

|  |  |
|--|--|
| <b>FORM 1</b><br><b>THE PATENTS ACT, 1970</b><br><b>(39 of 1970)</b><br><b>&amp;</b><br><b>THE PATENTS RULES, 2003</b><br><b>APPLICATION FOR GRANT OF PATENT</b><br><b>[See sections 7, 54 &amp; 135 and rule 20(1)]</b> | <b>(FOR OFFICE USE ONLY)</b><br><b>Application No.:</b> .....<br><b>Filing Date:</b> .....<br><b>Amount of Fee Paid:</b> .....<br><b>CBR No.:</b> .....<br><b>Signature:</b> ..... |
|--|--|

**1. APPLICANT(S):**

| Sr.No | Name                        | Nationality | Address  | Country | State          |
|-------|-----------------------------|-------------|--|---------|----------------|
| 1     | Prof.M.James Stephen        | Indian      | Professor, Department of CSE, Wellfare Institute of Science Technology and Management (WISTM), Visakhapatnam, Andhra Pradesh, India. Pin Code:530007 | India   | Andhra Pradesh |
| 2     | Mr.K.Nitalaksheswara Rao    | Indian      | Research Scholar, Department of CS & SE, A.U. College of Engineering (A), Andhra University, Visakhapatnam, Andhra Pradesh, India. Pin Code:530003   | India   | Andhra Pradesh |
| 3     | Prof. P.V.G.D. Prasad Reddy | Indian      | Senior Professor, Department of CS & SE, A.U. College of Engineering (A), Andhra University, Visakhapatnam, Andhra Pradesh, India. Pin Code:530003   | India   | Andhra Pradesh |
| 4     | Mr.Ch.V.Murali Krishna      | Indian      | Associate Professor, Department of CSE NRI Institute of Technology, Agiripalli, Krishna District, Andhra Pradesh, India. Pin Code: 521212            | India   | Andhra Pradesh |

**2. INVENTOR(S):**

| Sr.No | Name                        | Nationality | Address  | Country | State          |
|-------|-----------------------------|-------------|--|---------|----------------|
| 1     | Prof.M.James Stephen        | Indian      | Professor, Department of CSE, Wellfare Institute of Science Technology and Management (WISTM), Visakhapatnam, Andhra Pradesh, India. Pin Code:530007 | India   | Andhra Pradesh |
| 2     | Mr.K.Nitalaksheswara Rao    | Indian      | Research Scholar, Department of CS & SE, A.U. College of Engineering (A), Andhra University, Visakhapatnam, Andhra Pradesh, India. Pin Code:530003   | India   | Andhra Pradesh |
| 3     | Prof. P.V.G.D. Prasad Reddy | Indian      | Senior Professor, Department of CS & SE, A.U. College of Engineering (A), Andhra University, Visakhapatnam, Andhra Pradesh, India. Pin Code:530003   | India   | Andhra Pradesh |
| 4     | Mr.Ch.V.Murali Krishna      | Indian      | Associate Professor, Department of CSE NRI Institute of Technology, Agiripalli, Krishna District, Andhra Pradesh, India. Pin Code: 521212            | India   | Andhra Pradesh |

**3. TITLE OF THE INVENTION: A NOVEL IMPROVED INTEGRATED SAMPLING STRATEGY FOR SOFTWARE DEFECT PREDICTION****4. ADDRESS FOR CORRESPONDENCE OF APPLICANT/ AUTHORISED PATENT AGENT IN INDIA:**

**Prof. James Stephen Meka**, Professor, Department of CSE, Wellfare Institute of Science Technology and Management (WISTM), Visakhapatnam, Andhra Pradesh, India. Pin Code:530007

Telephone No.:  
 Fax No.:  
 Mobile No: .....  
 E-mail: [jamesstephenmm@yahoo.com](mailto:jamesstephenmm@yahoo.com)

**5. PRIORITY PARTICULARS OF THE APPLICATION(S) FILED IN CONVENTION COUNTRY:**

| Sr.No | Country | Application Number | Filing Date | Name of the Applicant | Title of the Invention |
|-------|---------|--------------------|-------------|-----------------------|------------------------|
|-------|---------|--------------------|-------------|-----------------------|------------------------|

**6. PARTICULARS FOR FILING PATENT COOPERATION TREATY (PCT) NATIONAL PHASE APPLICATION:**

|                                  |   |
|----------------------------------|---|
| International Application Number | International Filing Date as Allotted by the Receiving Office |
| PCT//                            |   |

**7. PARTICULARS FOR FILING DIVISIONAL APPLICATION**

|                                     |  |
|-------------------------------------|--|
| Original (first) Application Number | Date of Filing of Original (first) Application |
|-------------------------------------|--|

**8. PARTICULARS FOR FILING PATENT OF ADDITION:**

|                                   |                                    |
|-----------------------------------|------------------------------------|
| Main Application / Patent Number: | Date of Filing of Main Application |
|-----------------------------------|------------------------------------|

**9. DECLARATIONS:**

**(i) Declaration by the inventor(s):**

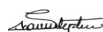
I/We, Prof.M.James Stephen, Mr.K.Nitalaksheswara Rao, Prof. P.V.G.D. Prasad Reddy, Mr.Ch.V.Murali Krishna is/are the true & first inventor(s) for this invention and declare that the applicant(s) herein is/are my/our assignee or legal representative.

(a)Date: ----- Dated this 08<sup>th</sup> day of February, 2022

(b)Signature(s) of the inventor(s): .....

(c)Name(s):

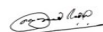
Prof.M.James Stephen



Mr.K.Nitalaksheswara Rao



Prof. P.V.G.D. Prasad Reddy



Mr.Ch.V.Murali Krishna



**(ii) Declaration by the applicant(s) in the convention country:**


I/We, the applicant(s) in the convention country declare that the applicant(s) herein is/are my/our assignee or legal representative.

(a)Date: ----- Dated this 08<sup>th</sup> day of February, 2022

(b)Signature(s) of the inventor(s): .....

(c)Name(s):

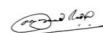
Prof.M.James Stephen



Mr.K.Nitalaksheswara Rao



Prof. P.V.G.D. Prasad Reddy



Mr.Ch.V.Murali Krishna



**(iii) Declaration by the Applicant(s):**

- The Complete specification relating to the invention is filed with this application.
- I am/ We are, in the possession of the above mentioned invention.
- There is no lawful ground of objection to the grant of the Patent to me/us.

**10. FOLLOWING ARE THE ATTACHMENTS WITH THE APPLICATION:**

| Sr.No | Document Description                  | File Name                       |
|-------|---------------------------------------|---------------------------------|
| 1     | Complete Specifications(Form-2)       | CompletespecificationsForm2.pdf |
| 2     | Drawings                              | Drawings.pdf                    |
| 3     | Request For Early Publication(Form-9) | Form9.pdf                       |
| 4     | Statement of Undertaking (Form 3)     | Form3.pdf                       |
| 5     | Declaration of Inventorship (Form 5)  | Form5.pdf                       |

I/We hereby declare that to the best of my/our knowledge, information and belief the fact and matters stated herein are correct and I/We request that a patent may be granted to me/us for the said invention.

Dated this (Final Payment Date): -----

Signature: .....

Name(s):

Prof.M.James Stephen

Mr.K.Nitalaksheswara Rao

Prof. P.V.G.D. Prasad Reddy

Mr.Ch.V.Murali Krishna









To

The Controller of Patents

**The Patent office at CHENNAI**

(54) Title of the invention : A Novel Improved Integrated Sampling Strategy for Software Defect Prediction

(51) International classification :G06F0011360000, G06N0007000000, G06N0005000000, G06N0020000000, G06N0005040000

(86) International Application No :PCT// /  
Filing Date :01/01/1900

(87) International Publication No : NA

(61) Patent of Addition to Application Number :NA  
Filing Date :NA

(62) Divisional to Application Number :NA  
Filing Date :NA

(71)Name of Applicant :  
**1)Prof.M.James Stephen**  
 Address of Applicant :Professor, Department of CSE, Wellfare Institute of Science Technology and Management (WISTM), Visakhapatnam, Andhra Pradesh, India. Pin Code:530007 -----

**2)Mr.K.Nitalaksheswara Rao**  
**3)Prof. P.V.G.D. Prasad Reddy**  
**4)Mr.Ch.V.Murali Krishna**

Name of Applicant : NA  
 Address of Applicant : NA

(72)Name of Inventor :  
**1)Prof.M.James Stephen**  
 Address of Applicant :Professor, Department of CSE, Wellfare Institute of Science Technology and Management (WISTM), Visakhapatnam, Andhra Pradesh, India. Pin Code:530007 -----

**2)Mr.K.Nitalaksheswara Rao**  
 Address of Applicant :Research Scholar, Department of CS & SE, A.U. College of Engineering (A), Andhra University, Visakhapatnam, Andhra Pradesh, India. Pin Code:530003 -----

**3)Prof. P.V.G.D. Prasad Reddy**  
 Address of Applicant :Senior Professor, Department of CS & SE, A.U. College of Engineering (A), Andhra University, Visakhapatnam, Andhra Pradesh, India. Pin Code:530003 -----

**4)Mr.Ch.V.Murali Krishna**  
 Address of Applicant :Associate Professor, Department of CSE, NRI Institute of Technology, Agiripalli, Krishna District, Andhra Pradesh, India. Pin Code: 521212 -----

(57) Abstract :  
 Software Defect Prediction using data mining techniques is one of the best practices for finding defective modules. On normal datasets, existing classification techniques can be applied for effective knowledge discovery. Most of the real world data sources are biased towards any one of the class and 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. There is a need for the invention that can increase the software defect prediction rate. The present invention disclosed here is a Novel Improved Integrated Sampling Strategy (IISS) for Software Defect Prediction comprising of three phases: Processing Phase (201); Adaptation Phase (202); and Building Predictive Model Phase (203). The present invention disclosed herein predicts the software defects on Skewed Data Distribution. The invention of this disclosure uses noisy removal strategy by integrating both over sampling and under sampling for software defect prediction. The experimental analysis of the present invention disclosed herein is conducted on skewed software defect prediction datasets by the proposed IISS and its performance is compared with C4.5, C4.5 with Balance dataset, RF (Random Forest) and RF with Balance dataset algorithms with various class imbalance evaluation measures.

