Analyzing the Impact of Ensemble Techniques and Resampling Techniques Over Multi Class Skewed Datasets

[1]Rose Mary Mathew and [2]R Gunasundari
[1,2] Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore, India.
[1]rosem.mathew@gmail.com, [2]gunasoundar@gmail.com

ArticleInfo
R. Arulmurugan et al. (eds.), *First International Conference on Machines, Computing and Management Technologies*, Advances in Intelligent Systems and Technologies
Doi: https://doi.org/10.53759/aist/978-9914-9946-0-5_1

Abstract - Machine Learning is having great importance in this era, since of its board spectrum of applications and its capability to adjust and give solutions to complex problems reliably, rapidly, and productively. Machine learning models trained with the data from past experiences and based on the learned data it produces outcomes. The data used for training with these machine learning models should be in balanced manner otherwise the model gives incorrect results. Data is having an important role in this scenario, and it is evident that most of the data are skewed towards some classes and this kind of skewness can be found in all sectors of data in real world. Multimajority datasets and multiminority datasets are the different types of imbalances viewed in multiclass datasets. In this study three different datasets from multimajority domain and three different datasets from multiminority domain are analysed. Six different resampling procedure were applied out of which three belongs to undersampling and three belongs to oversampling. Four different classifiers K-NN, SVM, Random Forest and XGBoost were used to create the various models and their performance were analysed in this study.

Keywords - Imbalanced, Multiclass, Multimajority, Multiminority Oversampling, Undersampling

I. INTRODUCTION

Data skewness is commonly determined in many real-world cases, and this arise if the distribution of data across one of the classes is smaller (minority class) than other classes (majority class). The hassle of skewed distribution is generally related to misclassification. The minority class data is incorrectly classified with respect to the majority class[1]. A difficulty occurs when minority class data contains crucial records, and it became the spotlight of research so as fallacy in classification tends to fallacy in decision-making and in particular the accuracy of the minority class prediction.

To resolve problems in data skewness in binary classification, there are two approaches that should be performed. The different approaches are at data level and at algorithm level. Solution for data level is carried out via way of means of balancing the data distribution by resampling techniques via methods of undersampling, oversampling or combination of both methods. Solution for algorithm level is carried out via way of means of change in the classifier techniques or optimizing the overall performance of the learning algorithm[2]. The benefit of the data level approach is independent of the classifier selected.

Multi-class imbalanced concerns are observed as considerably more troublesome than the binary partners for various reasons. Skewness can appear in various ways in the case of a multi class dataset. Multi class skewness can be either one minority class with a couple of majority classes (Multi -Majority cases) or one majority class with a couple of minority classes (Multi -Minority cases). It is difficult to make an accurate prediction from the multiclass imbalanced datasets[3]. This was considered as a challenging issue from the past years. Many algorithms are available to tackle this issue. For handling multiclass imbalanced data class decomposition techniques were used. In this technique multiple classes are decomposed into combinations of binary classes and handle the imbalanced issue. This technique of partitioning has different approaches. The two approaches are one against all and one against one. These two techniques are combined to form a hybrid approach termed all and one[4]. Data level approaches can be applied on the existing dataset to minimize the effect of majority classes and minority classes[5]. By using the resampling techniques, we can minimize the issue.

In this paper multiclass imbalance datasets from various domains are analyzed. The primary aim of this work is to identify resampling method that can be used for making the dataset balanced and produce the accurate results. In this study four different resampling techniques were analyzed. The secondary aim is to identify the classifier which works best with these multiclass imbalanced datasets. The effect of ensemble algorithms in classification process is also analyzed.

In this session the basic concepts of data skewness in binary and multiclass scenario are specified. The second section of this paper focuses on the materials and methods used in this experiment which involves the details of datasets, resampling techniques, algorithms used in this study and the performance measures used for analysis. The third section

discusses the stages in experiment and the different results obtained with various resampling techniques and classifiers. The fourth section concludes the work with future scope of the work.

II. MATERIALS AND METHODS

This section of the paper addresses the description about the dataset used for the study, the different resampling techniques used to make the dataset balanced, machine learning algorithms used for study and the different performance measures that were considered for the assessment.

Dataset Description

For this study six datasets from various domains were chosen from the Keel data repository and UCI Machine Learning repository[6][7]. All these six datasets are multiclass imbalanced sets. Each dataset is categorised under the 3-class classification. Out of the six datasets three of them are treated as multimajority classes and three of them are treated as multiminority cases. These selected six datasets are having data which belongs to three different classes. The details of various datasets are available in the table, Table.1. Distribution plot of data across various datasets to show the skewness of classes are plotted in the figure Fig.1.

