International Journal of Electrical and Computer Engineering (IJECE)
Vol. 13, No. 4, August 2023, pp. 4357~4368
ISSN: 2088-8708, DOI: 10.11591/ijece.v13i4.pp4357-4368

An advance extended binomial GLMBoost ensemble method with synthetic minority over-sampling technique for handling imbalanced datasets

Neelam Rout[1], Debahuti Mishra[1], Manas Kumar Mallick[2]
[1]Computer Science and Engineering, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, India
[2]Electrical Engineering, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, India

Article Info

Article history:

Received Jun 30, 2022
Revised Oct 29, 2022
Accepted Nov 6, 2022

Keywords:

Boosting ensemble method
generalized linear model
Extended binomial method
Imbalance data
Performance metrics
Synthetic minority oversampling technique
Wilcoxon test

ABSTRACT

Classification is an important activity in a variety of domains. Class imbalance problem have reduced the performance of the traditional classification approaches. An imbalance problem arises when mismatched class distributions are discovered among the instances of class of classification datasets. An advance extended binomial GLMBoost (EBGLMBoost) coupled with synthetic minority over-sampling technique (SMOTE) technique is the proposed model in the study to manage imbalance issues. The SMOTE is used to solve the proposed model, ensuring that the target variable's distribution is balanced, whereas the GLMBoost ensemble techniques are built to deal with imbalanced datasets. For the entire experiment, twenty different datasets are used, and support vector machine (SVM), Nu-SVM, bagging, and AdaBoost classification algorithms are used to compare with the suggested method. The model's sensitivity, specificity, geometric mean (G-mean), precision, recall, and F-measure resulted in percentages for training and testing datasets are 99.37, 66.95, 80.81, 99.21, 99.37, 99.29 and 98.61, 54.78, 69.88, 98.77, 96.61, 98.68, respectively. With the help of the Wilcoxon test, it is determined that the proposed technique performed well on unbalanced data. Finally, the proposed solutions are capable of efficiently dealing with the problem of class imbalance.

Corresponding Author:

Neelam Rout
Computer Science and Engineering, Siksha 'O' Anusandhan (Deemed to be) University
J-15, Khandagiri Marg, Dharam Vihar, Jagamara, Bhubaneswar, Odisha 751030, India
Email: neelamrout@yahoo.com

1. INTRODUCTION

The objective of classification is to study about input and target variables. Imbalanced can hinder the performance of the classification algorithms. Imbalanced problems outline a scenario when the number of examples of the majority class far more that of the minority class [1], [2]. In data mining imbalanced classification is an attentive topic to research. Some of the real-world problems are fraud detection, fault classification in manufacturing, text classification, disease diagnosis, event classification, oil spill detection, and intrusion detection [3]–[9]. Ensemble methods can handle the problems of imbalanced data [10]. Ensemble algorithm is used to integrate multiple classifiers into a single classifier to get more accurate results [11], [12]. A collection of techniques that can transform weak learners into strong ones are referred to as the "boosting ensemble method". A powerful learner is very close to fine performance, whereas a weak learner is only marginally finer than a random utterance on the surface. Boosting is a sequential ensemble strategy for reducing bias error and constructing powerful prediction models. The term "boosting" refers to a collection of algorithms

that help a weak learner become a strong learner. As the performance metrics accuracy and its complement, misclassification rate is not worked well on imbalanced data, so other metrics should take into consideration like geometric mean (G-mean)[13]. In this study, a generalized liner model (GLM) is used due to its good performance quality [14], [15]. Also, synthetic minority oversampling technique (SMOTE) approach is applied to obtain the new training set [16]. So, the Boosting GLM [17] method is proposed for the imbalanced datasets.

To get better insight into dealing with imbalanced data issues, few literatures have been studied and are discussed and reviews of handling imbalance datasets are described. Alibeigi *et al.* [18] have presented the advantages of genetic algorithm with neural classifiers in the ensemble method for the electrocardiography registration. In this [19] paper a new "synthetic informative minority over-sampling (SIMO) algorithm" with leveraging "support vector machine (SVM)" is proposed to show the best performance. A powerful ensemble classification algorithm called random under sampling Boost is suggested to handle the inequality problems [20]. To solve class [21] imbalance problems, a new weighted SVM is proposed using 10 types of imbalanced datasets and analyze the proposed method with SVM and other SVMs hybrid with sampling and boosting techniques. The article is focused on reducing the time by proposing a neural networks classifier (pattern recognition) for inaccurate data [22]. In the paper [23], Di Martino *et al.* have presented a new method developed for imbalanced problems to maximize F-measure. In [24], an improved ensemble method based on under-sampling, is helpful for the ensemble system and pruning procedure is included to remove the irrelevant models. Błaszczyński and Stefanowski [25] proposed a method called neighborhood balanced bagging, in which the sample probabilities of examples are adjusted based on the distribution of classes in the neighborhood. An integrated sampling method is proposed [26], which uses both the over-sampling and under-sampling and addition with an ensemble SVMs to increase the performance percentage of the classifier. A novel ensemble method [27] with specific ensemble rule adding 3 conventional sampling methods, 1 cost-sensitive learning method, 6 boosting and bagging ensemble methods with forty-six unbalanced datasets are taken for solving the highly unbalanced issues. A novel quicker positive-unlabeled (PU) learning methodology based on bootstrap aggregating (bagging) techniques with a conceptually simple implementation has been developed, and the method [28] has demonstrated the best PU learning performance.

To construct (main contribution of the paper) the model, the proposed method "extended binomial GLMBoost (EBGLMBoost) ensemble method" where the synthetic SMOTE [29], [30] sampling method is combined with the extended binomial GLMBoost algorithm [14]. SMOTE is an advanced sampling technique that goes far above simple under and over sampling. Using convex combinations of surrounding examples, this technique generates new instances of the minority class. Then the proposed method is compared with other methods. The rest of the text is formatted as: section 2 and 3 explains the suggested model for the unbalanced dataset with a general overview, system configuration, datasets preparation, and performance measures. The result analysis and statistical test are shown in section 4. The conclusion and suggestions for further research are presented in section 5.

2. PROPOSED MODEL FOR IMBALANCED DATASET

The flow chart of the proposed method is shown in Figure 1 to handle imbalanced datasets. A "10-fold cross-validation" technique is selected for evaluating the models [31] and obtaining unbiased outputs from the systems. There are 20 different types of unbalanced datasets Table 1 used as input to the model. The unbalanced datasets are sorted into training and testing datasets, where the training datasets are trained using the SMOTE method (to get the new balanced training set from the original imbalanced training set) with five types of classification approaches including the proposed method EBGLMBoost. The SVM, Nu-SVM, bagging, boosting and AdaBoost algorithms are taken. An extended binomial GLM and boosting method are combined to construct the model [32]. Finally, the results are analyzed using the performance metrics.

