24th European Conference on Artificial Intelligence - ECAI 2020
Santiago de Compostela, Spain

Sequencing, Combining and Sampling Classifiers to Help Find Needles in Haystacks

Jaebeen Lee [1] and **Léa A. Deleris** [2]

Abstract. Many binary prediction situations involve imbalanced datasets where the ratio of the minority class over the majority class is very low. This is especially true when dealing with problems looking to use machine learning to better detect fraud, errors or exceptions. In this paper, we address the problem of extreme imbalance, i.e. where the imbalance ratio of majority over minority instances exceeds 500. Given the scarcity of minority examples, oversampling is not sensible due to expensive computational cost. Hence, we explore and expand undersampling approaches. Specifically, we propose a modeling framework (i.e., sequence of modeling steps) that seeks to leverage as much training data as possible. Our results indicate the better trade-off between the false positives and false negatives, which makes it more suitable for real-life application.

1 Introduction

In binary classification, a dataset is considered to be suffering from class imbalance problem when the population of one class largely outnumbers another class. It is equivalent to having imbalance ratio above 1, which is defined to be the number of majority instances divided by the minority instances. Training on such dataset without addressing this issue can lead to excellent levels of accuracy overall and for the majority class but poor performance for minority classes due to the bias towards the majority instances introduced to the models [16]. This is a common problem for real-life datasets in a variety of domains such as fraud detection [26], network intrusion [6] and medical diagnosis [23]. The simplest and most popular way to deal with this challenge is data sampling, which aims at balancing the class representation by either undersampling the majority or oversampling the minority [22]. We can tackle this problem from the model perspective by assigning higher penalisation cost to the wrongly classified minority instances [1, 28]. Embedding data-sampling techniques to the ensemble models is also a popular solution [14].

In real world situations, it is not unusual to encounter datasets that are considered *extremely imbalanced*, where the imbalance ratio is higher than 500. This is especially true in fraud detection where huge number of normal transactions take place on a daily basis, with minimal number of fraudulent cases due to the preventative systems in place. Despite the scarcity of fraudulent cases however, the consequence of failing to detect such cases can be quite detrimental. This highlights the importance of developing a strategy to deal with such problem. Likewise, this is a common problem in medical diagnosis of the rare disease where the impact of the error is critical to the patients. Other contexts, for instance the detection of errors in highly reliable processes - the initial motivation for this work - also yields imbalance ratios that can be one or two orders of magnitude over 500.

The difficulty of extremely imbalanced dataset is not limited to the highly diluted minorities within the dataset, but also to the ensuing size of the associated dataset. A dataset with an imbalance ratio of 10,000 with 1,000 minority instances contains 10 million instances. Oversampling the minority to balance the data will lead to almost doubling the dataset size and the computational cost from both sampling and model training perspective may become prohibitively expensive [9]. This leaves us with little choice but to use undersampling or ensemble methods that leverage on undersampling.

The most common way of using undersampling is to randomly select majority instances so as to obtain the same number of minority instances and to use this much smaller dataset for training. The biggest downfall of random undersampling in extreme imbalanced context is that a large number of majority instances are discarded from the dataset, leading to the loss in valuable information. Moreover, the sample size of the majority class is inconsequential in comparison to the overall population. Therefore, the composition of the sampled dataset differ greatly at each sampling step, leading to high variance in the model prediction [7]. To alleviate this problem, undersampling with ensemble methods is regarded as the best option, with some research on applying such models in a context of highly imbalanced data, i.e., where the imbalance ratio is above 10 [13, 27]. Nevertheless, the proposals have not been tested on extremely imbalanced datasets. Through our initial experiments for an error detection task, we observed that these methods detected a good number of true positives, yet the high number of false positives made the model impractical in real-life scenario. We believe this is caused by too many majority instances still being discarded.

In this work, we propose a new novel approach designed specifically for extremely imbalanced datasets. The key idea behind our approach is to utilise as much of the training dataset as possible. Our approach should be thought more as an architecture for sequencing and combining models rather than fully specified model. Therefore, it is versatile in the sense that it allows to choose variety of techniques, including oversampling. The inspiration behind the proposal is *ensembling the ensemble models* from the EasyEnsemble algorithm [21]. The proposed architecture consists of two main stages. In the first step, we obtain a base undersampling embedded ensemble model with high recall score to filter out the obvious majority elements from the training dataset. In the second step, we use the reduced training dataset to look for an improved way to combine the ensemble the models from the first step. Benefits of the reduced dataset is the reduction in imbalance ratio and the possibility to use methods with higher computational cost during the second stage. To summarise, the main contributions of our research are as follows:

[1] BNP Paribas, United Kingdom, email: jaebeen.lee@uk.bnpparibas.com
[2] BNP Paribas, France, email: lea.deleris@bnpparibas.com

- A novel modification to EasyEnsemble which introduces extra model to maximise the utilisation of the dataset and ensembles the sub-models more effectively.
- A mechanism to filter the obvious majority instances within the intermediate step of the pipeline.
- The ability to use other techniques dealing with imbalanced ratio with lower computational cost.

