%. We conducted our experiments on these three text classification tasks for reducing the performance bottleneck due to the class imbalance issue. Later, we received an excellent outcome for each dataset in terms of f1-score for the minority class without hampering the overall performance, because **all of the experimental results in this work strictly derive 99-100% of precision, recall, and f1-score for the majority class.** Major contributions of this work are summarized below:

- This work addresses the class imbalance problem in the text classification task for Bengali, a low resource language.
- We deal with the problem by experimenting with several strategies that include data augmentation with synthetic samples, ensembling of deep learning models, applying focal loss, outlier detection technique, data resampling, and hidden feature extraction.
- We perform these strategies as mentioned above on three different datasets which contain critical class imbalance.
- Finally, we analyze the performance of each of those techniques thoroughly based on the three text classification tasks and reach more than **90%** of the minority-f1 score for each of the three datasets.

The remaining parts of the paper are organized as follows: Related Works are presented in Section 2, followed by Section 3, where detailed methodologies of the manifold techniques that we have undertaken in this work are illustrated. Next, the experimental settings, including dataset description, preprocessing as well as model implementations are described in Section 4. We then thoroughly discuss and compare the performance of those

techniques by analyzing the experimental results in Section 5. Finally, we conclude this work with future directions in Section 6.

2 RELATED WORK

Problems related to class imbalance have been prevalent for a while in machine learning, as real-world data hardly comes with balanced distributions for each class [Kotsiantis et al. 2006; Oksuz et al. 2020]. Not only limited to Natural Language Processing [Rumshisky et al. 2016; Tomanek and Hahn 2009], but this problem is also predominant in other domains including Object Detection for Image Domain [Oksuz et al. 2020]. Various techniques to handle class imbalance have been proposed in the literature [Liu et al. 2009]. Li et al. [2012] focused on the imbalanced class distribution scenario for sentiment classification using active learning. Besides, the skewed class distribution problem is addressed by Li and Nenkova [2014] in implicit discourse relation recognition. In addition, Seiffert et al. [2010] presented a new hybrid sampling/boosting algorithm, called RUSBoost, for learning from skewed training data. Apart from that, Zhou and Liu [2006] studies the effect of sampling and threshold-moving in training cost-sensitive neural networks in order to address the class imbalance problem. A prime example of class imbalance in the NLP domain is fake-news detection [Shu et al. 2017; Zhou et al. 2019]. For the English language, there have been several breakthroughs in this problem set with the aid of Machine Learning and Deep Learning techniques, such as using n-grams [Ahmed et al. 2017], Naive Bayes Classifier [Granik and Mesyura 2017], event adversarial network [Wang et al. 2018b], explainable fake news detection using sentence-comment co-attention sub-network [Shu et al. 2019], weakly supervised learning [Helmstetter and Paulheim 2018], geometric deep learning [Monti et al. 2019]. Another salient text classification task is sentiment analysis [Tang et al. 2014] [Pang et al. 2002] which refers to classifying positive and negative reviews, comments, and messages. In most of the sentiment classification datasets, the negative samples are found less in number which turns it into a class imbalance issue as well. Besides English, sentiment analysis has also been done in other languages Badaro et al. [2014] Badaro et al. [2019].

Since most of the NLP tasks are language-specific, class-imbalance can not be considered a generic problem for all languages. Many of the approaches we followed in this work may perform either well or poorly for different languages. For instance, the performance of the synthesis by character/n-gram/word and embedding generation approaches highly depends on the size and complexity of the alphabet of that language along with its syntax and semantics. Apart from that, during the fundamental preprocessing tasks in NLP, i.e., lemmatization, stemming, and stopword removal we can not be unaware of the language specificity. However, there are some approaches, like resampling, outlier detection, and ensembling which can be applied in other domains of class imbalance tasks and for other languages as well, but due to some unavoidable language-specific routines, as mentioned earlier, they may not necessarily perform in a similar manner in all such scenarios. For example, we will see in section 5 that despite being a generally well performant technique, outlier detection derives the worst result among all the approaches in our work. **The primary insight behind our work was that the class imbalance problem has never been addressed in the Bengali language for any NLP task**. Hence we attempt to tackle this problem for the text classification task by experimenting with a bunch of techniques and evaluate their performance in reducing the impact of class imbalance.

3 METHODOLOGY

This section discusses all of the techniques and methodologies that we utilized in this work to address the class-imbalance issue in Bangla text classification.

3.1 Dataset Augmentation

To deal with the scarcity of the minority class samples, we experiment with a couple of data augmentation techniques that include several text generation schemes such as next word prediction, next character prediction, and embedding generation schemes using Generative Adversarial Network, Variational Autoencoder, and LSTM Autoencoder.

3.1.1 Text generation by next word prediction. Predicting the word which will come next in a context is one of the most fundamental tasks in NLP [Ganai and Khursheed 2019]. We utilize this technique to generate new text samples. Given a seed text in the first place, we repeatedly generate the subsequent word in the context, supposing that the immediate previous word is predicted correctly. For this purpose, we train a deep BiLSTM architecture with the training samples from the minority class and reach an accuracy of around 70% for each of the three datasets. The prevailing prediction error works as an additional noise to the newly generated texts. We synthesize new fake samples of the same amount as the original fake samples so as to double the total number of samples in the minority class.

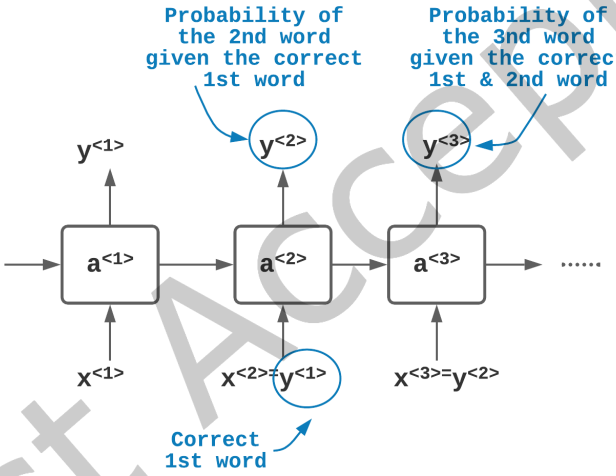

Fig. 1. Next word prediction

3.1.2 Text generation by next character prediction. Next word prediction is not a feasible option concerning memory usage when the vocabulary size of a corpus is very large. Instead, we can go for the next character prediction scheme in this case, as the total number of unique characters is always much lower than the total number of unique words in a dataset. Although the vocabulary size is not a concern for our respect, we also apply this technique to generate new text samples. Just like the next word prediction task, here we deploy an LSTM model and double the number of training samples of the minority class to use them for training the classifier models.

3.1.3 Text generation by Replacing with similar words. Common vector representation techniques of words such as Word2vec [Goldberg and Levy 2014] and GloVe [Pennington et al. 2014] have an exclusive application in finding words with the closest meaning or context from their cosine similarities. Our idea is to replace a bunch

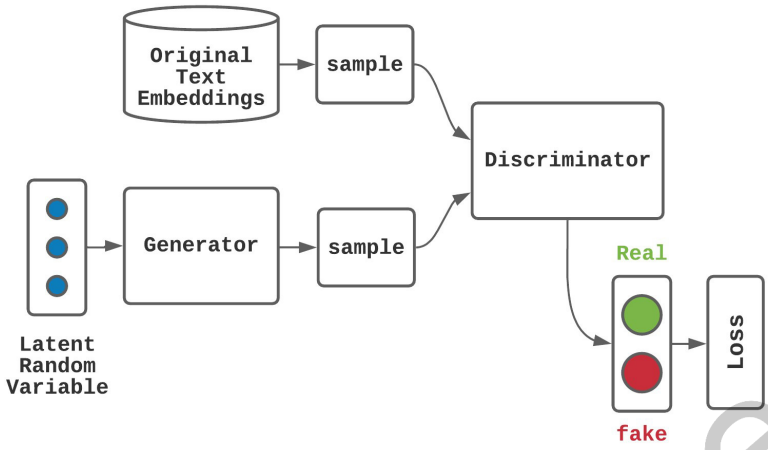

Fig. 2. Basic GAN mechanism for embedding synthesis

of words in the text with their closest counterparts, i.e., similar words in order to generate new text samples. In our experiment, we train **Word2vec embeddings** with the three text corpus respectively and utilize it for finding similar words for a given word. We perform our experimentation by replacing one-third and half of the total words in a text, respectively in order to synthesize new samples. In this manner also, we synthesize new minority class samples of the same amount as the original ones, doubling the total quantity.

