Anempirical comparative study of novel approaches for software defect prediction on class imbalance datasets

K. Nitalaksheswara Rao* Ch. Satyananda Reddy **

Department of Computer Science and Systems Engineering

Andhra University, Visakhapatnam.

*kolukulanitla@gmail.com**satyanandau@yahoo.com

Abstract: Software Defect Prediction using data mining techniques is one of the best practices for finding defective modules. The on hand classification techniques can be used for efficient knowledge discovery on class balance datasets. The data in the real world are not completely balance in nature as any one of the class predominantly increases in ratio with other class. This type of data sources are known as class imbalance or skewed data sources. The defect prediction rate for the class imbalance datasets reduces with the increases in the class imbalance nature. The proposed algorithms consists of a novel oversampling, under sampling techniques implemented by removing noisy and weak instances from both majority and minority for better performance of class imbalance data streams. We conduct experiments on software defect datasets with class imbalance nature on three methods using four evaluation measures. The generated results suggest that the problem of class imbalanced software defect datasets can be effectively solved.

***Index Terms*—Knowledge Discovery, Software Defect datasets, Imbalanced data.**

1. Introduction

Software engineering is the process of building software with the desired properties of the user. The complete process of software engineering consists of different phases such as requirement analysis, designing, coding and testing. The complete or exhaustive testing for finding all the errors in the software modules is a tedious job.

The performances of the existing algorithms on software defect prediction are not capable enough for mitigating the issues regarding class imbalance learning. There is a good gap/scope for need of novel algorithms for effective learning of software defect prediction on real world class imbalance data sources. The in depth understanding of the class imbalance nature such as class disjunct, class drift, class imbalance ratio, boarder line instance overlapping etc are to be studied for effective problem solution. The proposed novel algorithms have taken under consideration different perspectives and scenarios for complete solution of the class imbalance problem in software defect prediction.

The remainder of the paper is as follows: The main recent works related to software defect prediction are presented in section 2. The frameworks of the comparative approaches are present in the section 3. The experimental methodology, evaluation criteria's and results discussion is done in the section 4 and 5 respectively. The conclusion and future work is presented in section 6.

2. Related work:

Abeer S. Desuky et al., [1] have proposed a novel method using the concept of simulated annealing and under sampling techniques on various tasks for classification algorithm. Sahar K. Hussin et al., [2] have proposed the combination of minority sampling using SMOTE and k-means algorithm for improved predictive accuracy for majority and minority class. Moses A. Agebureet al., [3] have setup a software data set approach for class imbalance software defects analysis exclusively for better identification of software defect modules. SatyaSrinivasMaddipataet al., [4] have proposed a new approach for class imbalance learning using kernel principal component analysis and cost sensitive approach for dimensionality and class imbalance reduction.

Shamsul Hudaet al., [5] have proposed a novel software defect prediction ensemble model using oversampling technique for accurate prediction of defective minority samples. Victoria Lopez et al., [6] have introduced approaches for effective solutions regarding data intrinsic issues such as over lapping classes, borderline instances and small discounts etc.Ashwini N et al., [7] have investigated the effects of class imbalance nature on software defect prediction datasets on various under sampling approaches.Peter Gnipet al., [8] have proposed an effective synthetic and adaptive oversampling technique for effective outlier detection.

SikhaBaguiet al., [9] have investigated different re-sampling effect on class imbalance datasets using artificial neural network multi class classifier.Pradeep Kumaret al., [10] have presented a detailed view of different approaches for class imbalance data in improving the performance of both majority and minority sub classes. Minh ThanhVo et al., [11] have presented different techniques for identification of fake jobs from original ones which are in the class imbalance nature by using stop words and different unique techniques.

Ge Song et al., [12] have proposed ensemble clustering framework for textual imbalance data especially with concept drift by performing chunk based reuse of rare class instances.K. Sri Kavya et al., [13] have proposed an ensemble based deepboost classifier to predict software defect modules under scenario of class imbalance nature and high dimensionality.Cui Yin Huang et al., [14] have reviewed and compared effect of different re-sampling and decision tree classifiers on class imbalance data sources. The study also focuses on cost sensitive approaches to handle the class imbalance problem. M. Mostafizur Rahman et al., [15] have proposed an improved under sampling approach for applicability on cardio vascular data.Mateusz Ochal et al., [16] have performed analysis on few shot learning algorithms for solving the class imbalance learning problem.