The remainder of this paper is organised as follows. In Section 2, we provide more context into the existing literature related to highly imbalanced prediction problems. Section 3 describes our proposed method to tackle such problem in detail. We then test our approach on the extreme imbalanced dataset and rigorously analyse the result in Section 4, with the comparison against other approaches and discussion on the possible next steps.

2 Related Work

Data sampling is a common technique to deal with class imbalance problems. It involves either replicating minority class instances to add them to the training dataset (oversampling) or removing majority class instances (undersampling) from the training dataset, thus balancing the ratio of these two classes.

Random undersampling (RUS) is the most straightforward and computationally cheap undersampling technique. Variations to the random undersampling have been proposed to sample in a smarter way, such as using the nearest neighbour of the clustering centres for sampling [20]. However, many majority examples are discarded from the training dataset, especially with extreme imbalanced cases. In addition to losing valuable information on the majority instances, the variance of the prediction for the model with the same setting but different undersampled dataset will be very high [7].

Random oversampling is not considered as a viable option as the duplication of the same points can lead to overfitting. Synthetic minority oversampling technique (SMOTE) [4] is the common oversampling technique which adds synthetic minorities by interpolating several pre-existing minorities that lie close to each other. Once again, alternatives to create synthetic minorities have also been proposed, such as Modified SMOTE (MSMOTE) [17]. As mentioned previously however, the resulting training dataset will approximately double in size for the extreme imbalanced dataset which will make the process computationally expensive. Not to mention the conceptual problem related to training with hundreds of replications of the same data points.

Another method to deal with imbalanced dataset problem is to embed data sampling techniques within ensemble models [14]. RUSBoost [24] and SMOTEBoost [5] are variants of AdaBoost which applies RUS and SMOTE respectively to the training dataset at each iteration to learn base learners. EasyEnsemble [21] produces a single ensemble from multiple AdaBoost models trained with RUS-applied dataset. A slight adaptation to EasyEnsemble, called BalanceCascade [21] filters out the redundant majority class samples whilst training each AdaBoost sub-models to be ensembled. This leads to a restricted sample space of the majority instances for each sub-models, thus forcing them to explore as much useful information as possible. However, it is noted that the performance of BalanceCascade is inferior to EasyEnsemble on harder tasks, possibly due to the overfitting.

Recently, other interesting alternatives have been proposed which do not clearly fall into any of the traditional categories. Two sample representation (s2s) [11] involves representing the training set in such way that two arbitrary samples are considered at the same time during the training of the model. Disadvantage of this approach is the quadratic increase in the number of training samples, which makes it unsuitable for large datasets. Uncorrelated cost-sensitive multiset learning (UCML) [27] uses multiset feature learning (MFL) on multiple subsets of the data created from undersampling.

Although many methods have been motivated by the need to solve imbalanced or highly imbalanced dataset, not much work address imbalance ratio to such extremity as we require for instance in our error detetcion task. Oversampling the minority based on the distribution of the majority instances have been proposed [25] to overcome extremely imbalanced data, as opposed to other oversampling method which are based solely on the minority examples. However, this does not solve the issue regarding the computational cost and reliance on artificially produced data. Otherwise, proposals looking into highly imbalanced dataset do not get tested on the extremely imbalanced dataset. Abalone19 dataset in KEEL repository was the dataset with the highest imbalance ratio of 129 in almost all of the works [11, 27], with a clear deterioration in results observed as the imbalance ratio increased. In fact, a study on big data with imbalance ratio between 100 and 10000 [19] mentions that there are significant gaps in the current research in this topic. Currently, dealing with such problem mainly involves applying a MapReduce framework to the traditional algorithms or data-sampling methods [12] but the results indicate that it is inadequate to deploy such methods in practice.

The key focus of this paper is to propose a model architecture that specifically looks at extremely imbalanced datasets, with a ratio above 500. This is done by seeking to utilise minority and majority classes instance in a smarter way.

3 Proposed Approach

Our proposed approach to extreme imbalance dataset consists of two-step model strategy where first model filters out the obvious majority instances and the second model learns how to distinguish more challenging majority instances from the minority instances by leveraging results from previous step.