No. of Pages : 27 No. of Claims : 10

# **FORM 2**

THE PATENTS ACT, 1970  
(39 of 1970) &  
THE PATENTS RULES, 2003  
**COMPLETE SPECIFICATION**  
(See section 10, rule 13)

1. TITLE OF THE INVENTION:

**A NOVEL IMPROVED INTEGRATED SAMPLING STRATEGY  
FOR SOFTWARE DEFECT PREDICTION**

## 2. APPLICANT(S)

| Sr.No | Name                        | Nationality | Address   | Country | State          |
|-------|-----------------------------|-------------|---|---------|----------------|
| 1     | Prof.M.James Stephen        | Indian      | Professor, Department of CSE, Welfare Institute of Science Technology and Management (WISTM), Visakhapatnam, Andhra Pradesh, India. Pin Code:530007 | India   | Andhra Pradesh |
| 2     | Mr.K.Nitalaksheswara Rao    | Indian      | Research Scholar, Department of CS & SE, A.U. College of Engineering (A), Andhra University, Visakhapatnam, Andhra Pradesh, India. Pin Code:530003  | India   | Andhra Pradesh |
| 3     | Prof. P.V.G.D. Prasad Reddy | Indian      | Senior Professor, Department of CS & SE, A.U. College of Engineering (A), Andhra University, Visakhapatnam, Andhra Pradesh, India. Pin Code:530003  | India   | Andhra Pradesh |
| 4     | Mr.Ch.V.Murali Krishna      | Indian      | Associate Professor, Department of CSE NRI Institute of Technology, Agiripalli, Krishna District, Andhra Pradesh, India. Pin Code: 521212           | India   | Andhra Pradesh |

## 3. PREAMBLE TO THE DESCRIPTION:

### **COMPLETE SPECIFICATION**

The following specification particularly describes the invention and the manner in which it is to be performed.

# **A NOVEL IMPROVED INTEGRATED SAMPLING STRATEGY FOR SOFTWARE DEFECT PREDICTION**

## **FIELD OF INVENTION**

5 The present invention relates to the technical field of Data Mining.

Particularly, the present invention is related to a Novel Improved Integrated Sampling Strategy (IISS) for Software Defect Prediction of the broader field of Data Mining of Computer Science Engineering.

10 More particularly, the present invention relates to a Software Defect Prediction method which uses Novel Improved Integrated Sampling Strategy (IISS) for Software Defect Prediction on the Skewed Data sources which are class imbalance in nature.

## **BACKGROUND OF INVENTION**

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  
15 requirement analysis, designing, coding and testing. The complete or exhaustive testing for finding all the errors in the software modules is a tedious job. A common method for software defect prediction of class imbalance nature, need to be very accurate and precise, in spite of very less number of defective module instances. There by developing such a model is ineffective in the practical implementation due to a very high Imbalance ratio. In this study,  
20 we propose to use correlation based oversampling, instance ranges specific under sampling strategy and Improved integrated sampling techniques to help improve both majority and minority sub sets. The main rationale behind the approach is feature to feature correlation index and feature to class correlation index in the implementation of improved correlation based over sampling algorithm to learn range of instance. The proposals are supported with  
25 sound experimental setup for effective evaluation of class imbalance software defect datasets significantly improves classification over a decision tree as baseline. The recent research in software defect prediction learning has not laid much stress to consider the software defect



prediction as an efficient implementation in all the scenarios. The software defect prediction is also considered in the class balance framework where all the class are regard as equally. The main focus of our research is to overcome the issues with high imbalance ratio scenario in the knowledge discovery process of software defect prediction. The proposal, Improved  
5 Integrated Sampling Strategy (IISS) is well capable of handling effectively the process of knowledge discovery from the skewed software prediction datasets.

Maximally Collapsing Metric Learning (MCML) proposed by other inventors is an innovative approach to learning metric. The instances in the same class are close, and those in different classes are far. The metric after learned would make instances in the same class  
10 close and those in different classes far. To begin with, the approach assumes that distance between instances in the same class is zero while distance between instances in different classes is infinite. In other words, the approach tries to map same class instances into a single point using a linear projection. The goal of the approach is to find matrix  $M$  such that set of  
15 instances is as close as possible to another set of instances. In order to match those distributions, KL divergence [52], which measures the difference between two probability distributions, is minimized. The above said technique can be used for identifying noisy or borderline range of instances in majority subset for performing under sampling. The above said all techniques and other novel ideas are well utilized in the proposed model for improved performance on skewed software defect dataset predication.

20 Learning from class-imbalanced data continues to be a common and challenging problem in supervised learning as standard classification algorithms are designed to handle balanced class distributions. While different strategies exist to tackle this problem, methods which generate artificial data to achieve a balanced class distribution are more versatile than modifications to the classification algorithm. Such techniques, called over samplers, modify  
25 the training data, allowing any classifier to be used with class-imbalanced datasets. Many algorithms have been proposed for this task, but most are complex and tend to generate unnecessary noise. The two classes existing in the software defect datasets are: defective modules and non-defective modules.

The proper learning of the defective modules is very crucial for the improved reliability of  
30 the quality assurance. The benchmark classification algorithms build the model from the software engineering datasets which are having very limited instances for defective modules. The class imbalance nature of the data source can be reduced either by increasing the

instances in the minority subset, i.e. Oversampling or by decreasing the instances in the majority subset, i.e., under sampling. In the phase, it is proposed to oversample the minority subset, which is the number of instances in the minority subset will be increased. The oversampling will be performed using different techniques such as replication of some percentage of instances in the minority subset, synthetic instances generation, hybrid instances generation with properties of two or more instances.

## SUMMARY OF INVENTION

The proposed invention disclosed here uses a unique strategy for replicating and generating instances in the minority subset and at the same time reducing the instances from majority subset. The proposed technique is known as improved integrated sampling strategy (IISS) as it integrates both sampling strategies in a single method. This rationale behind combining both the strategies is to address the issues of both majority and minority subsets. The task of combining these strategies in the single class is a challenging task as the counter effects need to be properly under taken for consideration of the learning process for class imbalance problem of software defect prediction. In the present invention disclosed here, a novel hybrid algorithm for imbalanced data with the application of software defect prediction has been proposed. This method uses unique oversampling and intelligent under sampling technique to almost balance dataset such that to minimize the imbalance effect in the software defect prediction model. The set of results run on 16 skewed software defect datasets show that the proposed approach IISS have performed better than the benchmark compared algorithms. In future extension of the work, the proposed approach can be implemented on more than 2 class skewed datasets in another domain of applicability

Further, this present invention proposes a new IISS algorithm, has performed well on all the measures. However, IISS is better in the aspect of class imbalance measures, which is the problem in hand for real world datasets. Finally, IISS have generated favorable results in terms of class imbalance measures for software defect prediction. The total experimental simulation conducted on 16 class imbalance software defect datasets project that prominent recursive oversampling and intelligent under sampling approaches can improve the effectiveness when dealing with imbalanced data, as it has helped the IISS method to be the best performing algorithms when compared with benchmark algorithms.

## BRIEF DESCRIPTION OF SYSTEM

The accompanying illustrations are incorporated into and constitute part of this specification, and they are utilized to better understand the invention. When viewed with the discussion, the drawing depicts exemplary embodiments of the current disclosure and aids  
5 comprehension of its concepts. The drawings are solely for illustrative purposes and do not in any way limit the scope of the disclosure. As evidenced by the usage of the same reference numerals, the elements are comparable but not identical. On the other hand, different reference numerals might be used to define linked components. In some embodiments, such elements and/or components may not be present, while in others, they may be present.

10 Referring to Figure 1, illustrates General Software Defect Prediction Process comprising of: Software Archives (101); Instances with metrics and labels (102); Preprocessing (103); Training Instances (104); New Instances (105); Build a Prediction Model (106); and Classification (107), in accordance with an exemplary embodiment of the disclosure.

The present invention referring to Figure 2, illustrates Phases in Improved Integrated  
15 Sampling Strategy comprising of three phases: Processing Phase (201); Adaptation Phase (202); and Building Predictive Model Phase (203), in accordance with an exemplary main embodiment of the disclosure.

Referring to Figure 3, illustrates the AR1 Dataset, in accordance with an exemplary embodiment of the disclosure.

20 Referring to Figure 4, illustrates the KC1 Dataset, in accordance with an exemplary embodiment of the disclosure.

Referring to Figure 5, illustrates the MC1 Dataset, in accordance with an exemplary embodiment of the disclosure.

Referring to Figure 6, illustrates the PC1 Dataset, in accordance with an exemplary  
25 embodiment of the disclosure.

Referring to Figure 7, illustrates the trends in AUC for C4.5, REP, CART versus IISS on SDP Datasets, in accordance with an exemplary embodiment of the disclosure.

These accompanying illustrations are provided to aid comprehension of the disclosure and  
6

should not be construed as limiting the disclosure's breadth, scope, or applicability. The invention is not limited to these drawing, some elements and/or components, on the other hand, may not be present in embodiments, and others may be used in different ways than those shown in the designs. Depending on the context, the use of a single language to describe a component or element may contain a plural number of such components or elements, and vice versa.