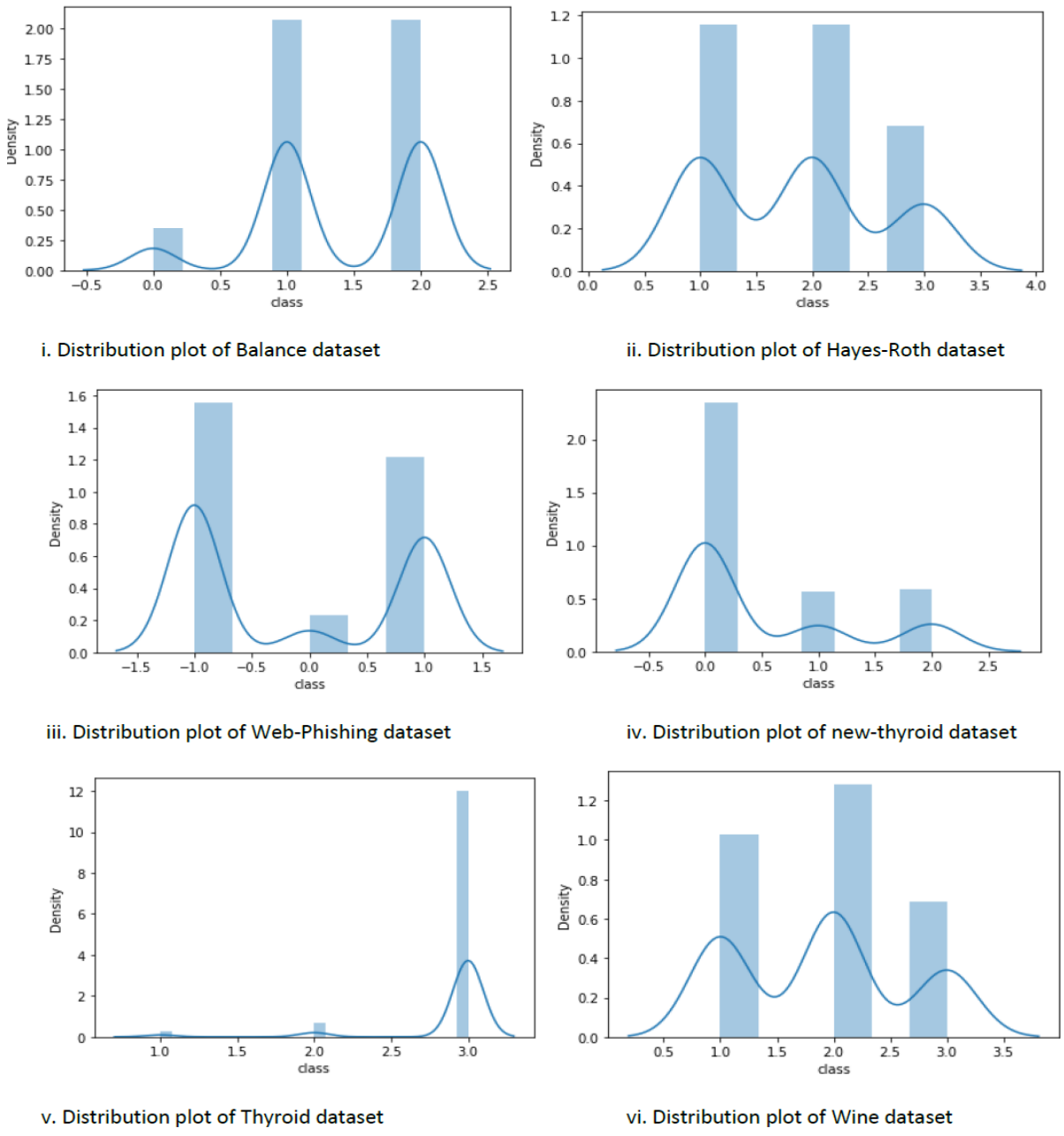

Fig 1. Represents the distribution plot of various datasets

Table 1. Description of Datasets

Name of the dataset	No. of Instances	Imbalance Ratio	No. of Attributes	Class Label	No. of records	Type of Imbalance
Balance	625	5.88	5	Balance	49	Multi-majority
				Left	288	
				Right	288	
Hayes-Roth	160	2.1	5	1	51	
				2	51	
				3	30	
Web-Phishing	1353	6	11	-1	702	
				0	103	
				1	548	
New - thyroid	215	5	6	Normal	150	Multi-minority
				Hyper	35	
				Hypo	30	
Thyroid	720	40.16	22	1	17	
				2	33	
				3	666	
Wine	178	1.48	14	1	59	
				2	71	
				3	48	

Resampling Techniques

To make the data set a balanced one resampling method were applied to the dataset during the preprocessing stage. This can be done by either adding synthetic samples to the dataset or by removing samples from the dataset. Addition of samples was done on minority class data and removal happened on majority class data. This procedure removed the skewness in the data. Adding new samples to the minority classes is termed as oversampling and eradicating samples from the majority classes is termed as undersampling.[2]

For this study we used six different resampling techniques like Random Oversampling, Random Undersampling, Synthetic Minority Oversampling Technique and Adaptive Synthetic sampling, Near Miss and Edited Nearest Neighbours to balance the skewed data.

Random OverSampling (ROS)

ROS is the lightest form of oversampling technique which is used in the data preprocessing stage. In this method the data from the minority classes are selected randomly and replicated to make a balance with the count of the majority class data[8]. This is applicable for all minority classes in the multiclass classification problem. Since the existing points are replicated there will be a possibility for increasing the overfitting of data.[9]

Random UnderSampling (RUS)

Like ROS, RUS is the lightest form of undersampling technique that exists. RUS is applied in the data preprocessing stage. In this technique the data in the majority classes are randomly selected and the chosen data are eradicated so that they must make a balance with the count of the minority class data[10]. This is applied for all majority classes in the multiclass classification problem. Since the original data points are eradicated that will leads to data loss in the learning process of the data [1].

Synthetic Minority Oversampling Technique (SMOTE)

SMOTE produces synthetic datapoints by photocopy the live datapoints. To generate new datapoints k-nearest neighbours' method is used[11]. The value of k in k-nn depends on the number of new datapoints created for making the dataset balanced. The distance between feature vector and neighbouring points are calculated with any of the available distance formulas. The variation in the distance is noted for different points and this variation is multiplied with a random value in the set (0,1). The value of the product is included to the feature vector as the new data value.

ADAptive Synthetic sampling (ADASYN)

Like SMOTE, ADASYN is an oversampling technique which produces artificial datapoints rather than duplicating the surviving datapoints[12]. The data points generated by this technique are harder to learn. ADASYN is an extended version of SMOTE in which artificial datapoints are produced with the aid of a weighted distribution of data in minority classes based on their hardness to learn them. This method generates new data which are more complex than other existing data. By using this strategy, the bias in the dataset can be reduced.

NearMiss

Near-miss is an under sampling technique. This technique randomly eliminates samples from the larger class. If two data values belong to different classes and are very close to one another in the data distribution, this strategy eliminates the datapoint of the majority class [8]. This technique finds the distance between all the points of majority class with the points in the minority class. Then it selects points of the majority class that is having the shortest distance with the points of the minority class[9]. These n points need to be taken for elimination.

Edited Nearest Neighbours

Edited Nearest Neighbours is used for finding ambiguous and noisy examples in a dataset. This algorithm uses nearest neighbours k=3 to locate the records in the dataset that are misclassified and that are eliminated before the k=1 classification rule is applied[13]. This is used as an under sampling method and the same can be applied to each record in the majority class, allowing those records that are wrongly classified as belonging to the minority class to be eliminated, and those correctly classified remains. This can be also applied to each record in the minority class where those records that are wrongly classified have their nearest neighbours from the majority class eliminated.