3.1 Mathematical Formulation

Before the description of the proposal, we set out the mathematical notation that will be used. Consider a binary classification problem with the imbalanced dataset D;

$$D = \left\{ (X_i, y_i) \mid i = 1, ..., N, X_i \in \mathbb{R}^d, y_i \in \{0, 1\} \right\} \qquad (1)$$

where 0 and 1 corresponds to the majority and minority class label respectively. X_i refers to a d dimensional feature vector. Let N_0 and N_1 be the number of majority and minority instances such that $N_0 + N_1 = N$ and we define the dataset to be suffering from extreme imbalance class if the imbalance ratio, defined as N_0/N_1, is larger than 500. Also note that 0 and 1 can be described as 'negative' and 'positive' instances. We finally denote D_{train} and D_{test} to be the training and testing dataset, which are subsets of D with $D_{\text{train}} \cap D_{\text{test}} = \emptyset$.

3.2 Training the Filtering Models

Ideally, we want to minimise the number of minority instances incorrectly filtered out after the first step so we need our first model to have the lowest possible false negative rate. Furthermore, it is common for extreme imbalance dataset to have a big data characteristic.

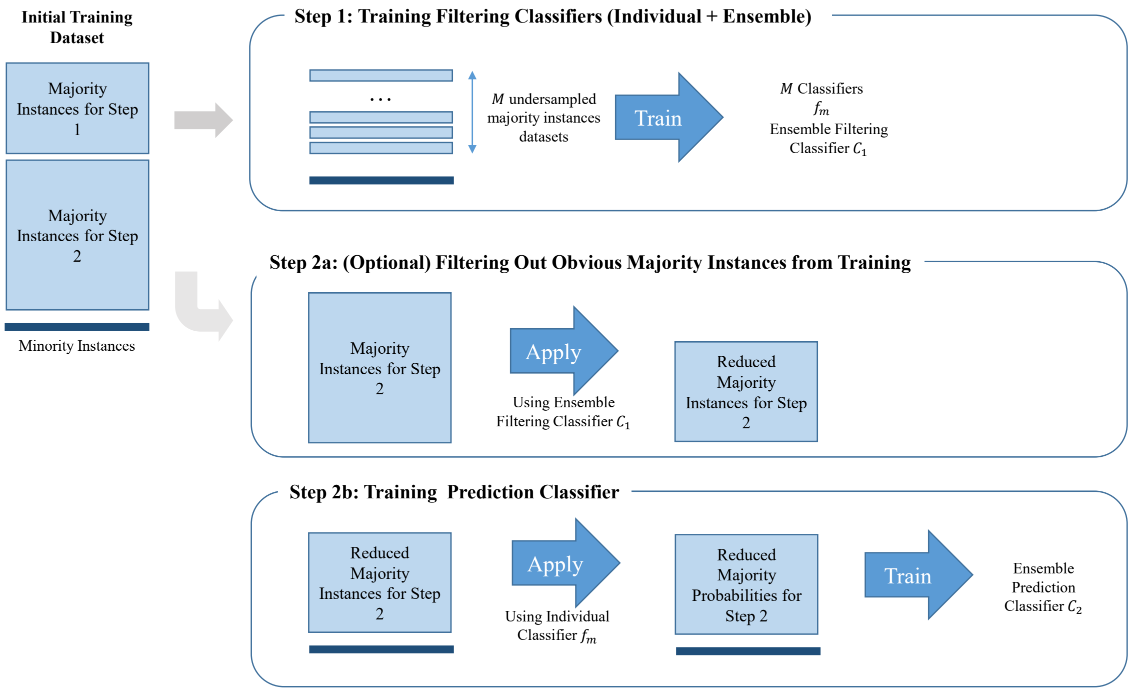
Initial Training Dataset
Majority Instances for Step 1
Majority Instances for Step 2
Minority Instances
Step 1: Training Filtering Classifiers (Individual + Ensemble)
M undersampled majority instances datasets
Train
M Classifiers f_m
Ensemble Filtering Classifier C_1
Step 2a: (Optional) Filtering Out Obvious Majority Instances from Training
Majority Instances for Step 2
Apply
Using Ensemble Filtering Classifier C_1
Reduced Majority Instances for Step 2
Step 2b: Training Prediction Classifier
Reduced Majority Instances for Step 2
Apply
Using Individual Classifier f_m
Reduced Majority Probabilities for Step 2
Train
Ensemble Prediction Classifier C_2

Figure 1. Architecture of our proposal.