3. Proposed Algorithms:

3.1 Improved Correlation Over Sampling (ICOS):

This approach uses the novelty of improved correlation based strategy for up sampling of minority subset of instances for improved performance of software defect datasets. Wang [20] have conducted an in depth study on class imbalance datasets of software defect prediction.

The proposed ICOS algorithm is summarized as below.

Algorithm: ICOS

Algorithm: New Decision Tree (D, A, GR)
Input: D – Data Partition
A – Attribute List
GR – Gain Ratio
Output : A Decision Tree

Procedure:
Processing Phase:
Step 1.Take the datasets and find the important features in the dataset by apply Correlation based feature subset filter.

Step 2.*Divide the dataset into majority and minority sub sets. Let the minority subset be P € pi (i = 1,2,..., pnum) and majority subset be N € ni (i = 1,2,..., nnum).*

Let us consider
m' = the number of majority nearest neighbors
T = the whole training set
m= the number of nearest neighbors

Step 3. *Find mostly misclassified instances pi*

$pi = m'$; where m' $(0 \leq m' \leq m)$

if m/ 2 ≤ m'< m then pi is a mostly misclassified instance. Then remove the instances m' from the minority set.

Let us consider
m' = the number of minority nearest neighbors

Step 4. *Find noisy instances pi'*

$pi' = m'$; where m' $(0 \leq m' \leq m)$
If m'= m, i.e. all the m nearest neighbors of pi are majority examples, pi' is considered to be noise or outliers or missing values and are to be removed.

Step 5. *In this step, we generate s × dnum synthetic minority examples from the minority sub set, where s is an integer between 1 and k. One percentage of synthetic examples generated is replica of minority examples and other are the hybrid of minority examples.*

Building Decision Tree:

1. *Create a node N*
2. *If samples in N are of same class, C* ***then***
3. *return N as a leaf node and mark class C;*
4. *If A is empty* ***then***
5. ***return*** *N as a leaf node and mark with majority class;*
6. ***else***
7. *apply Gain Ratio(*D_w *,* A_w *)*
8. *label root node N as f(A)*
9. ***for*** *each outcome j of f(A)* ***do***
10. *subtree j =New Decision Tree(Dj,A)*
11. *connect the root node N to subtree j*
12. ***endfor***
13. ***endif***
14. ***endif***
15. *Return N*

3.2 Under Sampling Strategy (USS):

The proposed USS algorithm is summarized as below.

```
Algorithm: USS

Algorithm:    New Decision Tree (D, A)
Input:        D      - Data Partition
              A       - Attribute List
Output:       A Decision Tree

Procedure:
Processing Phase:
Step 1.Take the datasets and find the important features in the dataset by apply
Correlation based feature subset filter.

Step 2.Divide the dataset  into majority and minority sub sets. Let the minority
subset be P € pi ±i = 1,2,..., pnum) and majority subset be N € ni ±i = 1,2,...,
nnum).

Let us consider
m' = the number of majority nearest neighbors
T = the whole training set
m= the number of nearest neighbors

Step 3. Find mostly misclassified instances pi

      pi = m'; where m' ±0 ≤ m'≤ m)

   if m/ 2 ≤ m'< m then pi is a mostly misclassified instance. Then remove the
instances m' from the minority set.

Let us consider
m' = the number of minority nearest neighbors

Step 4. Find noisy instances pi'

      pi' = m'; where m' ±0 ≤ m'≤ m)
If m'= m, i.e. all the m nearest neighbors of pi are majority examples, pi' is
considered to be noise or outliers or missing values and are to be removed.

Step 5.Take the datasets and find the important features in the dataset by apply
Correlation based feature subset filter.
begin
k ← 0,j←1.
Apply CFS on subset N,
Find Fj from N, k= number of features extracted in classifier subset evaluator
repeat
k=k+1
Select the range for weak or noises instances of Fj.
Remove ranges of weak attributes and form a set of major class examples Nstrong
Until j = k
Form a new dataset using P and Nstrong
End
Building Decision Tree:
    1. Create a node N
    2.    If samples in N are of same class, C then
    3.     return N as a leaf node and mark class C;
    4.        If A is empty then
    5. return N as a leaf node and mark with majority class;
    6. else
    7.             build decision trees
    8.              label root node N as f(A)
    9. for each outcome j of f(A)do
    10.          subtree j =New Decision Tree(Dj, random forest)
    11.           connect the root node N to subtree j
    12. endfor
    13.       endif
    14. endif
    15. Return N
```