## **DETAIL DESCRIPTION OF THE PRESENT INVENTION**

The accompanying illustrations are incorporated into and constitute part of this specification, and they are utilized to better understand the invention. When viewed with the discussion, the drawing depicts exemplary embodiments of the current disclosure and aids comprehension of its concepts. The drawings are solely for illustrative purposes and do not in any way limit the scope of the disclosure. As evidenced by the usage of the same reference numerals, the elements are comparable but not identical. On the other hand, different reference numerals might be used to define linked components. In some embodiments, such elements and/or components may not be present, while in others, they may be present.

Further, it will nevertheless be understood that no limitation in the scope of the invention is thereby intended, such alterations and further modifications in the figures and such further applications of the principles of the invention as illustrated herein being contemplated as would normally occur to one skilled in the art to which the invention relates.

Also, it is to be understood that the phraseology and terminology used herein is for the purpose of description and should not be regarded as limiting. Further, reference herein to "one embodiment" or "an embodiment" means that a particular feature, characteristic, or function described in connection with the embodiment is included in at least one embodiment of the invention.

Furthermore, the appearances of such phrase at various places herein are not necessarily all referring to the same embodiment. The terms "a" and "an" herein do not denote a limitation of quantity, but rather denote the presence of at least one of the referenced items.

Referring to Figure 1, illustrates General Software Defect Prediction Process comprising of: Software Archives (101); Instances with metrics and labels (102); Preprocessing (103); Training Instances (104); New Instances (105); Build a Prediction Model (106); and Classification (107). The Software Archives (101) contains multiple files and folders into a

single file, Instances with metrics and labels (102) is the process of collecting files (Instances) and labeling them based on the number of defects for each file. The Preprocessing (103) is for normalizing the files and data by removing the noise for giving equals weights for the features of the datasets to classify. The Training Instances (104) are used to train the model, and these training instances (104) contain instances features and labels obtained from the post released defect files. The New Instances (105) are used to predict the defect present or not by the trained model. The trained model is Build a Prediction Model (106) contains instances used to build models constitute a training set, whereas those used to test the learned models constitute a test set. The instances are classified (107) into defect or non-defect.

10 **Problem Statements:** Software development industry is one of the main technological applicability of the recent developments. Knowledge acquire from software developed can be used for different applications in real world scenario. The analyzed results of software defect analyses can be used for the below contexts:

- To identify the modules with errors.
- 15 • To identify total amount of defects.
- To analyze and identify the reasons and solutions for effective defective case prediction.
- To allocate error prone sub blocks further developing.

By using this algorithm minority classes can be oversampled and majority class is under sampled to populate the further required instances in the software defect prediction datasets for effective knowledge discovery.

20 The main objectives are:

- To apply novel under sampling technique to remove noisy and outlier instances.
- To apply novel over sampling strategies to over sample instances in the minority subset.
- To use advance filtering techniques to identify the correlated range of features from the data sources.
- 25 • To use efficient class imbalance validation measures for better results analysis.
- To decrease the overall time required for debugging the software.

In this research, we aim to answer several questions to analyze the performance of test models on software defect analysis datasets.

30 RQ1: What is the effect of class imbalance nature on the classification process?

RQ2: What is the level of class imbalance in software defect prediction datasets?

RQ3: What is the direct or indirect effect of imbalance nature in the software defect prediction?

RQ4: What is the effect of base learner on the class imbalance nature?

RQ5: What are the implications of class imbalance nature on the scalability of the dataset?

The present invention referring to Figure 2, illustrates Phases in Improved Integrated Sampling Strategy (IISS) comprising of three phases: Processing Phase (201); Adaptation Phase (202); and Building Predictive Model Phase (203). The integrated hybrid sampling approaches are required as the user cannot predict where the noise situation exists that is in the majority subset or minority subset. So, in the invention both over sampling and under sampling for a finer results generation are presented and advised. The IISS algorithm consists of three phases called Processing Phase, Adaptation phase and Building predictive Model Phase. They are as follow:

**Processing Phase (201):** In Processing Phase, the class imbalance software defect dataset which contains both majority and minority sub classes is divided into two separate classes. There by the majority class which has more percentage of instances is applied with under sampling and the minority class with less percentage of instances is applied with over sampling. In this proposed approach we have applied both majority and minority for better improvement of the data source. The under sampling technique used in the proposed approach uses correlation based feature selection technique to find the influential features of the data source. The weak range of instances for elimination can be identified using the irrelevant features identified in the filtering technique. Removal of these instances have the dual effect on the data source, one is the unnecessary instances ranges are removed helping for quality synthetic instances generation in the later on stages. The other is reducing the percentage of instances from the majority subset to limit the problem of class imbalance.

**Adaptation Phase (202):** The minority subset is further analyzed to form any existing outliers and noisy instances. The removal of noisy and outlier instances will improve the quality of the dataset and reduce the risk of magnifying the influence of outliers and noisy instances in the over sampling stage. In oversampling process different type of techniques are applied such as replicating the existing instances, generating synthetic instances, hybrid techniques for generating novel instances. The synthetic instances generation technique uses different level of generation, ranging from 10 to 100 percent instances generation depending on the unique properties of the datasets.

**Building Predictive Model Phase (203):** The improved majority and minority subsets are

combined to form a almost balanced software defect prediction dataset and is applied to a base algorithm. In this case we have selected random forest as the base algorithm and evaluate on different evaluation metrics.

The steps involved in the proposed IISS invention are elaborated step by step as follow:

| <b><i>Algorithm: Improved Integrated Sampling Strategy (IISS)</i></b>  |
|--|
| <p><b>Input:</b> S: data stream of examples partitioned into chunks,<br/> P: A set of minor class examples,<br/> N: A set of major class examples,<br/> <math>jP_j &lt; jN_j</math>, and <math>F_j</math>, the feature set, <math>j &gt; 0</math>.<br/> where <math>jP_j</math> are the number of minority instances and <math>jN_j</math> are the number of majority instances.</p> <p><b>Output:</b> Average Measure {AUC, Precision, F-Measure, TP Rate, TN Rate}</p> <p><b>Procedure:</b></p> <p><b>Processing Phase:</b></p> <p><b>Step 1.</b> Take the datasets and find the important features in the dataset by apply Correlation based feature subset filter.</p> <p><b>Step 2.</b> Divide the dataset into majority and minority subsets.<br/> Let the minority subset be <math>P \in p_i</math> (<math>i = 1, 2, \dots, pnum</math>) and majority subset be <math>N \in n_i</math> (<math>i = 1, 2, \dots, nnum</math>).<br/> Let us consider<br/> <math>m'</math> = the number of majority nearest neighbors<br/> <math>T</math> = the whole training set<br/> <math>m</math> = the number of nearest neighbors</p> <p><b>Step 3.</b> Find mostly misclassified instances <math>p_i</math><br/> <math>p_i = m'</math>; where <math>m' (0 \leq m' \leq m)</math><br/> if <math>m/2 \leq m' &lt; m</math> then <math>p_i</math> is a mostly misclassified instance. Then remove the instances <math>m'</math> from the minority set.<br/> Let us consider<br/> <math>m'</math> = the number of minority nearest neighbors</p> <p><b>Step 4.</b> Find noisy instances <math>p_i'</math><br/> <math>p_i' = m'</math>; where <math>m' (0 \leq m' \leq m)</math></p> |

If  $m' = m$ , i.e. all the  $m$  nearest neighbors of  $p_i$  are majority examples,  $p_i$  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**

$k \leftarrow 0, j \leftarrow 1$ .

**Apply** CFS on subset  $N$ ,

Find  $F_j$  from  $N$ ,  $k =$  number of features extracted in classifier subset evaluator

**repeat**

$k = k + 1$

Select the range for weak or noises instances of  $F_j$ .

Remove ranges of weak attributes and form a set of major class examples  $N_{strong}$

**Until**  $j = k$

Form a new dataset using  $P$  and  $N_{strong}$

**End**

**Adaptation Phase:**

**for all** data  $S$  **do**

**if** input data is empty **then**

**Generate** model

**else**

**Compute**

$MISSCLASS = P_{m'}$  using  $m/2 \leq P_{m'} < \min$  the minority class  $P$

$MISSCLASS = N_{m'}$  using  $m/2 \leq N_{m'} < \min$  the majority class  $N$

**Remove**

$P_{m'}$  &  $N_{m'}$  from minority class  $P$  and majority class  $N$  respectively

**Generate**

$PR = \{p^1, p^2, \dots, p^{dnum}\}, 0 \leq dnum \leq pnum$

$s \times dnum$ ;

    synthetic positive examples from the  $pr$  examples in minority set

    Update

$PR = s \times dnum$

**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  $N$

In the Processing Phase, the dataset is passed on through a attribute evaluation filter and important features are selected, thereby removing the unnecessary features from the dataset.

**Step 1:** Take the datasets and find the important features in the dataset by apply Correlation based feature subset filter. Then the dataset is split into majority and minority subsets for further processing. The minority subset is indicated with 'P' and the majority subset is indicated with 'N'.

**Step 2:** Divide the dataset into majority and minority subsets. Let the minority subset be  $P \in p_i (i = 1, 2, \dots, pnum)$  and majority subset be  $N \in n_i (i = 1, 2, \dots, nnum)$ .

The different terms used in the algorithm such as follows: Let us consider

$m'$  = the number of majority nearest neighbors

$T$  = the whole training set

$m$  = the number of nearest neighbors

The noisy and outlier instances from majority and minority subsets are detected and removed.

**Step 3:** Find mostly misclassified instances  $p_i$

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

if  $m/2 \leq m' < m$  then  $p_i$  is a mostly misclassified instance. Then remove the instances  $m'$  from the minority set.

Let us consider

$m'$  = the number of minority nearest neighbors 12

**Step 4:** Find noisy instances  $p_i'$

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

If  $m' = m$ , i.e. all the  $m$  nearest neighbors of  $p_i$  are majority examples,  $p_i'$  is considered to be noise or outliers or missing values and are to be removed. The weak range of instances from majority and minority subsets are detected and removed. Find  $F_j$  from  $N$ ,  $k =$  number of features extracted in classifier subset evaluator.