We have chosen EasyEnsemble [21] as the model to be used in our first step. EasyEnsemble is an ensemble of AdaBoost sub-models where each uses randomly undersampled dataset. Amongst the traditional ensemble and hybrid boosting models , EasyEnsemble was the model reported to achieve the optimal performance indicator scores with the dataset that has higher imbalance ratio [14]. We also felt that there were obvious modifications that could be made to make the model more suitable in extreme imbalanced cases.

Formally speaking, let f_m, where $m = 1, ..., M$, be the mth AdaBoost sub-model that returns the probability of the data belonging to the minority class. Each f_m is trained with dataset D_m, randomly undersampled from the training dataset D_{train}, that has the same number of majority and minority instances. The model that makes the overall prediction, f, is the average of the probabilities from the sub-models, i.e.

$$f(X_i) = \sum_{m=1}^{M} f_m(X_i) \tag{2}$$

The overall classifier for our first step, C_1, is then

$$C_1(X_i) = \mathbb{1}_{f(X_i)>0.5} \tag{3}$$

where $\mathbb{1}$ is an indicator function.

3.3 Constructing the Training Dataset for the Prediction Model

The downfall of EasyEnsemble in extreme imbalanced class context is that many majority instances are left unused. Indeed, the ratio of sampled majority instances to the total is approximately N_0/N_1, meaning in extreme imbalanced scenario, we need we need significantly large number of AdaBoost sub-models (M parameter) to be averaged to utilise all the majority instances. However, the benefit by increasing the number of sub-models is minimal against the trade-off of memory and computational cost. Furthermore, we believe that EasyEnsemble can benefit from having another machine learning model which learns how to aggrgate the probabilities from each sub-models rather than just a simple averaging, given that we still have unused dataset to use.

Formally speaking, let $D'_{\text{train}} \subset D_{\text{train}}$ be the training dataset that has not been used to train EasyEnsemble model from the first step, i.e.

$$D'_{\text{train}} \cap \left(\cup_{m=1}^{M} D_m\right) = \emptyset \tag{4}$$

We note that D'_{train} does not have any minority instances in the set so fron this point onward, we also include all of the minority instances to D'_{train}.

Let X'_i be the feature vector in D'_{train} which can be used to obtain probabilistic outputs from the sub-models trained already, $f_m(X'_i)$ for $m = 1, ..., M$. These will be the feature vector for our new, reduced training dataset for our second-step model, E_{train}. In other words,

$$E_{\text{train}} = \left\{(F_k, y_k) \mid k \in I, F_k \in \mathbb{R}^M\right\} \tag{5}$$

where the jth entry of the feature vector F_k is $f_j(X_k)$ and I is the set of the indices of the dataset in D_{train} that were not used to train the first model.

3.3.1 *Filter Obvious Majority Instances*

As mentioned above, the utilisation of majority instances is very small in extreme imbalance, meaning size reduction in E_{train} compared to the original dataset D_{train} is still minimal. We would need to reduce the size further to make it feasible to use oversampling or class weight.

We recall that during our prediction, we use the first model to filter out the obvious majority instances. Therefore, there is no reason why we cannot use the first model to also filter out the obvious majority instances from E_{train} as the majority instances in this training dataset were not seen during the training stage. Filtered training dataset, E'_{train} would then be

$$E'_{\text{train}} = \{(F_k, y_k) \mid C_1(X_k) = 1, k \in I\} \tag{6}$$

We observed during our experiments that this step can significantly reduce the dataset size to the level where oversampling and weight balance become feasible option for the second model. Another benefit of this approach is that we only present the majorities that are more difficult to distinguish from the minorities which prevents the second model to lazily overfit on the obvious instances.

We note that the notion of filtering out the obvious majority instances in our proposal is similar to the BalanceCascade algorithm [21]. The key difference is the timing of the filtering: for BalanceCascade, filtering is performed a priori for each sub-models i.e., before they are ensembled whereas in our proposal, it is applied in a second step after the first model for filtering out obvious examples has been fully trained. We believe that this difference addresses the pitfall of the former approach. It is likely that the majority instances that are regarded as obvious by weak sub-models may have happened by chance rather than from truly learning the characteristics of obvious majority examples. As the sub-models trained afterwards do not have access to such majorities, they can suffer from missing out on the useful information. This explanation may also explain the poorer performance of BalanceCascade in harder tasks identified in [21].

3.4 Making Prediction on Test Dataset

Once we have two trained classifiers, C_1 and C_2, we can then make a prediction on the test dataset D_{test}. This is done in the same way as constructing the training dataset in 3.3 where you use the final prediction from C_2 only if the data is predicted as the minority by C_1. If no filter is applied, then we simply take the result from C_2, regardless of what C_1 has predicted.