3.1.4 Text embedding generation. Apart from the text generation methods mentioned above, we extend our idea of synthesis to embedding generation. As we can feed the text embeddings [Levy and Goldberg 2014] to the deep learning classifiers, we thought of synthesizing the embedding vectors instead of the raw texts and see whether this approach leads to an improvement in balancing the train set. For this purpose, we utilize three deep learning generative models, i.e. Variational Autoencoder [Semeniuta et al. 2017], Generative Adversarial Network [Li et al. 2018], and LSTM Autoencoder [Tang et al. 2018]. The first two models are traditionally used in computer vision for generating image samples [Bao et al. 2017; Brock et al. 2018], and the last one is used for sequence reconstruction.
Our idea is to train these models with the text embeddings occupied from the training samples and generate a new set of embedding vectors that correspond to the minority class.

3.2 Minimizing Focal Loss

Due to the massive imbalance in the three datasets, an alternative approach may be to apply a different loss function instead of binary cross-entropy. With that in mind, we train the four deep learning classifiers using **focal loss**. Proposed by Lin et al. [2017], the focal loss was subsequently used in a wide variety of tasks in the Image domain [Trichet and Bremond 2018; Wang et al. 2018a; Yang et al. 2018]. This loss function was designed to deal with the huge class imbalance for Dense Object Detection. The primary objective of this custom loss function is to downplay the importance of well-classified examples, thus making it easier and more efficient for the neural network to learn the hard examples. Here, we briefly describe, per sample, the key difference in focal loss compared to binary cross-entropy.
Assuming y to be the true labels, and $\hat{y}$ to be predicted labels, breaking down binary cross-entropy to per sample,

we get:

$$BCE(y, \hat{y}) = \begin{cases} -log(\hat{y}) & \text{if } y = 1 \\ -log(1 - \hat{y}) & \text{else} \end{cases} \quad (1)$$

Breaking focal loss down to per sample, we get:

$$FL(y, \hat{y}) = \begin{cases} -\alpha(1 - \hat{y})^{\gamma} log(\hat{y}) & \text{if } y = 1 \\ -(1 - \alpha)\hat{y}^{\gamma} log(1 - \hat{y}) & \text{else} \end{cases} \quad (2)$$

Equation 2 is also known as *weighted* focal loss, since α becomes the weighting factor in the case of binary classification, balancing the importance of positive and negative labels. The focusing parameter γ controls the importance of misclassified examples. It adjusts the rate of down-weighting the easy samples. The higher its value, the less amount of loss will be propagated from the *easy* samples. We train the deep neural network models by setting γ to 2 and α to be the ratio of the number of samples in the minority and the number of samples in the majority.

3.3 Ensembling of Deep Learning Models

The Ensembling strategy aims to build a predictive model by integrating multiple models. It is well-known that ensembling of various models helps improve the performance of separate usage of individual models. The two common approaches for ensembling include bagging [Breiman 1996a] and boosting [Breiman 1996b]. Throughout the years, various boosting algorithms have been introduced and successfully employed in the field of machine learning, including AdaBoost [Freund and Schapire 1999], XGBoost [Chen and Guestrin 2016], LightGBM [Ke et al. 2017] and CatBoost [Prokhorenkova et al. 2018]. Besides, employing ensembling for imbalanced data has been prevalent for a while [Galar et al. 2012b; Nikulin et al. 2009; Sun et al. 2015]. By utilizing the strategy of *horizontal partitioning* [Chawla et al. 2004], training the models on partitions of data samples and taking an ensemble amongst them can augment the overall performance. Hence, we adopt this approach by using a subset of the majority-class samples along with all of the minority-class samples to train each of the four deep-learning models with simpler architectures.

Let T and F be our majority samples and minority samples, respectively. Also, let n and k be the number of models used for training and consensus (while prediction), respectively. To train the model M_i ($\forall_{i \in n}$), we use a subset of majority-samples T_i (where $|T_i| = \lfloor |T|/n \rfloor$) and all of minority samples F. For prediction, we employ a consensus strategy which is a variant of majority-voting [Brown and Kuncheva 2010], depending on the value k and λ (global threshold for prediction of each model M_i). Final prediction using consensus strategy is described in the algorithm 1. For our experimentation, we varied $k \in [n/2, n]$ and $\lambda \in [0.1, 0.35, 0.5]$.

3.4 Resampling

Data-level solutions are widely employed due to the benefit of being independent of the classification algorithms. The main idea of data-level solutions is resampling which is a simple yet effective solution to the class imbalance problems [Branco et al. 2015]. Random Over-Sampling (ROS) and Random Under-Sampling (RUS) are two of the most straightforward approaches of resampling. Here, we employ both oversampling and random undersampling to our class-imbalance problem.

3.4.1 Random Under-Sampling of majority class examples. Undersampling [Galar et al. 2012a] of majority class samples is a very popular method that aims to relatively balance the class distribution by randomly eliminating a chunk of majority class examples. This approach has the risk of losing valuable information, especially if a significant proportion of samples is removed [He and Ma 2013]. In order to observe the performance of RUS, we have varied the ratios of majority:minority samples by randomly eliminating majority-class samples while reserving all the minority class examples for training.

Algorithm 1: Ensembling Prediction

Inputs: x: Sample for prediction
$M_1, M_2, ..., M_n$: n trained models
k: Consensus Number
λ: Global threshold
Output: y: Final Binary Prediction (T/F)
for $i \in n$ **do**
  //CF_i shows confidence of F class of sample x for model M_i where $CF_i \in [0, 1]$
  $CF_i \leftarrow Predict(x)$;
  if $CF_i \geq \lambda$ **then** $Y_i \leftarrow F$;
  else $Y_i \leftarrow T$;
end
if $(\sum_{i=1}^{n} |Y_i == F|) \geq k$ **then** $y \leftarrow F$;
else $y \leftarrow T$;

3.4.2 Over-Sampling of minority class examples. Oversampling of minority classes aims to present a higher representation of the minority class examples in the training set. Random Over-Sampling (ROS) also happens to be one of the most popular methods of resampling. Randomly replicating copies may make the models more prone to overfitting [Branco et al. 2015]. In this problem, however, instead of randomly replicating minority class examples, we replicate those examples which have a good average dissimilarity with all of the majority class examples with respect to their text embeddings.

Let minority-class/fake-news samples and majority-class/true-news samples be represented by F and T respectively. Let $s_{i,j}$ denote the cosine-similarity obtained between the text embeddings of F_i minority class sample and T_j majority class sample. Next, we compute Avg_i using equation 3.

$$Avg_i = \frac{\sum_{j=0}^{|T|} s_{i,j}}{|T|} \tag{3}$$

Then, we only choose those minority-class samples to oversample for whom $Avg_i \geq \mu$. This implies that the overall average cosine-similarity between the F_i minority-class sample and all majority-class samples lies within $[-1, \mu]$. We have empirically determined $\mu = -0.45$.

3.5 Outlier Detection

Since the minority samples are very scarce in the dataset we are working with, one idea is to treat them as outliers [Aggarwal 2015]. Then, we can focus on learning the pattern of our regular data (e.g., authentic samples for the fake news dataset) and consider any major deviation from the regular pattern as an outlier. To implement this outlier detection scheme, we train an LSTM autoencoder model with the training samples of the majority class standalone. Later, we perform reconstruction on our test set with a view to differentiating the minority samples from the major ones, as they would supposedly lead to a higher reconstruction loss.