**Repeat:**

$k = k + 1$

Select the range for weak or noises instances of  $F_j$ .

10 Remove ranges of weak attributes and form a set of major class examples  $N_{strong}$

**Until**  $j = k$

Form a new dataset using  $P$  and  $N_{strong}$

The misclassified instances from majority and minority subsets are detected and removed.

$MISSCLASS = P_{m'}$  using  $m/2 \leq P_{m'} < \min$  the minority class  $P$

15  $MISSCLASS = N_{m'}$  using  $m/2 \leq N_{m'} < \min$  the majority class  $N$

The synthetic instances are generated in the minority subset depending upon the unique properties of the datasets

$PR = \{p'_1, p'_2, \dots, p'_{dnum}\}$ ,  $0 \leq dnum \leq pnum$   
 $s \times dnum$ ;

20 Synthetic positive examples from the  $pr$  examples in minority set. Then the decision tree is build using the Random forest base classifier.

There are many advantages of the proposed technique on the existing prediction methods. First, the proposed approach is an independent entity for implementation on any of the decision tree approaches as if it generates classification of classes. The only thing, which should be considered, is efficient model building techniques for defective modules prediction.

**Datasets and Evaluation Criteria's:** The experimental simulation is conducted on the 16 varied datasets of software defect prediction with class imbalance nature. The nature of the datasets is class imbalance, where one class has predominately more number of instances than the other class. The datasets are obtained from the PROMISE repository. The complete details of all the simulated datasets are presented in the Table 1.

**TABLE 1**

**Summary of the defect prediction datasets used in the invention**

| S.No       | 1    | 2    | 3   | 4    | 5     | 6   | 7    | 8    | 9    | 10    | 11   | 12    | 13   | 14   | 15    | 16   | 17    |
|------------|------|------|-----|------|-------|-----|------|------|------|-------|------|-------|------|------|-------|------|-------|
| System     | AR1  | AR3  | AR5 | CM1  | DR    | DE  | JM1  | KC1  | KC1D | KC1T  | KC3  | MC1   | MC21 | MW1  | PC1   | PC3  | REUSE |
| Attributes | 29   | 29   | 29  | 37   | 9     | 11  | 21   | 21   | 94   | 94    | 39   | 38    | 39   | 37   | 37    | 37   | 27    |
| Modules    | 121  | 63   | 36  | 327  | 130   | 81  | 7782 | 2109 | 145  | 145   | 194  | 1988  | 125  | 253  | 705   | 1077 | 24    |
| Defective  | 9    | 8    | 8   | 42   | 11    | 10  | 1672 | 326  | 60   | 8     | 36   | 46    | 44   | 27   | 61    | 134  | 9     |
| IR         | 13.4 | 7.87 | 4.5 | 7.78 | 11.81 | 8.1 | 6.46 | 6.46 | 2.41 | 18.12 | 5.38 | 43.21 | 2.84 | 9.37 | 11.55 | 8.03 | 2.66  |

5

The performance evaluation measures used in this invention are provided here as follow: in class imbalance learning scenario, the accuracy measure is not opt way to evaluate the performance of an algorithm. Since the accuracy provides the percentage of correctly classified instances from both the majority and minority subsets in a combined way. In the case of class imbalance scenario we need to find the exact percentage of instances classified correctly in individual classes that is subsets.

This can be done by calculating the values such as True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) respectively. The detail of these measures can be given as follows: True Positive (TP): The instance is a positive instance and is classified as a positive instance. True Negative (TN): The instance is a negative instance and is classified as a negative instance. False Positive (FP): The instance is a negative instance and is classified as a positive instance. False Negative (FN): The instance is a positive instance and is classified as a negative instance. This above basic units of performance can be used to build the evaluation measures such as AUC, Precision, Recall and F-measure

**Experimental Results:** As the part of the experimental results obtained by the invention, the empirical comparisons are carried out for the proposed algorithm with the benchmarks. The results comparison of the proposed approach is one of the important parts of the manuscript, which is presented for highlighting the strengths and weakness of the proposed framework. The mean and standard deviation results are summarized for all the experimental conducted.

The results of AUC are presented in Table 2, the proposed algorithm results are better for all 16 software defect datasets. The C4.5 algorithm has also not performed well than the proposed IISS algorithm in terms of AUC. IISS verses REP has won on all 16 software defect prediction datasets in terms of AUC. The results of AUC metric for IISS when compared with CART are also improved on all the datasets.

**TABLE 2**

**Summary of the defect prediction datasets used in the invention**

| Data set | C4.5          | REP           | CART          | IISS               |
|----------|---------------|---------------|---------------|--------------------|
| AR1      | 0.584 ±0.250● | 0.503 ±0.045● | 0.498 ±0.015● | 0.936±0.130        |
| AR3      | 0.738 ±0.259● | 0.570 ±0.191● | 0.604 ±0.218● | 0.893±0.171        |
| AR5      | 0.747 ±0.266● | 0.610 ±0.216● | 0.729 ±0.267● | 0.926±0.156        |
| DATA     | 0.481 ±0.068● | 0.509 ±0.075● | 0.500 ±0.048● | 0.772±0.095        |
| DESH     | 0.685 ±0.184● | 0.694 ±0.177● | 0.733 ±0.159● | 0.844±0.132        |
| KC1      | 0.696 ±0.067● | 0.749 ±0.052● | 0.683 ±0.064● | 0.888±0.021        |
| KC1DEF   | 0.704 ±0.126● | 0.751 ±0.119● | 0.704 ±0.110● | 0.860±0.092        |
| KC1TOP   | 0.654 ±0.230● | 0.557 ±0.152● | 0.500 ±0.000● | 0.921±0.146        |
| KC2      | 0.693±0.104●  | 0.767±0.087●  | 0.772±0.070●  | 0.871±0.046        |
| KC3      | 0.604 ±0.175● | 0.533 ±0.113● | 0.540 ±0.117● | 0.819±0.093        |
| MC1      | 0.582 ±0.133● | 0.585 ±0.135● | 0.503 ±0.023● | 0.918±0.057        |
| MC21     | 0.596 ±0.166● | 0.583 ±0.119● | 0.604 ±0.129● | 0.821±0.105        |
| MW1      | 0.461 ±0.219● | 0.586 ±0.149● | 0.512 ±0.082● | <b>0.865±0.080</b> |
| PC1      | 0.675 ±0.141● | 0.663 ±0.148● | 0.515 ±0.062● | 0.922±0.038        |
| PC3      | 0.626 ±0.110● | 0.657 ±0.133● | 0.502 ±0.019● | 0.874±0.039        |
| REUSE    | 0.939 ±0.165● | 0.500 ±0.000● | 0.500 ±0.000● | <b>0.995±0.050</b> |

5

In above Table:

●Bold dot indicate the win of IISS      ○Empty dot indicate the loss of IISS

The Results of IISS model on SDP datasets in terms of Precision is shown in the Table 3.

10 **TABLE 3**

**Results of IISS model on SDP datasets in terms of Precision**

| Data set | C4.5           | REP           | CART          | IISS        |
|----------|----------------|---------------|---------------|-------------|
| AR1      | 0.932 ±0.036●  | 0.925 ±0.025● | 0.925 ±0.025● | 0.954±0.054 |
| AR3      | 0.938 ±0.082○  | 0.894 ±0.078● | 0.902 ±0.081● | 0.917±0.116 |
| AR5      | 0.881 ±0.200 ○ | 0.829 ±0.167● | 0.884 ±0.174○ | 0.876±0.210 |

|        |               |               |               |             |
|--------|---------------|---------------|---------------|-------------|
| DATA   | 0.914 ±0.024○ | 0.915 ±0.024○ | 0.916 ±0.025○ | 0.838±0.027 |
| DESH   | 0.751 ±0.165● | 0.741 ±0.152● | 0.752 ±0.142● | 0.788±0.166 |
| KC1    | 0.880 ±0.013○ | 0.869 ±0.011○ | 0.864 ±0.011  | 0.864±0.016 |
| KC1DEF | 0.628 ±0.153● | 0.611 ±0.163● | 0.589 ±0.139● | 0.801±0.082 |
| KC1TOP | 0.049 ±0.144● | 0.045 ±0.160● | 0.000 ±0.000● | 0.635±0.402 |
| KC2    | 0.870±0.041○  | 0.861±0.038○  | 0.891±0.039○  | 0.842±0.040 |
| KC3    | 0.374 ±0.298● | 0.165 ±0.305● | 0.184 ±0.324● | 0.706±0.148 |
| MC1    | 0.158 ±0.287● | 0.066 ±0.207● | 0.000 ±0.000● | 0.780±0.172 |
| MC21   | 0.503 ±0.271● | 0.484 ±0.398● | 0.567 ±0.352● | 0.752±0.117 |
| MW1    | 0.249 ±0.370● | 0.140 ±0.305● | 0.021 ±0.082● | 0.685±0.188 |
| PC1    | 0.359 ±0.246● | 0.216 ±0.317● | 0.077 ±0.218● | 0.704±0.125 |
| PC3    | 0.384 ±0.170● | 0.176 ±0.293● | 0.000 ±0.000● | 0.626±0.079 |
| REUSE  | 0.953 ±0.136○ | 0.617 ±0.151● | 0.617 ±0.151● | 0.935±0.183 |

In above Table:

●**Bold dot indicate the win of IISS**

○**Empty dot indicate the loss of IISS**

5 Thus, compared to C4.5, REP and CART algorithms, the IISS algorithm lays more stress on identifying and improving the minority and majority subsets. In this situation, our proposed approach IISS have gained significant improvement in terms of precision (Table 3). The efficiency of the IISS can be shown in the form of potential learning technique as it has performed well in the performance evaluation of recall measure (Table 4). The results of IISS  
10 algorithms are efficient when compared with C4.5; out of the all 16 software defect prediction datasets, our proposed IISS algorithm have won on 10 datasets and losses on 6 datasets. When compared with REP Tree our proposed IISS algorithm have achieved 12 wins, 1 tie and 3 losses out of 16 software defect prediction datasets. The trends of AUC are shown in the Figure 7 for C4.5, REP, CART and IISS on software defect prediction datasets  
15 respectively.

