International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol. 9, Issue.8, August 2021 E-ISSN: 2347-2693

Comparative Study of Techniques for Alleviating Class Imbalance in Spam Classification

Gopalkrishna Waja

IT Department, KJ Somaiya College of Engineering (KJSCE), Mumbai, India

Author's Mail Id: gopalkrishnawaja@gmail.com, Mobile: +91 8169012079

DOI: https://doi.org/10.26438/ijcse/v9i8.3845 | Available online at: www.ijcseonline.org

Received: 05/Aug/2021, Accepted: 10/Aug/2021, Published: 31/Aug/2021

***Abstract*—** Class Imbalance is inarguably one of the most significant and common problem faced while training supervised machine learning models to identify anomalies. In paradigms like spam filtering, medical diagnosis, intrusion detection etc. the amount of data available on negative class is much greater than that on the positive class and hence training traditional machine learning model on such data biases it in favor of the negative class at the cost of the positive class leading the model to give a false sense of accuracy and hence undermine its own purpose. Owing to the importance of this problem several techniques have been developed to tackle it and this paper is aimed to provide an empirical comparative evaluation of a gamut of these techniques to mitigate the adverse effect of class imbalance pertaining to spam classification. In this paper I have compared the effect of 8 resampling techniques including ROS, SMOTE, ADASYN, Near-Miss and TOMEK-LINKS on the performance of eight different learning classifiers which were selected cautiously to incorporate diverse strategies used for classification. In addition to this the performance of four Ensemble learning methods, including EasyEnsemble and SMOTEBoost, are contrasted when trained on an imbalanced dataset. The AUC-ROC performance metric calculated using a stratified 5-fold cross validation was used to evaluate the effect of different imbalance handling techniques. Furthermore, Statistical tests were performed on the results obtained to posit the best model for spam classification for the dataset used.

***Keywords*—**Imbalance, spam classification, resampling, ensemble learners, statistical test.

I. INTRODUCTION

In today's modern world E-mail is one of the most ubiquitously used and cost-effective methods for official communication and has been profoundly beneficial for the growth of various business sectors. However, over the years, emails have become the primary source of dissemination of spam and malicious contents. Spam mailing is primarily the distribution of bulk unsolicited messages [1]. As per [2] spammers, on an average, earn around 3.5 USD million every year by incurring monetary loss to both business and personal users. This has motivated researchers and academicians to proffer different email spam classification techniques for spam classification. Though, these proposed techniques have provided a fairly good performance they are still plagued by the Class imbalance problem and hence there lies an opportunity to improve them.

Class imbalance refers to a situation, where some classes are highly underrepresented compared to other classes [3]. In case of binary classification like spam identification the positive class is often underrepresented and hence this skew in the distribution makes it difficult for many conventional ML algorithms to predict effectively the examples belonging to the minority class. When trained on an unbalanced dataset, the accuracy of traditional ML models tends to be biased towards the majority class and therefore even though if the model is highly accurate it may not be correct. For example, consider a dataset which has an imbalance ratio of 99:1 (i.e., for every spam example there are 99 non spam example presents) then in such a situation if the model always predicts the label corresponding to the non-spam class, then it would have an accuracy of 99% despite the fact that the model is incorrect.

In this paper, I have compared several techniques mentioned in [4] and [5] aimed at mitigating the effects of class imbalance from the perspective of spam detection. For spam classification I have used a combination of Natural language processing and supervised learning models.

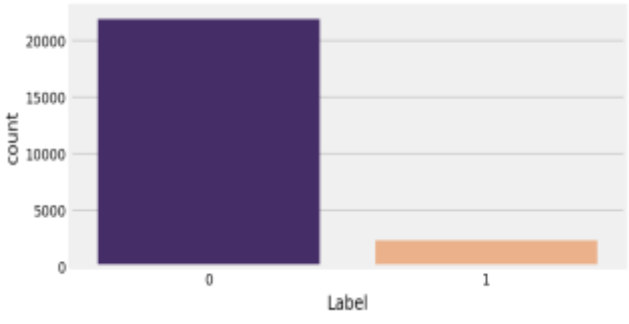

Figure 1. Class Distribution

The models have been trained on a dataset containing 24236 data items created by merging the SpamAssassin and Enron-1 dataset. This dataset has 21904 instances of non-spam examples and 2332 instances of spam examples giving an imbalance ratio of approximately 9:1 as shown in the Fig 1. In the upcoming sections I have explained the methodology used in detail and then compared the results obtained by application of eight resampling techniques and multiple classifier ensembles to analyze their ability to reduce the effect of the unbalance and posit the best model for Spam classification when trained on this dataset.

The rest of this paper is divided into the following sections: Related work provides a brief review of some of the previous work done on spam classification and dealing with the problem of class imbalance; Overview of Methodologies briefly explains the technique used for spam classification and also talks about the class imbalance handling techniques; Results and discussion, focuses on the results and detailed discussion of results obtained as the result of the methodology followed and the last sections provides the conclusion.