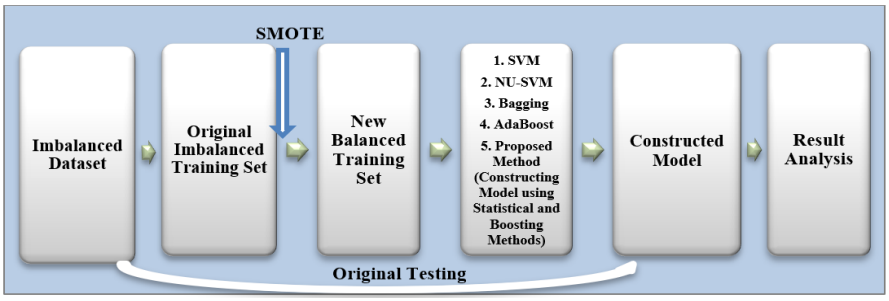

Figure 1. Diagram for the proposed method [an advance EBGLMBoost coupled with SMOTE technique]

Table 1. Information of imbalanced datasets

IDS	Datasets	Attributes	Examples	Classes	Imbalance Ratio (IR)
ID1	Kr_vs_k_three_vs_eleven	6	2935	2	35.23
ID2	winequality_red_8_vs_6	11	656	2	35.44
ID3	abalone_17_vs_7_8_9_10	8	2338	2	39.31
ID4	abalone_21_vs_8	8	581	2	40.50
ID5	winequality_white_3_vs_7	11	900	2	44.00
ID6	winequality_red_8_vs_6_7	11	855	2	46.50
ID7	kddcup_land_vs_portsweep	41	1061	2	49.52
ID8	abalone_19_vs_10_11_12_13	8	1622	2	49.69
ID9	kr_vs_k_zero_vs_eight	6	1460	2	53.07
ID10	Winequality_white_3_9_vs_5	11	1482	2	58.28
ID11	poker_8_9_vs_6	10	1485	2	58.40
ID12	shuttle_2_vs_5	9	3316	2	66.67
ID13	winequality_red_3_vs_5	11	691	2	68.10
ID14	abalone_20_vs_8_9_10	8	1916	2	72.69
ID15	Kddcup_buffer_overflow_vs_back	41	2233	2	73.43
ID16	kddcup_land_vs_satan	41	1610	2	75.67
ID17	Kr_vs_k_zero_vs_fifteen	6	2193	2	80.22
ID18	poker_8_9_vs_5	10	2075	2	82.00
ID19	poker_8_vs_6	10	1477	2	85.88
ID20	kddcup_rootkit_imap_vs_back	41	2225	2	100.14

3. METHOD

SMOTE [16] is a data augmentation algorithm that creates synthetic data points depending on the original data points. SMOTE can be thought of as a more advanced variant of oversampling or as a specific data augmentation process. SMOTE has the advantage of not creating duplicate data points, but rather synthetic data points that are somewhat different from the original data points. SVM "is a supervised machine learning technique that can be used to classify and predict data. The SVM algorithm seeks to locate a hyperplane in an M-dimensional [33] space that clearly categorizes data points". The number of features determines the hyperplane's size. The hyperplane is essentially a line if there are just two input features. The hyperplane becomes a two-dimensional plane when there are three input features. It gets impossible to imagine when the number of features exceeds three [34]. In Nu-SVM [35], a new class of support vector algorithms for classification and regression is proposed. A parameter ν in these methods allows one to effectively regulate the number of "support vectors". While this is important in and of itself, the parameterization also allows us to remove one of the algorithm's two free parameters: the accuracy parameter ε in the regression case and the regularization "constant C" in the classification case [36]. In ν-support vector classification, the primal optimization problem [37] is to minimize.

In bagging, a decision tree's variance is reduced by bagging (Bootstrap Aggregation). In the case of a set E of e tuples, a "training set E_i" of e tuples is sampled with replacing from E at each iteration "i" (That is, bootstrap). Then, for each training set "$E < i$" a classifier model N_i is learned. N_i returns the class prediction for each classifier. N*, a bagged classifier, calculates the votes and assigns Y to the most popular class (unidentified sample) [38]. Boosting is an ensemble modelling technique that seeks to combine many weak classifiers into one strong one. It is done by building a model out of a series of weak models. The training data is first used to build a model. The second model is then developed in an effort to fix the previous model's flaws. This process is repeated until either the maximum number of models have been added or the whole training data set has been correctly predicted. AdaBoost was the first truly successful [39] boosting algorithm created specifically for binary classification. Adaptive Boosting, or AdaBoost, is a powerful boosting strategy that combines several "weak classifiers" into a single "strong classifier". Robert Schapire and Yoav Freund came up with the idea. They are also awarded the Godel Prize in 2003 for their efforts.

3.1. Datasets preparation and parameters

In the experiment, there are twenty types of imbalanced datasets are used. Table 1 has shown the information of Imbalanced datasets. All the datasets are freely available in Keel dataset repository site [40]. The experiments consider the SMOTE algorithm settings listed below:

- k: the number of nearest neighbors is fixed to five.
- The distribution of classes will be rebalanced to 50–50%.
- The distance function uses the Euclidean distance to determine which neighbors are nearest.

3.2. SMOTE

Synthetic minority oversampling technique (SMOTE) is one of the oversampling strategies [41], which creates minority class instances. As a result, it is frequently used to solve problems of class inconsistence

and gives better outcomes than basic oversampling strategies. The SMOTE technique is a valuable and strong approach that has been employed in a variety of medicinal applications. Synthetic data were developed as per to the attribute area in order to apply this approach. "SMOTE" over-samples the minority class by inserting synthetic instances along the segmentation of a line connecting any or all of the $k - minority$ class nearest neighbours for each and every minority class sample. Synthetic instances from the k nearest neighbours are picked at random based on the quantity of over-sampling necessary. The value of "k" is set to 5 in all SMOTE [42] calculations.

3.3. Statistical performance of extended binomial GLMBoost combined with SMOTE

GLMBoost, which is based on penalized loglikelihood, is one of the boosting methods. Because it is one of the most significant ensembles learning algorithms, this technique can be used to address a wide variety of regression or classification issues. GLMBoost has a lot of advantages when it comes to implementation. GLMBoost lists a number of benefits in addition [43] to the ease of calculation. It has a large calculating capacity and does not require complex tuning processes [44]. It [45] provides more thorough information on this ensemble learning approach. A GLM is a regression strategy that adds response distributions that are not normal and modelling functions to regular regression models. Using a link function (logistic model) a GLM can model binary datasets dependent on presence or absence data. It is interesting to show the performance of "GLMBoost and SMOTE" to validate the influence of SMOTE on the GLMBoost algorithm. However, estimating the performance of different classifiers using objective statistical metrics is not always easy, and how to solve the evaluation problem [46] is still a hot research topic [47]. The extended binomial distribution is applied in this study. The conditional distribution of one of two independent Poisson random variables given the sum of these two variables is known to be binomial. It is interesting to find the conditional distribution of one of two independent Poisson difference random variables given the sum of these two variables [48]. There are four main steps in GLM algorithm which are given below.

a. Model and parameters estimation

By minimizing some objective function, fit the model once to all observations in the current node.

$$\sum_{i=1}^{n} \varphi(Y_i, \theta) \quad (1)$$

By solving the first order requirements, the estimation of the vector of parameters can be computed:

$$\sum_{i=1}^{n} \varphi\left(Y_i, \hat{\theta}\right) = 0, \quad \varphi(Y, \theta) = \frac{\partial \varphi(Y,\theta)}{\partial \theta} \quad (2)$$

The score function is then checked for systematic deviations at the estimated parameters $\hat{\varphi}_i = \varphi(Y_i, \hat{\theta})$.

b. Instability tests

The broad class of score-based fluctuation tests is used to determine whether node splitting is required. Depending on whether the partitioning variable is categorical or numerical, the test used varies. For dividing the node, *Zj* with the lowest p-value is picked.

c. Partitioning

The model is estimated on the two resulting subsets and the resulting objective functions are summed for each conceivable split. The ideal split is the one that optimizes the segmented objective function.

d. Pruning

For determine the optimal size of the tree, one can either use a "*pre-pruning or post-pruning*" strategy. Extended Binomial Theorem:

Euclid, a prominent Greek mathematician, first mentioned the binomial theorem in the 4th century BC. The binomial theorem shows how to use individual [49] exponents of variables *x* and *y* to expand the algebraic statement $(x + y)^n$ to a sum of terms. Each word in a binomial expansion is given a numerical [50] value called coefficient [51]. A special case of the Binomial Theorem is very useful that is given in (3).

$$(1 + x)^n = \sum_{k=0}^{n} \binom{n}{k} x^k \quad (3)$$

for any positive integer *n*, which is the Taylor series for $(1 + x)^n$.