Answer for RQ1: The RQ1 can be justified as follows that there is direct effect of class imbalance nature on the classification process. The claim can be observed in the results of the experimental simulation where there is less class imbalance nature the variation of performance is less and for high level of class imbalance the performance is high, directly  
20 proportional to the class imbalance nature of the data source.

Answer for RQ2: There is high level of class imbalance nature in the software defect prediction datasets as it is also clearly indicate that any real world data sources will be in class imbalance nature. The need of the proposed study on class imbalance software defect data source is necessary for proper extraction of the knowledge.

5

The Results of IISS model on SDP datasets in terms of Recall is shown in the Table 4.

**TABLE 4**

**Results of IISS model on SDP datasets in terms of Recall**

10

| Data set | C4.5          | REP           | CART          | IISS        |
|----------|---------------|---------------|---------------|-------------|
| AR1      | 0.964 ±0.057● | 0.997 ±0.020○ | 0.995 ±0.034○ | 0.985±0.044 |
| AR3      | 0.944 ±0.087● | 0.973 ±0.078○ | 0.962 ±0.084○ | 0.925±0.133 |
| AR5      | 0.867 ±0.226○ | 0.913 ±0.192○ | 0.903 ±0.203○ | 0.847±0.249 |
| DATA     | 0.984 ±0.039● | 0.992 ±0.034● | 0.990 ±0.032● | 1.000±0.000 |
| DESH     | 0.799 ±0.198○ | 0.817 ±0.182○ | 0.855 ±0.172○ | 0.778±0.195 |
| KC1      | 0.940 ±0.021○ | 0.964 ±0.022○ | 0.970 ±0.018○ | 0.932±0.019 |
| KC1DEF   | 0.602 ±0.174● | 0.745 ±0.255● | 0.745 ±0.227● | 0.910±0.083 |
| KC1TOP   | 0.110 ±0.314● | 0.080 ±0.273● | 0.000 ±0.000● | 0.635±0.401 |
| KC2      | 0.901±0.057○  | 0.939±0.045○  | 0.913±0.054○  | 0.881±0.054 |
| KC3      | 0.308 ±0.243● | 0.115 ±0.216● | 0.125 ±0.218● | 0.689±0.163 |
| MC1      | 0.077 ±0.135● | 0.027 ±0.078● | 0.000 ±0.000● | 0.467±0.174 |
| MC21     | 0.421 ±0.244● | 0.263 ±0.242● | 0.332 ±0.242● | 0.809±0.129 |
| MW1      | 0.170 ±0.239● | 0.103 ±0.218● | 0.027 ±0.100● | 0.609±0.194 |
| PC1      | 0.273 ±0.196● | 0.099 ±0.150● | 0.030 ±0.082● | 0.649±0.140 |
| PC3      | 0.284 ±0.138● | 0.049 ±0.082● | 0.000 ±0.000● | 0.634±0.103 |
| REUSE    | 0.995 ±0.050○ | 1.000 ±0.000○ | 1.000 ±0.000○ | 0.990±0.100 |

In above Table:

●**Bold dot indicate the win of IISS**

○**Empty dot indicate the loss of IISS**

15

The Results of IISS model on SDP datasets in terms of F-measure is shown in the Table 5.

**TABLE 5**

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

5

| Data set | C4.5          | REP           | CART          | IISS        |
|----------|---------------|---------------|---------------|-------------|
| AR1      | 0.947 ±0.039● | 0.960 ±0.017● | 0.958 ±0.024● | 0.968±0.036 |
| AR3      | 0.938 ±0.067○ | 0.929 ±0.058○ | 0.927 ±0.062○ | 0.913±0.099 |
| AR5      | 0.861 ±0.195○ | 0.851 ±0.147○ | 0.879 ±0.169○ | 0.835±0.198 |
| DATA     | 0.947 ±0.025○ | 0.952 ±0.023○ | 0.951 ±0.019○ | 0.912±0.016 |
| DESH     | 0.759 ±0.152● | 0.765 ±0.136  | 0.790 ±0.129○ | 0.765±0.145 |
| KC1      | 0.909 ±0.011○ | 0.914 ±0.010○ | 0.914 ±0.007○ | 0.896±0.012 |
| KC1DEF   | 0.602 ±0.134● | 0.651 ±0.173● | 0.646 ±0.155● | 0.849±0.064 |
| KC1TOP   | 0.067 ±0.195● | 0.057 ±0.196● | 0.000 ±0.000● | 0.607±0.369 |
| KC2      | 0.883±0.029○  | 0.897±0.024○  | 0.900±0.027○  | 0.860±0.033 |
| KC3      | 0.322 ±0.236● | 0.125 ±0.220● | 0.140 ±0.237● | 0.685±0.125 |
| MC1      | 0.096 ±0.161● | 0.037 ±0.109● | 0.000 ±0.000● | 0.563±0.156 |
| MC21     | 0.433 ±0.213● | 0.315 ±0.257● | 0.388 ±0.242● | 0.772±0.095 |
| MW1      | 0.188 ±0.260● | 0.107 ±0.216● | 0.023 ±0.086● | 0.623±0.150 |
| PC1      | 0.293 ±0.190● | 0.127 ±0.179● | 0.041 ±0.108● | 0.667±0.108 |
| PC3      | 0.317 ±0.136● | 0.069 ±0.108○ | 0.000 ±0.000● | 0.626±0.077 |
| REUSE    | 0.968 ±0.096○ | 0.753 ±0.104● | 0.753 ±0.104● | 0.953±0.142 |

In above Table:

●**Bold dot indicate the win of IISS**

○**Empty dot indicate the loss of IISS**

10 Answer for RQ3: The prediction accuracy of the modules with high probability of software defects can be easily identified for class balance data sources. Whereas the data sources with high level of class imbalance nature, are not well identified in the case of traditional approaches. The proposed approach has mitigated the problem by performing efficient level of sampling in both the subset of classes.

15 Answer for RQ4: The effect of base classifier can be clearly noticed for different approaches which use their unique ways for model building. The less percentage of instances in the minority subsets raises the problem of improper model building leading to inefficient

performance, exclusively for minority subset.

Answer for RQ5: The scalability of the software prediction datasets is a virtue for minority subsets where there is scarcity of instances. The larger the data sources the more instance are available for better performance of the models which use training subset of instances for performing testing.

Referring to Figure 3,4,5,6, illustrates the AR1 Dataset, KC1 Dataset, MC1 Dataset, and PC1 Dataset respectively. These visualization results of AR1, KC1, MC1 and PC1 datasets generated using the IISS algorithm. The results show the improvement gained by the IISS algorithm on the software defect prediction datasets. The following Table 6, summaries the results of wins, ties and losses of compared algorithms on IISS on all the datasets. The first row and first column of the Table 6, presents the wins, ties and losses and evaluation measures of AUC, Precision, Recall and F-measure.

**TABLE 6**

**15 Summary of experimental results for IISS**

| <b>Results</b>   | <b>Systems</b> | <b>Wins</b> | <b>Ties</b> | <b>Losses</b> |
|------------------|----------------|-------------|-------------|---------------|
| <b>AUC</b>       | IISS v/s C4.5  | 16          | 0           | 0             |
|                  | IISS v/s REP   | 16          | 0           | 0             |
|                  | IISS v/s CART  | 16          | 0           | 0             |
| <b>Precision</b> | IISS v/s C4.5  | 10          | 0           | 6             |
|                  | IISS v/s REP   | 13          | 0           | 3             |
|                  | IISS v/s CART  | 12          | 1           | 3             |
| <b>Recall</b>    | IISS v/s C4.5  | 11          | 0           | 5             |
|                  | IISS v/s REP   | 9           | 0           | 7             |
|                  | IISS v/s CART  | 9           | 0           | 7             |
| <b>F-measure</b> | IISS v/s C4.5  | 10          | 0           | 6             |
|                  | IISS v/s REP   | 9           | 1           | 6             |
|                  | IISS v/s CART  | 10          | 0           | 6             |

Referring to Figure 7, illustrates the trends in AUC for C4.5, REP, CART versus IISS on SDP Datasets. The IISS algorithm has performed well on all the measures. However, IISS is better in the aspect of class imbalance measures, which is the problem in hand for real world datasets. Finally, IISS have generated favorable results in terms of class imbalance measures



for software defect prediction. The total experimental simulation conducted on 16 class imbalance software defect datasets project that prominent recursive oversampling and intelligent under sampling approaches can improve the effectiveness when dealing with imbalanced data, as it has helped the IISS method to be the best performing algorithms when compared with benchmark algorithms.