3.3 Improved Integrated Sampling Strategy for Software Defect Prediction (IISS)

The detailed procedure of IISS is given in the form of algorithm as follows.

```
Algorithm: Improved Integrated Sampling Strategy (IISS)

Input: S: data stream of examples partitioned into chunks,
P: A set of minor class examples,
N: A set of major class examples,
jPj<jNj, and Fj,the feature set, j > 0.

Output: Average Measure {AUC, Precision, F-Measure, TP Rate, TN Rate}

Procedure:
Processing Phase:
Step 1.Take the datasets and find the important features in the dataset by apply
Correlation based feature subset filter.
Step 2.Divide the dataset into majority and minority subsets.
Let the minority subset be P € pi (i = 1,2,..., pnum) and majority subset be N € ni
(i = 1,2,..., nnum).
Let us consider
m' = the number of majority nearest neighbors
T = the whole training set
m= the number of nearest neighbors
Step 3. Find mostly misclassified instances pi
      pi = m'; where m' (0 ≤ m'≤ m)
   if m/ 2 ≤ m'< m then pi is a mostly misclassified instance. Then remove the
instances m' from the minority set.
Let us consider
m' = the number of minority nearest neighbors

Step 4. Find noisy instances pi'
      pi' = m'; where m' (0 ≤ m'≤ m)
If m'= m, i.e. all the m nearest neighbors of pi are majority examples, pi' is
considered to be noise or outliers or missing values and are to be removed.

Step 5.Take the datasets and find the important features in the dataset by apply
Correlation based feature subset filter.
begin
k ← 0,j←1.
Apply CFS on subset N,
Find Fj from N, k= number of features extracted in classifier subset evaluator
repeat
k=k+1
Select the range for weak or noises instances of Fj.
Remove ranges of weak attributes and form a set of major class examples Nstrong
Until j = k
Form a new dataset using P and Nstrong
End

Adaptation Phase:

for all data S do
if  input data is empty then
        Generate model
else
Compute
MISSCLASS=Pm' using m/ 2 ≤ Pm'<min the minority class P
MISSCLASS=Nm'using m/ 2≤ Nm'<min the majority class N
Remove
Pm'&Nm' from minority class P and majority class N respectively
Generate
PR= {p'1 ,p'2 ,..., p'dnum}, 0 ≤ dnum≤ pnum
s× dnum;
synthetic positive examples from the pr examples in minority set
Update
PR=s × dnum
endif
endfor
```

```
Building Predictive Model Phase:
    1. Create a node N
        2.    If samples in N are of same class, C then
        3.     return N as a leaf node and mark class C;
        4.        If A is empty then
        5.  return N as a leaf node and mark with majority class;
        6.  else
        7.              apply Random Forest
        8.  endif
        9.  endif
        10. Return N
```