This formula can be extended to any number of real powers α:

$$(1 + x)^\alpha = \sum_{k=0}^{\infty} \binom{\alpha}{k} x^k \quad (4)$$

where, for any real number α,

$$\binom{\alpha}{k} = \frac{(\alpha)(\alpha-1)(\alpha-2)\ldots(\alpha-(k-1))}{k!} = \frac{\alpha!}{k!(\alpha-k)!} \quad (5)$$

Now the formula gives an infinite series: when $\alpha=n$ is $a+ve$ integer, all but the first (n+1) terms are 0 then, $n-n$ \$=0\$ appears in each numerator. This expansion is very useful for approximating $(1 + x)^{\alpha}$ for $|x| \ll 1$:

$$(1+x)^{\alpha} = 1 + \alpha x + \frac{\alpha(\alpha-1)}{2!}x^2 + \frac{\alpha(\alpha-1)(\alpha-2)}{3!}x^3 + \cdots \quad (6)$$

However, because higher powers of x get very small quickly for $|x| \ll 1$, $(1 + x)^{\alpha}$ can be approximated to any accuracy then it is required by truncating the series after a finite number of entries.

3.4. System configuration

The entire experiment is carried out on a single-language version of Windows 10 with an Intel® Core (TM) i5-7300HQ processor running at 2.50 GHz. The operating system is 64-bit, with an x64-based processor and 8.00 GB of installed memory (RAM). The hardware in a computing device called random access memory (RAM) stores the operating system (OS), application programs, and data that are currently in use so that the processor of the device can access them rapidly. The primary memory of a computer is RAM.

3.5. Performance metrics

The confusion matrix [52] is used for evaluation process and many metrics are based on this for effectiveness evaluation on classification problem. The matrix overall accuracy does not work on imbalanced datasets. Different metrics are given below. Sensitivity (7) is known as true positive rate and specificity (8) is known as true negative rate, G–mean (9) is the geometric mean of sensitivity and specificity. F-measure (12) is used to integrate precision (10) and recall (11) into a single metric for the support of the modelling and β is a coefficient to manage the relative importance of precision vs. recall [53], [54].

$$Sensitivity = \frac{TP}{TP+FN} \quad (7)$$

$$Specificity = \frac{TN}{TN+FP} \quad (8)$$

$$G - Mean = \sqrt{Sensitivity * Specificity} \quad (9)$$

$$Precision = \frac{TP}{TP+FP} \quad (10)$$

$$Recall = \frac{TP}{TP+FN} \quad (11)$$

$$F - Measure = \frac{(1+\beta^2) \times Precision \times Recall}{\beta^2 \times Recall + Precision}, \text{ where, } \beta = 1 \quad (12)$$

4. RESULTS AND DISCUSSION

Indeed, total classification accuracy is typically not a significant measure of performance for the imbalanced classification problem, because in extremely skewed domains, a simple classifier that assigns every sample to the majority class can achieve exceptionally high accuracy. Instead of sophisticated metrics, six simple and useful measurements (sensitivity, specificity, G-mean, recall, and precision) are utilized in this study [55]. MATLAB and Keel [56] are used to perform the calculations. Table 2 has been created using the confusion matrix to display the results of the training datasets for the SVM algorithm using pre-processed SMOTE. The average percentages for sensitivity, specificity, G-mean, precision, recall, and F-measure are 99.26%, 32.46%, 47.09%, 85.59%, 99.26%, and 91.40%, respectively. The results of the testing datasets for the SVM with SMOTE method are displayed in Table 3. 98.84%, 31.08%, 43.12%, 79.26%, 98.84%, and 86.96% are the average values for sensitivity, specificity, G-mean, precision, recall, and F-measure, respectively.

The training datasets for the Nu-SVM method with pre-processed SMOTE have results, which are shown in Table 4. 98.84%, 52.73%, 68.52%, 95.19%, 98.84%, and 96.38% are the average values for sensitivity, specificity, G-mean, precision, recall, and F-measure, respectively. The results of the testing datasets for the Nu-SVM with SMOTE method are displayed in Table 5. 98.75%, 22.94%, 43.88%, 93.73%, 98.74%, and 95.94% are the average values for sensitivity, specificity, G-mean, precision, recall, and F-measure, respectively.

Table 2. Results of the training datasets for SVM with SMOTE method

Datasets	Sensitivity	Specificity	G-mean	Precision	Recall	F-measure
Kr_vs_k_three_vs_eleven	99.06	72.09	84.51	99.26	99.06	99.16
winequality_red_8_vs_6	99.24	05.54	23.45	86.23	99.24	92.28
abalone_17_vs_7_8_9_10	99.31	07.35	27.01	74.42	99.31	85.08
abalone_21_vs_8	98.66	03.74	19.22	74.14	98.66	84.66
winequality_white_3_vs_7	98.68	08.87	29.59	82.29	98.68	89.74
winequality_red_8_vs_6_7	98.44	04.37	20.73	81.03	98.44	88.89
kddcup_land_vs_portsweep	99.90	57.14	75.56	98.58	99.90	99.24
abalone_19_vs_10_11_12_13	99.36	03.55	18.79	75.95	99.36	86.09
kr_vs_k_zero_vs_eight	99.27	89.19	94.10	99.86	99.27	99.57
Winequality_white_3_9_vs_5	98.90	06.27	24.91	79.20	98.90	87.96
poker_8_9_vs_6	99.03	01.44	11.95	61.39	99.03	75.79
shuttle_2_vs_5	99.95	73.64	85.80	99.48	99.95	99.72
winequality_red_3_vs_5	98.70	03.18	17.71	79.62	98.70	88.14
abalone_20_vs_8_9_10	99.36	07.39	27.10	75.13	99.36	85.56
Kddcup_buffer_overflow_vs_back	99.85	13.04	36.09	90.91	99.85	95.17
kddcup_land_vs_satan	99.08	21.88	46.55	95.24	99.08	97.12
Kr_vs_k_zero_vs_fifteen	99.91	94.34	97.08	99.93	99.91	99.92
poker_8_9_vs_5	99.91	86.96	93.21	99.86	99.91	99.89
poker_8_vs_6	98.78	02.35	15.23	59.42	98.78	74.20
kddcup_rootkit_imap_vs_back	99.91	86.96	93.21	99.86	99.91	99.89
Averages	99.26	32.46	47.09	85.59	99.26	91.40

Table 3. Results of the testing datasets for SVM with SMOTE method

Datasets	Sensitivity	Specificity	G-mean	Precision	Recall	F-measure
Kr_vs_k_three_vs_eleven	98.74	98.95	98.84	98.95	98.74	98.84
winequality_red_8_vs_6	99.46	05.04	22.38	80.62	99.46	89.05
abalone_17_vs_7_8_9_10	99.11	13.25	36.24	73.29	99.11	84.27
abalone_21_vs_8	98.60	03.40	18.30	71.20	98.60	82.69
winequality_white_3_vs_7	97.97	04.91	21.93	75.71	97.97	85.41
winequality_red_8_vs_6_7	98.45	03.81	19.37	75.87	98.45	85.70
kddcup_land_vs_portsweep	96.25	95.45	95.85	99.90	96.25	98.04
abalone_19_vs_10_11_12_13	99.16	02.86	16.83	74.81	99.16	85.28
kr_vs_k_zero_vs_eight	98.74	32.14	56.34	98.67	98.74	98.71
Winequality_white_3_9_vs_5	98.48	04.13	20.17	73.64	98.48	84.27
poker_8_9_vs_6	99.12	01.47	12.08	54.18	99.12	70.06
shuttle_2_vs_5	99.73	93.88	96.76	99.91	99.73	99.82
winequality_red_3_vs_5	98.48	02.21	14.75	75.70	98.48	85.60
abalone_20_vs_8_9_10	98.81	05.59	23.50	73.19	98.81	84.09
Kddcup_buffer_overflow_vs_back	99.95	90.63	95.18	99.86	99.95	99.91
kddcup_land_vs_satan	99.94	90.00	94.84	99.87	99.94	99.91
Kr_vs_k_zero_vs_fifteen	99.81	69.70	83.41	99.54	99.81	99.68
poker_8_9_vs_5	98.82	01.23	11.03	53.07	98.82	69.06
poker_8_vs_6	98.13	01.49	12.10	50.27	98.13	66.49
kddcup_rootkit_imap_vs_back	99.09	01.59	12.56	57.02	99.09	72.38
Averages	98.84	31.08	43.12	79.26	98.84	86.96