Meka James  
Stephen

Digitally signed by Meka  
James Stephen  
Date: 2022.02.08 17:12:55  
+05'30'

## CLAIMS

### **We claim:**


1. A Novel Improved Integrated Sampling Strategy (IISS) for Software Defect Prediction comprising of three phases: Processing Phase (201); Adaptation Phase (202); and Building Predictive Model Phase (203) used to predict the software defects on Skewed Data Distribution.
2. A Novel Improved Integrated Sampling Strategy (IISS) for Software Defect Prediction as claimed in claim 1, wherein it uses noisy removal strategy by integrating both over sampling and under sampling for software defect prediction.
3. A Novel Improved Integrated Sampling Strategy (IISS) for Software Defect Prediction as claimed in claim 1, wherein the class imbalance software defect dataset which contains both majority and minority sub classes is divided into two separate classes in the processing phase,
4. A Novel Improved Integrated Sampling Strategy (IISS) for Software Defect Prediction as claimed in claim 1, wherein the minority subset is further analyzed to form any existing outliers and noisy instances. The removal of noisy and outlier instances will improve the quality of the dataset and reduce the risk of magnifying the influence of outliers and noisy instances in the over sampling stage in adaptation phase.
5. A Novel Improved Integrated Sampling Strategy (IISS) for Software Defect Prediction as claimed in claim 1, wherein the improved majority and minority subsets are combined to form an almost balanced software defect prediction dataset and is applied to a base algorithm. In this case, random forest is selected as the base algorithm and evaluates on different evaluation metrics in building predictive Model phase.
6. A Novel Improved Integrated Sampling Strategy (IISS) for Software Defect Prediction as claimed in claim 1, wherein there is direct effect of class imbalance nature on the classification process.
7. A Novel Improved Integrated Sampling Strategy (IISS) for Software Defect

Prediction as claimed in claim 1, wherein there is high level of class imbalance nature in the software defect prediction datasets as it is also clearly indicate that any real world data sources will be in class imbalance nature.

8. A Novel Improved Integrated Sampling Strategy (IISS) for Software Defect Prediction as claimed in claim 1, wherein the prediction accuracy of the modules with high probability of software defects can be easily identified for class balance data sources. Whereas the data sources with high level of class imbalance nature, are not well identified in the case of traditional approaches. The proposed approach has mitigated the problem by performing efficient level of sampling in both the subset of classes.
9. A Novel Improved Integrated Sampling Strategy (IISS) for Software Defect Prediction as claimed in claim 1, wherein the less percentage of instances in the minority subsets raises the problem of improper model building leading to inefficient performance, exclusively for minority subset. The scalability of the software prediction datasets is a virtue for minority subsets where there is scarcity of instances. The larger the data sources the more instance are available for better performance of the models which use training subset of instances for performing testing.
10. A Novel Improved Integrated Sampling Strategy (IISS) for Software Defect Prediction as claimed in claim 1, wherein it is conducted on skewed software defect prediction datasets by the proposed IISS and its performance is compared with C4.5, C4.5 with Balance dataset, RF (Random Forest) and RF with Balance dataset algorithms with various class imbalance evaluation measures.

Dated this 08<sup>th</sup> day of February, 2022

Meka James  
Stephen

 Digitally signed by Meka James  
Stephen  
Date: 2022.02.08 17:12:38 +05'30'

# **A NOVEL IMPROVED INTEGRATED SAMPLING STRATEGY FOR SOFTWARE DEFECT PREDICTION**

## **ABSTRACT**

Software Defect Prediction using data mining techniques is one of the best practices for finding defective modules. On normal datasets, existing classification techniques can be applied for effective knowledge discovery. Most of the real world data sources are biased towards any one of the class and 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. There is a need for the invention that can increase the software defect prediction rate. The present invention disclosed here is a Novel Improved Integrated Sampling Strategy (IISS) for Software Defect Prediction comprising of three phases: Processing Phase (201); Adaptation Phase (202); and Building Predictive Model Phase (203). The present invention disclosed herein predicts the software defects on Skewed Data Distribution. The invention of this disclosure uses noisy removal strategy by integrating both over sampling and under sampling for software defect prediction. The experimental analysis of the present invention disclosed herein is conducted on skewed software defect prediction datasets by the proposed IISS and its performance is compared with C4.5, C4.5 with Balance dataset, RF (Random Forest) and RF with Balance dataset algorithms with various class imbalance evaluation measures.

Dated this 08<sup>th</sup> day of February, 2022

Meka James Stephen

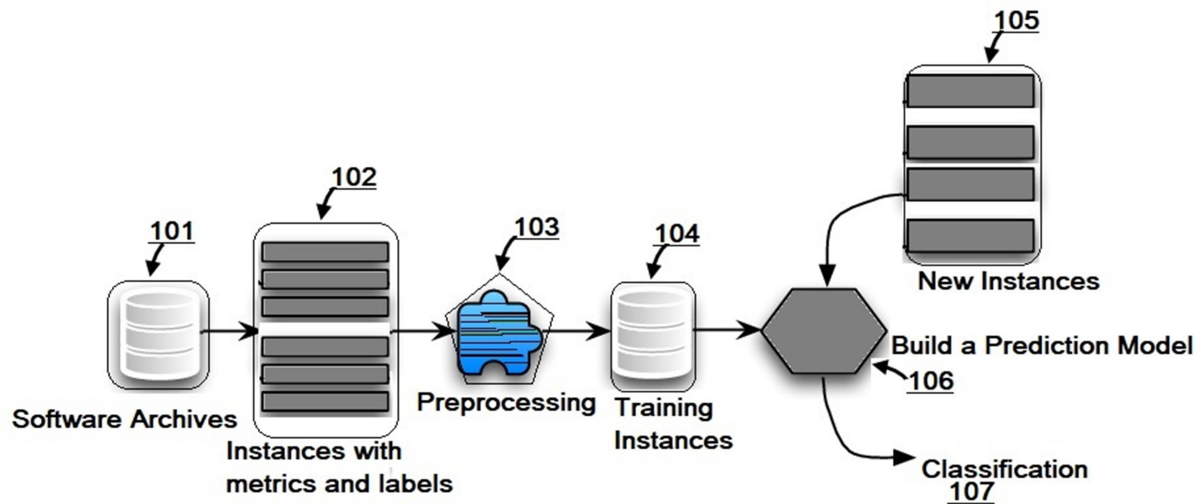
Digitally signed by Meka James  
Stephen  
Date: 2022.02.08 17:12:18 +05'30'

# DRAWINGS

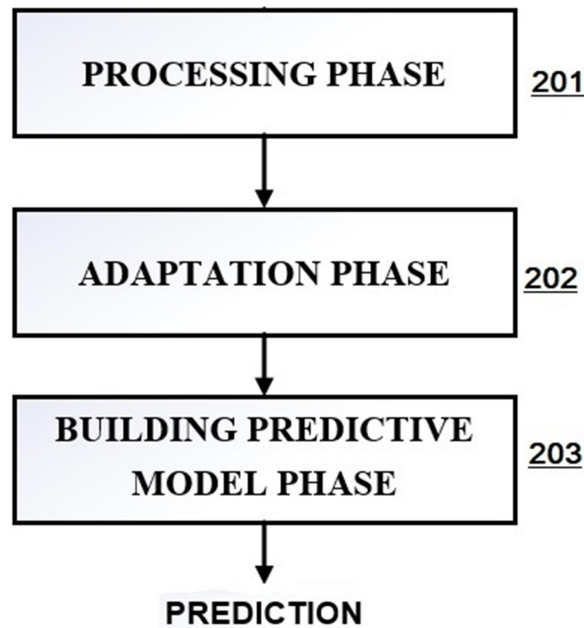
Total No of Sheets: 04

Sheet No.1

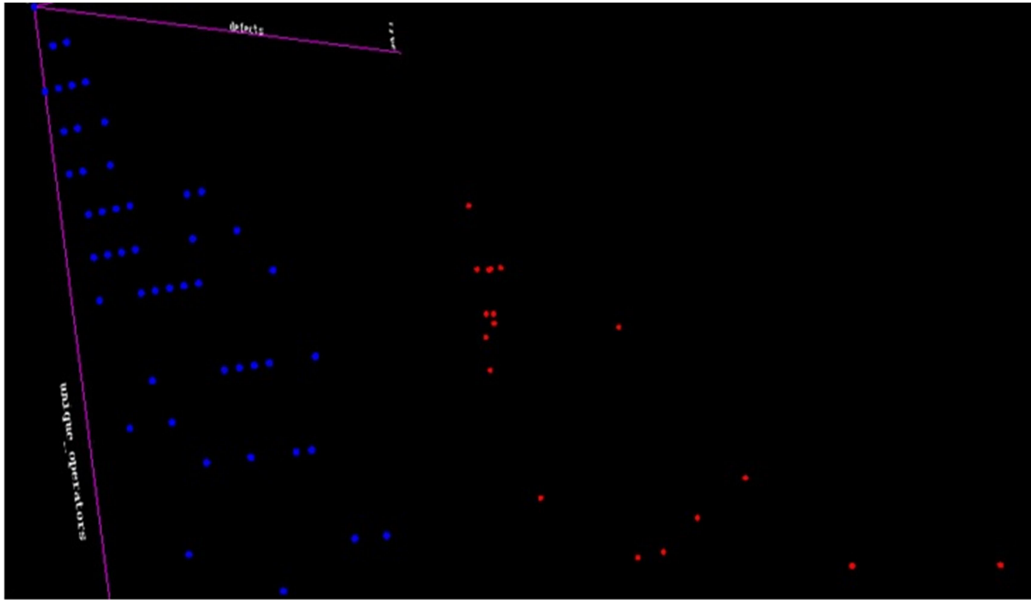
**Applicants:** Prof.M.James Stephen, Mr.K.Nitalaksheswara Rao, Prof. P.V.G.D. Prasad Reddy, Mr.Ch.V.Murali Krishna.



