# Comparing ELMo and BERT in Toxic Comments Classification Task

**Justin Hsia** and **Arnaldo Tavares**
MIDS, UC Berkeley
{justin.hsia, arnaldotav}@berkeley.edu

**Abstract -** Identifying toxic comments is a challenging task that attracts study and competitors for the best approach. But this brings challenges in making direct comparison among different Natural Language Processing models. This work leverages Kaggle's "Toxic Comment Classification Challenge" dataset to compare the performance between ELMo and BERT models. We describe how we addressed the unbalanced data and multi-class/multi-label dataset and how we fine-tuned BERT to deal with overfitting. The results show that BERT performs better at this task than ELMo, demonstrating lower losses during training, better F1 score, and better accuracy in the testing phase, while being less computationally demanding.

## 1. Introduction

Toxic comments on the internet are a source of concern for social media companies and their users. Both should be able to identify toxic comments to flag, ignore, respond or even take legal action. A solution to this problem comes from NLP (Natural Language Processing) systems that can go through comments and identify those that are considered toxic.

This topic called the attention of the scientific community and new workshops, and online challenges were organized and created in order to address this problem. Kaggle was one of the organizers of such online challenges and, in 2018, created the "Toxic Comment Classification Challenge". This challenge attracted 4,551 teams that competed to create the better detector of toxicity, using a dataset of almost 160,000 comments labelled according to 6 different classes: 'toxic', 'severe_toxic', 'obscene', 'threat', 'insult' and 'identity_hate'.

Such competitions are a great opportunity to evaluate different approaches for a specific task. But it is usually challenging for practitioners to compare them to define which model performs the best, as each competitor uses different combinations of models, methods, approaches, topologies, hyperparameters and computing resources.

In this work, we will compare the performance of two modern NLP models side-by-side: ELMo (Embeddings from Language Models, Peters et al., 2018a) and BERT (Bidirectional Encoder Representations from Transformers, Devlin et al., 2018). We will leverage the same dataset from Kaggle's "Toxic Comment Classification Challenge".

## 2. Background

During this work, we used different references to better understand both NLP models and their differences. Also, we leveraged work from researchers in order to solve challenges we faced in the project, such as dealing with unbalanced data and fine-tuning the models to achieve their best results.

### 2.1 Comparing EMLo and BERT

Devlin explained that there are two main strategies for applying pre-trained language representations to NLP tasks: feature-based and fine-tuning. ELMo leverages the former using different architectures for specific tasks that include the pre-trained representations as additional features. BERT, on the other hand, uses the latter approach and is trained on the specific tasks by fine-tuning all its pre-trained parameters.

Devlin also explained that BERT uses marked language models to allow pre-trained bi-directional representations, while ELMo uses a concatenation of independently trained left-to-right and right-to-left language models. The author stated that this reduces the need for heavily-engineered task-specific architectures, allowing BERT to achieve good performance in many sentence- and token-level tasks.

### 2.2 Dealing with unbalanced data

One of the problems identified during this work was on the extreme unbalanceness of the dataset. According to Mountassir et al, 2012, to solve the problem of unbalanced data, two different approaches can be employed. The first approach focuses on modifying the classifier, while the second approach focuses on modifying the dataset itself. In this work, as we would like to compare both ELMo and BERT as unchanged as possible, we decided to use the second approach and modify the dataset in order to improve its balance. Also according to Mountassir et al., 2012 and Madabushi et al., 2019, when modifying the dataset, one can oversample the underrepresented class(es), undersample the overrepresented class(es), or a combination of both. We decided to use random undersampling (Tahir et al., 2009) as it is a simpler method and is proven to provide good results. We avoided the use of oversampling due to the risk of overfitting (Chawla et al., 2002).

### 2.3 Fine-tuning BERT

We expected to encounter issues with BERT when working on unbalanced data: per Madabushi, BERT fails to perform well when training and test sets are dissimilar (much more examples of one class versus the other(s)). The author also stated that this is typical in tasks that deal with social data, which is the case in this work. We leveraged the undersampling technique to address this, as previously described.