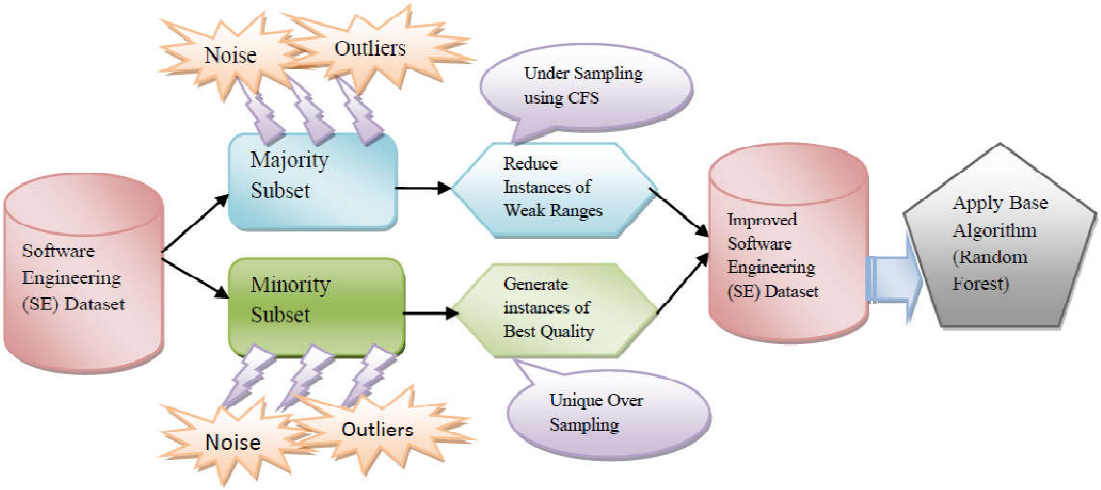
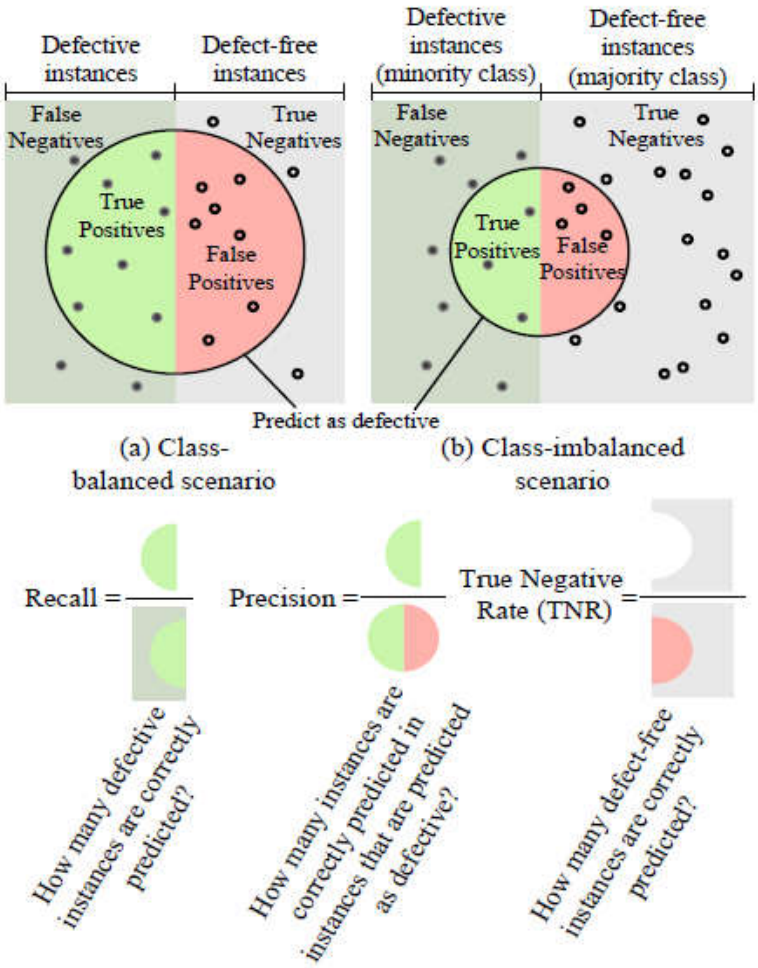


Figure 1. Framework of Improved Integrated Sampling Strategy (IISS)

4. Methodology

Table 1 Details of the seventeen datasets used in the study

No.	Name	#Features	# Modules	# Defects	IR ()
1	AR1	29	121	9	13.4
2	AR3	29	63	8	7.87
3	AR5	29	36	8	4.5
4	CM1	37	327	42	7.78
5	DATATRIEVE	9	130	11	11.81
6	DESHARNAIS	11	81	10	8.1
7	JM1	21	7782	1672	4.65
8	KC1	21	2109	326	6.46
9	KC1-DEFECTIVE	94	145	60	2.41
10	KC1-TOP5	94	145	8	18.12
11	KC3	39	194	36	5.38
12	MC1	38	1988	46	43.21
13	MC21	39	125	44	2.84
14	MW1	37	253	27	9.37
15	PC1	37	705	61	11.55
16	PC3	37	1077	134	8.03
17	REUSE	27	24	9	2.66



5. Results and Discussion:

The experimental results of comparative study is presented in table 3,,4, 5and 6. The results suggest that IISS algorithm have performed better for all the datasets on the compared ICOS and USS algorithms. The reasons for better performance is both the up sampling and down sampling are integrated in the IISS algorithm.