Classifiers Selected

For this study, four different classifiers are used to create the models. The different classifiers are K-NN, Support Vector Machines, Random Forest and XGBoost.

K-NN

K-Nearest Neighbour follows the lazy learning technique. This classifier performs well for predictive analysis[14]. In this technique for the test data 'k 'number of closest neighbours are identified, and the classes are identified for these k neighbours and the class occurs with high frequency is fixed as the class of test data. Fig.2 shows the representation of k-nn classification.

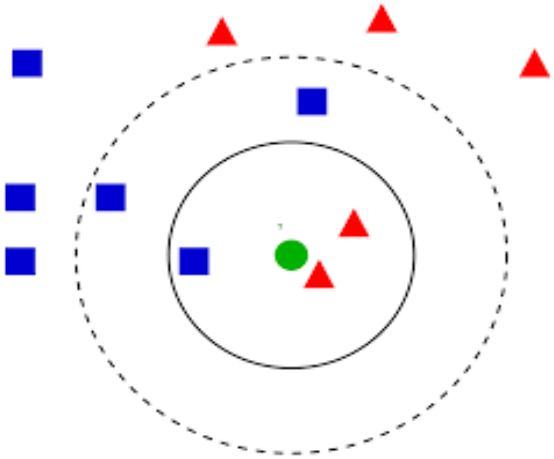

Fig 2. K-NN Classification [21]

Support Vector Machines

SVM is one of the most popularly used classifier. In SVM the algorithm generates a decision boundary for the data points and this decision boundary is called as hyperplane [15]. This hyperplane is made in such a manner that it keeps maximum distance away from the data values. This hyperplane is termed as maximum margin hyperplane(MMH)[16] . Fig.3 represents the graphical interpretation of SVM.

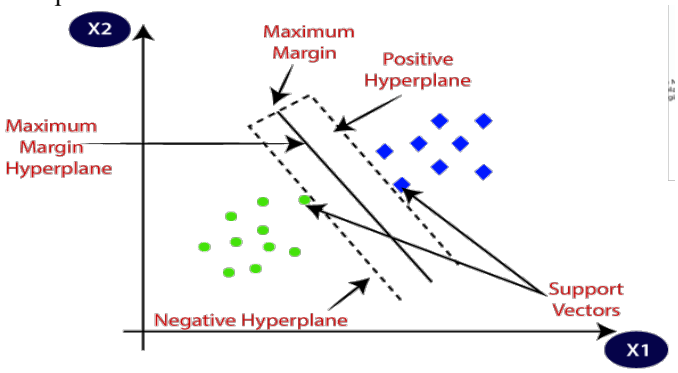

Fig 3. Support Vector Machine[22]

Random Forest

Random forest is an ensemble technique. This is very much popular and powerful classifier. Random forest is a collection of various tree structures. In this technique the random forest is collecting the results from all the decision trees, and it takes the average or finds a majority poll to identify the result. This algorithm definitively increases the predictive accuracy of the model[17]. Fig.4 shows the representation of random forest in general.

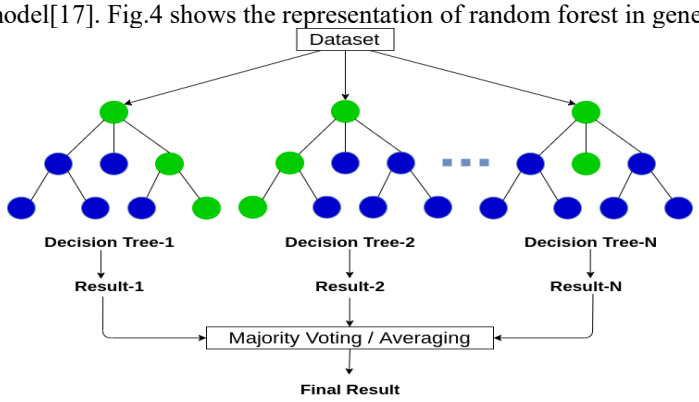

Fig 4. Random Forest [23]

XGBOOST

XGBOOST is an ensemble classifier based on gradient boosted trees algorithm. This algorithm combines the predictions from a set of weak classifiers and produces the final prediction. Speed and performance are comparatively better than other machine learning algorithms[18]. In this approach new models are generated which predicts the errors and residuals of the previous models and they added together to produce the result. This is termed as gradient boosting because it minimizes the loss when creating new models. The figure Fig.5 Shows working of XGBoost Algorithm.

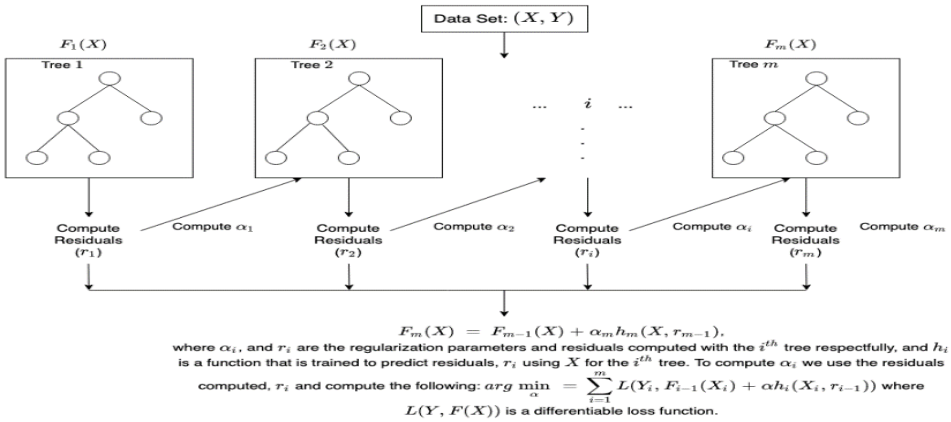

Fig 5. Working of XGBoost Algorithm [24]

Performance Measures

The different models are developed with a few resampling procedures and classifiers. The exhibition of these models is to be assessed. For doing the assessment, confusion matrices are utilized. Confusion matrices is based in actual values and predicted values. If the prediction and actual data are same, it is termed as correct classification otherwise it is termed as incorrect classification. The data present in the confusion matrix are True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN). By using these values, the accuracy of the model can be predicted. This experiment is on imbalanced datasets so that the performance can be evaluated with some other metrics like F1-Score, Precision, Recall, cross validation score, roc_auc curve.