We also identified overfitting issues in the first few epochs, so we proceeded to apply some fine tuning and comparisons, initially following the typical hyperparameters ranges (Devlin et. al., 2018):

- Batch size: 16, 32
- Learning rate (Adam): 5e-5, 3e-5, 2e-5
- Number of epochs: 2, 3, 4

Per C. Sun et al., 2019, we needed to consider and adjust for overfitting problems when adapting BERT to a target task. In order to address this, they recommended working with the appropriate learning rate and evaluating different ones. In our work, we went beyond initial typical hyperparameters ranges mentioned above to identify the best setting for this task.

## 3. Methods

### 3.1 Dataset

The dataset used came from Kaggle's "Toxic Comment Classification Challenge" in 2018. This competition used a dataset from Wikipedia, comprising 159,571 comments in total, 16,225 of them are considered toxic by human evaluators. The toxic comments are further divided in 6 classes: toxic, severe-toxic, obscene, threat, insult, and identity-hate. The distribution of the classification is shown in Figure 1 below.

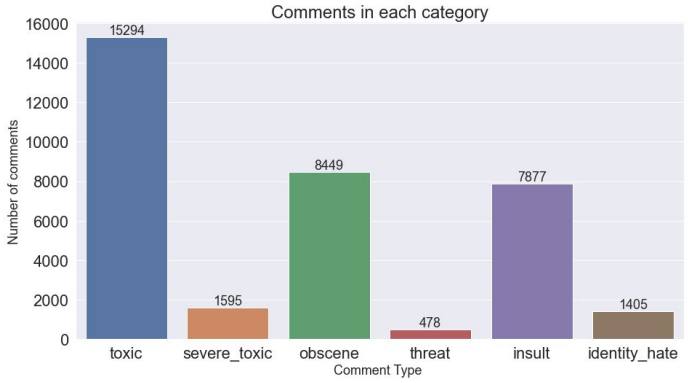


**Figure 1.** Distribution of toxic comment classifications

Also, this dataset allowed for multi-label entries up to 6 class labels. The distribution of multi-labels is as followed, on Figure 2:

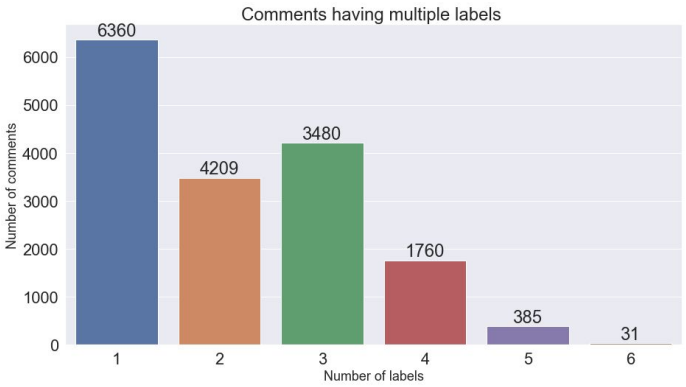


**Figure 2.** Distribution of number of labels

Here we can clearly see 2 challenges: a) the use of multi-labels and b) the unbalanceness in the dataset between labeled and unlabeled comments.

To address the multi-labelling ambiguity and considering the limited time and computing resources available for this project, we decided to combine all classes that had at least 1 label into a consolidated 'toxic' class. All entries that had no labels were consolidated into an 'no-toxic' class, transforming the problem into a binary classification task.

Even with this consolidation, we still needed to address the unbalanceness between the 'toxic' and the 'no-toxic' classes. In order to do that, we used undersampling of the overrepresented 'no-toxic' class to achieve balance in the dataset:

- We created a balanced 'new_train' dataset, with 14,602 entries with 'toxic' label and the same number with 'non-toxic' labels, breaking it down into 80% training set and 20% validation set
- We created an unbalanced 'new_test' test set, with the remaining 1,623 'toxic' entries and the remaining 128,744 'no-toxic' entries.