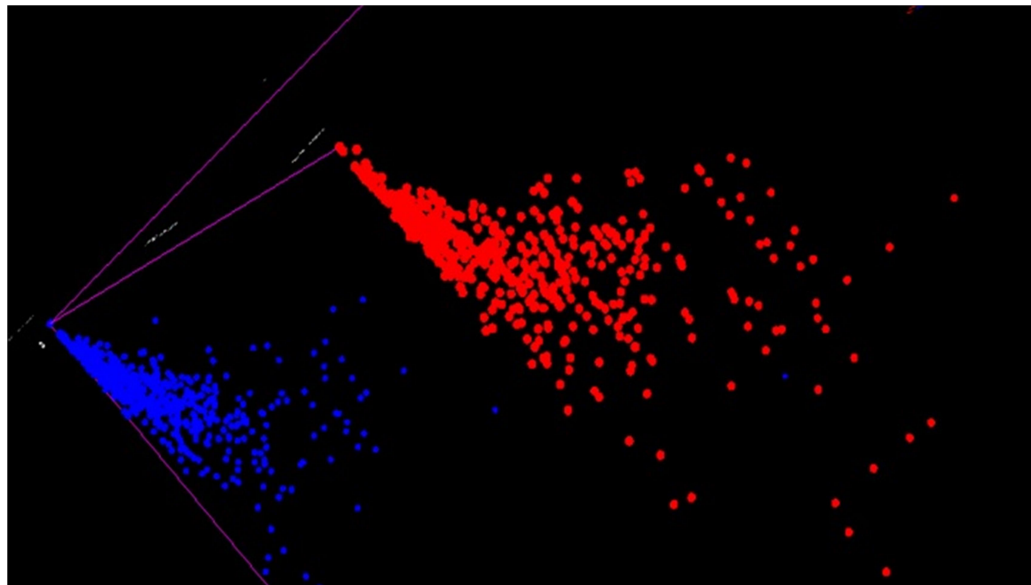
**FIGURE 1:** General Software Defect Prediction Process.



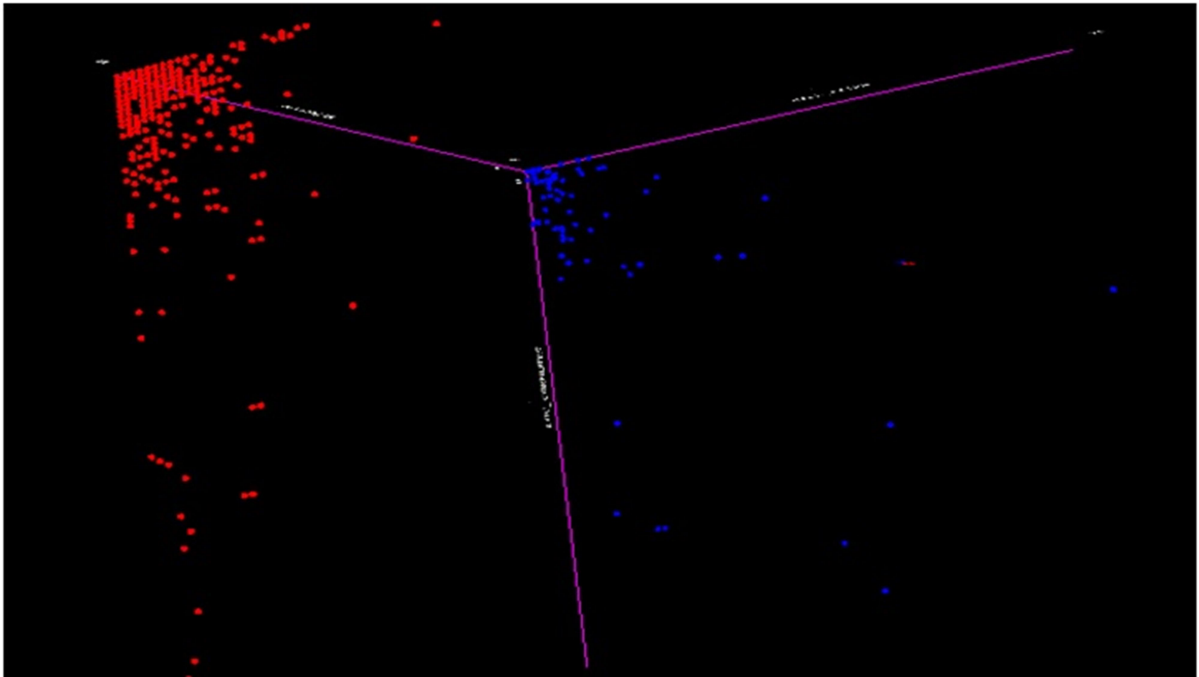
**FIGURE 2:** Phases in Improved Integrated Sampling Strategy.



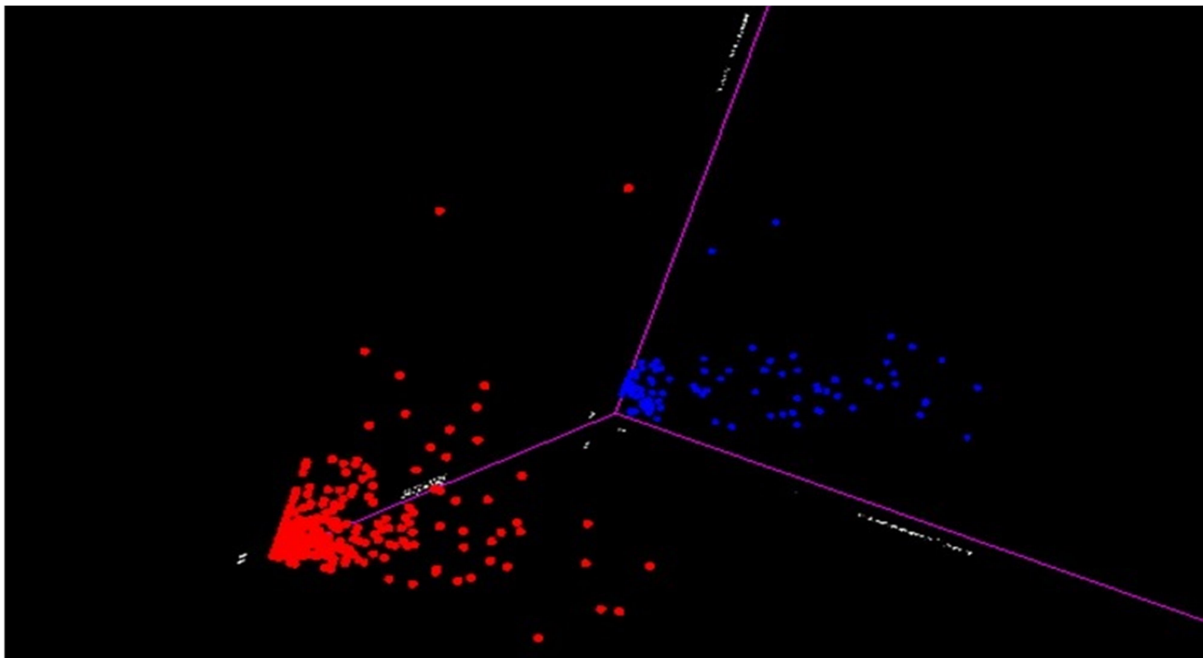
**FIGURE 3:** AR1 Dataset.



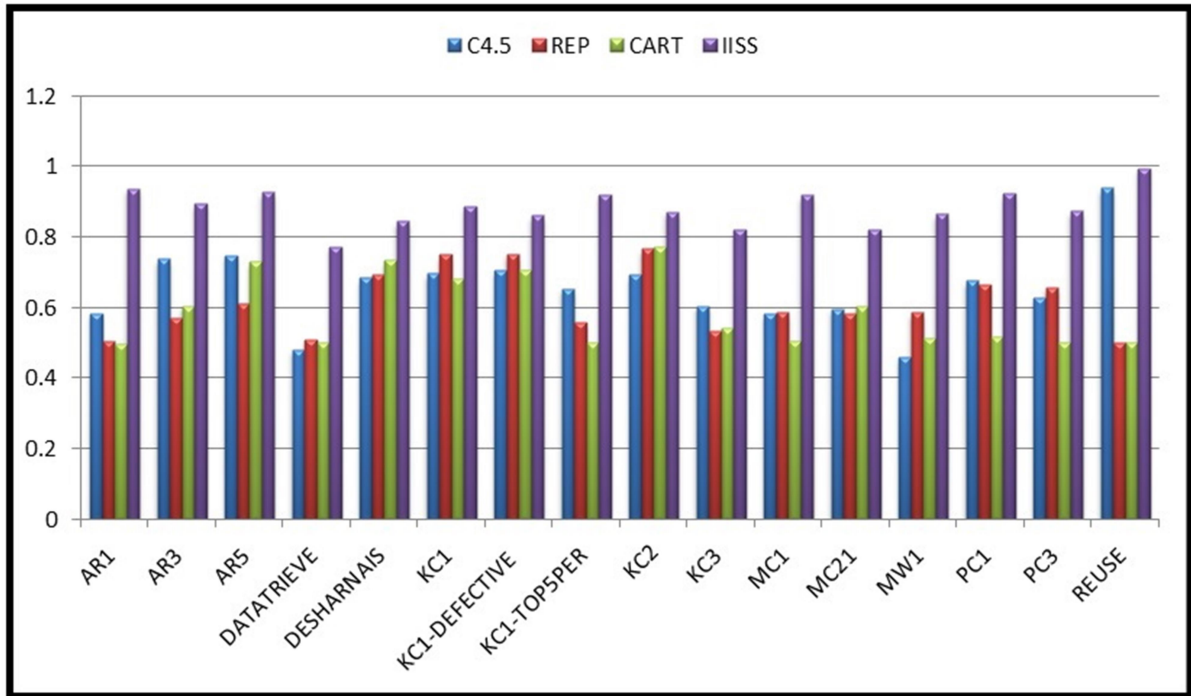
**FIGURE 4:** KC1 Dataset:



**FIGURE 5:** MC1 Dataset.



**FIGURE 6:** PC1 Dataset.



**FIGURE 7:** Trends in AUC for C4.5, REP, CART versus IISS on SDP Datasets.

Meka James Stephen  Digitally signed by Meka James Stephen  
 Date: 2022.02.08 17:14:28 +05'30'



# FORM 9

THE PATENT ACT, 1970

(39 of 1970)

&

THE PATENTS RULES, 2003

## REQUEST FOR PUBLICATION

[See section 11A (2); rule 24A]

I/We **Prof.M.James Stephen, Mr.K.Nitalaksheswara Rao, Prof. P.V.G.D. Prasad Reddy, Mr.Ch.V.Murali Krishna.**, hereby request for early publication of my/our application for patent, titled **“A Novel Improved Integrated Sampling Strategy for Software Defect Prediction”** dated 08-02-2022, under section 11A(2) of the act.