3.6 Exploring the feature-space of a hidden layer of the DL model

Apart from directly using the BiLSTM model as a classifier, we also use its hidden layer's feature space to extract the embeddings generated from the 32-Dimension layer. Then we feed this into linear and tree-based classifiers in the hopes that they would be able to classify these compressed features (embeddings) better. For experimentation, we have used state-of-the-art classifiers including Support Vector Classifier of SVM [Cortes and Vapnik 1995],

Dataset	Label	Total Samples	Mean Sentence Count	Mean Word Count	Mean Character Count
Fake News [Hossain et al. 2020]	Fake	1649	23	279	1437
	Authentic	48678	21	271	1479
Sentiment Analysis [Sazzed 2020]	Negative	1807	3	17	146
	Positive	8500	3	16	139
Music Lyrics [Kaggle]	Rabindra	576	23	204	1444
	Others	3511	19	187	1292

Table 1. Corpus details

Bernoulli Naive Bayes [Friedman et al. 1997] for linear models and RandomForest [Breiman 2001], XGBoost [Chen and Guestrin 2016], LightGBM [Ke et al. 2017] for tree-based models.

4 EXPERIMENTAL SETUP

4.1 Dataset

We utilize three benchmark datasets in this work. The idea behind experimenting with multiple datasets is to ensure domain variety and reach a convincing conclusion, which are well-performing techniques for tackling the class imbalance issue in Bangla text classification. The first dataset is *Bangla fake news dataset* developed by Hossain et al. [2020]. It contains around 50*K* annotated news distributed in two classes, i.e., fake and authentic. The sample ratio of the two classes in the original dataset is 97.41%-2.59%. In order to reduce this massive imbalance a little bit, We added 350 new fake samples to that corpus which we have manually curated from Facebook, notification click baits, and some disputed news portals. Consequently, the contribution from the true class measures to 96.73%, and for the fake class, it is 3.27%. That clearly shows the existing imbalance in this dataset. Although the original dataset has eight attributes, including various metadata (e.g., source, domain, headline), we only consider two of them, i.e., *Article* and *Label* as our objective in this work exclusively focuses on **content based classification**.

The second dataset we used in this work is *Bangla sentiment analysis classification benchmark dataset* prepared by Sazzed [2020]. This dataset contains 3307 Negative reviews and 8500 Positive reviews collected and manually annotated from Youtube Bengali drama. To increase the class imbalance in that corpus, we randomly removed 1.5*K* negative reviews. It made the sample ratio 85%-15%. The reason behind choosing this dataset is that here the text samples are comparatively shorter in length unlike the fake news dataset. So, we wanted to observe how our proposed techniques perform in reducing the impact of class imbalance for such short text classification tasks.

The third dataset is *Bangla Song Lyrics* from Kaggle. This dataset consists of 4087 Bangla song lyrics from 21 different genres. We adopt an *one versus all* classification approach where we considered Rabindra songs versus all other genres. In the original dataset, 19% of the lyrics are of Rabindra songs. To increase the class imbalance a bit more, we randomly removed 200 samples from that minority class. It made the sample ratio 86%-14%. Song lyrics are somewhat different from the regular text data. They have got their identical sentence pattern and usage of words. We were particularly interested to see how the synthesis techniques work for such text data. Also, this dataset has comparatively fewer samples in total than the other two datasets. Hence, it will help us observe how the proposed models in this work perform in reducing the impact of class imbalance for such small datasets. Table 1 shows the details of the three datasets that we used in this work. During the experiments, we split our corpus into 75% train, 10% dev, and 15% test.

4.2 Text Preprocessing

A couple of essential preprocessing steps need to be done before extracting some numerical representation from the text, which we will feed to the classifier. First of all, punctuation marks, digits, English literals, and special characters are removed. Then each document is tokenized using blank space, which results in a list of words. Furthermore, we have done some lemmatizations and excluded the stop words from that list. We have collected 430 Bangla stopwords that hold no importance in classifying the documents rather than increasing haziness during our models' learning session. Since the sentiment text samples are short in length, they became shorter after the stopword removal phase, and we need to remove some of the shortest text samples from the training set.

After the aforementioned preprocessing steps, we convert each document into a set of filtered words. Then we one hot encode these words and feed this numerical representation of the documents into the learning models.

4.3 Training Models

We train three deep learning models as the benchmark classifiers in our work, i.e., LSTMs, GRUs, and CNNs.

LSTMs and GRUs: These are the two variants of RNN [Mikolov et al. 2011] that have ad-hoc applications in text classification tasks. They are widely used for their âĂŸmemoryâĂŹ features that can efficiently capture sequential information. Apart from the classic LSTM [Hochreiter and Schmidhuber 1997], we also used Bidirectional LSTM [Schuster and Paliwal 1997] that puts two independent LSTM networks together so as to have both backward and forward information about the sequence at every time step. Besides, we also performed the classification with both classic and bidirectional GRUs, but only mentioned the results occupied from the latter, as classic GRU didnâĂŹt seem to perform well on this dataset.

For each model, we use five hidden layers with an embedding layer on top of them. We varied the number of corresponding recurrent units in each hidden layer from 32 to 256. Apart from that, we also experimented with different learning rates, batch sizes, number of epochs, and dropout regularization.

CNNs: Traditionally, CNNs are thought to be specialized for processing a grid of values such as image data [Krizhevsky et al. 2012]. The idea behind using CNNs in NLP [Kim 2014] is to make use of their ability to extract features. In this work, the CNN model is constructed by stacking four 1D Convolution layers, interleaved with four 1D Max-pooling layers, each having a pool size of 2 and one 1D Global average pooling layer followed by them. For each convolution layer, we adjusted the number of filters up to 512 and window length up to 7 based on the model's performance.

For all of these deep learning models, we used adam optimizer and binary-cross-entropy loss function with sigmoid activation. During training, we made use of a **weighted version of binary cross entropy** by assigning class weights of inverse proportions to each label (F/T). Let $w(F)$ and $w(T)$ be the weights assigned to labels F and T respectively. We set $w(F) = \frac{|T|}{|T|+|F|}$ and $w(T) = \frac{|F|}{|T|+|F|}$

4.4 Dataset Augmentation models

For the **next word prediction** task, we use a BiLSTM model that consists of two bidirectional LSTM, two vanilla LSTM, and two fully connected layers with an embedding layer stacked on top of them. Here, each of the BiLSTM layers has up to 60 recurrent units, and the LSTM layers have up to 200 units. We deal with the vocabulary size of $23K$, $12K$, and $8K$ respectively for the three datasets and use adam optimizer with a learning rate of 0.01. It derives around 70% of prediction accuracy. Later we take seed text of varying length from each of the minority class documents based on their size and synthesize new texts by predicting next words. Since the sentiment corpus texts are short in length, they showed somewhat lower prediction accuracy (66%). Also, their seed text length was smaller than the other two which might have an impact on the quality of their synthetic samples.

We use almost the same model architecture as the next word prediction task for **predicting next character**. The