II. RELATED WORK

In [6] RAZA has performed a comprehensive analysis of different supervised, unsupervised and semi-supervised techniques for spam classification and has come to the conclusion that the supervised learning approach for spam classification gives a better performance than the other methods. In line with this ample research has been done over the last few years to automate spam classification systems using supervised machine learning [7]. In [8] Gomes and Saroar have proposed a combination of NLP with Naive Bayes Classifier and Hidden Markov Model while in [9] Junnarkar has performed vectorization using word2vec to learn the association between the words before using SVM for classification. Paper [10] has taken spam classification with supervised learning to a new level by combining NLP with ensemble learning where the overall ensemble learning classifier was made up of a combination of supervised and unsupervised classifiers. Though the supervised learning models proposed in these papers have been able to achieve a high performance in classifying the emails, they all in common face the problem of class imbalance. In terms of work done in using the techniques to improve model performance by handling class imbalance work done by E. M. Dogo in [5] stands out. In that paper the authors have compared several class imbalance reduction techniques to improve the performance in Water Quality Anomaly Detection and have come to the conclusion that combining oversampling techniques like SMOTE with missing value methods like miss forest and using them on customized DNN can significantly improve the performance. In [7] spam filtering on forums is discussed with the combination of SMOTE with Naïve Bayes, Decision Trees, Logistic regression and SVM and have come to the conclusion that Naïve bayes networks with SMOTE was the best approach. In [11] the authors have analyzed the impact of class imbalance in in Intrusion detection were the minority class attacks like R2L and U2R are highly misclassified solely due to the inadequate number of examples in the respective classes. In [12] the authors have explored the problem of imbalance in the credit rating domain and have significantly improved the performance by using SMOTE with SVC and C5.0. Similarly, researchers have explored the problem of class imbalance in several domains but there are not many who have performed a comprehensive comparison of various resampling and ensemble-based techniques for spam classification.

III. OVERVIEW OF THE METHODOLOGIES

A. RESAMPLING

Resampling is one of the most widely used technique for handling class imbalance and it aims to alter the dataset distribution to minimize the skewness in the class labels so as to effectively apprehend the decision boundary between the majority and the minority classes. Specifically, resampling techniques try to produce a dataset which is a reasonable approximation of the original dataset so as to make the classification of minority class easier. They reduce the effect of class imbalance, by either Undersampling the majority class, Oversampling the minority class, or using a hybrid approach of Undersampling and Oversampling. In addition to these methods, a data-level ensemble-based learning approach can also be effective for learning from an imbalanced dataset [13]. All these methods are explained below.

1) *Oversampling:* Oversampling refers to the technique of supplementing the dataset with additional instances of the minority class to reduce the imbalance between the two classes. The simplest and the most common form of Oversampling is Random Oversampling (ROS) where the instances from the minority class are randomly duplicated to reduce skewness in the data. Though this method is computationally inexpensive, it suffers from the problem of overfitting and hence to tackle this several other strategies can be used. Synthetic Minority Over Sampling technique (SMOTE) involves creation of 'synthetic' samples from existing minority instances by finding the K-neighbours of the instance and then randomly selecting a vector point between the current point and the K neighbours [14]. Adaptive Synthetic Sampling Approach (ADSYN) builds on SMOTE to use a weighted distribution for minority class to generate more synthetic examples for minority instances that are harder to learn compared to those which are easier to learn [15]. In the upcoming section I have compared the effect of all these techniques on the performance of supervised learning models and contrasted them with other resampling techniques.