Table 4. Results of the training datasets for Nu-SVM with SMOTE method

Datasets	Sensitivity	Specificity	G-mean	Precision	Recall	F-measure
Kr_vs_k_three_vs_eleven	99.45	38.73	62.06	96.36	99.45	97.88
winequality_red_8_vs_6	99.08	41.03	63.76	99.16	99.08	99.12
abalone_17_vs_7_8_9_10	98.61	91.78	95.13	98.44	98.61	98.52
abalone_21_vs_8	99.37	72.73	85.01	99.86	99.37	99.62
winequality_white_3_vs_7	97.66	18.75	42.79	97.96	97.66	97.81
winequality_red_8_vs_6_7	98.19	18.64	42.79	98.57	98.19	98.38
kddcup_land_vs_portsweep	99.51	51.61	71.67	98.56	99.51	99.03
abalone_19_vs_10_11_12_13	99.43	54.62	73.70	98.11	99.43	98.77
kr_vs_k_zero_vs_eight	99.05	98.15	98.60	99.98	99.05	99.51
Winequality_white_3_9_vs_5	98.95	68.25	82.18	99.43	98.95	99.19
poker_8_9_vs_6	93.98	04.11	19.65	37.31	93.98	53.42
shuttle_2_vs_5	99.44	64.40	80.02	99.48	99.44	99.46
winequality_red_3_vs_5	98.76	49.09	69.63	99.52	98.76	99.14
abalone_20_vs_8_9_10	98.69	80.77	89.28	99.78	98.69	99.23
Kddcup_buffer_overflow_vs_back	99.77	89.29	94.38	99.86	99.77	99.82
kddcup_land_vs_satan	99.68	53.33	72.91	99.12	99.68	99.40
Kr_vs_k_zero_vs_fifteen	99.91	60.98	78.05	99.26	99.91	99.58
poker_8_9_vs_5	99.34	19.03	43.48	97.56	99.34	98.44
poker_8_vs_6	98.62	03.45	18.44	85.62	98.62	91.66
kddcup_rootkit_imap_vs_back	99.44	76.00	86.93	99.86	99.44	99.65
Averages	98.84	52.73	68.52	95.19	98.84	96.38

Table 5. Results of the testing datasets for Nu-SVM with SMOTE method

Datasets	Sensitivity	Specificity	G-mean	Precision	Recall	F-measure
Kr_vs_k_three_vs_eleven	99.32	13.13	36.11	95.17	99.32	97.20
winequality_red_8_vs_6	98.98	60.00	77.06	99.71	98.98	99.34
abalone_17_vs_7_8_9_10	98.41	07.01	26.27	84.30	98.41	90.81
abalone_21_vs_8	98.41	01.98	13.96	68.93	98.21	81.01
winequality_white_3_vs_7	97.70	60.00	76.56	99.69	97.70	98.68
winequality_red_8_vs_6_7	98.23	33.33	57.22	99.52	98.23	98.87
kddcup_land_vs_portsweep	99.50	28.57	53.32	96.15	99.50	97.80
abalone_19_vs_10_11_12_13	99.17	10.78	32.70	95.19	99.17	97.14
kr_vs_k_zero_vs_eight	98.94	26.67	51.37	97.70	98.94	98.31
Winequality_white_3_9_vs_5	97.99	33.33	57.15	99.77	97.99	98.88
poker_8_9_vs_6	99.29	10.45	32.21	95.89	99.29	97.56
shuttle_2_vs_5	99.16	24.72	49.51	97.95	99.16	98.55
winequality_red_3_vs_5	98.44	33.33	57.28	99.86	98.44	99.15
abalone_20_vs_8_9_10	98.23	06.94	26.12	88.18	98.23	92.94
Kddcup_buffer_overflow_vs_back	98.88	40.00	62.89	99.86	98.88	99.37
kddcup_land_vs_satan	99.67	15.24	38.97	94.40	99.67	96.96
Kr_vs_k_zero_vs_fifteen	99.67	23.26	48.14	96.95	99.67	98.29
poker_8_9_vs_5	98.84	25.00	49.71	99.85	98.84	99.34
poker_8_vs_6	98.38	03.23	17.81	95.89	98.38	97.12
kddcup_rootkit_imap_vs_back	97.97	01.84	13.43	69.81	97.97	81.53
Averages	98.75	22.94	43.88	93.73	98.74	95.94

The training datasets for the bagging method with pre-processed SMOTE have results, which are shown in Table 6. 98.26%, 23.08%, 40.27%, 91.74%, 98.26%, and 92.66% are the average values for sensitivity, specificity, G-mean, precision, recall, and F-measure, respectively. The results of the testing datasets for the bagging using SMOTE method are displayed in Table 7. The average values of the sensitivity, specificity, G-mean, precision, recall, and F-measure are respectively 97.76%, 20.58%, 33.89%, 88.81%, 97.76%, and 90.92%.

The results of the training datasets for the AdaBoost algorithm using pre-processed SMOTE have been prepared in Table 8. The average percentages for the following metrics are 99.44%, 62.02%, 74.02%, 97.99%, 99.44%, and 98.68%, respectively. The results of the testing datasets for the AdaBoost with SMOTE approach are displayed in Table 9. 96.61%, 47.90%, 65.43%, 98.10%, 96.61%, and 97.22% are the average values for sensitivity, specificity, G-mean, precision, recall, and F-measure, respectively.

Table 10 has been created to display the training dataset's results for the suggested method (EBGLMBoost ensemble method with SMOTE). The average percentages for the following metrics are 99.37%, 66.95%, 80.81%, 99.21%, 99.37%, and 99.29%, respectively. Table 11 displays the findings from the datasets used to test the suggested strategy (EBGLMBoost ensemble method with SMOTE). 98.61%, 54.78%, 69.88%, 98.77%, 96.61%, and 98.68% are the average values for sensitivity, specificity, G-mean, precision, recall, and F-measure, respectively. Finally, the suggested technique demonstrated the highest G-mean and F-measure accuracy for the training datasets, respectively, of 80.8% and 99.29%.