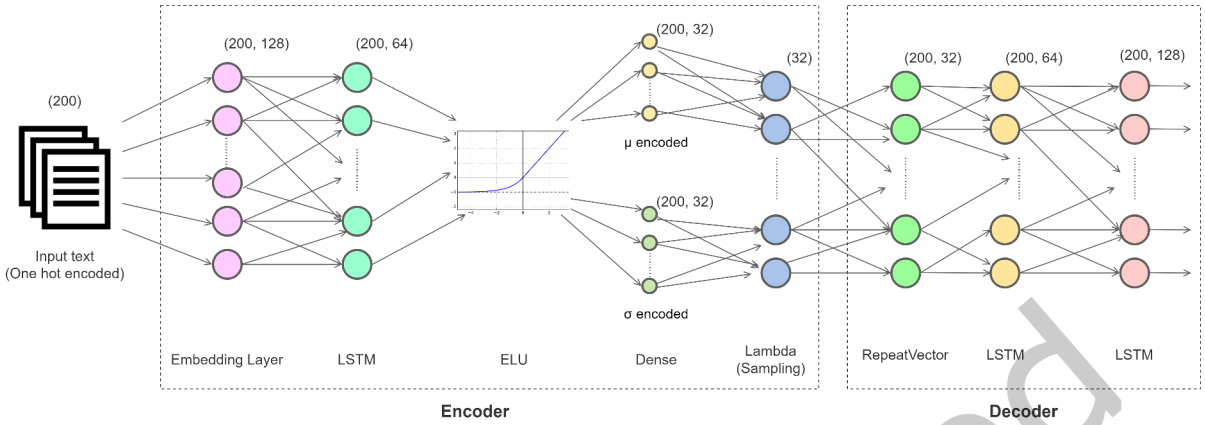

Fig. 3. The VAE architecture

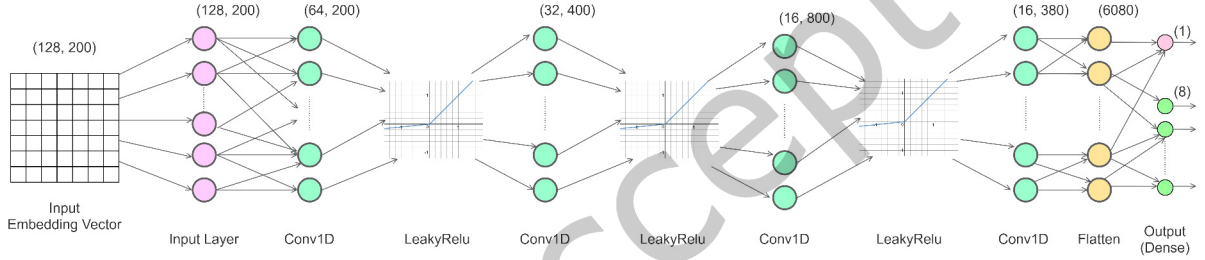

Fig. 4. The discriminator model of AC-GAN

only difference is that here we replace the bidirectional layers with LSTMs where each of them has up to 512 recurrent cells. After the preprocessing step, it yields a vocabulary of 368 unique characters for the fake news dataset, which is much less than the number of unique words in that corpus. Training the model, therefore, derives around 65% of accuracy in predicting the next character, which we utilize for synthesizing new text samples subsequently.

4.5 Embedding generation models

Word embeddings provide a dense (vector) representation of words and their relative meanings. We adopted an *Embedding layer* approach where we let the embeddings be learned jointly as a part of our deep learning models. For this purpose, at first, each preprocessed word is one-hot encoded and then fed to the embedding layer. Output of this layer is a vector of a certain length for each of the words. We specified the embedding dimension (D) to 128 and the number of words (W) for the embedding vectors to 200, 12, and 150, respectively, for fake news, sentiment analysis, and song lyrics dataset. Hence, each document of the minority class is represented as a $[D, W]$dimensional vector. Later we utilize them for synthesizing new embeddings by training three deep generative models, i.e. variational autoencoder, Generative Adversarial Network, and LSTM autoencoder.

In this work, we tried to replicate the VAE model proposed by Bowman et al. [2016]. It is composed of one recurrent LSTM encoder network, a standard fully connected network for the variational inference, and two

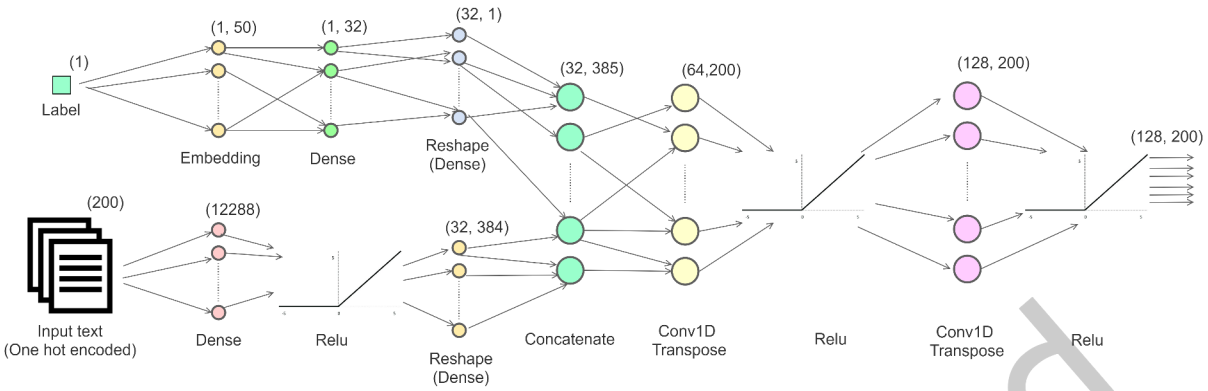

Fig. 5. The generator model of AC-GAN

Recurrent LSTM decoder networks. All the network is trained simultaneously via SGD. Figure 3 shows the detailed architecture of the variational autoencoder model used for fake news embedding generation. Moreover, we choose to train the auxiliary classifier generative adversarial network (AC-GAN) for the embedding synthesis task. The architecture of the discriminator and the generator of the AC-GAN model for fake news embedding generation are separately demonstrated in Figure 4 and 5. Apart from that, we also utilize an LSTM autoencoder model for this purpose, which consists of a basic encoder-decoder architecture made of six LSTMs, one repeat vector, and one time distributed layer. Mean squared error is considered for the loss calculation along with adam optimizer, and the number of recurrent units per LSTM layer is tuned empirically up to 512.

5 EXPERIMENTAL RESULTS

5.1 Metrics

There are quite a few metrics to choose from in order to evaluate the decisions of a classifier [Ferri et al. 2009]. Having a substantial difference in distributions of majority and minority class labels, evaluation using traditional metrics such as *precision*, *recall* and *Micro-F1* scores that include both classes may not reflect the best performance. Moreover, **all the experimental results in this work derive 99-100% of precision, recall, and f1-score with respect to the majority class.** Hence, we show the performance of each classification model throughout our experiments **only for the minority class** using the metrics mentioned above.

5.2 Weighted binary cross entropy

For the original corpus, we can observe from Table 2 that CNN model outperforms all the deep models in terms of both f1 and recall for each of the three datasets.

Using text synthesis with next word prediction (TGW), the f1 scores of all the deep models improve, suggesting that using additional synthetic text corpus via TGW improves the performance of the classification task. In this case, both CNN and Bidirectional LSTM models achieve 88% f1 score for the fake news dataset. Using text synthesis with predicting next character (TGC), just like synthesis using TGW, the f1 scores improve in comparison to the original corpus, with CNN model achieving the best f1 score (**90%**) for the fake news dataset. Using text synthesis via similar words replacement(SWR), almost all models have a superior performance in regards to f1 with an improvement in recall compared to the original corpus. However, it did not perform as well as the other two synthesis techniques for this dataset.

Dataset	Model	P	R	F1	P	R	F1	P	R	F1	P	R	F1
		Original Corpus			Original Corpus + TGW			Original Corpus + TGC			Original Corpus + SWR		
Fake News	BiLSTM	0.94	0.73	0.82	0.95	0.81	0.88	0.89	**0.85**	0.87	0.89	0.77	0.83
	CNN	0.93	0.77	0.85	0.93	0.84	0.88	**0.95**	0.84	**0.90**	0.94	0.81	0.87
	BiGRU	0.95	0.65	0.77	0.91	0.76	0.85	0.94	0.73	0.82	0.89	0.70	0.78
	LSTM	0.87	0.74	0.79	0.95	0.73	0.85	0.90	0.78	0.83	0.88	0.75	0.81
Sentiment	BiLSTM	0.94	0.83	0.88	0.90	0.82	0.86	0.92	0.84	0.88	**0.95**	**0.85**	0.90
	CNN	**0.97**	0.85	0.90	0.94	0.84	0.89	0.92	0.84	0.88	0.96	**0.87**	**0.91**
	BiGRU	0.95	0.76	0.84	0.92	0.79	0.83	0.88	0.75	0.81	0.92	0.81	0.85
	LSTM	0.90	0.82	0.86	0.94	0.80	0.85	0.89	0.82	0.85	0.91	0.81	0.86
Song Lyrics	BiLSTM	0.88	0.78	0.83	**0.90**	**0.81**	**0.87**	0.89	0.81	0.86	0.88	0.76	0.82
	CNN	0.90	0.80	0.85	0.92	**0.84**	0.88	**0.94**	0.83	**0.89**	0.92	0.81	0.86
	BiGRU	0.93	0.76	0.84	0.91	0.79	0.85	0.91	0.81	0.86	0.89	0.77	0.83
	LSTM	0.86	0.79	0.82	0.88	0.81	0.84	0.90	0.81	0.85	0.88	0.78	0.83