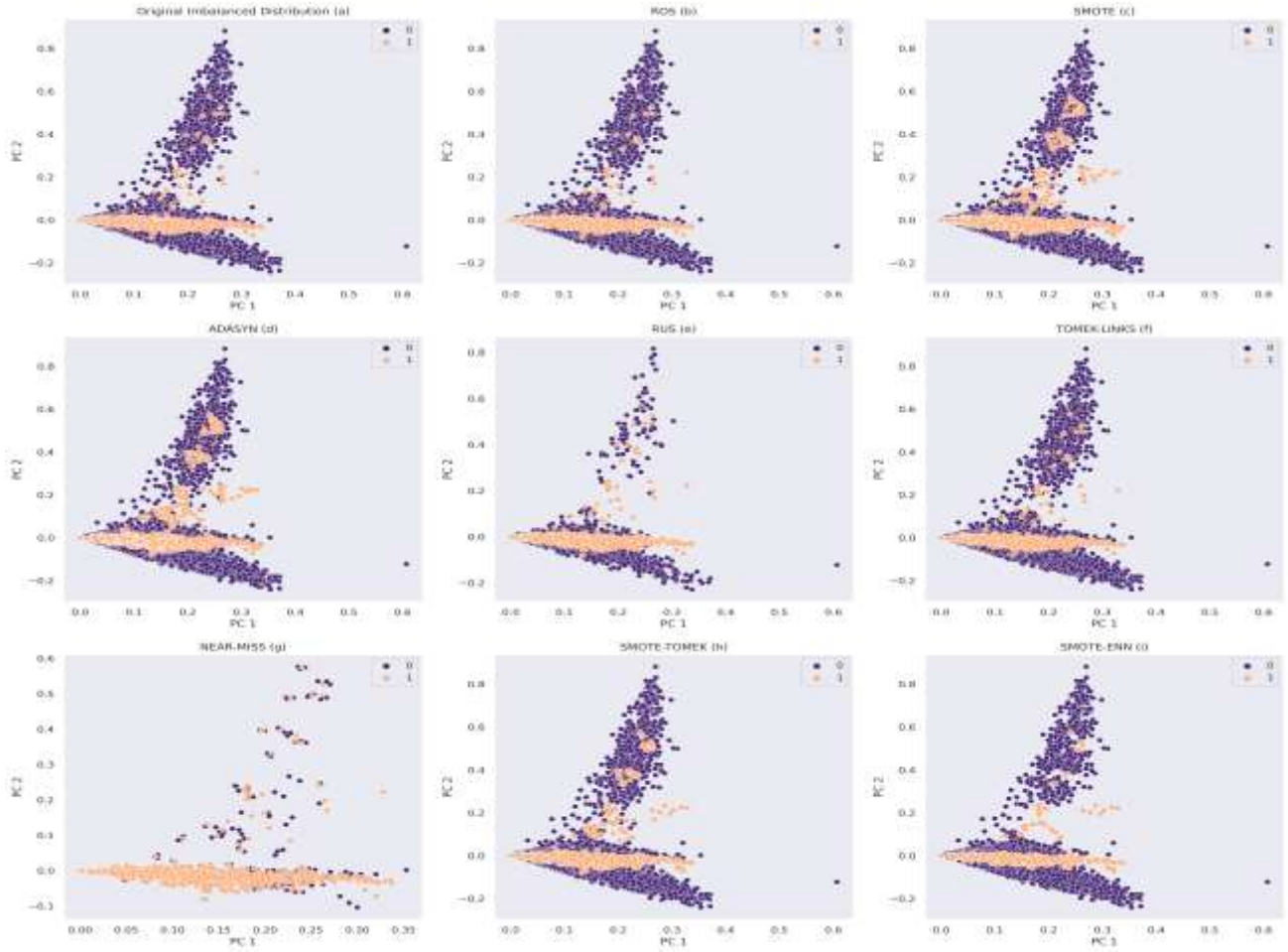

Figure 2. Training data distribution maps with the application of different resampling methods.
(a) Original Distribution [(0, 21904), (1, 2332)]; (b) ROS [(0, 21904),(1, 21904)]; (c) SMOTE [(0, 21904),(0, 21904)]; (d) ADASYN [(0, 21904),(0, 21906)]; (e) RUS [(0, 2332), (1, 2332)]; (f) TOMEK LINKS [(0, 21414), (1, 2332)] ; (g) Near-Miss [(0, 2332), (1, 2332)]; (h) SMOTE+TOMEK [(0, 14581), (1, 14581)]; (i) SMOTE+ENN [(0, 10263), (1, 11491)]

2) Undersampling: Undersampling refers to the technique used for shrinking the dataset by removing instances belonging to the majority class with the aim of reducing the skewness in data [16]. From the various available techniques, I have contrasted the effect of three Undersampling techniques on the imbalanced dataset and the model performance. The first technique is Random Undersampling (RUS) in which the majority class examples are discarded at random till balanced distribution is obtained. The advantage of this technique is that it is computationally inexpensive but the disadvantage is that sometimes random deletion of instances can lead to loss of critical information making learning of decision boundary more difficult. In order to circumvent this problem various other techniques have been developed, which remove only redundant, borderline and noisy data from the majority class, and retain all useful information. One of such a method is Tomek Links where cross-class nearest neighbours are found and eliminated to remove borderline, ambiguous and noisy examples making identifying the decision boundary much easier. Another Undersampling method I have compared is Near Miss which, unlike Tomek Links, selects examples to keep based on the distance of majority class examples to minority class examples and tends to only keep examples from the majority class that are on the decision boundary [16].

3) Hybrid sampling: As discussed earlier, excessive Oversampling can lead to overfitting, while excessive Undersampling can result in the loss of important information and hence several researchers like [17] and [18] have used a balanced hybrid mixture of both the techniques for applications like student performance classification and phishing URL classification. For my study I have compared two hybrid methods, SMOTE-TOMEK and SMOTE-ENN respectively. Here SMOTE is used for controlled Oversampling through synthetic sampling while techniques like TOMEK Links and ENN (Edit Nearest Neighbour) are used for downsampling. This combination of an Oversampling and an Undersampling technique increases or decreases the amount of data by just the right amount such that neither overfitting takes place nor there is any loss of information.