4 Experiment

Table 1. Summary of the datasets used for the experiment.

Dataset	# Minority	# Majority	Imbalance Ratio
Cashout	674	11,317,134	16,791
Credit Fraud	492	284,807	578
Forest	2,747	578,265	211
Walking Activity	911	148,421	163
Abalone	32	4,142	129

4.1 Datasets

To evaluate our proposed modelling approach, we conducted the experiments on three datasets: two extremely imbalanced ones (Cashout and Credut Fraud) and three 'highly' imbalanced one (Walking Activity, Forest and Abalone19). We describe each one in further details in the following sections.

4.1.1 Cashout

A cashout error is defined as an erroneous payment sent to a given counterparty. Examples of error include sending the wrong amount, sending a payment to the wrong counterparty and late payment. Erros are mainly caused by so called fat finger (human processing error) or IT issues. The impact of such erroneous payments can include significant reputational damage. Whilst it is important to detect as many of them as possible, we also have to consider the operational cost associated with checking a large number of false positives. To the best of our knowledge, no research has been done related to such extreme imbalance ratios. For our experiment, we trained the model using the transaction dataset with labels from 3 consecutive months and tested our model on the following month.

4.1.2 Credit Fraud

This dataset was obtained from OpenML[3], which contains transactions record asscoaited with credit cards payments over two days in September 2013 [8]. The minority class in this dataset corresponds to fraudulent payments. We have split the dataset into half by the feature 'Time' where we have earlier half as the training and later half as the testing dataset.

4.1.3 Forest

This dataset was obtained from UCI Repository [10] which consists in information about different forest cover types for a given observation based on the US Geological Survey and US Forest Service data. There are 7 different forest cover types in total and to make this a binary classification problem with imbalance issue, we have relabelled the cover type 4, the least represented class, as the positive instance and the rest as the negative instances. The scores are the average of the five-fold cross validation via random selection.

4.1.4 Walking Activity

This dataset consists in a collection of data points from Android smartphones. It corresponds to accelerometer information from phones positioned in the chest pocket of their owners for 22 participants walking in the wild over a predefined path [3]. Similar to the Forest dataset, the target variable, the participant numbered between 1 to 22, has been transformed so that the participant 7, which had the least amount of the data, is labelled as the positive and the rest as the negative.

4.1.5 Abalone19

Abalone19 comes from the KEEL repository[4]. It is an imbalanced version of the Abalone dataset where minority examples belong to the class 19 [10].

Despite Abalone19 being the dataset with the highest imbalanced ratio for almost all of the research efforts that deal with 'highly' imbalanced datasets [11, 27], its imbalance ratio, along with Forest and Walking Activity data, is significantly lower than our scope of interest. However, we still include it within our experiments to determine whether our proposed approach remain relevant and also as a reference point. The scores are the average of five-fold cross validation

[3] https://www.openml.org/d/1597
[4] https://sci2s.ugr.es/keel/dataset.php?cod=115

via random selection. It is worth noting that there are only 32 minority instances in this dataset which is very small number for the machine learning model to learn from, meaning approximately 25 of them were present in the training dataset per fold. So even if the imbalance ratio is lower than the other dataset, we expect this dataset to yield lower scores.

4.2 Evaluation Metrics

By considering minority and majority instances as the positive and negative instances respectively, we have four possible outcomes from the prediction; True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). To quantify the performance of the model, we use precision, recall and F1-score as our metrics. The definitions of these metrics are as follows:

$$\text{Precision} = \frac{TP}{TP+FP} \tag{7}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{8}$$

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

4.3 Configuration

As mentioned previously, the first classifier used is EasyEnsemble, with $M = 100$ AdaBoost sub-models. We will use logistic regression (LR) as our second step model classifier, C_2, applied with filtering from 3.3.1 with one of undersampling, SMOTE and weight balance to handle the class imbalance in E'_{train}. For different class weight, we use the imbalance ratio as the weight of the minority. We also experiment with the option not to apply the filter on obvious majority instances (NF), although note that only undersampling has been used for this case due to the expensive computational cost for other methods. We benchmark our proposal against three traditional undersampling-based techniques; random undersampling(RUS), RUSBoost and EasyEnsemble(EE).