Table 2. DL Models with binary cross entropy

There are some noticeable properties found in the results for the sentiment analysis dataset. Since this dataset has a less severe class imbalance, its overall scores are better than the fake news dataset. However, as we mentioned in section 4.4, due to the short seed texts, synthesis by word and character prediction did not add much improvement to the performance in this case. Rather, we can see a better set of scores in Table 2 for synthesis by similar word replacement than the other two techniques. Here the best score (91% f1) is derived by CNN with SWR. Apart from that, performance of these three synthesis techniques for the song lyrics dataset is quite similar to the fake news dataset. However, due to its smaller size, the overall scores for this dataset turn out to be slightly worse (best f1 of 89%) than sentiment classification. Analyzing the results of the synthesis techniques on different datasets, it can be summarized that prediction-based text synthesis techniques require a considerable amount of text length. In contrast, the synthesis by word replacement works fine even with short texts.

5.3 Weighted focal loss

Compared to the weighted binary cross-entropy in Table 2, all the deep models showed a superior performance while trained to minimize focal loss. Consequently, **we achieve the best scores in this work for each of the three datasets by applying focal loss function with text synthesis techniques.** As shown in Table 3, CNN model along with TGW synthesis attains the best result (91% f1) for fake news classification, while the same classifier derives the best f1 of 94% with SWR for sentiment classification. Like the fake news dataset, CNN, along with TGW synthesis, derives the best result (91% f1) for song lyrics classification as well. Among the other models, bidirectional GRU also shows a promising performance for sentiment and song lyrics classification using focal loss, having attained the best f1 of 88% for both cases, whereas the best possible f1 for this model using binary cross-entropy was a mere 85%.

The striking improvements in both f1 and recall for all of the deep learning models using focal loss instead of binary cross-entropy show that focal loss is a better custom loss to classify the samples on a significant amount of class imbalance. Moreover, with the addition of synthesized text (TGW, TGC or SWR), the performance plateau using focal loss suggests that it gets even better.

Dataset	Model	P	R	F1	P	R	F1	P	R	F1	P	R	F1
		Original Corpus			Original Corpus + TGW			Original Corpus + TGC			Original Corpus + SWR		
Fake News	BiLSTM	0.91	0.75	0.83	0.94	0.76	0.84	**0.95**	0.74	0.83	0.90	0.77	0.83
	CNN	**0.97**	0.80	0.88	0.97	**0.85**	**0.91**	0.96	0.82	0.90	0.91	0.82	0.87
	BiGRU	0.89	0.80	0.84	0.94	0.73	0.82	0.93	0.75	0.83	0.97	0.73	0.83
	LSTM	0.88	0.79	0.83	0.93	0.75	0.83	0.88	0.76	0.82	0.95	0.73	0.83
Sentiment	BiLSTM	0.95	0.85	0.90	**0.96**	**0.87**	0.91	0.94	0.87	0.90	0.92	0.83	0.87
	CNN	**0.97**	0.87	0.92	0.96	0.87	0.92	0.94	0.86	0.91	0.97	**0.92**	**0.94**
	BiGRU	0.96	0.79	0.86	0.96	0.80	0.88	0.89	0.79	0.83	0.94	0.83	0.88
	LSTM	0.92	0.84	0.88	0.95	0.82	0.87	0.90	0.84	0.86	0.92	0.82	0.86
Song Lyrics	BiLSTM	0.90	0.81	0.86	**0.92**	0.82	**0.88**	0.91	**0.83**	0.88	0.90	0.77	0.83
	CNN	0.92	0.81	0.86	**0.96**	**0.86**	**0.91**	0.94	0.85	0.90	0.93	0.82	0.88
	BiGRU	0.95	0.79	0.86	0.93	0.81	0.87	0.93	0.83	0.88	0.90	0.79	0.85
	LSTM	0.89	0.81	0.84	0.90	0.82	0.85	0.93	0.82	0.87	0.90	0.80	0.85

Table 3. Deep models with Focal Loss

Dataset	Model	P	R	F1	P	R	F1	P	R	F1	P	R	F1
		Original Corpus			Original Corpus + TGW			Original Corpus + TGC			Original Corpus + SWR		
Fake News	BiLSTM	0.91	0.76	0.83	0.84	0.81	0.82	0.91	0.82	0.87	0.88	0.79	0.84
	CNN	0.93	0.82	0.88	**0.96**	0.82	**0.90**	0.86	**0.83**	0.85	0.90	0.82	0.87
	BiGRU	0.83	0.78	0.81	0.86	0.78	0.82	0.93	0.77	0.84	0.84	0.78	0.81
	LSTM	0.87	0.75	0.81	0.88	0.77	0.82	0.86	0.80	0.83	0.86	0.78	0.82
Sentiment Analysis	BiLSTM	0.95	0.83	0.89	**0.95**	0.86	0.92	0.92	0.85	0.89	0.92	0.82	**0.87**
	CNN	0.95	0.86	0.90	**0.97**	**0.88**	**0.92**	0.95	0.84	0.90	0.91	0.86	0.89
	BiGRU	0.97	0.78	0.85	0.93	0.82	0.87	0.96	0.81	0.88	0.88	0.77	0.82
	LSTM	0.90	0.84	0.87	0.95	0.81	0.86	0.91	0.82	0.86	0.91	0.82	0.86
Song Lyrics	BiLSTM	0.88	0.80	0.84	**0.91**	**0.82**	**0.89**	0.90	0.82	0.88	0.88	0.79	0.84
	CNN	0.92	0.81	0.87	0.93	0.86	0.90	**0.96**	**0.86**	**0.91**	0.92	0.83	0.87
	BiGRU	0.94	0.79	0.86	0.92	0.80	0.86	0.91	0.83	0.88	0.90	0.79	0.85
	LSTM	0.88	0.80	0.84	0.88	0.83	0.85	0.91	0.82	0.87	0.88	0.80	0.84

Table 4. Ensembling of smaller architecture Deep Models.

5.4 Ensembling

Observing from Table 4, for the original corpus, all the models showed a significant improvement using ensembling (with simpler architectures) with respect to both f1 and recall in comparison to using binary cross-entropy as observed in Table 2. Using ensembling, CNN model attained an impressive f1 of 88% for the fake news dataset using a configuration of n=5, k=5, $\lambda = 0.1$.

In TGW, the f1-scores are very much similar with deep models using binary cross-entropy in Table 2, however, the BiLSTM model achieves a degrade in performance using ensembling, having attained 82% f1 as opposed to

Dataset	Model	P	R	F1	P	R	F1	P	R	F1
		Original Corpus + EG_LSTM-AE			Original Corpus + EG_AC-GAN			Original Corpus + EG_VAE		
Fake News	BiLSTM	0.87	0.79	0.83	0.89	0.81	0.85	0.89	0.77	0.83
	CNN	0.89	0.80	0.85	0.84	0.79	0.83	0.88	0.82	0.85
	BiGRU	0.85	0.81	0.83	0.86	0.80	0.83	0.84	0.78	0.81
	LSTM	**0.91**	0.78	0.84	0.91	**0.84**	**0.88**	0.87	0.81	0.84
Song Lyrics	BiLSTM	0.86	0.81	0.84	**0.89**	**0.82**	**0.87**	0.87	0.81	0.85
	CNN	0.90	0.82	0.86	0.90	0.86	0.89	**0.94**	**0.86**	**0.92**
	BiGRU	0.91	0.79	0.85	0.90	0.79	0.84	0.92	0.82	0.87
	LSTM	0.84	0.80	0.81	0.89	0.81	0.85	0.90	0.82	0.86

Table 5. DL Models on Synthesized Embeddings

88% for the fake news dataset using binary cross-entropy. Additionally, ensembling of simpler architecture CNN models achieved 90% f1 for the fake news dataset and 92% for the sentiment analysis dataset, which is the best possible values attained using ensembling of any model, configuration.
In TGC, just like in TGW, the f1-scores for ensembling in Table 4 are also very much similar to those in Table 2. Although the CNN model did not achieve performance improvement for the first two datasets, in this case, there is, however, a boost in f1 for the song lyrics dataset, increasing from 87% to 91%. On the other hand, using SWR, as observed in Table 4, all deep classifier models attained a superior performance with respect to both f1 and recall compared to using binary cross-entropy.