4) Ensemble-Based Methods: Ensemble based methods use the classification power of multiple weak learners, which are trained on separate sub-sets of the data, to augment the classification performance and hence ensemble learning leads to better accuracy and generalizability when compared to an individual classifier. Most of the data-level ensemble methods can be categorized, depending on whether they use a bagging or boosting framework. For this paper I have contrasted SMOTEBagging, RUSBagging, RUSBoost and EasyEnsemble. In SMOTEBagging each bootstrap sample is further Oversampled using SMOTE to achieve the desired balanced stage before providing it to the base classifier [19]. Similarly in case of RUSBagging instead of Oversampling the desired balance is achieved through Undersampling using RUS. For both SMOTEBagging and RUSBagging, the Decision tree has been used as the base classifier. RUSBoost is similar to the SMOTEBoost algorithm and is an improvement on the AdaBoost. RUS aids in balancing the class distribution, while AdaBoost ameliorates the performance of the classifier using these balanced data and since RUS is much faster than SMOTE comparable performance is received in with a much less training time [20]. Lastly, EasyEnsemble builds on the concept of RUSBoost by sampling several subsets from the majority class and training a learner using each of them, and combines the results of those learners [21].

B. SPAM CLASSIFICATION

As discussed earlier, I have used Resampling techniques for handling class imbalance and Natural Language processing in combination with Supervised machine learning for Spam classification. This section gives a detailed explanation of all the steps I have followed from pre-processing to final classification.

1) Pre-Processing: Pre-processing and cleaning the data involves performing operations on the raw text data to transform it in a proper format so that meaningful information can be extracted from it [22]. In this case, the following pre-processing was involved:

- Conversion of text into lowercase to make text consistent throughout the dataset.
- Removal of accented letters like \`a, \`e etc. by converting them to standard English alphabets.
- Expansion of contractions like 'I've' to 'I have', and 'don't' to 'do not'.
- Removal of special characters and non-alphanumeric characters.

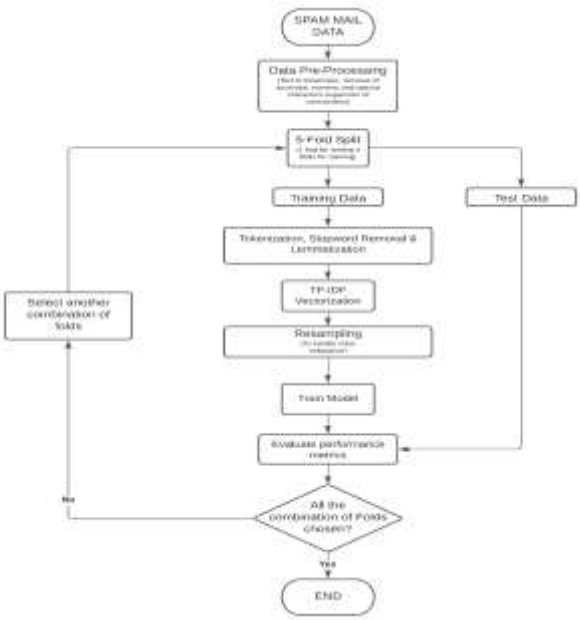

Figure 3. Flow for Spam Classification

2) Tokenization and Stopword Removal: This is the next step after initial pre-processing is done. Tokenization is the process of breaking the email body into individual characters, words or n-grams. The tokenizer makes use of white-space as delimiters for forming tokens from the text. Since in some research papers word tokenization is done [22] while in others n-grams are used [9], here, I have made use of both words and n-grams for tokenization. Stopwords refer to the most commonly used words in a language which generally do not provide any meaningful information required for the classification and hence these unwanted words can be removed and kept out of the corpus.

3) Lemmatization: Lemmatization aims to convert inflectional forms to a common base form. An email can contain, for grammatical reasons, different inflections such as playing, plays, played, etc of the same base word "play". Lemmatization makes use of dictionary words and morphological analysis on tokens, to remove inflections in a manner such that the resultant/base word is meaningful [9]. Lemmatization essentially reduces the number of words in the corpus and hence helps in dimensionality reduction and development of a better model.

4) Feature Extraction with TF-IDF: After Lemmatization the next step is Feature extraction through vectorization. Based on the comparison done in paper [8] between Bag of words and TF-IDF the TF-IDF technique gave a better accuracy and hence instead of using the traditional bag of words technique I have used the TF-IDF technique. TF-IDF technique involves calculating the Term Frequency and Inverse Document Frequency. The TF-IDF will assign weights to the terms such that the weight assigned is directly-proportional to the probability that word will appear in the document, and is inversely proportional to the number of documents in which the

words appear. Thus, TF-IDF helps to give weights to the terms based on their information content by penalizing the terms which occur most frequently through most of the documents and hence servers as a better technique for vectorization when compared to Bag of Words which solely rely of term frequency.