Accuracy of the model is termed as the correctness of the model. The values in confusion matrix are used for calculating the accuracy.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Precision of a model is defined as the rate in which samples are correctly classified.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Recall is measuring the completeness of the results. It is called as the sensitivity of the model. It represents the rate in which positive samples are correctly classified.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

F1-Score is a measure which is obtained by combining precision and recall values. This is evaluated d as the harmonic mean between recall and precision [19].

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

roc_auc score is a metric that discusses the ability of the model to distinguish among the various classes. ROC is receiver operating characteristics which represents a probability curve and AUC (area under curve) represents the degree of separability.

III. EXPERIMENTAL STUDY AND RESULTS

This study is related to analyse the efficiency of various resampling techniques together with different machine learning algorithms over multiclass skewed data. The learning data for the model is from Keel data repository and UCI Machine Learning repository. Implementation of this work is done in the python language and the environment used for this work is performed in Pythons Jupyter Notebook[20]. The different stages of this work are represented in the figure. The Fig.6 represents the workflow of this experiment. The different steps that are followed in this study are,

i. Selection of dataset from the KEEL Repository. We choose three multimajority and three multi minority datasets of 3-class classification.
ii. Selected datasets undergone data pre-processing stage to remove the noise data.
iii. Spliting the dataset into training dataset and testing dataset in the ratio 80:20.
iv. Evaluate the performance of classifiers like k-NN, SVM, Random Forest and XGBoost.
v. Apply resampling techniques like ROS, RUS, SMOTE and ADASYN to the to the training dataset one by one.
vi. After applying each resampling technique, it evaluates the performance of the various classifiers like k-NN, SVM, Random Forest and XGBoost.
vii. Analyse the performance of these classifiers before applying resampling technique and after applying the resampling technique.

For this experimental study multiclass datasets are used. All the selected datasets are having three class labels. So, the predicted results will be one from any of these three classes. In multiclass classification process. the confusion matrix generated by the datasets are of the form 3X3. Performance metrics values are considered for all the class combinations. For the easiness of assessment, the average produced by the results are considered. The average data can be found using the library functions of Python.

For this experiment six different datasets from various domains were chosen for study. Out of these six, three are of multimajority and three are following multiminority case. The imbalance ratio of these datasets will vary from 1.48 to 40.16. That is work was performed on the different imbalanced scenarios. The detailed description of the datasets together with number of records, imbalance ratio, number of attributes, the various class labels present in the dataset, number of records following particular class label and the type of imbalance noticed in the dataset are represented in the following table.

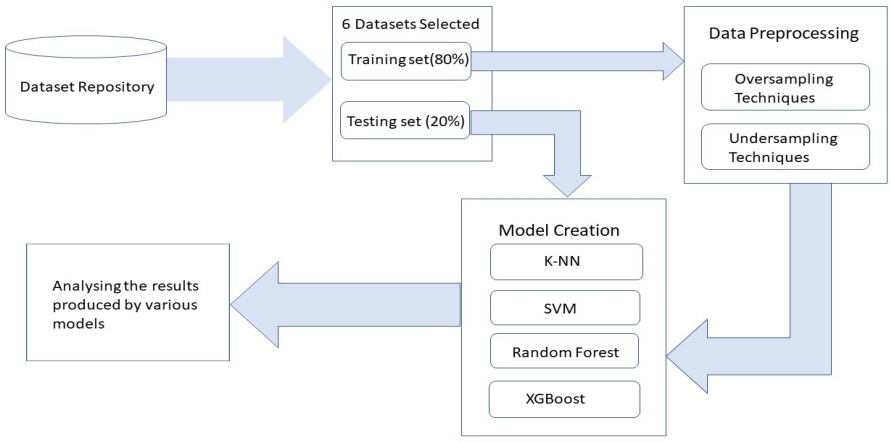

Fig 6. Shows the different stages in experimental study

The model performance can be evaluated using the confusion matrix. For each dataset the order of the confusion matrix is 3*3, as our dataset is having three class entry, The evaluation metrics like precision, recall, F-score and accuracy will be evaluated for these multiple classes. The detailed evaluation is shown in the following tables. The following table Table.1. is showing the Accuracy, Precision, Recall and F-score of the various classes by applying the different classifiers on original dataset.

Table 2. Represents the performance metrics of unbalanced datasets

Name of the dataset	Name of the classifier	Accuracy	Precision	Recall	F-Score
	K-NN	0.8	0.57	0.59	0.58
	SVM	0.89	0.597	0.66	0.63
	Random Forest	0.82	0.59	0.61	0.6
Balance	XGBoost	0.85	0.65	0.656	0.647
	K-NN	0.51	0.595	0.52	0.525
	SVM	0.81	0.849	0.83	0.829
	Random Forest	0.81	0.849	0.83	0.829
Hayes-Roth	XGBoost	0.81	0.835	0.83	0.83
	K-NN	0.834	0.857	0.747	0.78
	SVM	0.81	0.817	0.652	0.81
	Random Forest	0.867	0.888	0.849	0.867
Web-Phishing	XGBoost	0.885	0.875	0.86	0.868
	K-NN	0.9	0.956	0.84	0.88
	SVM	0.9	0.956	0.84	0.887
	Random Forest	0.93	0.94	0.898	0.918
New-Thyroid	XGBoost	0.95	0.977	0.88	0.92
	K-NN	0.937	0.869	0.519	0.584
	SVM	0.93	0.81	0.45	0.515
	Random Forest	0.97	0.82	0.92	0.86
Thyroid	XGBoost	0.979	0.86	0.99	0.918
	K-NN	0.97	0.97	0.969	0.97
	SVM	0.97	0.977	0.958	0.966
	Random Forest	0.97	0.976	0.969	0.97
Wine	XGBoost	0.97	0.977	0.958	0.966

In the previous table Table.2 the Precision, Recall and F-score value for the skewed datasets are very low. These models were not able to predict an event correctly. As the data is skewed, the results produced by these models were not trustworthy. For eliminating the skewness resampling should be used. Resampling techniques can appear into two ways, one is oversampling and other is under sampling. In this experimental study we choose four different techniques to evaluate the performance of skewed data. They are ROS, RUS, SMOTE and ADASYN.