Table 1 summarizes the final dataset we used to train, validate and test our models. Note that we avoided the use of oversampling techniques for the underrepresented class to avoid issues of overfitting, as this is very common, especially with the BERT model.

| Class | Training Set | Validation Set | Test Set |
|---|---|---|---|
| *non-toxic* | 11,682 | 2,920 | 128,744 |
| *toxic* | 11,682 | 2,920 | 1,623 |

**Table 1.** Final dataset training, validation and test split

### 3.2 Models

We chose ELMo and BERT for this work because both models use context-based embeddings, and there are readily available pre-trained models we can employ. Here is a brief description of each model:

- **ELMo:** use a pre-trained ELMo embedding module available in tensorflow-hub. This module supports both raw text strings or tokenized text strings as input. The module outputs fixed embeddings at each LSTM layer, a learnable aggregation of the 3 layers, and a fixed mean-pooled vector representation of the input (for sentences). Total number of features for each embedding is 1024. We added a single dense layer with 10 neurons to help train the model and a final one-neuron layer for classification.

- **BERT:** used BERT-Base pre-trained model with 12 Transformer blocks, 12 self-attention heads and dimension vector size of 768 features, totaling 110M parameters. We inserted a single fine tune layer with two neurons, one of each class.

We used binary cross entropy loss as our loss function and F1 score as our metric during validation to evaluate the performance of each model. This metric was selected to make a better comparison between the models as it equally weights the performance of each class (Madabushi et al., 2019).

## 4. Results and Discussion

We did not encounter any abnormalities when running the ELMo model despite the fact that each epoch took approximately 35 minutes to run on Google Colab with P100 GPU. We reduced the batch size to 16 in order to avoid the OOM (out-of-memory) issue. The model was able to run well on the dataset, and both training and validation losses reduced as expected despite some minor variations. As shown in Figure 3, validation loss was fairly stable after epoch 2. We also did not observe much variation in the F1 score, with a value of 0.88 at epoch 10.

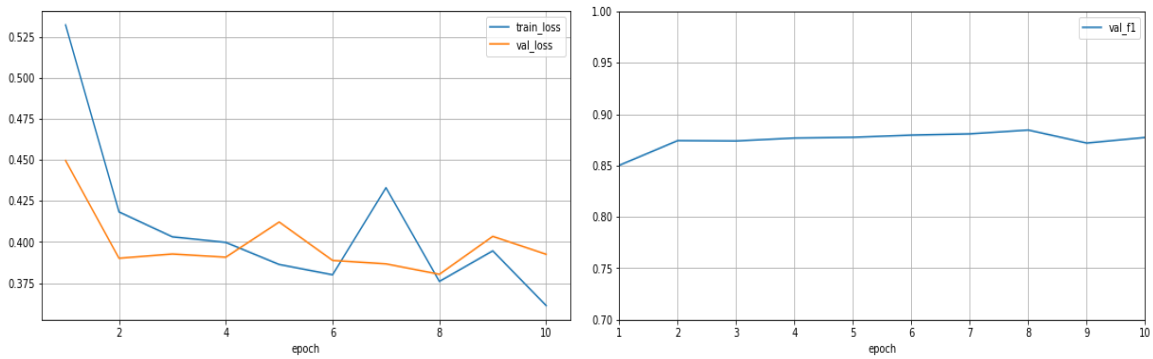


**Figure 3.** ELMo test results

We selected the weights from epoch 10 to conduct the final test on classification. Recall from Table 1 that the test dataset is highly unbalanced with 128,744 samples in the non-toxic class and 1,623 samples in the toxic class. The results are captured in Table 2.

| Class | Accuracy |
|---|---|
| non-toxic | 104845 / 128744 (81.44%) |
| toxic | 1493 / 1623 (91.99%) |

**Table 2.** ELMo classification results on test dataset

We ran the BERT model with the learning rate set at 2e-5 as suggested by Devlin. The batch size was set to 32 We ran into the overfitting problem (described in section 2.3) right out of the gate using the 80-20 training-validation dataset split. We ran the model a second time using 5e-5 as the new learning rate. However, the overfitting problem persisted. Figure 4 captures the result with both learning rates.