These results suggest that ensembling of smaller architecture deep models obtained by training a subset of majority label samples and all minority label samples cause a substantial improvement, especially in the original corpus, as opposed to training single models with weighted binary cross-entropy.

5.5 Embedding Generation Techniques

This technique was performed using a cascading models approach, where M_1, a Bidirectional LSTM model, was trained on original corpus embeddings, and M_2, a deep classifier model, was trained on both original corpus embeddings as well as the embeddings obtained by using LSTM AE, LSTM VAE, AC-GAN on M_1 embeddings. The performance variation of M_2 is shown in Table 5. Compared to deep learning models trained on original corpus using binary cross-entropy as in Table 2, we can see that almost all the classification models M_2 achieved superior performance with respect to both f1 and recall when trained on additional synthetic embeddings. Especially, the LSTM and Bi-LSTM model with AC-GAN embeddings achieves more than 87% of f1 score as shown in Table 5. However, the embedding vectors have a very small number of words (12) for the sentiment analysis dataset. As a result, each of the three generative models underfits while trained with these embeddings producing some low-quality synthetic samples. Subsequently, these synthetic embeddings did not add any improvement to the classification performance; rather, it resulted in less than 40% f1 for all of the deep learning models. Hence, we did not include these results in Table 5.

5.6 Outlier Detection

Treating minority class samples as outliers using LSTM Autoencoder does not yield good scores for any of the classification tasks as observed in Table 6. It may indicate the limitations on the embeddings generated by the first model M_1, as if it produces embeddings which is not good enough for the outlier strategy to perform well.

Threshold	P	R	F1	P	R	F1	P	R	F1
	Fake News			Sentiment Analysis			Song Lyrics		
0.0007	0.64	0.81	0.70	0.67	0.83	0.73	0.65	0.78	0.69
0.001	0.70	0.67	0.68	0.72	0.70	0.71	0.68	0.70	0.69
0.004	0.89	0.48	0.62	0.86	0.61	0.71	0.84	0.43	0.57

Table 6. Outlier Detection

Model	P	R	F1	P	R	F1	P	R	F1
	Fake News			Sentiment Analysis			Song Lyrics		
BiLSTM	0.96	0.70	0.81	0.93	0.75	0.83	0.91	0.75	0.82
CNN	0.94	0.68	0.79	0.92	0.75	0.83	0.88	0.74	0.80
BiGRU	0.96	0.72	0.82	0.93	0.77	0.84	0.89	0.74	0.81
LSTM	0.81	0.78	0.79	0.84	0.79	0.82	0.81	0.73	0.77

Table 7. Performance of oversampling of minority class samples using $\mu = -0.45$

5.7 Resampling Techniques

Here, we explore Oversampling and Random Undersampling (RUS) respectively described in subsections 3.4.2 and 3.4.1.

Oversampling: Using the strategy of oversampling, we observe from Table 7 that apart from Bidirectional GRU model, there is no significant improvement compared with training on original corpus as observed in Table 2.

RUS: After comparing RUS performances from Table 8 with the original corpus of Table 2, we observe that only using $15K$ majority samples and all minority samples, there is a significant improvement in both f1 and recall for the fake news dataset. However, all other combinations of subsets of majority samples do not yield good results in this case. Apart from that, since both the sentiment classification and song lyrics datasets have smaller data samples, applying the undersampling strategy further worsened the overall results instead of improving.

5.8 Tree and Linear Models

Comparing the Bidirectional LSTM model in Table 2 with the tree models trained on 32-D feature space of this BiLSTM model as shown in Table 9, we observe that for the original corpus, RandomForest (RF), Bernoulli Naive Bayes (BNB), and LightGBM (LGBM) outperform the BiLSTM model for all three datasets in terms of f1 and recall.

For synthetic texts using TGW and TGC, the linear and tree models do not show superior performance compared to the BiLSTM model. However, they do show an improvement with respect to their original corpus performance, achieving an f1 score of 88% (Random Forest with TGW) for the fake news dataset and 89% (BernouliNB with

Dataset	Model	P	R	F1	P	R	F1	P	R	F1
		30k majority class samples			15k majority class samples			10k majority class samples		
Fake News	BiLSTM	0.82	0.70	0.75	0.86	0.75	0.80	0.66	0.76	0.71
	CNN	0.90	0.75	0.82	**0.92**	0.78	**0.85**	0.80	**0.79**	0.80
	LSTM	0.80	0.71	0.75	0.83	0.76	0.79	0.78	0.78	0.78
		6.5k majority class samples			5k majority class samples			3.5k majority class samples		
Sentiment	BiLSTM	0.95	0.86	0.91	0.88	0.79	0.83	0.84	0.75	0.80
	CNN	0.95	0.86	0.90	0.89	0.83	0.87	0.81	0.82	0.81
	LSTM	0.90	**0.86**	**0.89**	0.88	0.76	0.82	0.82	0.70	0.76
		3k majority class samples			2k majority class samples			1.5k majority class samples		
Song Lyrics	BiLSTM	**0.85**	**0.81**	**0.83**	0.83	0.78	0.80	0.81	0.72	0.76
	CNN	0.93	0.86	0.91	0.90	0.75	0.82	0.83	0.74	0.79
	LSTM	0.88	0.82	0.85	0.83	0.75	0.79	0.80	0.71	0.74

Table 8. Variation in performance of DL models after undersampling & keeping various subsets of majority samples (along with all minority samples)

Dataset	Model	P	R	F1	P	R	F1	P	R	F1
		Original Corpus			Original Corpus + TGW			Original Corpus + TGC		
Fake News	Random Forest	0.94	0.75	0.83	0.95	**0.82**	**0.88**	0.96	0.79	0.87
	SVC	0.97	0.65	0.78	0.97	0.71	0.82	0.95	0.73	0.83
	BernoulliNB	0.95	0.75	0.84	0.88	0.82	0.85	0.89	0.79	0.84
	XGBoost	0.96	0.66	0.78	**0.98**	0.71	0.82	0.95	0.72	0.82
	LightGBM	0.96	0.72	0.82	0.95	0.77	0.84	0.95	0.74	0.83
Sentiment	Random Forest	0.93	0.81	0.87	**0.95**	0.82	0.89	0.94	0.80	0.87
	SVC	0.90	0.77	0.83	0.92	0.78	0.84	0.90	0.78	0.83
	BernoulliNB	0.92	0.81	0.86	0.93	**0.83**	**0.89**	0.93	0.81	0.87
	XGBoost	0.89	0.76	0.82	0.92	0.79	0.84	0.92	0.78	0.83
	LightGBM	0.90	0.81	0.85	0.91	0.83	0.87	0.90	0.82	0.86
Song Lyrics	Random Forest	0.88	0.78	0.82	0.90	0.81	0.85	0.91	0.79	0.83
	SVC	0.90	0.74	0.82	0.90	0.78	0.83	0.89	0.79	0.83
	BernoulliNB	0.90	0.79	0.83	0.92	0.80	0.85	0.89	0.80	0.83
	XGBoost	0.87	0.75	0.80	0.89	0.78	0.82	0.91	0.77	0.82
	LightGBM	0.92	0.80	0.84	**0.93**	**0.82**	**0.86**	0.92	0.81	0.85

Table 9. Linear and Tree based models on 32-D feature space

TGW) for sentiment classification. Again, none of the classifier models show promising results using SWR; hence it is not included in Table 9.