Dated this 08<sup>th</sup> day of February, 2022 **18:00:00** under section 11A (2) of the Act.

### 1. Name, Nationality and address of Applicants:

| Sr.No | Name                        | Nationality | Address   | Country | State          |
|-------|-----------------------------|-------------|---|---------|----------------|
| 1     | Prof.M.James Stephen        | Indian      | Professor, Department of CSE, Welfare Institute of Science Technology and Management (WISTM), Visakhapatnam, Andhra Pradesh, India. Pin Code:530007 | India   | Andhra Pradesh |
| 2     | Mr.K.Nitalaksheswara Rao    | Indian      | Research Scholar, Department of CS & SE, A.U. College of Engineering (A), Andhra University, Visakhapatnam, Andhra Pradesh, India. Pin Code:530003  | India   | Andhra Pradesh |
| 3     | Prof. P.V.G.D. Prasad Reddy | Indian      | Senior Professor, Department of CS & SE, A.U. College of Engineering (A), Andhra University, Visakhapatnam, Andhra Pradesh, India. Pin Code:530003  | India   | Andhra Pradesh |
| 4     | Mr.Ch.V.Murali Krishna      | Indian      | Associate Professor, Department of CSE NRI Institute of Technology, Agiripalli, Krishna District, Andhra Pradesh, India. Pin Code: 521212           | India   | Andhra Pradesh |

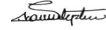
### 2. To be signed by the applicant or authorized registered patent

Dated this 08<sup>th</sup> day of February, 2022

**3. Name of Applicant(s)/ Inventor(s) Signature(s):**

Name of the natural person who has signed.      Signature:-

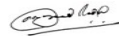
Prof.M.James Stephen



Mr.K.Nitalaksheswara Rao



Prof. P.V.G.D. Prasad Reddy



Mr.Ch.V.Murali Krishna



To

The Controller of Patents

**The Patent office at CHENNAI**

## FORM 3

THE PATENTS ACT, 1970  
(39 of 1970)  
and  
THE PATENTS RULES, 2003

### STATEMENT AND UNDERTAKING UNDER SECTION 8

(See section 8; Rule 12)

#### 1. Name of Applicant(s):

Prof.M.James Stephen, Mr.K.Nitalaksheswara Rao, Prof. P.V.G.D. Prasad Reddy, Mr.Ch.V.Murali Krishna

#### 2. Name, Address and Nationality of the Applicant(s):

| Sr.No | Name                        | Nationality | Address   | Country | State          |
|-------|-----------------------------|-------------|---|---------|----------------|
| 1     | Prof.M.James Stephen        | Indian      | Professor, Department of CSE, Welfare Institute of Science Technology and Management (WISTM), Visakhapatnam, Andhra Pradesh, India. Pin Code:530007 | India   | Andhra Pradesh |
| 2     | Mr.K.Nitalaksheswara Rao    | Indian      | Research Scholar, Department of CS & SE, A.U. College of Engineering (A), Andhra University, Visakhapatnam, Andhra Pradesh, India. Pin Code:530003  | India   | Andhra Pradesh |
| 3     | Prof. P.V.G.D. Prasad Reddy | Indian      | Senior Professor, Department of CS & SE, A.U. College of Engineering (A), Andhra University, Visakhapatnam, Andhra Pradesh, India. Pin Code:530003  | India   | Andhra Pradesh |
| 4     | Mr.Ch.V.Murali Krishna      | Indian      | Associate Professor, Department of CSE NRI Institute of Technology, Agiripalli, Krishna District, Andhra Pradesh, India. Pin Code: 521212           | India   | Andhra Pradesh |

I/We, Prof.M.James Stephen, Mr.K.Nitalaksheswara Rao, Prof. P.V.G.D. Prasad Reddy, Mr.Ch.V.Murali Krishna., is/are the true & first inventor(s) for this invention and declare that the applicant(s) herein is/are my/our assignee or legal representative.

(i) that I/We have not made any application for the same/substantially the same invention outside India.

OR





~~(ii) that I/We who have made this application No.....dated .....alone/jointly with....., made for the same/substantially same invention, application(s) for patent in the other countries, the particulars of which are given below:~~

| Name of the country | Date of application | Application No. | Status of the application | Date of publication | Date of grant |
|---------------------|---------------------|-----------------|---------------------------|---------------------|---------------|
|---------------------|---------------------|-----------------|---------------------------|---------------------|---------------|



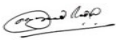

~~3. Name and address of the assignee~~

(iii) that the rights in the application(s) has/have been assigned to..... that I/We undertake that upto the date of grant of the patent by the Controller, I/We would keep him informed in writing the details regarding corresponding applications for patents filed outside India within six months from the date of filing of such application.  
Dated this: 08<sup>th</sup> Day of February, 2022

~~4. To be signed by the applicant or his authorized registered patent agent.~~

Signature:.....  
Prof.M.James Stephen   
Mr.K.Nitalaksheswara Rao   
Prof. P.V.G.D. Prasad Reddy   
Mr.Ch.V.Murali Krishna 

~~5. Name of the natural person who has Signed.~~

Prof.M.James Stephen   
Mr.K.Nitalaksheswara Rao   
Prof. P.V.G.D. Prasad Reddy   
Mr.Ch.V.Murali Krishna 

To  
The Controller of Patents  
The Patent office at CHENNAI

# FORM 5

THE PATENT ACT, 1970

(39 OF 1970) &

The Patent Rules, 2003

## DECLARATION AS TO INVENTORSHIP

[See sections 10(6) and Rule 13(6)]

### 1. NAME OF APPLICANT(S):

| Sr.No | Name                        | Nationality | Address  | Country | State          |
|-------|-----------------------------|-------------|--|---------|----------------|
| 1     | Prof.M.James Stephen        | Indian      | Professor, Department of CSE, Wellfare Institute of Science Technology and Management (WISTM), Visakhapatnam, Andhra Pradesh, India. Pin Code:530007 | India   | Andhra Pradesh |
| 2     | Mr.K.Nitalaksheswara Rao    | Indian      | Research Scholar, Department of CS & SE, A.U. College of Engineering (A), Andhra University, Visakhapatnam, Andhra Pradesh, India. Pin Code:530003   | India   | Andhra Pradesh |
| 3     | Prof. P.V.G.D. Prasad Reddy | Indian      | Senior Professor, Department of CS & SE, A.U. College of Engineering (A), Andhra University, Visakhapatnam, Andhra Pradesh, India. Pin Code:530003   | India   | Andhra Pradesh |
| 4     | Mr.Ch.V.Murali Krishna      | Indian      | Associate Professor, Department of CSE NRI Institute of Technology, Agiripalli, Krishna District, Andhra Pradesh, India. Pin Code: 521212            | India   | Andhra Pradesh |

I/We Prof.M.James Stephen, Mr.K.Nitalaksheswara Rao, Prof. P.V.G.D. Prasad Reddy, Mr.Ch.V.Murali Krishna., hereby declare that the true and first inventor(s) of the invention disclosed in the complete specification filed in pursuance of my/our application numbered.....dated is/are:

**2. INVENTOR(s):**

| Sr.No | Name                        | Nationality | Address   | Country | State          |
|-------|-----------------------------|-------------|---|---------|----------------|
| 1     | Prof.M.James Stephen        | Indian      | Professor, Department of CSE, Welfare Institute of Science Technology and Management (WISTM), Visakhapatnam, Andhra Pradesh, India. Pin Code:530007 | India   | Andhra Pradesh |
| 2     | Mr.K.Nitalaksheswara Rao    | Indian      | Research Scholar, Department of CS & SE, A.U. College of Engineering (A), Andhra University, Visakhapatnam, Andhra Pradesh, India. Pin Code:530003  | India   | Andhra Pradesh |
| 3     | Prof. P.V.G.D. Prasad Reddy | Indian      | Senior Professor, Department of CS & SE, A.U. College of Engineering (A), Andhra University, Visakhapatnam, Andhra Pradesh, India. Pin Code:530003  | India   | Andhra Pradesh |
| 4     | Mr.Ch.V.Murali Krishna      | Indian      | Associate Professor, Department of CSE NRI Institute of Technology, Agiripalli, Krishna District, Andhra Pradesh, India. Pin Code: 521212           | India   | Andhra Pradesh |

Dated this...08<sup>th</sup> day of February, 2022

Name of the Signatory

Signature:-

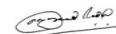
Prof.M.James Stephen



Mr.K.Nitalaksheswara Rao



Prof. P.V.G.D. Prasad Reddy



Mr.Ch.V.Murali Krishna

**3. DECLARATION TO BE GIVEN WHEN THE APPLICATION IN INDIA IS FILED BY THE APPLICANT (S) IN THE CONVENTION COUNTRY:-**

We the applicant(s) in the convention country hereby declare that our right to apply for a patent in India is by way of assignment from the true and first inventor(s).

Dated this .....day of 2020

Signature:-

Name of Signatory:-

**4. STATEMENT (to be signed by the additional inventor(s) not mentioned in the application form)**

~~I/we assent to invention referred to in the above declaration, being included in the complete specification filed in pursuance of the stated application.~~

~~Dated this .....day of 2020~~

~~Signature of the additional inventor (s)~~

~~Name :~~

To

The Controller of Patents

**The Patent office at CHENNAI**