In the world of machine learning it was assumed that data loss should affect the performance of model so that only one under sampling technique was chosen for study that is random under sampling (RUS). To preserve the original data, we chosen oversampling techniques for assessment.

The following table Table.3 shows the data distribution of various datasets before and after the application of resampling techniques like ROS, SMOTE, ADASYN and RUS. In Random Oversampling and SMOTE, both produces same number of samples after resampling. The resampling techniques makes the datasets almost balanced. In the case of oversampling, it performs oversampling in minority classes. In the case of under sampling it performs under sampling in majority classes.

Table 3. Represents the distribution of data over various classes before and after resampling

Dataset	Imbalanced data distribution			After Oversampling						After Under sampling					
				ROS/SMOTE			ADASYN			RUS/NearMiss			ENN		
	class 1	class 2	class 3	class 1	class 2	class 3	class 1	class 2	class 3	class 1	class 2	class 3	class 1	class 2	class 3
Balance	49	288	288	233	233	233	240	233	231	36	36	36	38	156	158
Hayes-Roth	51	51	30	41	41	41	41	41	42	23	23	23	9	6	23
Web-Phishing	702	103	548	565	565	565	565	565	435	82	82	82	430	82	290
New-thyroid	150	35	30	122	122	122	122	119	123	22	22	22	116	23	22
Thyroid	17	37	666	533	533	533	532	527	533	12	12	12	3	12	480
Wine	59	71	38	57	57	57	57	57	58	29	29	29	40	45	30

To find the effectiveness of resampling and classifiers we must consider the results of each dataset one by one. Firstly, we consider the multi majority datasets. Assessment of Balance dataset is considered first. The following table Table.4 shows the performance metric values for Balance dataset in the case of different resampling situations over various classifiers.

Table 4. Represents the performance metrics of Balance dataset

Name of the resampling	Name of the classifier	Accuracy	Precision	Recall	F-Score
RUS	K-NN	0.696	0.715	0.735	0.696
	SVM	0.744	0.76	0.81	0.7
	Random Forest	0.728	0.65	0.62	0.621
	XGBoost	0.76	0.687	0.68	0.665
ROS	K-NN	0.71	0.57	0.52	0.549
	SVM	0.87	0.808	0.8853	0.815
	Random Forest	0.816	0.591	0.607	0.599
	XGBoost	0.848	0.693	0.69	0.69
SMOTE	K-NN	0.736	0.61	0.587	0.596
	SVM	0.848	0.79	0.867	0.79
	Random Forest	0.808	0.59	0.6	0.596
	XGBoost	0.864	0.655	0.66	0.65
ADASYN	K-NN	0.736	0.629	0.607	0.611
	SVM	0.856	0.796	0.797	0.799
	Random Forest	**0.93**	**0.94**	**0.898**	**0.918**
	XGBoost	0.856	0.65	0.656	0.647
NearMiss	K-NN	0.67	0.698	0.677	0.625
	SVM	0.776	0.736	0.77	0.71
	Random Forest	0.656	0.616	0.566	0.57
	XGBoost	0.744	0.659	0.65	0.64
ENN	K-NN	0.83	0.659	0.658	0.657
	SVM	0.856	0.797	0.87	0.799
	Random Forest	0.768	0.626	0.61	0.616
	XGBoost	0.832	0.733	0.756	0.736

Table 5. Represents the performance metrics of Hayes-Roth dataset

Name of the resampling	Name of the classifier	Accuracy	Precision	Recall	F-Score
	K-NN	0.44	0.55	0.457	0.479
	SVM	0.81	0.86	0.83	0.81
	Random Forest	**0.888**	**0.9**	**0.9**	**0.899**
RUS	XGBoost	0.85	0.875	0.866	0.865
	K-NN	0.592	0.63	0.619	0.6
	SVM	0.81	0.849	0.83	0.829
	Random Forest	0.81	0.835	0.83	0.83
ROS	XGBoost	0.81	0.835	0.83	0.83
	K-NN	0.592	0.63	0.619	0.6
	SVM	0.81	0.849	0.83	0.829
	Random Forest	0.81	0.849	0.83	0.829
SMOTE	XGBoost	0.81	0.835	0.83	0.83
	K-NN	0.555	0.61	0.57	0.566
	SVM	0.81	0.849	0.833	0.829
	Random Forest	0.81	0.849	0.833	0.829
ADASYN	XGBoost	0.81	0.835	0.833	0.83
	K-NN	0.59	0.66	0.59	0.59
	SVM	0.703	0.733	0.733	0.733
	Random Forest	0.74	0.767	0.766	0.766
NearMiss	XGBoost	0.777	0.799	0.805	0.797
	K-NN	0.44	0.41	0.5	0.41
	SVM	0.518	0.533	0.566	0.5
	Random Forest	0.59	0.64	0.633	0.585
ENN	XGBoost	0.66	0.694	0.7	0.664

From table Table.4 of performance metrics of Balance dataset, the classification model with oversampling technique ADASYN together with the classifier Random Forest had the better performance. Accuracy, precision, recall and F1-score high for ADASYN resampled dataset with Random Forest classifier. Hayes-roth is another multimajority dataset that is in account. The following table Table.5 shows the performance metric values for Hayes-Roth dataset in the case of different resampling situations over various classifiers.