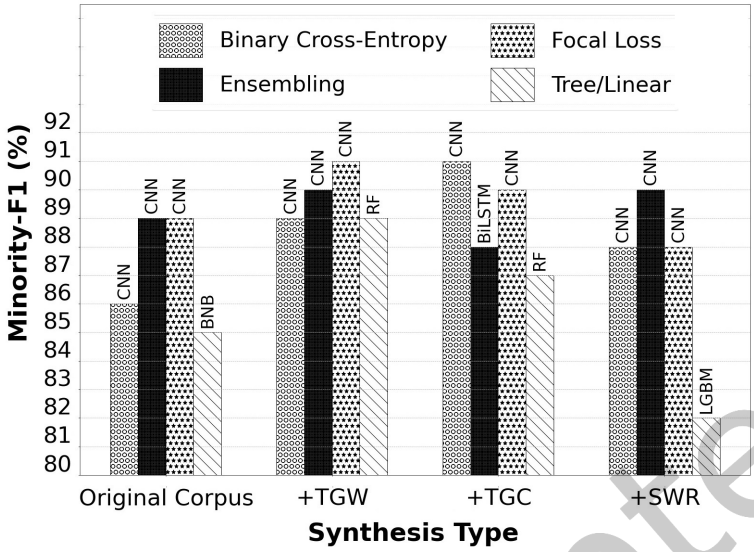

Fig. 6. Performances of the best models for fake news classification

5.9 Results Summary

In this work, we have experimented with three different datasets. All the techniques that we applied to tackle class imbalance did not perform equally for these datasets. From the experimental results we have just discussed, there are certain observations that we can summarize as follows:

The text synthesis techniques we applied in this work turn out to be well performant for all three datasets. TGW and TGC methods, however, seem to work good only for long texts, where SWR performs well for both short and long text data. In addition, the deep learning classifiers achieve better training to reduce the impact of class imbalance with focal loss function instead of binary-cross-entropy loss. In fact, **we obtained the best results (more than 90 % f1) for each of the three classification tasks by applying focal loss along with the text synthesis techniques**. Apart from that, ensembling of simple architectures added some improvement to the performance showing its ability to reduce the impact of class imbalance for each of the datasets. Although the embedding synthesis techniques derived some good scores, especially with the AC-GAN model, they did not hold good for the sentiment analysis dataset due to the small number of words per text. In almost all scenarios, CNN outperformed the other RNN based deep classifiers with respect to their minority f1. On the other hand, outlier detection and resampling strategies did not show any potential to reduce the impact of class imbalance in Bangla text classification. Finally, among the state-of-the-art machine learning classifiers, random forest, Bernoulli naive Bayes, and light gradient boosting machine showed some noticeable improvement in the performance, particularly with synthetic text samples for all three datasets. Since the fake news dataset has the highest class imbalance among the three, we demonstrate the summary of the best-performing methods for this particular corpus in Figure 6.

6 CONCLUSION

In this work, we used three benchmark datasets as the use case for addressing the class imbalance issue in Bangla text classification. We have experimented with various techniques to tackle the class-imbalance problem. Firstly, we performed a few synthesis techniques on both raw texts and embeddings for dataset augmentation. Next, we trained four deep learning models, i.e., BiLSTM, CNN, BiGRU, and LSTM, using weighted binary cross-entropy

and weighted focal loss. Moreover, we experimented with a horizontal ensembling strategy using the simpler architecture of the deep learning models. Apart from that, we trained several linear and tree models using the hidden feature space of a BiLSTM model. Finally, we made a thorough analysis on the performance of each of these techniques for the three text classification tasks achieving more than 90% f1-score for the minority class without hampering the overall performance.

As a future extension of this work, we look forward to applying relatively newer text generation methods for data augmentation. These techniques can be applied to some other class imbalanced datasets of different low resource languages to evaluate their general use-cases. Besides, the enforcement of several discriminative models with attention mechanisms on such a class-imbalanced dataset can be a potential target for future research.

ACKNOWLEDGMENT

This work was supported by the ICT Division, Government of the People's Republic of Bangladesh.

REFERENCES

Charu C Aggarwal. 2015. Outlier analysis. In *Data mining*. Springer, 237–263.

Charu C Aggarwal and Philip S Yu. 2001. Outlier detection for high dimensional data. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*. 37–46.

Hadeer Ahmed, Issa Traore, and Sherif Saad. 2017. Detection of online fake news using n-gram analysis and machine learning techniques. In *International conference on intelligent, secure, and dependable systems in distributed and cloud environments*. Springer, 127–138.

Gilbert Badaro, Ramy Baly, Hazem Hajj, Wassim El-Hajj, Khaled Bashir Shaban, Nizar Habash, Ahmad Al-Sallab, and Ali Hamdi. 2019. A survey of opinion mining in Arabic: A comprehensive system perspective covering challenges and advances in tools, resources, models, applications, and visualizations. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)* 18, 3 (2019), 1–52.

Gilbert Badaro, Ramy Baly, Hazem Hajj, Nizar Habash, and Wassim El-Hajj. 2014. A large scale Arabic sentiment lexicon for Arabic opinion mining. In *Proceedings of the EMNLP 2014 workshop on arabic natural language processing (ANLP)*. 165–173.

Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. 2017. CVAE-GAN: fine-grained image generation through asymmetric training. In *Proceedings of the IEEE international conference on computer vision*. 2745–2754.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating Sentences from a Continuous Space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Berlin, Germany, 10–21. https://doi.org/10.18653/v1/K16-1002

Paula Branco, L. Torgo, and Rita P. Ribeiro. 2015. A Survey of Predictive Modelling under Imbalanced Distributions. *ArXiv* abs/1505.01658 (2015).

Leo Breiman. 1996a. Bagging Predictors. *Machine Learning* 24, 2 (01 Aug 1996), 123–140. https://doi.org/10.1023/A:1018054314350

Leo Breiman. 1996b. Stacked regressions. *Machine Learning* 24, 1 (01 Jul 1996), 49–64. https://doi.org/10.1007/BF00117832

Leo Breiman. 2001. Random Forests. *Machine Learning* 45, 1 (01 Oct 2001), 5–32. https://doi.org/10.1023/A:1010933404324

Andrew Brock, Jeff Donahue, and Karen Simonyan. 2018. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096* (2018).

Gavin Brown and Ludmila I. Kuncheva. 2010. "Good" and "Bad" Diversity in Majority Vote Ensembles. In *Multiple Classifier Systems*, Neamat El Gayar, Josef Kittler, and Fabio Roli (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 124–133.

Nitesh V. Chawla, Lawrence O. Hall, Kevin W. Bowyer, and W. Philip Kegelmeyer. 2004. Learning Ensembles from Bites: A Scalable and Accurate Approach. *J. Mach. Learn. Res.* 5 (Dec. 2004), 421âĂŞ451.

Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16)*. Association for Computing Machinery, New York, NY, USA, 785âĂŞ794. https://doi.org/10.1145/2939672.2939785

D. A. Cieslak, N. V. Chawla, and A. Striegel. 2006. Combating imbalance in network intrusion datasets. In *2006 IEEE International Conference on Granular Computing*. 732–737.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning* 20, 3 (01 Sep 1995), 273–297. https://doi.org/10.1007/BF00994018

C. Ferri, J. HernÃąndez-Orallo, and R. Modroiu. 2009. An experimental comparison of performance measures for classification. *Pattern Recognition Letters* 30, 1 (2009), 27 – 38. https://doi.org/10.1016/j.patrec.2008.08.010

Yoav Freund and Robert E. Schapire. 1999. A Short Introduction to Boosting. In *In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*. Morgan Kaufmann, 1401–1406.