5) *Classification:* Supervised learning models can be categorized as parametric and non-parametric depending on the assumptions made by the model and this paper has made use of models from both the categories. I have compared the performance of eight different supervised learning models namely Logistic Regression, KNN, SVM, Naive Bayes, Decision Tree, Random Forest, LightGBM and Artificial Neural Network. Table 1 shows the parameters used for these models. These models have been selected to include a combination of different learning paradigms like linear, instance-based, tree-based, density-based, and neural network-based models in order to provide a holistic comparison and ensure a comprehensive assessment of the effects of the class imbalance on these models [23].

Table 1. Model Parameters

No.	Model	*Parameters*
1	Logistic Regression	Regularization= L2
2	KNN	K=7
3	SVM	Kernel= Linear
4	Naïve Bayes	Default
5	Decision Tree	Criterion= Entropy
6	Random Forest	No. of estimators= 100 Criterion= Entropy
7	LightGBM	Default
8	ANN	Hidden layer= 512 neurons Loss= Binary Cross Entropy Optimizer= Adam

C. PERFORMANCE METRICS

Performance metric provides us an idea about how well a model can predict an outcome given some input. A performance metric used to evaluate a model trained on a balanced model might not necessarily be ideal for evaluating the performance of a model trained on an imbalance dataset. As we have seen earlier, accuracy, which is widely used to evaluate performance of a model trained on relatively balanced datasets, becomes biased towards the majority class when used for a model trained on an unbalanced data. Therefore, it is quintessential to select an appropriate metric for model evaluation. For this paper, I have used AUC-ROC as the performance metric as it takes class distribution into consideration to provide a good measure of separability and is also one of the most popular metrics for imbalanced classification problems. In addition to this I have used a 5-Fold Cross Validation Strategy for to calculate these metrics in order to be sure that the model does not overfit the training data.

IV. RESULTS AND DISCUSSION

In this section, I have discussed the results obtained by using the aforementioned methodology for spam classification and then I have made use of the statistical tests approach mention in paper [5] involving Friedman test in paper [24] followed by the *post-hoc* Nemenyi test to validate the results obtained from the experiments conducted and select the best model for spam classification.

Table 2. AUC-ROC Measure with Resampling methods

			OVER-SAMPLING			UNDER-SAMPLING			HYBRID	
	Model	No Sampling	ROS	SMOTE	ADASYN	RUS	TOMEK	NM	SMOTE+ TOMEK	SMOTE+ ENN
1	Logistic Regression	0.840657	0.928357	0.941080	0.939785	0.925891	0.841127	0.910951	0.943345	0.705337
2	KNN	0.491257	0.758820	0.626510	0.624726	0.680007	0.680521	0.680296	0.625890	0.582177
3	SVM	0.905667	0.920812	0.918998	0.918763	0.930934	0.906465	0.921964	0.923709	0.806563
4	Naive Bayes	0.665961	0.923983	0.920638	0.929807	0.909854	0.663640	0.863047	0.925282	0.662742
5	Decision Tree	0.858108	0.880728	0.885620	0.873644	0.857282	0.854564	0.854760	0.882753	0.828287
6	Random Forest	0.850084	0.902915	0.914918	0.901417	0.927923	0.848358	0.890418	0.912125	0.833084
7	Light GBM	0.899191	0.924063	0.915798	0.918357	0.912302	0.897900	0.890224	0.918839	0.866287
8	ANN	0.937075	0.988428	0.987883	0.988406	0.986697	0.994693	0.967624	0.988490	0.985963

The Fig. 2 shows the results of application of different Oversampling, Undersampling and hybrid sampling techniques on the imbalanced dataset to produce eight different training sets which are used to train each of the eight-classifier mentioned in the previous section. We can see that in Fig. 2 (b)-(d) the oversampling techniques do not affect the distribution of the majority class while augmenting the minority class examples. ROS tends to give the same distribution pattern as the original dataset as here the minority examples are randomly duplicated causing a change in their density but not in pattern. For SMOTE and ADASYN we can see a change in density as well as distribution pattern of the minority class. The difference between the distribution pattern of SMOTE and ADASYN is quite subtle with the difference mainly visible at the boundary of the two classes. Fig. 2 (e)-(g) show the plots for Undersampling where the distribution of minority examples is kept constant. We can see in case of RUS (e) the density of the majority class is significantly reduced with examples mainly being removed from the upper

branch. In case of TOMEK-LINKS we can see that not many majority examples are removed and the distribution is very similar to the original distribution. Here, TOMEK-LINKS has focused on removal of only the examples which might have made classification difficult i.e., the borderline, ambiguous and noisy examples. Quite opposite to TOMEK-LINKS in Fig. 2 (e) for Near-Miss we can see a major decrease in the upper cluster of majority examples as only examples from the majority class that are on the decision boundary are kept. Fig. 2 (h)-(i) show the distribution for the hybrid methods which have ensured a balance between Oversampling and Undersampling and combination of patterns visible in SMOTE (b) and TOMEK-LINKS(f) are visible in SMOTE-TOMEK.