Table 6. Represents the performance metrics of Web-Phishing dataset

Name of the resampling	Name of the classifier	Accuracy	Precision	Recall	F-Score
	K-NN	0.69	0.63	0.7	0.62
	SVM	0.79	0.718	0.82	0.74
	Random Forest	0.84	0.78	0.88	0.81
RUS	XGBoost	0.848	0.786	0.885	0.81
	K-NN	0.79	0.72	0.798	0.745
	SVM	0.837	0.78	0.85	0.808
	Random Forest	0.867	0.889	0.86	0.875
ROS	XGBoost	0.88	0.87	0.86	0.866
	K-NN	0.79	0.72	0.827	0.75
	SVM	0.83	0.787	0.85	0.81
	Random Forest	0.87	0.88	0.878	0.879
SMOTE	XGBoost	0.87	0.867	0.865	0.866
	K-NN	0.782	0.707	0.81	0.729
	SVM	0.83	0.78	0.838	0.805
	Random Forest	0.859	0.87	0.857	0.86
ADASYN	XGBoost	**0.889**	**0.88**	**0.89**	**0.887**
	K-NN	0.58	0.525	0.567	0.515
	SVM	0.66	0.597	0.63	0.579
	Random Forest	0.77	0.696	0.78	0.718
NearMiss	XGBoost	0.79	0.717	0.81	0.74
	K-NN	0.8	0.755	0.72	0.73
	SVM	0.811	0.75	0.81	0.775
	Random Forest	0.86	0.865	0.87	0.867
ENN	XGBoost	0.87	0.828	0.9	0.855

For the Hayes-roth dataset the classification model with resampling technique RUS together with random forest algorithm produces the better performance. Random undersampling technique performs well for Hayes dataset. Random Forest classifier over undersampled data performed better than other models. The third multi-majority dataset under consideration is Web-Phishing dataset. The following table Table.6 shows the performance metric values for Web-Phishing dataset in the case of different resampling situations over various classifiers.

For Web-Phishing dataset the classification model with resampling technique ADASYN together with XGBoost algorithm produces the better performance. The boosting algorithm XGBoost over oversampled data with ADASYN produced better results.

Next, we take the new-thyroid dataset into consideration. This is a multiminority dataset. The following table Table.7 shows the performance metric values for new-thyroid dataset in the case of different resampling situations over various classifiers. For the new-thyroid dataset the classification model with all oversampling techniques together with ensemble algorithms produce better performance. Random forest algorithm and XGBoost shows similar performances in oversampled data of new-thyroid dataset.

Table 7. Represents the performance metrics of New-thyroid dataset

Name of the resampling	Name of the classifier	Accuracy	Precision	Recall	F-Score
	K-NN	0.95	0.975	0.888	0.92
	SVM	0.95	0.93	0.93	0.93
	Random Forest	0.977	0.95	0.986	0.967
RUS	XGBoost	0.977	0.95	0.986	0.967
	K-NN	0.95	0.939	0.96	0.947
	SVM	0.977	0.95	0.986	0.967
	Random Forest	**0.977**	**0.987**	**0.97**	**0.98**
ROS	XGBoost	0.95	0.975	0.948	0.959
	K-NN	0.977	0.95	0.986	0.967
	SVM	0.977	0.95	0.986	0.967
	Random Forest	**0.977**	**0.987**	**0.97**	**0.98**
SMOTE	XGBoost	**0.977**	**0.987**	**0.97**	**0.98**
	K-NN	0.977	0.95	0.986	0.967
	SVM	0.977	0.95	0.986	0.967
	Random Forest	**0.977**	**0.987**	**0.97**	**0.98**
ADASYN	XGBoost	**0.977**	**0.987**	**0.97**	**0.98**
	K-NN	0.86	0.83	0.907	0.845
	SVM	0.84	0.79	0.86	0.8
	Random Forest	0.84	0.82	0.9	0.822
NearMiss	XGBoost	0.84	0.816	0.89	0.82
	K-NN	0.909	0.95	0.83	0.88
	SVM	0.93	0.96	0.86	0.9
	Random Forest	0.93	0.96	0.86	0.9
ENN	XGBoost	0.909	0.95	0.837	0.88

Another multiminority dataset under consideration is thyroid dataset. The following table Table.8 shows the performance metric values for thyroid dataset in the case of different resampling situations over various classifiers. For the thyroid dataset the classification model with all oversampling techniques together with ensemble algorithm XGBoost produce better performance. XGBoost algorithm shows similar performances in oversampled data of thyroid dataset.

Another multiminority dataset under consideration is wine dataset. The following table Table.9 shows the performance metric values for wine dataset in the case of different resampling situations over various classifiers. For the wine dataset the classification model with all oversampling techniques produces similar results for all classifiers. Random forest algorithm shows good performances in oversampled and under sampled data of wine dataset.

Table 8. Represents the performance metrics of Thyroid dataset

Name of the resampling	Name of the classifier	Accuracy	Precision	Recall	F-Score
	K-NN	0.34	0.37	0.4	0.24
	SVM	0.368	0.366	0.48	0.255
	Random Forest	0.937	0.7	0.977	0.8
RUS	XGBoost	0.95	0.75	0.98	0.837
	K-NN	0.85	0.498	0.54	0.507
	SVM	0.88	0.56	0.608	0.568
	Random Forest	0.97	0.82	0.92	0.86
ROS	XGBoost	**0.979**	**0.86**	**0.99**	**0.918**
	K-NN	0.84	0.52	0.537	0.51
	SVM	0.909	0.62	0.679	0.64
	Random Forest	0.97	0.82	0.925	0.86
SMOTE	XGBoost	**0.979**	**0.86**	**0.99**	**0.918**
	K-NN	0.805	0.48	0.52	0.48
	SVM	0.9	0.61	0.67	0.63
	Random Forest	0.97	0.82	0.925	0.86
ADASYN	XGBoost	**0.979**	**0.86**	**0.99**	**0.918**
	K-NN	0.14	0.39	0.49	0.17
	SVM	0.17	0.33	0.467	0.15
	Random Forest	0.9	0.705	0.9	0.74
NearMiss	XGBoost	0.95	0.766	0.918	0.81
	K-NN	0.91	0.42	0.39	0.4
	SVM	0.93	0.64	0.39	0.43
	Random Forest	0.93	0.869	0.519	0.58
ENN	XGBoost	0.93	0.7	0.517	0.57