Table 3 Results of IISS model on SDP datasets in terms of AUC

Data set	ICOS	USS	IISS
AR1	0.794 ± 0.232	0.722±0.251	**0.936±0.130**
AR3	0.836 ±0.202	0.793±0.252	**0.893±0.171**
AR5	0.847 ±0.184	0.853±0.272	**0.926±0.156**
DATATRIEVE	0.500 ±0.000	0.500±0.000	**0.772±0.095**
DESHARNAIS	0.787 ±0.136	0.812±0.180	**0.844±0.132**
KC1	0.817 ±0.031	0.785±0.040	**0.888±0.021**
KC1-DEFECTIVE	0.771 ±0.099	---------	**0.860±0.092**
KC1-TOP5PER	0.754 ±0.263	---------	**0.921±0.146**
KC3	0.761 ±0.104	0.714±0.142	**0.819±0.093**
MC1	0.759 ±0.111	0.855±0.114	**0.918±0.057**
MC21	0.701 ±0.123	0.671±0.178	**0.821±0.105**
MW1	0.688 ±0.132	0.721±0.146	**0.865±0.080**
PC1	0.816 ±0.087	0.841±0.084	**0.922±0.038**
PC3	0.753 ±0.093	0.766±0.073	**0.874±0.039**
REUSE	0.969 ±0.093	0.989±0.105	**0.995±0.050**

Bold face value indicate the best performing algorithm

Table 4 Results of IISS model on SDP datasets in terms of Precision

Data set	ICOS	USS	IISS
AR1	0.944 ±0.055	0.954±0.046	**0.954±0.054**
AR3	0.924 ±0.098	**0.924±0.100**	0.917±0.116
AR5	0.895 ±0.147	**0.898±0.177**	0.876±0.210
DATATRIEVE	0.844 ±0.026	**0.912±0.022**	0.838±0.027
DESHARNAIS	0.773 ±0.140	0.778±0.176	**0.788±0.166**
KC1	0.870 ±0.024	**0.878±0.011**	0.864±0.016
KC1-DEFECTIVE	0.786 ±0.080	---------	**0.801±0.082**
KC1-TOP5PER	0.515 ±0.412	---------	**0.635±0.402**
KC3	**0.770 ±0.146**	0.463±0.294	0.706±0.148
MC1	0.627 ±0.221	0.670±0.336	**0.780±0.172**
MC21	0.726 ±0.130	0.555±0.247	**0.752±0.117**
MW1	0.643 ±0.223	0.368±0.325	**0.685±0.188**
PC1	0.658 ±0.138	0.526±0.214	**0.704±0.125**
PC3	0.481 ±0.194	0.376±0.156	**0.626±0.079**
REUSE	**0.957 ±0.133**	0.910±0.206	0.935±0.183

Bold face value indicate the best performing algorithm

Table 5 Results of IISS model on SDP datasets in terms of Recall

Data set	ICOS	USS	IISS
AR1	0.973 ±0.053	**0.996±0.020**	0.985±0.044
AR3	**0.959 ±0.092**	0.936±0.120	0.925±0.133
AR5	0.863 ±0.203	**0.910±0.203**	0.847±0.249
DATATRIEVE	1.000 ±0.000	1.000±0.000	1.000±0.000
DESHARNAIS	0.855 ±0.155	0.771±0.215	**0.778±0.195**
KC1	0.870 ±0.037	**0.953±0.015**	0.932±0.019
KC1-DEFECTIVE	**0.930 ±0.077**	---------	0.910±0.083
KC1-TOP5PER	0.570 ±0.421	---------	**0.635±0.401**
KC3	0.606 ±0.179	0.412±0.257	**0.689±0.163**
MC1	0.351 ±0.174	0.433±0.257	**0.467±0.174**
MC21	0.642 ±0.155	0.524±0.258	**0.809±0.129**
MW1	0.509 ±0.212	0.330±0.288	**0.609±0.194**
PC1	0.590 ±0.151	0.363±0.182	**0.649±0.140**
PC3	0.381 ±0.210	0.261±0.120	**0.634±0.103**
REUSE	**0.995 ±0.050**	0.985±0.111	0.990±0.100