The Table 2 displays the AUC-ROC measures for all the eight models corresponding to different sampling techniques. It can see that SMOTE+TOMEK improves the AUC-ROC score for Logistic Regression by almost 10% by aiding the Logistic regression model to better classify the spam examples and minimizing miss classification at the boundaries. For KNN we can see that initially without sampling a very low score of 0.49125 was obtained and ROS gave the maximum increase from 75.882% indicating that KNN might not be a good model for this particular dataset. For SVM we can see that it give a reasonably good score of 0.905 even without the application of sampling techniques and this is because the SVM makes use of edge observations known as support vectors to find the hyperplane which separates the two classes and works well if reasonably good number of minority class are available and hence we see only a slight improvement in the performance on application of sampling techniques, in fact SMOTE+ENN leads to a reduction in the score probably because of removal of some support vectors by ENN which were used by SVM. In case of Naive Bayes, it observed that overall, the oversampling techniques work better that the undersampling techniques and ADASYN provides the highest score of 0.929807. For Decision Tree and Random Forest, it is seen that without sampling they give almost the same score but decision trees do not show any significant improvement in the score even on application of the resampling techniques while the Random Forest model show an improvement in the AUC-ROC score with RUS giving the maximum score of 0.927923, in case of LightGBM maximum increase in AUC-ROC score is seen for ROS with a score of 0.924063. Finally, it is observed that ANN gives the best overall performance with AUC-ROC score of 0.994693 for TOMEK-LINKS and hence it seems that this combination will be the best choice for this particular imbalanced dataset. However, before to reaching this conclusion I needed to make sure that the difference between different classifiers was statistically significant and were not merely a coincidence. In order to do this, I have made use of the Friedman test used in [5], [24]. In the Friedman test the null hypothesis maintains that there is no significant difference in the performance of the classifiers any difference is just a coincidence, while the alternate hypothesis is that there is a significant difference in performance of at least two of the classifiers being tested. If as the result of Friedman Test the null hypothesis is rejected then the post hoc Nemenyi test is performed in which we compare each classifier to the others to find which classifiers performances are statistically different from others. According to this test there is a significant variation in the performance of two classifiers if the difference between their average ranks more than a critical distance (CD) which not only depends on the performance metric but also the number of algorithms, sampling techniques and a critical value. It must be noted over here that each of eight models are trained on eight different independent datasets generated as the result of different sampling techniques and hence this test can be as comparing multiple classifiers over multiple datasets [5], [24].

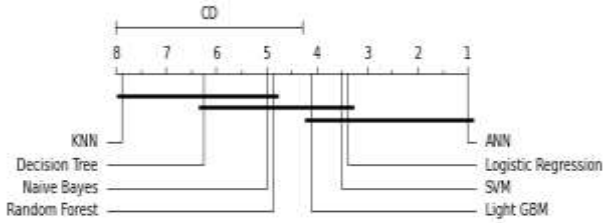

Figure 4 Average classifier rank for resampling technique

Performing the Friedman Test on the results obtained for degree of freedom = (8-1) *(8-1) = 49 and 0.05 as the level of significance, I received Friedman test statistic = 21.5416 and p-value = 0.003045. Therefore, this test showed that since p-value < 0.05 the null hypothesis is rejected and hence there were at least two classifiers who had a significant difference in their performance. Table 3 shows the Average rank obtained for each classifier during the Friedman Test. It can be observed from the ranks that ANN has rank 1 indicating that it has the best performance while KNN with a rank of 7.875 has the worst performance.

Table 3. Average Ranking of the classifier

Model	***Ranking***
Logistic Regression	3.375
KNN	7.875
SVM	3.5
Naïve Bayes	5.0
Decision Tree	6.25
Random Forest	4.875
LightGBM	4.125
ANN	1.0

Following this performing the post hoc Nemenyi test at 0.05 level of significance gave a Critical Distance (CD) = 3.712 and the Fig. 4 shows the set of classifiers that do not differ significantly connected with a bold horizontally line. We can see ANN is the best performing classifier with Rank 1 and its performance is significantly higher than the KNN, Decision Tree, Naive Bayes and Random Forest

classifiers shown on the left and hence these statistic tests indicate that ANN with TOMEK-LINKS is a potentially good candidate for spam classification for this particular dataset.

Table 4. AUC-ROC Measures for Ensemble Techniques

Model	*Ranking*
Smote Bagging	91.979578
RusBagging	88.908503
RusBoost	84.111377
EasyEnsemble	88.371849

Also, the Table 4 shows the AUC-ROC measures for Ensemble based techniques. The maximum AUC-ROC score obtained for Ensemble based techniques is 0.895725 for SMOTEBagging and is comparatively much less than the AUC-ROC score obtained for ANN. This can be attributed to the fact that these ensemble classifiers are sensitive to the presence of class overlaps, which leads to difficulty in distinguishing between the two classes because of nearly equal probabilities estimates of both classes.