4.4 Result

The results from our experiments are provided in Table 2. For extreme imbalance dataset (Cashout and Credit Fraud), we observe that our proposal always achieves better F1 score than any of the traditional undersampling-based models regardless of the sampling techniques used in the second step. Even with just highly imbalanced dataset, our proposal with weight balance achieves the best F1 score. This is because from the minority class perspective, the dramatic improvement in the precision of the models occured with a comparatively small trade-off with the recall. This is exactly what we want from a practical standpoint; our proposal reduces the number of false positives to minimise the operational cost to carry out the verifications without losing too much of the ability to pick out the true positives. High recall but very low precision for the traditional undersampling-based approaches means they are unrealistic to be implemented in practice.

Table 3 demonstrates that this benefit comes from having second step model where we can observe the ratio of the increase in precision from first to second model larger than the ratio of reduction in recall for almost all cases. The strength of such characteristic is particularly highlighted in the credit fraud dataset.

We also note that the best F1 score across all datasets was achieved (with an exception for the Walking Activity) when we applied

Table 2. Results from the experiments rounded to 3 decimal places.

Cashout	Precision	Recall	F1
RUS	0.001	0.814	0.001
RUSBoost	0.001	0.725	0.001
EE	0.001	**0.857**	0.001
EE+LR (NF)	0.001	0.647	0.003
EE+LR (RUS)	0.003	0.349	0.006
EE+LR (SMOTE)	**0.004**	0.151	**0.007**
EE+LR (Weight)	0.003	0.403	0.006
Credit Fraud	Precision	Recall	F1
RUS	0.035	**0.900**	0.067
RUSBoost	0.041	0.888	0.078
EE	0.040	0.897	0.076
EE+LR (NF)	0.223	0.857	0.354
EE+LR (RUS)	0.743	0.816	0.778
EE+LR (SMOTE)	0.820	0.798	0.809
EE+LR (Weight)	**0.827**	0.794	**0.810**
Forest	Precision	Recall	F1
RUS	0.183	0.994	0.310
RUSBoost	0.150	0.979	0.261
EE	0.191	**0.998**	0.320
EE+LR (NF)	0.395	0.983	0.563
EE+LR (RUS)	0.653	0.951	0.774
EE+LR (SMOTE)	**0.660**	0.948	**0.778**
EE+LR (Weight)	0.654	0.950	0.774
Walking Activity	Precision	Recall	F1
RUS	0.280	0.998	0.426
RUSBoost	0.253	0.771	0.380
EE	0.259	0.998	0.411
EE+LR (NF)	0.461	**0.998**	0.616
EE+LR (RUS)	**0.935**	0.897	**0.976**
EE+LR (SMOTE)	0.910	0.969	0.939
EE+LR (Weight)	0.906	0.968	0.936
Abalone19	Precision	Recall	F1
RUS	0.015	0.656	0.029
RUSBoost	0.038	0.429	0.070
EE	0.020	**0.818**	0.040
EE+LR (NF)	0.018	0.593	0.034
EE+LR (RUS)	0.030	0.274	0.054
EE+LR (SMOTE)	0.031	0.098	0.047
EE+LR (Weight)	**0.045**	0.185	**0.072**

Table 3. Ratio of the increase in precision and reduction in recall between the first and second model within the pipeline in the extreme imbalance dataset. Benefit ratio is defined as the former divided by the latter.

Cashout	Increase in Pre.	Reduction in Rec.	Benefit
EE+LR (NF)	2.01	1.32	1.52
EE+LR (RUS)	3.87	2.46	1.57
EE+LR (SMOTE)	4.99	5.68	0.88
EE+LR (Weight)	4.30	2.13	2.02
Credit Fraud	Increase in Pre.	Reduction in Rec.	Benefit
EE+LR (NF)	6.35	1.05	6.05
EE+LR (RUS)	21.17	1.10	19.25
EE+LR (SMOTE)	23.37	1.13	20.68
EE+LR (Weight)	23.56	1.13	20.85

SMOTE or weight in the second step, unlike EasyEnsemble used in the first step where the architecture is based on random undersampling. We believe this observation is due to the different imbalanced data handling technique used between the first and second model step in the pipeline, thus reducing the overfitting of the model.

Table 4. Size of the extreme imbalanced training dataset at each step with imbalance ratio in bracket.

Dataset	First step	Second step (NF)	Second step
Cashout	9,012,239 (21,663)	8,605,650 (20,685)	859,972 (2187)
Credit Fraud	142,404 (528)	117,894 (437)	2611 (9)

Table 2 and 3 clearly demonstrate the be benefit of the filtering step before the second classification. By comparing no filter against the filter applied with RUS (selected the same data-sampling technique for fairness), F1 score and benefit ratio were larger with the filtering than without. We can also see from Table 4 that both the size of the training dataset and imbalance ratio reduces significantly when the filter is applied after the first step. This means also that we are able to apply SMOTE or weight balance more efficiently and thus allows for more variations in the balancing technique. Another interesting aspect linked to filtering is that the imbalanced ratio for Credit Fraud data actually went below the threshold for extreme imbalanced ratio after the filter is applied. This may explain why the performance on this dataset of our model is quite strong.