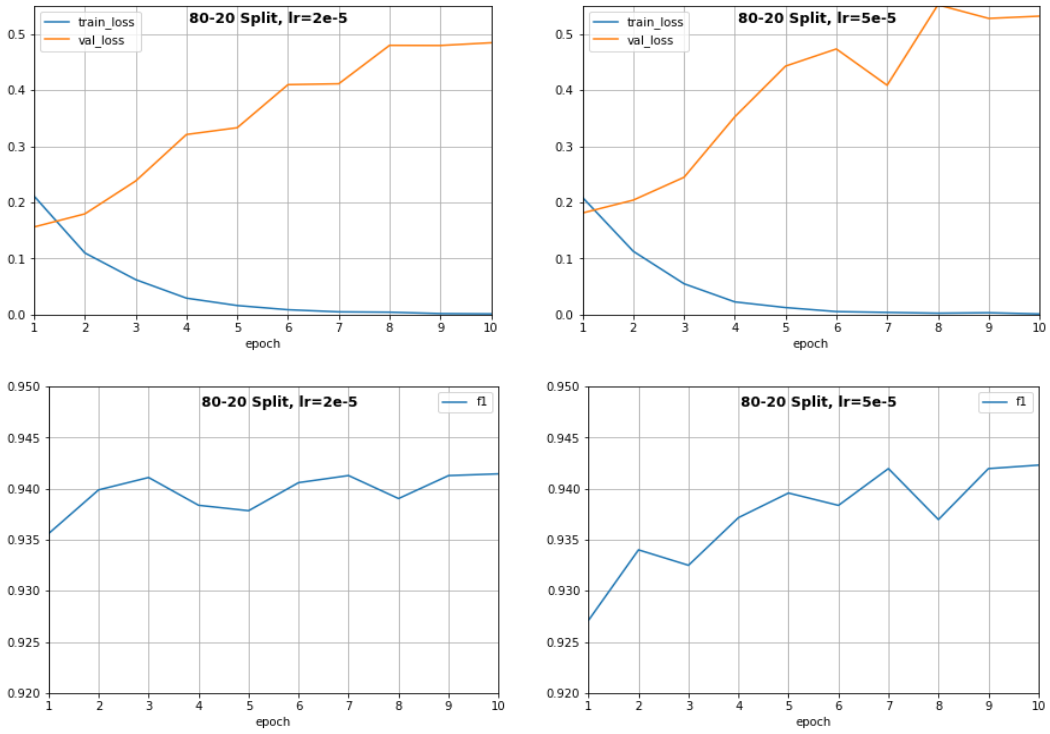


**Figure 4.** BERT training results using 80/20 split

As shown in Figure 4, after the first epoch the model already started overfitting with training loss continuing to decrease while validation loss started to grow. Also, as described in section 2.3 and per reference work of other researchers, we started a series of tests variating batch sizes, learning rates and testing over multiple epochs. In terms of batch sizes, we increased it to 64 and retrained the model. We encountered the OOM error at the completion of epoch 8. The results are captured in Figure 5. As shown in the figure, we were still not able to overcome the overfitting problem by varying the batch size. When increasing batch size to 128 or beyond, we again encountered

the OOM error due to the limited amount of GPU memory available in this project. Here, we clearly see that BERT generates very large models and the use of multiple GPUs with more memory is important to increase batch sizes.