Table 6. Results of the training datasets for bagging with SMOTE method

Datasets	Sensitivity	Specificity	G-mean	Precision	Recall	F-measure
Kr_vs_k_three_vs_eleven	97.35	05.45	23.04	96.36	97.35	96.85
winequality_red_8_vs_6	98.58	04.00	19.86	99.12	98.58	98.85
abalone_17_vs_7_8_9_10	97.69	45.95	67.00	99.78	97.69	98.73
abalone_21_vs_8	99.10	40.00	62.96	99.86	99.10	99.48
winequality_white_3_vs_7	97.31	05.45	23.04	97.96	97.31	97.64
winequality_red_8_vs_6_7	98.04	03.33	18.08	89.61	98.04	93.63
kddcup_land_vs_portsweep	98.88	62.50	78.61	99.86	98.88	99.37
abalone_19_vs_10_11_12_13	99.36	03.55	18.79	75.95	99.36	86.09
kr_vs_k_zero_vs_eight	98.04	01.08	10.30	87.23	98.04	92.32
Winequality_white_3_9_vs_5	97.59	23.08	47.46	99.34	97.59	98.46
poker_8_9_vs_6	98.96	15.22	38.81	99.33	98.96	99.15
shuttle_2_vs_5	99.57	95.24	97.38	99.95	99.57	99.76
winequality_red_3_vs_5	98.32	03.45	18.41	99.52	98.32	98.92
abalone_20_vs_8_9_10	98.61	26.88	51.49	97.00	98.61	97.8
Kddcup_buffer_overflow_vs_back	98.75	07.41	27.05	99.21	98.75	98.98
kddcup_land_vs_satan	93.75	01.91	13.39	01.44	93.75	02.84
Kr_vs_k_zero_vs_fifteen	98.86	66.67	81.18	99.95	98.86	99.40
poker_8_9_vs_5	98.91	05.77	23.89	97.61	98.91	98.26
poker_8_vs_6	98.52	04.84	21.83	95.96	98.52	97.22
kddcup_rootkit_imap_vs_back	99.10	40.00	62.96	99.86	99.10	99.48
Averages	98.26	23.08	40.27	91.74	98.26	92.66

Table 7. Results of the testing datasets for bagging with SMOTE method

Datasets	Sensitivity	Specificity	G-mean	Precision	Recall	F-measure
Kr_vs_k_three_vs_eleven	97.22	01.82	13.30	98.11	97.22	97.66
winequality_red_8_vs_6	98.52	01.22	10.96	88.11	98.52	93.02
abalone_17_vs_7_8_9_10	97.65	05.88	23.97	96.49	97.65	97.07
abalone_21_vs_8	97.98	94.34	96.14	94.34	97.98	96.12
winequality_white_3_vs_7	97.56	07.32	26.72	94.04	97.56	95.77
winequality_red_8_vs_6_7	98.77	17.78	41.90	95.58	98.77	97.15
kddcup_land_vs_portsweep	98.74	66.67	81.13	99.95	98.74	99.35
abalone_19_vs_10_11_12_13	99.16	02.86	16.83	74.81	99.16	85.28
kr_vs_k_zero_vs_eight	98.59	17.50	41.54	97.70	98.59	98.14
Winequality_white_3_9_vs_5	97.68	01.25	11.05	91.02	97.68	94.24
poker_8_9_vs_6	98.94	03.23	17.87	95.89	98.94	97.39
shuttle_2_vs_5	99.57	85.37	92.20	99.82	99.57	99.69
winequality_red_3_vs_5	98.38	05.00	22.18	96.09	98.38	97.22
abalone_20_vs_8_9_10	98.61	85.71	91.93	99.82	98.61	99.21
Kddcup_buffer_overflow_vs_back	98.68	01.11	10.47	94.40	98.68	96.49
kddcup_land_vs_satan	97.56	01.96	13.83	03.85	97.56	07.40
Kr_vs_k_zero_vs_fifteen	98.82	02.94	17.05	96.95	98.82	97.88
poker_8_9_vs_5	98.79	01.00	09.91	97.57	98.79	98.18
poker_8_vs_6	98.32	02.50	15.68	99.33	98.32	98.82
kddcup_rootkit_imap_vs_back	85.84	06.25	23.16	62.50	85.84	72.33
Averages	97.76	20.58	33.89	88.81	97.76	90.92

Table 8. Results of the training datasets for AdaBoost with SMOTE method

Datasets	Sensitivity	Specificity	G-mean	Precision	Recall	F-measure
Kr_vs_k_three_vs_eleven	99.82	42.22	64.92	96.36	99.82	98.06
winequality_red_8_vs_6	99.26	45.45	67.17	99.12	99.26	99.19
abalone_17_vs_7_8_9_10	99.91	99.55	99.73	99.99	99.91	99.95
abalone_21_vs_8	99.83	99.15	99.49	99.98	99.83	99.91
winequality_white_3_vs_7	98.58	40.91	63.50	97.96	98.58	98.27
winequality_red_8_vs_6_7	98.74	38.46	61.63	98.57	98.74	98.65
kddcup_land_vs_portsweep	99.92	96.00	97.94	99.94	99.92	99.93
abalone_19_vs_10_11_12_13	99.97	99.02	99.50	99.99	99.97	99.98
kr_vs_k_zero_vs_eight	98.04	01.08	10.30	87.23	98.04	92.32
Winequality_white_3_9_vs_5	99.74	78.02	88.22	99.43	99.74	99.59
poker_8_9_vs_6	99.50	49.37	70.09	99.32	99.50	99.41
shuttle_2_vs_5	99.95	99.49	99.72	100.0	99.95	99.97
winequality_red_3_vs_5	99.37	98.44	98.9	99.98	99.37	99.67
abalone_20_vs_8_9_10	99.91	65.85	81.11	98.77	99.91	99.33
Kddcup_buffer_overflow_vs_back	99.95	59.15	76.89	98.61	99.95	99.27
kddcup_land_vs_satan	99.05	66.67	81.26	99.93	99.05	99.49
Kr_vs_k_zero_vs_fifteen	99.72	56.76	75.23	99.26	99.72	99.49
poker_8_9_vs_5	99.26	16.67	40.67	97.56	99.26	98.4
poker_8_vs_6	98.52	01.22	10.96	88.11	98.52	93.02
kddcup_rootkit_imap_vs_back	99.91	86.96	93.21	99.86	99.91	99.89
Averages	99.44	62.02	74.02	97.99	99.44	98.68

Table 9. Results of the testing datasets for AdaBoost with SMOTE method

Datasets	Sensitivity	Specificity	G-mean	Precision	Recall	F-measure
Kr_vs_k_three_vs_eleven	99.82	58.46	76.39	98.11	99.82	98.96
winequality_red_8_vs_6	99.21	06.67	25.72	85.62	99.21	91.91
abalone_17_vs_7_8_9_10	98.22	47.22	68.10	99.17	98.22	98.69
abalone_21_vs_8	98.14	33.33	57.20	99.75	98.14	98.94
winequality_white_3_vs_7	97.39	25.00	49.34	99.53	97.39	98.45
winequality_red_8_vs_6_7	98.00	25.00	49.50	99.64	98.00	98.82
kddcup_land_vs_portsweep	90.91	48.39	66.32	98.43	90.91	94.52
abalone_19_vs_10_11_12_13	98.85	40.00	62.88	99.68	98.85	99.26
kr_vs_k_zero_vs_eight	98.59	17.50	41.54	97.70	98.59	98.14
Winequality_white_3_9_vs_5	98.43	66.67	81.01	99.66	98.43	99.04
poker_8_9_vs_6	66.67	60.00	63.25	91.46	66.67	77.12
shuttle_2_vs_5	99.72	37.38	61.06	97.95	99.72	98.83
winequality_red_3_vs_5	98.58	50.00	70.21	99.73	98.58	99.15
abalone_20_vs_8_9_10	98.77	63.64	79.28	99.29	98.77	99.03
Kddcup_buffer_overflow_vs_back	99.01	41.18	63.85	97.09	99.01	98.04
kddcup_land_vs_satan	94.70	95.24	94.97	99.94	94.70	97.25
Kr_vs_k_zero_vs_fifteen	99.72	95.45	97.57	99.95	99.72	99.84
poker_8_9_vs_5	99.03	20.00	44.50	99.81	99.03	99.42
poker_8_vs_6	98.65	40.00	62.82	99.79	98.65	99.22
kddcup_rootkit_imap_vs_back	99.91	86.96	93.21	99.86	99.91	99.89
Averages	96.61	47.90	65.43	98.10	96.61	97.22