Bold face value indicate the best performing algorithm

Table 6 Results of IISS model on SDP datasets in terms of F-measure

Data set	ICOS	USS	IISS
AR1	0.956 ±0.037	**0.974±0.028**	0.968±0.036
AR3	**0.937 ±0.079**	0.923±0.086	0.913±0.099
AR5	0.858 ±0.146	**0.883±0.161**	0.835±0.198
DATATRIEVE	0.915 ±0.016	**0.954±0.013**	0.912±0.016
DESHARNAIS	**0.799 ±0.108**	0.754±0.158	0.765±0.145
KC1	0.869 ±0.017	**0.914±0.009**	0.896±0.012
KC1-DEFECTIVE	0.848 ±0.055	---------	**0.849±0.064**
KC1-TOP5PER	0.513 ±0.384	---------	**0.607±0.369**
KC3	0.660 ±0.138	0.405±0.220	**0.685±0.125**
MC1	0.427 ±0.167	0.497±0.256	**0.563±0.156**
MC21	0.671 ±0.121	0.519±0.223	**0.772±0.095**
MW1	0.534 ±0.166	0.320±0.251	**0.623±0.150**
PC1	0.612 ±0.120	0.409±0.170	**0.667±0.108**
PC3	0.403 ±0.182	0.301±0.124	**0.626±0.077**
REUSE	**0.969 ±0.092**	0.933±0.157	0.953±0.142

Bold face value indicate the best performing algorithm

The AUC evaluation metric is one of the popular metric used in many benchmark research studies of class imbalance nature on software defect prediction datasets. The AUC results of the IISS algorithm are far better than the ICOS and USS algorithm for all the datasets. Precision, Recall and F-measure are the some of the other well known criteria used for evaluation on software defect prediction for class imbalance learning. The performance of IISS was also good on almost all the datasets in comparison with ICOS and USS. We can conclude that IISS approach is one of the best approaches for effective knowledge discovery for software defect prediction of class imbalance datasets.

6. Conclusion:

In this paper, novel software defect classification algorithms are compared with each other to find their respective strengths and limitations. This method uses unique oversampling and up sampling strategies for efficient solution of class driftsin data streams. Empirical results have shown that the proposed algorithms are effective in terms of AUC, accuracy, precision and f-measurethan other compared approaches.

References:

1. Abeer S. Desuky · SadiqHussain," An Improved Hybrid Approach for Handling Class Imbalance Problem", Arabian Journal for Science and Engineering (2021) 46:3853–3864.https://doi.org/10.1007/s13369-021-05347-7.
2. GgSahar K. Hussin , Salah M. Abdelmageid, Adel Alkhalil, Yasser M. Omar,Mahmoud I. Marie, and Rabie A. Ramadan," Handling Imbalance Classification Virtual Screening Big DataUsing Machine Learning Algorithms",HindawiComplexity, Volume 2021, Article ID 6675279, 15 pages, https://doi.org/10.1155/2021/6675279
3. Moses A. Agebure, Peter A. Agbedemnab," Addressing Class Imbalance in Software Defect Prediction by Averaging", International Journal of Software and Web Sciences, 19(1), December 2016- February 2017, pp. 09-14
4. SatyaSrinivasMaddipata and MalladiSrinivas,"Software Defect Prediction using KPCA & CSANFIS",*Turkish Journal of Computer and Mathematics Education Vol.12 No.9 (2021), 2429– 2436* .
5. Shamsul Huda, Kevin Liu (Shigang Liu), Mohamed Abdelrazek, Amani Ibrahim, Sultan Alyahya, Hmood Al-Dossari and Shafiq Ahmad," An ensemble oversampling model for class imbalance problem in software defect prediction", 2169-3536 (c) 2018 IEEE. Translations, DOI 10.1109/ACCESS.2018.2817572, IEEE Access.
6. Victoria Lopez, Alberto Fernandez, Salvador Garcia, Vasile Palade, Francisco Herrera," An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics", Information Sciences 250 (2013) 113–141.
7. Ashwini N, Bharathi R," Class Imbalance Learning for Software Defect Prediction", International Journal of Engineering Research & Technology (IJERT), www.ijert.org, NCRTS`14 Conference Proceedings, ISSN: 2278-0181.
8. Peter Gnip, LiberiosVokorokos and Peter Drotár, "Selective oversampling approach for strongly imbalanced data",*PeerJComput. Sci.*, DOI 10.7717/peerj-cs.604,2021.
9. SikhaBagui and Kunqi Li," Resampling imbalanced data for network intrusion detection datasets", *(2021) 8:6*, https://doi.org/10.1186/s40537-020-00390-x.
10. Pradeep Kumar, RoheetBhatnagar, Kuntal Gaur, and AnuragBhatnagar," Classification of Imbalanced Data:Review ofMethods and Applications", IOP Conf. Series: Materials Science and Engineering 1099 (2021) 012077, IOP Publishing, doi:10.1088/1757-899X/1099/1/012077.
11. Minh Thanh Vo, Anh H. Vo, Trang Nguyen, Rohit Sharmaand Tuong Le," Dealing with the Class Imbalance Problem in the Detection of Fake Job Descriptions", CMC, 2021, vol.68, no.1,*Computers,Materials& Continua,* DOI:10.32604/cmc.2021.015645.
12. Ge Song and Yunming Ye," A Dynamic Ensemble Framework for Mining Textual Streams with Class Imbalance",Hindawi Publishing Corporation Scientific World