V. CONCLUSION AND FUTURE SCOPE

Through this paper I was able to provide an empirical comparison between the effect of different Imbalance handling techniques on the model performance and was able to cogently use statistical tests to identify that ANN with TOMEK, which gave an AUC-ROC score of 0. 994693, was the best approach for spam classification for the imbalanced dataset used in this paper and was statistically better than the other models. In the results section it can be seen that for except SVM and decision trees the resampling techniques had an overall positive effect on the performance of the classifiers which was indicated by a more than 10% improvement in the AUC-ROC score of each classifier. Also it was seen that the general performance obtained by using ensemble methods was relatively low for this particular dataset when compared to best Oversampling, Undersampling and hybrid sampling techniques for different classifiers and hence from the results obtained from the comparative analysis done in the previous section I conclude that the best approach for spam classification using the given imbalanced dataset is to use ANN supervised learning model in combination with TOMEK-LINKS as the technique for handling class imbalance. Future work can include the evaluation of this technique on different imbalanced dataset and also evaluate the effect of the resampling techniques discussed in this paper on the performance of Deep Neural Network architectures.

REFERENCES

[1] A. D. R. F. Omar Saad, "*A survey of machine learning techniques for Spam filtering,*" International Journal of Computer Science and Network Security (IJCSNS), **Vol.12 No.2, p. 66, 2012.**

[2] A. Karim, S. Azam, B. Shanmugam, K. Kannoorpatti and M. Alazab, "*A Comprehensive Survey for Intelligent Spam Email Detection*," in IEEE Access**, vol. s7, pp. 168261-168295, 2019.**

[3] S. Wang and X. Yao, "*Multiclass Imbalance Problems: Analysis and Potential Solutions*," in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), **vol. 42, no. 4, pp. 1119-1130, 2012.**

[4] Leevy, J.L., Khoshgoftaar, T.M., Bauder, R.A. *et a* "*A survey on addressing high-class imbalance in big data,*" in Journal of Big Data, **Vol 5, pp.42, 2018.**

[5] E. M. Dogo, N. I. Nwulu, B. Twala and C. O. Aigbavboa, "*Empirical Comparison of Approaches for Mitigating Effects of Class Imbalances in Water Quality Anomaly Detection*," in IEEE Access, **vol. 8, pp. 218015-218036, 2020.**

[6] M. RAZA, N. D. Jayasinghe and M. M. A. Muslam, "A Comprehensive Review on Email Spam Classification using Machine Learning Algorithms," in the Proceedings of the 2021 International Conference on Information Networking (ICOIN), **pp. 327-332, 2021.**

[7] P. Ratadiya and R. Moorthy. "*Spam filtering on forums: A synthetic oversampling based approach for imbalanced data classification,*" in *CoRR 2019*, abs/1909.04826.

[8] S. R. Gomes *et al.*, "*A comparative approach to email classification using Naive Bayes classifier and hidden Markov model*," in the Proceedings of the 2017 4th International Conference on Advances in Electrical Engineering (ICAEE), **pp. 482-487, 2017.**

[9] A. Junnarkar, S. Adhikari, J. Fagania, P. Chimurkar and D. Karia, "*E-Mail Spam Classification via Machine Learning and Natural Language Processing,*" in the Proceedings of the 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), **pp. 693-699, 2021.**

[10] J. Fattahi and M. Mejri, "*SpaML: a Bimodal Ensemble Learning Spam Detector based on NLP Techniques*," in the Proceedings of the 2021 IEEE 5th International Conference on Cryptography, Security and Privacy (CSP), **pp. 107-112, 2021.**

[11] S. Rodda and U. S. R. Erothi, "*Class imbalance problem in the Network Intrusion Detection Systems,"* in the Proceedings of the 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), **pp. 2685-2688, 2016**

[12] L. Zhang and W. Wang, "*A Re-sampling Method for Class Imbalance Learning with Credit Data*," in the Proceedings of the 2011 International Conference of Information Technology, Computer Engineering and Management Sciences, **pp. 393-397, 2011.**

[13] G. Lemaitre, F. Nogueira, and C. Aridas, ‘‘*Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning,*’’ in Journal of Machine Learning Research., **vol. 18, no. 1, pp. 559–563, 2017.**

[14] S. Sharma, C. Bellinger, B. Krawczyk, O. Zaiane and N. Japkowicz, "*Synthetic Oversampling with the Majority Class: A New Perspective on Handling Extreme Imbalance*," in the Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM), **pp. 447-456, 2018.**

[15] Haibo He, Yang Bai, E. A. Garcia and Shutao Li, "*ADASYN: Adaptive synthetic sampling approach for imbalanced learning*," in the Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), **pp. 1322-1328, 2008.**