Table 10. Results of the training datasets for the EBGLMBoost ensemble method with SMOTE

Datasets	Sensitivity	Specificity	G-mean	Precision	Recall	F-measure
Kr_vs_k_three_vs_eleven	99.78	41.90	64.66	96.36	99.78	98.04
winequality_red_8_vs_6	99.48	52.00	71.92	99.12	99.48	99.30
abalone_17_vs_7_8_9_10	97.53	50.00	69.83	99.99	97.53	98.74
abalone_21_vs_8	98.33	95.24	96.77	99.98	98.33	99.15
winequality_white_3_vs_7	98.66	42.22	64.54	97.96	98.66	98.31
winequality_red_8_vs_6_7	98.92	42.86	65.11	98.57	98.92	98.74
kddcup_land_vs_portsweep	99.90	57.14	75.56	98.56	99.90	99.23
abalone_19_vs_10_11_12_13	98.59	99.29	98.94	99.94	98.59	99.26
kr_vs_k_zero_vs_eight	99.84	98.02	98.93	99.97	99.84	99.90
Winequality_white_3_9_vs_5	99.97	79.80	89.32	99.44	99.97	99.70
poker_8_9_vs_6	99.78	57.89	76.00	99.32	99.78	99.55
shuttle_2_vs_5	99.98	74.05	86.04	99.48	99.98	99.73
winequality_red_3_vs_5	99.98	77.95	88.28	99.52	99.98	99.75
abalone_20_vs_8_9_10	97.72	66.67	80.71	99.96	97.72	98.82
Kddcup_buffer_overflow_vs_back	99.99	90.84	95.30	99.86	99.99	99.93
kddcup_land_vs_satan	99.98	58.82	76.69	99.12	99.98	99.55
Kr_vs_k_zero_vs_fifteen	99.90	61.11	78.13	99.27	99.90	99.58
poker_8_9_vs_5	99.96	49.24	70.16	98.78	99.96	99.37
poker_8_vs_6	99.69	67.21	81.86	99.32	99.69	99.50
kddcup_rootkit_imap_vs_back	99.46	76.92	87.47	99.86	99.46	99.66
Averages	99.37	66.95	80.81	99.21	99.37	99.29

Table 11. Results of the testing datasets for the EBGLMBoost ensemble method with SMOTE

Datasets	Sensitivity	Specificity	G-mean	Precision	Recall	F-measure
Kr_vs_k_three_vs_eleven	99.96	97.56	98.76	99.93	99.96	99.95
winequality_red_8_vs_6	98.69	25.00	49.67	99.56	98.69	99.12
abalone_17_vs_7_8_9_10	97.78	90.91	94.28	99.64	97.78	98.70
abalone_21_vs_8	97.09	84.80	90.73	91.74	97.09	94.34
winequality_white_3_vs_7	97.67	27.27	51.61	98.75	97.67	98.21
winequality_red_8_vs_6_7	97.99	12.50	35.00	99.16	97.99	98.57
kddcup_land_vs_portsweep	99.81	65.52	80.86	99.04	99.81	99.42
abalone_19_vs_10_11_12_13	97.56	99.89	98.72	99.90	97.56	98.72
kr_vs_k_zero_vs_eight	99.79	96.00	97.88	99.93	99.79	99.86
Winequality_white_3_9_vs_5	97.98	22.22	46.66	99.20	97.98	98.59
poker_8_9_vs_6	98.92	50.00	70.33	99.93	98.92	99.42
shuttle_2_vs_5	99.79	93.33	96.51	99.91	99.79	99.85
winequality_red_3_vs_5	98.43	11.76	34.03	98.97	98.43	98.70
abalone_20_vs_8_9_10	97.75	50.00	69.91	99.82	97.75	98.78
Kddcup_buffer_overflow_vs_back	98.74	40.00	62.85	99.86	98.74	99.30
kddcup_land_vs_satan	99.87	17.89	41.92	94.40	99.87	97.06
Kr_vs_k_zero_vs_fifteen	97.00	74.07	84.77	99.67	97.00	98.32
poker_8_9_vs_5	98.98	33.33	57.44	99.61	98.98	99.29
poker_8_vs_6	98.65	83.33	90.67	99.93	98.65	99.29
kddcup_rootkit_imap_vs_back	99.91	20.41	45.15	96.46	99.91	98.15
Averages	98.61	54.78	69.88	98.77	98.61	98.68

4.1. Statistical test

The nonparametric (Wilcoxon signed rank) test is utilized in this research to compare two paired or related samples [57]. When the samples are not in normal distributions and are small, this method is applied. This approach is used when the samples are small and do not have normal distributions. The results are sorted by leaving out the zero values after first calculating the differences between the two techniques (with n objects) [58], followed by calculating the absolute values. The values of R+ and R- are computed where R+ is the sum of ranks of positive differences and R- is the sum of ranks of negative differences. The classifier's significant differences are also usefully revealed by the p-value, which is set at 0.05 along with the significance value "α". Table 12 summarizes all of the calculations. The selection column displays the techniques that were chosen based on whether the hypothesis is rejected or not. Among the techniques tested, the proposed method "EBGLMBoost with pre-processed SMOTE" is selected as the best.

Table 12. Test Wilcoxon (for pair wise comparison)

Comparison	R^+	R^-	p-value	Hypothesis ($\alpha = 0.05$)	Selection
SVM vs. EBGLMBoost	52	158	0.0489	Rejected	EBGLMBoost
Nu-SVM vs. EBGLMBoost	57.5	426.5	0.0142	Rejected	EBGLMBoost
EBGLMBoost vs. Bagging	157.5	457.5	0.0778	Not Rejected	EBGLMBoost
EBGLMBoost vs. AdaBoost	168	212	0.6653	Not Rejection	EBGLMBoost

5. CONCLUSION

In many different industries, such as fraud detection, video surveillance, genetic data analysis, and many others, data imbalance is a critical issue. It could be caused by a particularly expensive or challenging data collection method, a rare natural occurrence, insufficient and/or skewed data sources, mistakes, or an uneven sensor placement. Because the typical accuracy metric assumes that real positives and true negatives are of equal importance, it is erroneous in these situations because the cost of misclassification is never equalized for positive and negative instances. Twenty unbalanced datasets are used in this experiment, with 'SMOTE' as the pre-processing approach. SVM, Nu-SVM, bagging, Boosting, and EBGLMBoost are the five classification methods are employed after then. The classification of imbalanced datasets is proposed using a SMOTE combined with the EBGLMBoost ensemble method algorithm in this paper. Using different metrics, the proposed method has shown the best result for G-mean and F-measure, among others. Furthermore, the statistical test shows that the EBGLMBoost Ensemble approach is the most effective.

Other sampling approaches, as well as various ensemble methods, may be explored in future study to handle imbalanced datasets. Future research will focus on developing a more efficient parameter, other statistical and ensemble techniques. This proposed method has a lot of interesting potential study areas. The categorization of imbalanced data with multiple class labels is another intriguing research topic that will be investigated further in the future. Due to the complex issues and numerous possible applications, the classification of unbalanced data will continue to garner interest in both the "science and industrial" sectors.

REFERENCES

[1] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Special issue on learning from imbalanced data sets," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 1–6, Jun. 2004, doi: 10.1145/1007730.1007733.

[2] U. Bhowan, M. Johnston, and M. Zhang, "Developing new fitness functions in genetic programming for classification with unbalanced data," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 406–421, Apr. 2012, doi: 10.1109/TSMCB.2011.2167144.

[3] H. C. Koh and G. Tan, "Data mining in healthcare," *Journal of healthcare information management*, vol. 19, no. 2, pp. 64-72, 2011.