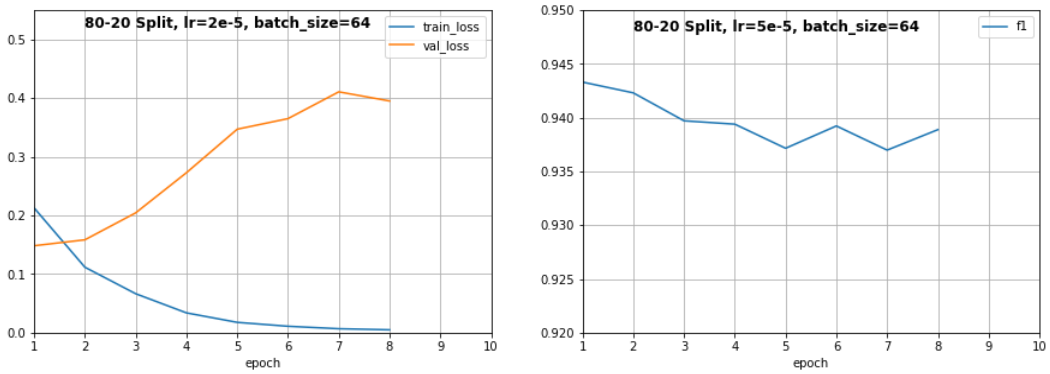


**Figure 5.** BERT training results with batch size = 64

An interesting learning occurred when we were evaluating the entire unbalanced dataset (almost 160,000 entries) versus the balanced dataset (with approximately 28,000 entries): by reducing the dataset size, we noticed that the model began overfitting a bit later. So we decided to try different split ratios between training and validation datasets (from a 80-20 split down to 05-95 split). Depending on the size of the training dataset, each epoch takes roughly 5 to 10 minutes to run with larger training datasets taking longer time. As shown in Figure 5, with a 10-90 and a 05-95 split, the model overfits after the second epoch. Although the model started to overfit later, we still obtained better results in terms of training loss, validation losses, and F1 score with the original 80-20 split. Our understanding here is that BERT seems to be a very powerful model that does not require a very large amount of data to train on. (Actually, too much data may make it overfit earlier!) The model can also learn the right parameters very quickly (in our case, in the very first epoch). This is in line with what Devlin suggested in his paper, where the author demonstrated optimal results were obtained in the first two for four epochs.

We chose the model weights from the first epoch for the 80-20 split and from the second epoch for the 10-90 split to conduct the final classification test. The test dataset is the same as that used on ELMo, which is highly unbalanced. The results are captured in Tables 3 and 4 below. As shown in Table 2, 3, and 4, BERT outperformed ELMo in both classes. Particularly in the non-toxic class, ELMo result was around 10% lower than the BERT results. When comparing the two BERT results, BERT model trained on the 10-90 split was able to capture more toxic comments in trade with misclassifying some non-toxic comments as toxic.

| Class | Accuracy |
|---|---|
| non-toxic | 120088 / 128744 (93.28%) |
| toxic | 1514 / 1623 (93.28%) |

**Table 3.** BERT classification results on test dataset, 80/20 split, LR=2e-5

| Class | Accuracy |
|---|---|
| non-toxic | 116425 / 128744 (90.43%) |
| toxic | 1534 / 1623 (94.52%) |