JournalVolume 2014, Article ID 497354, 11 pages,http://dx.doi.org/10.1155/2014/497354.

13. K. Sri Kavya, Dr. Y. Prasanth," An Ensemble DeepBoost Classifier for Software Defect Prediction", International Journal of Advanced Trends in Computer Science and Engineering, 9(2), March - April 2020, 2021 – 2028.
14. Cui Yin Huang and Hong Liang Dai," Learning from class-imbalanced data: review of data driven methods and algorithm driven methods", *Data Science in Finance and Economics* Volume 1, Issue 1, 21–36.
15. M. Mostafizur Rahman and D. N. Davis," Addressing the Class Imbalance Problem in Medical Datasets", *International Journal of Machine Learning and Computing, Vol. 3, No. 2, April 2013,*
16. Mateusz Ochal, MassimilianoPatacchiola, Jose Vazquez, Amos Storkey, Sen Wang," Few-Shot Learning with Class Imbalance",

Mr K. Nitalaksheswara Rao is pursuing Ph.D. In Computer Science and Systems Engineering at Andhra University Visakhapatnam India. He received his Master's degree M.Tech in Computer Science and engineering in 2009. He is the topper of the school at SSC level and batch topper at M.Tech level. Now, he is an Assistant Professor in department of Computer Science and engineering, Narasaraopeta engineering college, Guntur Andhra Pradesh India. He has 10 plus years of teaching and 5 years of research experience. He published one text book titled "3D Printing and It's Applications" with international German publisher. He published 6 patents intellectual property of India. Mr Rao published his research work in reputed international SCI journals like springer, AJSE, EVOS, IJSEIA, and also in international conferences. He is the reviewer of the Journals "journal of supercomputing" and "Arabian journal for Science and Engineering" International SCI Springer Journals. He is the life member of ISTE and member of IAENG. His current research interest includes Software Engineering, Data Engineering, Quality Assurance, Machine learning and Deep learning.

Dr.Ch. Satyananda Reddy is aProfessor in the Department of Computer Science and Systems Engineering, College of Engineering (A), Andhra University, Visakhapatnam. He obtained his Ph.D. in Computer Science Engineering from the Faculty of Engineering, Andhra University, INDIA. Dr.Reddy is mainly interested in the fields of Software Engineering i.e. Software Engineering, Software Project and Process Management, Software Estimation, Software Metrics, Software Quality Assurance, Human Computer Interaction, Software Requirements Engineering, Software Testing, Data Engineering. Having research experience of 20 years, he hasPublished3 patents with Intellectual property of India and was published 60+ papers in the International Journals in the field of Computer Science and Software Engineering.

He is the Reviewer for several International Journals and Software Engineering Journals like IEEE Transactions on Software Engineering, IEEE Transactions on Computers, IET Software, IJSEKE (World Scientific), International Journal of Software Engineering (Software Engineering Competence Centre), International Journal of Information Technology and Software Engineering. He is the Editorial Board Member for JECI, IJCER, Journals Pub and many other journals. He was the Program Committee Member/ Reviewer for a number of International Conferences. He is a Life Member of IAENG, IDES, ISTE, and ACEEE.