[4] Y. Sun, M. S. Kamel, A. K. C. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," *Pattern Recognition*, vol. 40, no. 12, pp. 3358–3378, Dec. 2007, doi: 10.1016/j.patcog.2007.04.009.

[5] X. Deng, X. Tian, S. Chen, and C. J. Harris, "Statistics local fisher discriminant analysis for industrial process fault classification," in *2016 UKACC 11th International Conference on Control (CONTROL)*, Aug. 2016, pp. 1–6, doi: 10.1109/CONTROL.2016.7737588.

[6] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *SIGIR '94*, London: Springer London, 1994, pp. 3–12, doi: 10.1007/978-1-4471-2099-5_1.

[7] P. K. Chan, W. Fan, A. L. Prodromidis, and S. J. Stolfo, "Distributed data mining in credit card fraud detection," *IEEE Intelligent Systems*, vol. 14, no. 6, pp. 67–74, Nov. 1999, doi: 10.1109/5254.809570.

[8] M. Kubat, R. C. Holte, and S. Matwin, "Machine learning for the detection of oil spills in satellite radar images," *Machine Learning*, vol. 30, pp. 195–215, 1998.

[9] K.-C. Khor, C.-Y. Ting, and S. Phon-Amnuaisuk, "A cascaded classifier approach for improving detection rates on rare attack categories in network intrusion detection," *Applied Intelligence*, vol. 36, no. 2, pp. 320–329, Mar. 2012, doi: 10.1007/s10489-010-0263-y.

[10] B. Sluban and N. Lavrač, "Relating ensemble diversity and performance: A study in class noise detection," *Neurocomputing*, vol. 160, pp. 120–131, Jul. 2015, doi: 10.1016/j.neucom.2014.10.086.

[11] J. Zhao, J. Jin, S. Chen, R. Zhang, B. Yu, and Q. Liu, "A weighted hybrid ensemble method for classifying imbalanced data," *Knowledge-Based Systems*, vol. 203, Sep. 2020, doi: 10.1016/j.knosys.2020.106087.

[12] N. Liu, X. Li, E. Qi, M. Xu, L. Li, and B. Gao, "A novel ensemble learning paradigm for medical diagnosis with imbalanced data," *IEEE Access*, vol. 8, pp. 171263–171280, 2020, doi: 10.1109/ACCESS.2020.3014362.

[13] V. Garcia, R. A. Mollineda, and J. S. Sanchez, "Theoretical analysis of a performance measure for imbalanced data," in *2010 20th International Conference on Pattern Recognition*, Aug. 2010, pp. 617–620, doi: 10.1109/ICPR.2010.156.

[14] J. A. Nelder and R. W. M. Wedderburn, "Generalized linear models," *Journal of the Royal Statistical Society. Series A (General)*, vol. 135, no. 3, 1972, doi: 10.2307/2344614.

[15] G. E. P. Box, J. S. Hunter, and W. G. Hunter, *Statistics for Experimenters*. New York: John Wiley and sons, 1978.

[16] J. A. Sáez, J. Luengo, J. Stefanowski, and F. Herrera, "SMOTE–IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering," *Information Sciences*, vol. 291, pp. 184–203, Jan. 2015, doi: 10.1016/j.ins.2014.08.051.

[17] T. M. Khoshgoftaar, S. Zhong, and V. Joshi, "Enhancing software quality estimation using ensemble-classifier based noise filtering," *Intelligent Data Analysis*, vol. 9, no. 1, pp. 3–27, Mar. 2005, doi: 10.3233/IDA-2005-9102.

[18] M. Alibeigi, S. Hashemi, and A. Hamzeh, "DBFS: An effective density based feature selection scheme for small sample size and high dimensional imbalanced data sets," *Data & Knowledge Engineering*, vol. 81–82, pp. 67–103, Nov. 2012, doi: 10.1016/j.datak.2012.08.001.

[19] S. Piri, D. Delen, and T. Liu, "A synthetic informative minority over-sampling (SIMO) algorithm leveraging support vector machine to enhance learning from imbalanced datasets," *Decision Support Systems*, vol. 106, pp. 15–29, Feb. 2018, doi: 10.1016/j.dss.2017.11.006.

[20] J. Gong and H. Kim, "RHSBoost: Improving classification performance in imbalance data," *Computational Statistics & Data Analysis*, vol. 111, pp. 1–13, Jul. 2017, doi: 10.1016/j.csda.2017.01.005.

[21] W. Lee, C.-H. Jun, and J.-S. Lee, "Instance categorization by support vector machines to adjust weights in AdaBoost for imbalanced data classification," *Information Sciences*, vol. 381, pp. 92–103, Mar. 2017, doi: 10.1016/j.ins.2016.11.014.