Nir Friedman, Dan Geiger, and Moises Goldszmidt. 1997. Bayesian Network Classifiers. *Machine Learning* 29, 2 (01 Nov 1997), 131–163. https://doi.org/10.1023/A:1007465528199

Mikel Galar, Alberto Fernandez, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. 2012a. A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches. *Trans. Sys. Man Cyber Part C* 42, 4 (July 2012), 463âĂŞ484. https://doi.org/10.1109/TSMCC.2011.2161285

Mikel Galar, Alberto FernÃąndez, Edurne Barrenechea, Humberto Sola, and Francisco Herrera. 2012b. A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 42 (07 2012), 463 – 484. https://doi.org/10.1109/TSMCC.2011.2161285

A. F. Ganai and F. Khursheed. 2019. Predicting next Word using RNN and LSTM cells: Stastical Language Modeling. In *2019 Fifth International Conference on Image Information Processing (ICIIP)*. 469–474.

Yoav Goldberg and Omer Levy. 2014. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. (02 2014).

Mykhailo Granik and Volodymyr Mesyura. 2017. Fake news detection using naive Bayes classifier. In *2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)*. IEEE, 900–903.

Haibo He and Yunqian Ma. 2013. *Imbalanced Learning: Foundations, Algorithms, and Applications* (1st ed.). Wiley-IEEE Press.

S. Helmstetter and H. Paulheim. 2018. Weakly Supervised Learning for Fake News Detection on Twitter. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. 274–277.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

Md Zobaer Hossain, Md Ashraful Rahman, Md Saiful Islam, and Sudipta Kar. 2020. BanFakeNews: A Dataset for Detecting Fake News in Bangla. In *Proceedings of The 12th Language Resources and Evaluation Conference*. European Language Resources Association, Marseille, France, 2862–2871. https://www.aclweb.org/anthology/2020.lrec-1.349

C. Huang, Y. Li, C. C. Loy, and X. Tang. 2016. Learning Deep Representation for Imbalanced Classification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5375–5384.

Minhajul Islam. 2009. Research on Bangla Language Processing in Bangladesh: Progress and Challenges. (07 2009).

Nathalie Japkowicz and Shaju Stephen. 2002. The Class Imbalance Problem: A Systematic Study. *Intell. Data Anal.* 6, 5 (Oct. 2002), 429âĂŞ449.

Justin M. Johnson and Taghi M. Khoshgoftaar. 2019. Survey on deep learning with class imbalance. *Journal of Big Data* 6, 1 (19 Mar 2019), 27. https://doi.org/10.1186/s40537-019-0192-5

Mohammad Karim, Mohammad Kaykobad, and M. Murshed. 2012. *Technical Challenges and Design Issues in Bangla Language Processing*.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 3146–3154. http://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree.pdf

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1746–1751. https://doi.org/10.3115/v1/D14-1181

S. Kotsiantis, D. Kanellopoulos, and P. Pintelas. 2006. Handling imbalanced datasets: A review.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 302–308.

Changliang Li, Yixin Su, and Wenju Liu. 2018. Text-to-text generative adversarial networks. In *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–7.

Der-Chiang Li, Chiao-Wen Liu, and Susan C. Hu. 2010. A Learning Method for the Class Imbalance Problem with Medical Data Sets. *Comput. Biol. Med.* 40, 5 (May 2010), 509âĂŞ518. https://doi.org/10.1016/j.compbiomed.2010.03.005

Junyi Jessy Li and Ani Nenkova. 2014. Addressing Class Imbalance for Improved Recognition of Implicit Discourse Relations. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*. Association for Computational Linguistics, Philadelphia, PA, U.S.A., 142–150. https://doi.org/10.3115/v1/W14-4320

Shoushan Li, Shengfeng Ju, Guodong Zhou, and Xiaojun Lin. 2012. Active learning for imbalanced sentiment classification. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. 139–148.

Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and P. Dollár. 2017. Focal Loss for Dense Object Detection. *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), 2999–3007.

Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. 2009. Exploratory Undersampling for Class-Imbalance Learning. *Trans. Sys. Man Cyber. Part B* 39, 2 (April 2009), 539âĂŞ550. https://doi.org/10.1109/TSMCB.2008.2007853

Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 5528–5531.

Federico Monti, Fabrizio Frasca, Davide Eynard, Damon Mannion, and Michael M. Bronstein. 2019. Fake News Detection on Social Media using Geometric Deep Learning. arXiv:1902.06673 [cs.SI]

Vladimir Nikulin, Geoffrey J. McLachlan, and Shu Kay Ng. 2009. Ensemble Approach for the Classification of Imbalanced Data. In *AI 2009: Advances in Artificial Intelligence*, Ann Nicholson and Xiaodong Li (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 291–300.

Kemal Oksuz, Baris Can Cam, Sinan Kalkan, and Emre Akbas. 2020. Imbalance problems in object detection: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. *arXiv preprint cs/0205070* (2002).

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1532–1543. https://doi.org/10.3115/v1/D14-1162

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. CatBoost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.). Curran Associates, Inc., 6638–6648. http://papers.nips.cc/paper/7898-catboost-unbiased-boosting-with-categorical-features.pdf

Anna Rumshisky, Marzyeh Ghassemi, Tristan Naumann, Peter Szolovits, VM Castro, TH McCoy, and RH Perlis. 2016. Predicting early psychiatric readmission with natural language processing of narrative discharge summaries. *Translational psychiatry* 6, 10 (2016), e921–e921.

Salim Sazzed. 2020. Cross-lingual sentiment classification in low-resource Bengali language. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*. 50–60.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing* 45, 11 (1997), 2673–2681.

C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano. 2010. RUSBoost: A Hybrid Approach to Alleviating Class Imbalance. *Trans. Sys. Man Cyber. Part A* 40, 1 (Jan. 2010), 185âĂŞ197. https://doi.org/10.1109/TSMCA.2009.2029559

Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. 2017. A Hybrid Convolutional Variational Autoencoder for Text Generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, 627–637. https://doi.org/10.18653/v1/D17-1066

Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. 2019. DEFEND: Explainable Fake News Detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and amp; Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 395âĂŞ405. https://doi.org/10.1145/3292500.3330935

Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter* 19, 1 (2017), 22–36.

Zhongbin Sun, Qinbao Song, Xiaoyan Zhu, Heli Sun, Baowen Xu, and Yuming Zhou. 2015. A novel ensemble method for classifying imbalanced data. *Pattern Recognition* 48, 5 (2015), 1623 – 1637. https://doi.org/10.1016/j.patcog.2014.11.014

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1555–1565.

Taoran Tang, Jia Jia, and Hanyang Mao. 2018. Dance with melody: An LSTM-autoencoder approach to music-oriented dance synthesis. In *Proceedings of the 26th ACM international conference on Multimedia*. 1598–1606.

Katrin Tomanek and Udo Hahn. 2009. Reducing class imbalance during active learning for named entity annotation. In *Proceedings of the fifth international conference on Knowledge capture*. 105–112.

Remi Trichet and Francois Bremond. 2018. Dataset optimization for real-time pedestrian detection. *IEEE access* 6 (2018), 7719–7727.

S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, and P. J. Kennedy. 2016. Training deep neural networks on imbalanced data sets. In *2016 International Joint Conference on Neural Networks (IJCNN)*. 4368–4374.

Xizi Wang, Feng Cheng, Shilin Wang, Huanrong Sun, Gongshen Liu, and Cheng Zhou. 2018a. Adult image classification by a local-context aware network. In *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2989–2993.

Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su, and Jing Gao. 2018b. EANN: Event Adversarial Neural Networks for Multi-Modal Fake News Detection. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge*

Discovery and amp; Data Mining (London, United Kingdom) *(KDD '18)*. Association for Computing Machinery, New York, NY, USA, 849âĂŞ857. https://doi.org/10.1145/3219819.3219903

Michael Ying Yang, Wentong Liao, Xinbo Li, and Bodo Rosenhahn. 2018. Deep learning for vehicle detection in aerial images. In *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 3079–3083.

Xinyi Zhou, Reza Zafarani, Kai Shu, and Huan Liu. 2019. Fake news: Fundamental theories, detection strategies and challenges. In *Proceedings of the twelfth ACM international conference on web search and data mining*. 836–837.

Zhi-Hua Zhou and Xu-Ying Liu. 2006. Training Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem. *IEEE Trans. on Knowl. and Data Eng.* 18, 1 (Jan. 2006), 63âĂŞ77. https://doi.org/10.1109/TKDE.2006.17