4.5 Discussion

Many of the works which tested their proposal on the dataset we have used only reported the Area Under the ROC Curve (AUC) as their performance metric so we could not benchmark our results against theirs. AUC score in extreme imbalanced setting is not suitable because it does not get affected by the disproportionate representation of the classes. This can mask the poor performances of the models [18], hence the avoidance of using AUC in our experiment.

We note that s2s reports an F1 score of 0.08 for the Abalone19 [11], which was 0.01 higher than our proposal. However, the improvement of their model is relatively limited considering the significant increase in computational cost due to the quadratic increase in the dataset size. Whilst the Abalone19 dataset is small enough for computers to handle the larger computational cost, it may not be straightforward for larger datasets such as Cashout and Credit Fraud, thus highlighting a clear advantage of our architecture.

Although we were able to maximise the use of the majority instances and made best effort to minimise the risk of overfitting, this meant that the same minority instances were shown to all of the sub-models and to the the second step model. The repeated use of the such instances can lead to the overfitting to these specific examples. Therefore, it would be interesting to think about how we can add variation for the minority instances, not just the imbalanced dataset handling technique. One possible method could be to mix the minority instances with the SMOTE samples in each subset of the data used to train the sub-models.

It will also be interesting to explore other architectures for the second-step model. The most natural next step from logistic regression would be neural network due to their similarities. Various works have been done to handle imbalance classes in neural network context, such as particle swarm optimisation (PSO) [2, 15] and apply them in the real-life dataset [23]. Given the big data characteristics of the extreme imbalanced dataset, aforementioned method could lead to further improvement.

Finally, we have applied undersampling process for the first model C_1 such that the number of the samples of the majorities matched the minorities exactly. Alternatively, we could look at increasing the imbalanced ratio for each sampled dataset and study the effect. This would mean that even more majority instances will be used during the first step which will lead to a further reduction in the imbalanced ratio for the second step. Moreover, there will be a smoother transition in the imbalance ratio between the first and the second step due to the smaller difference in the imbalanced ratio.

5 Conclusion

In this paper, we proposed a two-step model architecture to address the challenges related to extreme imbalance dataset. The purpose of the first model, the ensemble model, is to filter out the obvious majority instances, and doing so to reduce the size of the training dataset and the imbalance ratio. Second model then utilised the smaller, unused dataset to find a better way to combine the outcomes from the sub-models in the first step. This approach also allowed us to use different tactics that were not feasible with the initial dataset due to the computational cost. The experiments with the extreme imbalanced dataset demonstrated our proposal's ability to achieve significantly better balance between precision and recall than the existing method.

REFERENCES

[1] Rehan Akbani, Stephen Kwek, and Nathalie Japkowicz, 'Applying support vector machines to imbalanced datasets', in *European conference on machine learning*, pp. 39–50. Springer, (2004).
[2] Peng Cao, Dazhe Zhao, and Osmar R Zaïane, 'A pso-based cost-sensitive neural network for imbalanced data classification', in *Pacific-Asia conference on knowledge discovery and data mining*, pp. 452–463. Springer, (2013).
[3] Pierluigi Casale, Oriol Pujol, and Petia Radeva, 'Personalization and user verification in wearable systems using biometric walking patterns', *Personal and Ubiquitous Computing*, **16**(5), 563–580, (2012).
[4] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer, 'Smote: synthetic minority over-sampling technique', *Journal of artificial intelligence research*, **16**, 321–357, (2002).
[5] Nitesh V Chawla, Aleksandar Lazarevic, Lawrence O Hall, and Kevin W Bowyer, 'Smoteboost: Improving prediction of the minority class in boosting', in *European conference on principles of data mining and knowledge discovery*, pp. 107–119. Springer, (2003).
[6] David A Cieslak, Nitesh V Chawla, and Aaron Striegel, 'Combating imbalance in network intrusion datasets.', in *GrC*, pp. 732–737, (2006).

[7] Andrea Dal Pozzolo, Olivier Caelen, and Gianluca Bontempi, 'When is undersampling effective in unbalanced classification tasks?', in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 200–215. Springer, (2015).