[16] F. Alberto, G. Salvador, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, "*Learning from imbalanced data sets"* Springer Science+Business Media, New York, **pp. 19-46 2018.**

[17] Y. Pristyanto, N. A. Setiawan and I. Ardiyanto, "*Hybrid resampling to handle imbalanced class on classification of student performance in classroom*" in the Proceedings of the 2017 1st International Conference on Informatics and Computational Sciences (ICICoS), **pp. 207-212, 2017.**

[18] Y. Pristyanto and A. Dahlan, "*Hybrid Resampling for Imbalanced Class Handling on Web Phishing Classification Dataset*," in the Proceedings of the 2019 4th International

Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), **pp. 401-406, 2019.**

[19] S. Ahmed, A. Mahbub, F. Rayhan, R. Jani, S. Shatabda and D. M. Farid, "*Hybrid Methods for Class Imbalance Learning Employing Bagging with Sampling Techniques*," in the Proceedings of the 2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS), **pp. 1-5, 2017.**

[20] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse and A. Napolitano, "*RUSBoost: A Hybrid Approach to Alleviating Class Imbalance,"* in IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, **vol. 40, no. 1, pp. 185-197, 2010.**

[21] A. Sarmanova and S. Albayrak, "*Alleviating class imbalance problem in data mining*," in the Proceedings of the 21st Signal Processing and Communications Applications Conference (SIU), **pp. 1-4, 2013.**

[22] S. R. a. V. F. a. N. J. Mirhoseini, "*E-Mail phishing detection using natural language processing and machine learning techniques*," in the Proceedings of the 7th National Congress of New Findings of in Electrical Engineering, Iran, **2021.**

[23] B. Twala and F. Mekuria, "*Ensemble multisensor data using state-of-the-art classification methods*," in the Proceedings of the 2013 Africon, **pp. 1-6, 2013.**

[24] J. Demšar, ''*Statistical comparisons of classifiers over multiple data sets*,'' in Journal of Machine Learning Research, **vol. 7, pp. 1–30, 2006.**

[25] S. Shumaly, P. Neysaryan and Y. Guo, "*Handling Class Imbalance in Customer Churn Prediction in Telecom Sector Using Sampling Techniques, Bagging and Boosting Trees*," in the Proceedings of the 2020 10th International Conference on Computer and Knowledge Engineering (ICCKE), **pp. 082-087, 2020**

[26] A. Abdullah ALFRHAN, R. Hamad ALHUSAIN and R. Ulah Khan, "*SMOTE: Class Imbalance Problem In Intrusion Detection System*," in the Proceedings of the 2020 International Conference on Computing and Information Technology (ICCIT-1441), **pp. 1-5, 2020.**

[25] P. Lim, C. K. Goh and K. C. Tan, "*Evolutionary Cluster-Based Synthetic Oversampling Ensemble (ECO-Ensemble) for Imbalance Learning*," in IEEE Transactions on Cybernetics, **vol. 47, no. 9, pp. 2850-2861, 2017.**

[26] Z. Yuan and P. Zhao, "*An Improved Ensemble Learning for Imbalanced Data Classification,"* in the Proceedings of the 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), **pp. 408-41, 2019.**

[27] S.S. Patil, S. P. Sonavane, "*Handling of Class Imbalanced Problem in Big Data Sets: An Experimental Evaluation (UCPMOT),*" International Journal of Computer Sciences and Engineering, **Vol.06, Issue.01, pp.1-9, 2018.**

[28] S. Rodda and U. S. R. Erothi, "*Class imbalance problem in the Network Intrusion Detection Systems*," in the Proceedings of the 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), **pp. 2685-2688, 2016.**

[29] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince and F. Herrera, "A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches," in *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, **vol. 42, no. 4, pp. 463-484, 2012.**

[30] H. He and E. A. Garcia, "Learning from Imbalanced Data," in *IEEE Transactions on Knowledge and Data Engineering*, **vol. 21, no. 9, pp. 1263-1284, 2009.**

[31] P. Yerawar, G. Pakle, "*A Survey of Different Techniques to Handle An Unbalanced Dataset*," in International Journal of Computer Sciences and Engineering, **Vol.6, Issue.12, pp.818-824, 2018.**

[32] Y. Zhang, G. Liu, W. Luan, C. Yan and C. Jiang, "*An approach to class imbalance problem based on stacking and inverse random under sampling methods*," in the Proceedings of *the* 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), **pp. 1-6, 2018.**

AUTHORS PROFILE

Mr. Gopalkrishna Waja is curently pursuing a bachelors degree in Information Technology form K.J Somaiya College Of Engineering affiliated to Mumbai University. His areas of expertise are Cryptography, Cybersecurity and Machine Learning and he has performed significant amount of research in application of machine learning to the domain of healthcare and security. Futhermore, after completion of his undergrad studies he will be pursuing a Masters in Computer with a specialization in AI and ML.