Table 9. Represents the performance metrics of Wine dataset

Name of the resampling	Name of the classifier	Accuracy	Precision	Recall	F-Score
	K-NN	0.97	0.966	0.979	0.97
	SVM	0.97	0.977	0.958	0.966
	Random Forest	**1**	**1**	**1**	**1**
RUS	XGBoost	0.976	0.98	0.96	0.97
	K-NN	**1**	**1**	**1**	**1**
	SVM	0.97	0.977	0.958	0.966
	Random Forest	**1**	**1**	**1**	**1**
ROS	XGBoost	0.97	0.977	0.958	0.966
	K-NN	**1**	**1**	**1**	**1**
	SVM	0.97	0.977	0.958	0.966
	Random Forest	**1**	**1**	**1**	**1**
SMOTE	XGBoost	0.97	0.977	0.958	0.966
	K-NN	**1**	**1**	**1**	**1**
	SVM	0.97	0.977	0.958	0.966
	Random Forest	**1**	**1**	**1**	**1**
ADASYN	XGBoost	0.97	0.977	0.958	0.966
	K-NN	**1**	**1**	**1**	**1**
	SVM	0.97	0.977	0.958	0.966
	Random Forest	**1**	**1**	**1**	**1**
NearMiss	XGBoost	**1**	**1**	**1**	**1**
	K-NN	**1**	**1**	**1**	**1**
	SVM	**1**	**1**	**1**	**1**
	Random Forest	**1**	**1**	**1**	**1**
ENN	XGBoost	**1**	**1**	**1**	**1**

The performance metrics of various resampled dataset with different algorithms are already specified. From the metric tables it was clear that oversampled data produces good results for all the datasets except Hayes-Roth. Hayes Roth dataset performs well with undersampled data. For all other datasets random undersampling, SMOTE and ADASYN performance was better. Oversampling techniques preserves all the data that are available in the dataset.

AUC - ROC curve is the measurement for assessing the performance of classification problems at different threshold values. ROC represents the probability curve and AUC specifies the proportion of separability. It addresses how much the model can distinguish classes. Higher the AUC score, the better the model is at prediction. AUC - ROC curve for the different dataset with best performance are plotted in the following figure. The model which had better performance with resampling technique together with machine learning algorithm for different datasets are shown in the figure Fig.7.

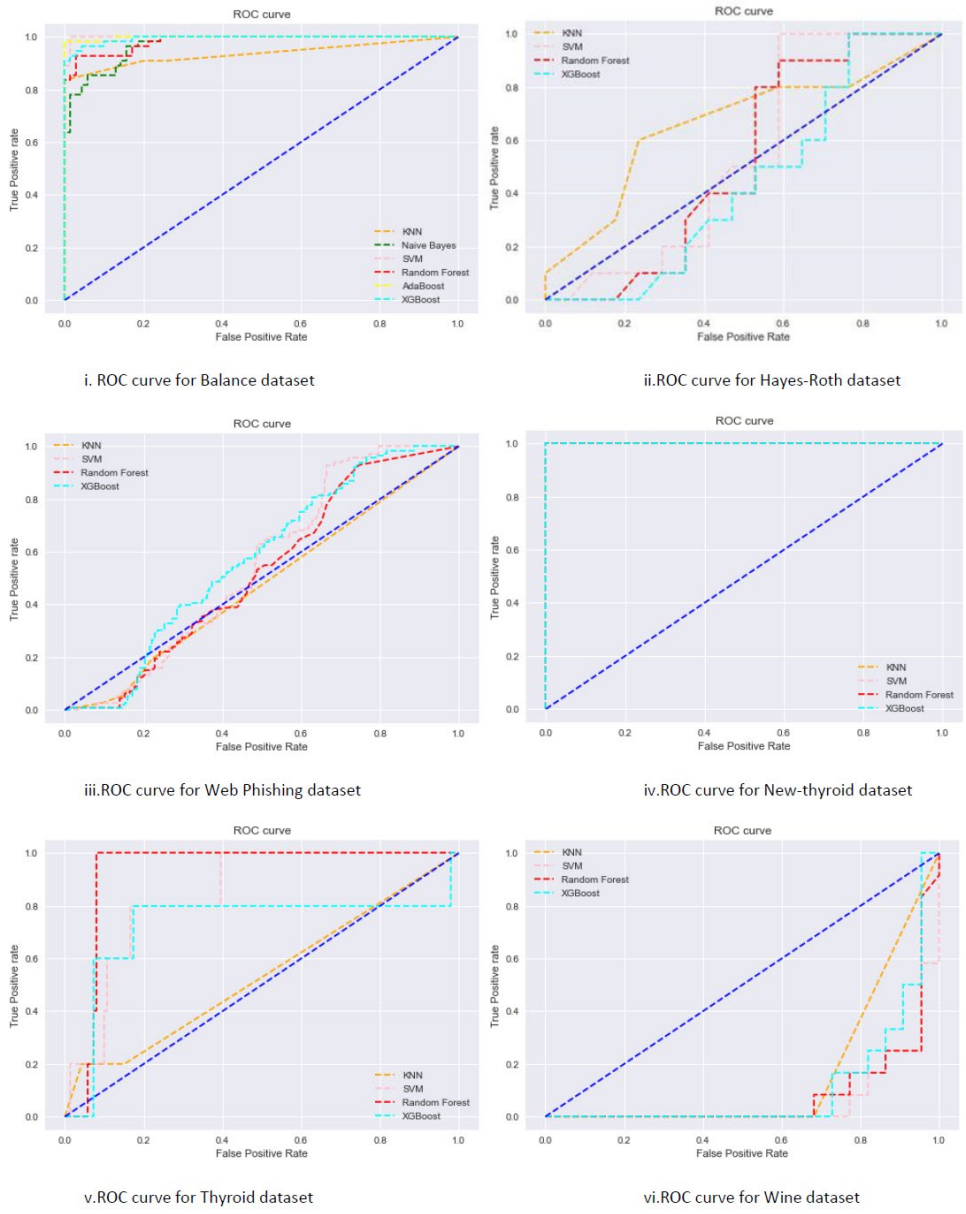

Fig 7. Shows the AUC-ROC curve for the different datasets

In this study, primary importance is given for resampling techniques and secondary importance is given for the classifiers. Multiclass imbalanced data from various domains are assessed in this study. It is higher to use oversampling approach to balance the data than making use of under sampling techniques. Under sampling strategies will every so often results in critical data loss and this may influence the overall performance of the model. Regarding the case of classifiers, the ensemble classifiers like random forest and XGBoost shows best in the classification process. Ensemble algorithms are performing better than the other machine algorithms.