[8] Andrea Dal Pozzolo, Olivier Caelen, Reid A Johnson, and Gianluca Bontempi, 'Calibrating probability with undersampling for unbalanced classification', in *2015 IEEE Symposium Series on Computational Intelligence*, pp. 159–166. IEEE, (2015).

[9] David J Dittman, Taghi M Khoshgoftaar, Randall Wald, and Amri Napolitano, 'Comparison of data sampling approaches for imbalanced bioinformatics data', in *The Twenty-Seventh International Flairs Conference*, (2014).

[10] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[11] Sri Harsha Dumpala, Rupayan Chakraborty, Sunil Kumar Kopparapu, and TCS Reseach, 'A novel data representation for effective learning in class imbalanced scenarios.', in *IJCAI*, pp. 2100–2106, (2018).

[12] Alberto Fernández, Sara del Río, Nitesh V Chawla, and Francisco Herrera, 'An insight into imbalanced big data classification: outcomes and challenges', *Complex & Intelligent Systems*, **3**(2), 105–120, (2017).

[13] Alberto Fernández, Salvador García, María José del Jesus, and Francisco Herrera, 'A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets', *Fuzzy Sets and Systems*, **159**(18), 2378–2398, (2008).

[14] Mikel Galar, Alberto Fernandez, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera, 'A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches', *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, **42**(4), 463–484, (2011).

[15] Piotr A Habas, Jacek M Zurada, Adel S Elmaghraby, and Georgia D Tourassi, 'Particle swarm optimization of neural network cad systems with clinically relevant objectives', in *Medical Imaging 2007: Computer-Aided Diagnosis*, volume 6514, p. 65140M. International Society for Optics and Photonics, (2007).

[16] Haibo He and Edwardo A Garcia, 'Learning from imbalanced data', *IEEE Transactions on knowledge and data engineering*, **21**(9), 1263–1284, (2009).

[17] Shengguo Hu, Yanfeng Liang, Lintao Ma, and Ying He, 'Msmote: improving classification performance when training data is imbalanced', in *2009 second international workshop on computer science and engineering*, volume 2, pp. 13–17. IEEE, (2009).

[18] László A Jeni, Jeffrey F Cohn, and Fernando De La Torre, 'Facing imbalanced data–recommendations for the use of performance metrics', in *2013 Humaine association conference on affective computing and intelligent interaction*, pp. 245–251. IEEE, (2013).

[19] Joffrey L Leevy, Taghi M Khoshgoftaar, Richard A Bauder, and Naeem Seliya, 'A survey on addressing high-class imbalance in big data', *Journal of Big Data*, **5**(1), 42, (2018).

[20] Wei-Chao Lin, Chih-Fong Tsai, Ya-Han Hu, and Jing-Shang Jhang, 'Clustering-based undersampling in class-imbalanced data', *Information Sciences*, **409**, 17–26, (2017).

[21] Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou, 'Exploratory undersampling for class-imbalance learning', *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, **39**(2), 539–550, (2008).

[22] Rushi Longadge and Snehalata Dongre, 'Class imbalance problem in data mining review', *arXiv preprint arXiv:1305.1707*, (2013).

[23] Maciej A Mazurowski, Piotr A Habas, Jacek M Zurada, Joseph Y Lo, Jay A Baker, and Georgia D Tourassi, 'Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance', *Neural networks*, **21**(2-3), 427–436, (2008).

[24] Chris Seiffert, Taghi M Khoshgoftaar, Jason Van Hulse, and Amri Napolitano, 'Rusboost: A hybrid approach to alleviating class imbalance', *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, **40**(1), 185–197, (2009).

[25] Shiven Sharma, Colin Bellinger, Bartosz Krawczyk, Osmar Zaiane, and Nathalie Japkowicz, 'Synthetic oversampling with the majority class: A new perspective on handling extreme imbalance', in *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 447–456. IEEE, (2018).

[26] Wei Wei, Jinjiu Li, Longbing Cao, Yuming Ou, and Jiahang Chen, 'Effective detection of sophisticated online banking fraud on extremely imbalanced data', *World Wide Web*, **16**(4), 449–475, (2013).

[27] Fei Wu, Xiao-Yuan Jing, Shiguang Shan, Wangmeng Zuo, and Jing-Yu Yang, 'Multiset feature learning for highly imbalanced data classification', in *Thirty-First AAAI Conference on Artificial Intelligence*, (2017).

[28] Zhi-Hua Zhou and Xu-Ying Liu, 'Training cost-sensitive neural networks with methods addressing the class imbalance problem', *IEEE Transactions on knowledge and data engineering*, **18**(1), 63–77, (2005).