[22] E. Volna and M. Kotyrba, "Enhanced ensemble-based classifier with boosting for pattern recognition," *Applied Mathematics and Computation*, vol. 310, pp. 1–14, Oct. 2017, doi: 10.1016/j.amc.2017.04.019.
[23] M. Di Martino, A. Fernández, P. Iturralde, and F. Lecumberry, "Novel classifier scheme for imbalanced problems," *Pattern Recognition Letters*, vol. 34, no. 10, pp. 1146–1151, Jul. 2013, doi: 10.1016/j.patrec.2013.03.012.
[24] B. Krawczyk and G. Schaefer, "An improved ensemble approach for imbalanced classification problems," in *2013 IEEE 8th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, May 2013, pp. 423–426, doi: 10.1109/SACI.2013.6609011.
[25] J. Błaszczyński and J. Stefanowski, "Neighbourhood sampling in bagging for imbalanced data," *Neurocomputing*, vol. 150, pp. 529–542, Feb. 2015, doi: 10.1016/j.neucom.2014.07.064.
[26] Y. Liu, X. Yu, J. X. Huang, and A. An, "Combining integrated sampling with SVM ensembles for learning from imbalanced datasets," *Information Processing & Management*, vol. 47, no. 4, pp. 617–631, Jul. 2011, doi: 10.1016/j.ipm.2010.11.007.
[27] Z. Sun, Q. Song, X. Zhu, H. Sun, B. Xu, and Y. Zhou, "A novel ensemble method for classifying imbalanced data," *Pattern Recognition*, vol. 48, no. 5, pp. 1623–1637, May 2015, doi: 10.1016/j.patcog.2014.11.014.
[28] F. Mordelet and J.-P. Vert, "A bagging SVM to learn from positive and unlabeled examples," *Pattern Recognition Letters*, vol. 37, pp. 201–209, Feb. 2014, doi: 10.1016/j.patrec.2013.06.010.
[29] A. Fernandez, S. Garcia, 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, pp. 863–905, Apr. 2018, doi: 10.1613/jair.1.11192.
[30] J. Han, M. Kamber, and J. Pei, *Data mining: Concepts and techniques*. Elsevier B.V., 2011.
[31] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.
[32] M. Hao, Y. Wang, and S. H. Bryant, "An efficient algorithm coupled with synthetic minority over-sampling technique to classify imbalanced PubChem BioAssay data," *Analytica Chimica Acta*, vol. 806, pp. 117–127, Jan. 2014, doi: 10.1016/j.aca.2013.10.050.
[33] B. Schölkopf, C. J. C. Burges, and A. J. Smola, *Advances in kernel methods: Support vector learning*. The MIT Press, 1998.
[34] Alokesh985, "Introduction to support vector machines (SVM)," Geeksforgeeks. https://www.geeksforgeeks.org/introduction-to-support-vector-machines-svm/?ref=rp (accessed Apr. 25, 2022).
[35] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/BF00994018.
[36] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009, doi: 10.1109/TKDE.2008.239.
[37] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: one-sided selection," *Fourteenth International Conference on Machine Learning*, vol. 97, 1997, pp. 179–186.
[38] D. Dey, "ML, bagging classifier," Geeksforgeeks. https://www.geeksforgeeks.org/ml-bagging-classifier/ (accessed Feb. 20, 2022).
[39] Raman_257, "Boosting in machine learning, boosting and AdaBoost," Geeksforgeeks, https://www.geeksforgeeks.org/boosting-in-machine-learning-boosting-and-adaboost/ (accessed Jan. 28, 2022).
[40] J. Alcala-Fdez *et al.*, "KEEL data-mining software Tool: data set repository, integration of algorithms and experimental analysis framework," *Journal of Multiple-valued Logic and Soft Computing*, vol. 17, no. 2, pp. 255–287, 2010.
[41] G. Tutz and H. Binder, "Generalized additive modeling with implicit variable selection by likelihood-based boosting," *Biometrics*, vol. 62, no. 4, pp. 961–971, Dec. 2006, doi: 10.1111/j.1541-0420.2006.00578.x.
[42] M. Kukar, "Quality assessment of individual classifications in machine learning and data mining," *Knowledge and Information Systems*, vol. 9, no. 3, pp. 364–384, Mar. 2006, doi: 10.1007/s10115-005-0203-z.
[43] Tyagikartik4282, "ML, handling imbalanced data with SMOTE and near miss algorithm in python," Geeksforgeeks. https://www.geeksforgeeks.org/ml-handling-imbalanced-data-with-smote-and-near-miss-algorithm-in-python/ (accessed Mar. 24, 2022).
[44] B. X. Wang and N. Japkowicz, "Boosting support vector machines for imbalanced data sets," *Knowledge and Information Systems*, vol. 25, no. 1, pp. 1–20, Oct. 2010, doi: 10.1007/s10115-009-0198-y.
[45] P. de Jong and G. Z. Heller, *Generalized linear models for insurance data*. Cambridge Books, 2008.
[46] Agresti, A., *Foundations of linear and generalized linear models*. John Wiley & Sons, 2015.
[47] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, "New support vector algorithms," *Neural Computation*, vol. 12, no. 5, pp. 1207–1245, May 2000, doi: 10.1162/089976600300015565.
[48] A. A. Alzaid and M. A. Omair, "An extended binomial distribution with applications," *Communications in Statistics - Theory and Methods*, vol. 41, no. 19, pp. 3511–3527, Oct. 2012, doi: 10.1080/03610926.2011.566974.
[49] "Generalized linear models (GLM) with binomial family," H2O.ai, https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/glm.html (accessed Jan. 30, 2022).
[50] Cuemath, "Binomial theorem," Cuemath. https://www.cuemath.com/algebra/binomial-theorem (accessed Feb. 02, 2022).
[51] E. W. Weisstein, "Binomial distribution," MathWorld. https://mathworld.wolfram.com/ (accessed Feb. 02, 2022).
[52] N. D. Marom, L. Rokach, and A. Shmilovici, "Using the confusion matrix for improving ensemble classifiers," in *2010 IEEE 26-th Convention of Electrical and Electronics Engineers in Israel*, Nov. 2010, pp. 000555–000559, doi: 10.1109/EEEI.2010.5662159.
[53] P.-N. Tan, M. Steinbach, A. Karpatne, and V. Kumar, *Introduction to data mining*. Pearson Education India, 2016.
[54] C. C. M, E. Karaaslan, C. Colak, A. K. Arslan, and N. Erdil, "Handling imbalanced class problem for the prediction of atrial fibrillation inobese patient," *Biomedical Research*, vol. 28, no. 7, pp. 3293–3299, 2017.
[55] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International Journal of Data Mining & Knowledge Management Process*, vol. 5, no. 2, pp. 01–11, Mar. 2015, doi: 10.5121/ijdkp.2015.5201.
[56] MathWorld, "MathWorks," MathWorld. https://in.mathworks.com/products/matlab.html (accessed Jan. 10, 2022).
[57] KEEL, "Reference manual." Knowledge Extraction based on Evolutionary Learning, https://sci2s.ugr.es/keel/development.php (accessed Jan. 02, 2022).
[58] R. M. Sundrum, "The power of wilcoxon's 2-sample test," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 15, no. 2, pp. 246–252, Jul. 1953, doi: 10.1111/j.2517-6161.1953.tb00139.x.

BIOGRAPHIES OF AUTHORS

Neelam Rout is a research scholar at the Department of Computer Science and Engineering Technology (Institute of Technical Education and Research, ITER) under Siksha 'O' Anusandhan (Deemed to be) University, Bhubaneswar, Odisha. She has completed her M.Tech from S'O'A (Deemed to be) University and B.Tech from Biju Pattnaik University, Bhubaneswar. She is working in the field of Machine learning. She can be contacted at email: neelamrout@yahoo.com.

Debahuti Mishra is currently a Professor and Head of the Department of Computer Science and Engineering at Faculty Engineering and Technology (Institute of Technical Education and Research, ITER) under Siksha 'O' Anusandhan (Deemed to be) University, Bhubaneswar, Odisha. She has approximately 27 years of experience in teaching and research. She has contributed seven books of International and National level on several core subjects and research domain in Computer Science and has around 200 research publications to her credit in several International and National Journals, Conferences of repute. Around 150 publications of her are indexed in Scopus database and overall h-index is 12. Most acclaimed research work of Prof. Mishra includes two IEEE transactions on Industrial Informatics, around six journal articles published with impact factor above 10 and around 25 numbers of SCI/SCIE/ESCI research articles. She has successfully supervised sixteen Ph.D. scholars so far under her supervision under Siksha 'O' Anusandhan (Deemed to be) University. She serves as the Editor in several journals of National and International repute. She also has organized more than 25 International/National conferences, workshops, symposiums, and winter schools. Prof. Mishra has a wider domain of research which includes health and bio-informatics, financial market data analysis, medical image processing, agriculture and crop prediction, meta-heuristic optimization, cloud containerization and defense signal transmission. The diversified societal contributions of her work include gene expression data analysis, protein structure prediction, gene network discovery and cholesterol motif finding, forecasting of Sensex, mutual fund, gold, crude oil, currency exchange, commodity market analysis, brain disease diagnosis through MRI, agricultural forecast and yield prediction based on climatic changes, and software defined radio for defense data transmission. She has successfully completed one in-house R&D project from SOA University, one DST-NIMAT Project for Entrepreneurship Awareness Camp, a research consultancy services in multi-objective in handover in 5G network for IoT/IoV communications worth Rs.28,50,000/- and one Start-Up grant of Rs. 31, 94, 900/- from Indian Council of Medical Research (ICMR) has been granted by the authority. She was awarded the BEST RESEARCHER on the eve of Engineers Day in 2018 and the WOMEN ICON on Women's Day in 2016 by Ever Green Forum, Bhubaneswar for her contribution towards technical education and research. She can be contacted at email: debahutimishra@soauniversity.ac.in.

Manas Kumar Mallick is currently a professor and director of Institute of Technical Education and Research (ITER) under Siksha 'O' Anusandhan (Deemed to be) University, Bhubaneswar, Odisha. He has published many research papers in various journals and conferences. He has also participated as organizing member of several conferences and workshops in international and national level. His publication topics are wavelet transforms, computation, feature extraction, power engineering computing, power system economics, Weibull distribution, cost reduction, electricity supply industry, fuzzy systems, load dispatching, load flow, load forecasting, mathematical operators, neural nets, optimization, pattern clustering, power generation dispatch, power generation economics, power supply quality, power system faults, pricing, probability, search problems, signal classification, and support vector machines He can be contacted at email: director.iter@soauniversity.ac.in.