**Table 4.** BERT classification results on test dataset, 10/90 split, LR=2e-5

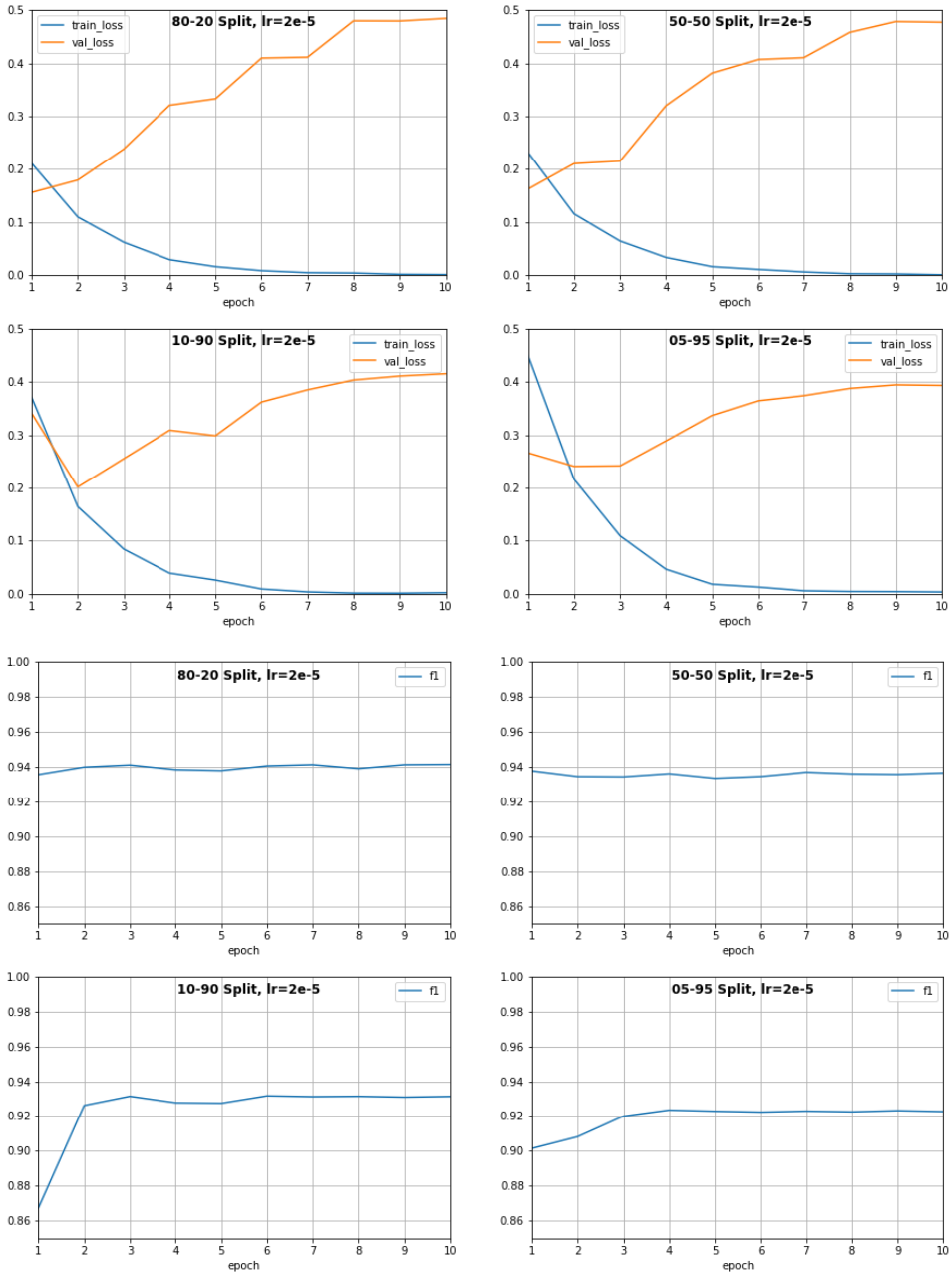


**Figure 5.** BERT test results with varying training validation dataset split

## 5. Conclusion and Future Work

We confirmed that classification of toxic comments add new complexities to the task due to the very unbalanced nature of the data and also due to its semantic similarity, making it harder for even human labelers to correctly classify them and having to use multi-labeling to address this extra degree of ambiguity.

In terms of the models, BERT has proven to be a better solution for this task. Despite its overfitting issues during training, the model demonstrated great ability to quickly learn the important features even with a small amount of training data. Also, it proved to be a more computationally efficient model, performing its training and test process much faster than ELMo.

### 5.1 Future Work

As future work, we consider trying different data balancing techniques, such as oversampling using random technique or constructing synthetic minority-class samples. In undersampling, we consider the use of other techniques such as Remove Similar or Remove Farthest.

We also consider in the future addressing the multi-label problem, and comparing how different modern NLP models deal with this additional challenge.

Finally, we plan to add new state-of-the-art models that are presented by the scientific community, such as the recently released GPT-3, created by OpenAI.

## References


Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer. 2002. W. P. SMOTE: Synthetic Minority Over- sampling Technique. Journal of Artificial Intelligence Research (JAIR), Volume 16, pp. 321-357.

Harish Tayyar Madabushi, Elena Kochkina, and Michael Castelle. 2019. Cost-Sensitive BERT for Generalisable Sentence Classification on Imbalanced Data. Association for Computational Linguistics, Volume: Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda, pp. 125-134.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Kubat, M., Matwin, S., 1997. Addressing the Curse of Imbalanced Data Sets: One-Sided Sampling. In Proceedings of the Fourteenth International Conference on Machine Learning, pp. 179-186.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. Deep contextualized word representations. In NAACL.

Mountassir, Asmaa & Benbrahim, Houda & Berrada, Ilham, 2012. An empirical study to address the problem of Unbalanced Data Sets in sentiment classification. 3298-3303. 10.1109/ICSMC.2012.6378300.

Na Pang, Li Qian, Weimin Lyu, Jin-Dong Yang, 2019. Transfer Learning for Scientific Data Chain Extraction in Small Chemical Corpus with BERT-CRF Model. arXiv preprint arXiv:1905.05615v1

Sun, Haitong & Yang, Liyang & Bai, Xiaoxia & Shen, Guofeng & Fei, Jie & Wang, Yonghui & Chen, An & Chen, Yuanchen & Zhao, Meirong, 2019. Supplementary Materials.

Tahir, M. A., Kittler, J., Mikolajczyk, K., & Yan, F., 2009. A multiple expert approach to the class imbalance problem using inverse random under sampling. Multiple Classifier Systems, pp. 82–91. Springer.