IV. CONCLUSION

Data of most of the real-world scenarios are imbalanced for a few classes. For restructuring the data balanced it is higher to apply resampling strategies on the data during the data pre-processing stage. In this study, it is recognised that oversampling techniques like RUS, SMOTE and ADASYN produces higher results than the under sampled data. It is applicable to apply oversampling strategies on skewed data for making it balanced. Under sampling constantly results in data loss which can be truly important. This study looks at additionally proven that the classifiers overall performance stepped forward after making use of the resampling strategies. Ensemble algorithms like random forest and XGBoost outperforms other classifiers. Bagging technique in random forest algorithm makes the result an excellent one. Likewise boosting strategy in XGBoost also outperforms the other algorithms. It is highly recommended to use oversampling algorithm together with ensemble algorithms to achieve good results. As a future work, it is hoped that a hybrid method of resampling techniques and ensemble algorithm for classification of multi class imbalanced datasets.

References

[1]. S. Vluymans, "Learning from imbalanced data," in *Studies in Computational Intelligence*, vol. 807, Springer Verlag, 2019, pp. 81–110.

[2]. G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Systems with Applications*, vol. 73. Elsevier Ltd, pp. 220–239, May 01, 2017, doi: 10.1016/j.eswa.2016.12.035.

[3]. S. Wang and X. Yao, "Multiclass imbalance problems: Analysis and potential solutions," *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 42, no. 4, pp. 1119–1130, 2012, doi: 10.1109/TSMCB.2012.2187280.

[4]. Y. Pristyanto, I. Pratama, and A. F. Nugraha, "Data level approach for imbalanced class handling on educational data mining multiclass classification," in *2018 International Conference on Information and Communications Technology, ICOIACT 2018*, 2018, vol. 2018-Janua, doi: 10.1109/ICOIACT.2018.8350792.

[5]. R. M. Mathew and R.Gunasundari, "A review on handling multiclass imbalanced data classification in education domain," in *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE*, 2021, pp. 752–755, doi: 10.1109/ICACITE51222.2021.9404626.

[6]. J. Alcalá-Fdez *et al.*, "KEEL: A software tool to assess evolutionary algorithms for data mining problems," *Soft Comput.*, vol. 13, no. 3, pp. 307–318, 2009, doi: 10.1007/s00500-008-0323-y.

[7]. J. Alcalá-Fdez *et al.*, "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *J. Mult. Log. Soft Comput.*, vol. 17, no. 2–3, pp. 255–287, 2011.

[8]. V. S. Spelmen and R. Porkodi, "A Review on Handling Imbalanced Data," *Proc. 2018 Int. Conf. Curr. Trends Towar. Converging Technol. ICCTCT 2018*, no. December, pp. 1–11, 2018, doi: 10.1109/ICCTCT.2018.8551020.

[9]. V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Inf. Sci. (Ny).*, vol. 250, pp. 113–141, 2013, doi: 10.1016/j.ins.2013.07.007.

[10]. R. M. Mathew and R.Gunasundari, "AN EXPERIMENTAL STUDY ON THE EFFECT OF RESAMPLING TECHNIQUES IN MULTICLASS IMBALANCED DATA IN LEARNING SECTOR," *Des. Eng.*, no. 8, pp. 16216–16231, 2021, [Online]. Available: http://www.thedesignengineering.com/index.php/DE/article/view/6768.

[11]. A. Fernández, S. García, F. Herrera, and N. V. Chawla, "SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary," *Journal of Artificial Intelligence Research*, vol. 61. AI Access Foundation, pp. 863–905, Apr. 01, 2018, doi: 10.1613/jair.1.11192.

[12]. A. De and N. Do, "Techniques to deal with imbalanced data in multi-class problems : A review of existing methods," 2020.

[13]. X. Ai, J. Wu, V. S. Sheng, P. Zhao, and Z. Cui, "Immune centroids oversampling method for binary classification," *Comput. Intell. Neurosci.*, vol. 2015, 2015, doi: 10.1155/2015/109806.

[14]. Y. Pristyanto, N. A. Setiawan, and I. Ardiyanto, "Hybrid resampling to handle imbalanced class on classification of student performance in classroom," *Proc. - 2017 1st Int. Conf. Informatics Comput. Sci. ICICoS 2017*, vol. 2018-Janua, pp. 207–212, 2017, doi: 10.1109/ICICOS.2017.8276363.

[15]. B. S. Raghuwanshi and S. Shukla, "Class imbalance learning using UnderBagging based kernelized extreme learning machine," *Neurocomputing*, vol. 329, pp. 172–187, Feb. 2019, doi: 10.1016/j.neucom.2018.10.056.

[16]. X. Li, S. Wu, X. Li, H. Yuan, and D. Zhao, "Particle Swarm Optimization-Support Vector Machine Model for Machinery Fault Diagnoses in High-Voltage Circuit Breakers," *J. Mech. Eng*, vol. 33, p. 6, 2020, doi: 10.1186/s10033-019-0428-5.

[17]. Y. Pristyanto, A. F. Nugraha, I. Pratama, and A. Dahlan, "Ensemble Model Approach for Imbalanced Class Handling on Dataset," *2020 3rd Int. Conf. Inf. Commun. Technol. ICOIACT 2020*, pp. 17–21, 2020, doi: 10.1109/ICOIACT50329.2020.9331984.

[18]. S. González, S. García, J. Del Ser, L. Rokach, and F. Herrera, "A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities," *Inf. Fusion*, vol. 64, no. May, pp. 205–237, 2020, doi: 10.1016/j.inffus.2020.07.007.

[19]. E. Mortaz, "Imbalance accuracy metric for model selection in multi-class imbalance classification problems," *Knowledge-Based Syst.*, vol. 210, Dec. 2020, doi: 10.1016/j.knosys.2020.106490.

[20]. R. M. Mathew and R.Gunasundari, "Techniques and Tools to Tackle Imbalanced Learning," *Karpagam J. Comput. Sci.*, vol. 16, no. 3 May-June2021, 2021, [Online]. Available: https://karpagampublications.com/archives-kjcs/paper-list-may-june-2021/.

[21]. Available at https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/

[22]. Available at https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm

[23]. Available at https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/

[24]. Available at https://docs.aws.amazon.com/sagemaker/latest/dg/xgboost-HowItWorks.html.