**A Pre-processing Method to Identify and Correct Mislabelled Data**

Project report submitted for

**VIth Semester Minor Project-II**

**in**

**Department of Computer Science and Engineering**

By,

**Balram Rathore (16100018)**

**Priyanka Mall (16101041)**

**Sakshita Jayaswal (16101045)**

**

**Department of Computer Science and Engineering**

**Dr. Shyama Prasad Mukherjee**

**International Institute of Information Technology, Naya Raipur**

**(A Joint Initiative of Govt. of Chhattisgarh and NTPC)**

**Email:** [**iiitnr@iiitnr.ac.in**](mailto:iiitnr@iiitnr.ac.in)**, Tel: (0771) 2474040, Web:** [**www.iiitnr.ac.in**](http://www.iiitnr.ac.in)

**CERTIFICATE**

This is to certify that the project titled “**A Pre-processing Method To Identify And Correct Mislabelled Data**” by “**Balram Rathore, Priyanka Mall, Sakshita Jaiswal**” has been carried out under my/our supervision and that this work has not been submitted elsewhere for a degree/diploma.

(Signature of Guide)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Guide Name**

**Designation of Guide**

**Department of Computer Science and Engineering**

**Dr. SPM IIIT-NR**

**May, 2019**

**DECLARATION**

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature of Author)

**Balram Rathore**

**Author Name**

**(16100018)**

(Signature of Author)

**Priyanka Mall**

**Author Name**

**(16101041)**

(Signature of Author)

**Sakshita Jaiswal**

**Author Name**

**(16101045)**

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Plagiarism Report**

**Approval Sheet**

This project report entitled “**A Pre-processing Method Using Adaboost To Identify and Correct Mislabelled Data**” by “**Balram Rathore, Priyanka Mall, Sakshita Jaiswal**” is approved for VIth Semester Minor Project-II.

(Signature of Examiner - I)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name of Examiner -I

(Signature of Examiner - II)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name of Examiner -II

(Signature of Chair)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name of Chair

Date: \_\_\_\_\_\_\_\_\_\_\_\_ Place: \_\_\_\_\_\_\_\_\_\_\_\_

**Table of Contents**

**Title Page No.**

**TABLE OF CONTENTS…………………………………………………….. 05**

**LIST OF TABLES……………………………………………………………. 06**

**LIST OF FIGURES…………………………………………………………... 07**

**ABSTRACT…………………………………………………………………… 08**

**CHAPTER 1 INTRODUCTION……………………………………………. 08**

**CHAPTER 2 LITERATURE REVIEW………………………………...….. 09**

**CHAPTER 3 METHEDOLOGIES …………………………………………. 10**

* 1. Problem Statement…………………………………………………… 10
  2. d/m method (Existing Method)………………………………………. 10
  3. Mode Method………………………………………………………… 10
  4. Mean Method……………………………………………………….... 12
  5. Cluster based Boosting………………………………………………. 14
  6. Clustering as a Feature………………………………………………. 14
  7. Clustering as a Feature with Semi-Supervised Learning……………. 16
  8. Double Addition of Cluster in Semi-Supervised Learning………...... 18

**CHAPTER 4 RESULTS 20**

**CHAPTER 5 CONCLUSIONS 21**

**CHAPTER 6 ACKNOWLEDGEMENT 21**

**REFERENCES 21**

**List of All Tables**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Table Title** | **Page Number** |
| 1 | Table  Title: Overall Global performance analysis with respect to Noisy dataset | 20 |

**List of All Figures**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Figure Title** | **Page Number** |
| 1 | Figure Title: Demonstration of The Number of Culprit Points Caught | 10 |
| 2 | Figure Title:Flow chart of the Mode Method | 11 |
| 3 | Figure Title: Variation in Accuracy across the dataset | 11 |
| 4 | Figure Title:Flow chart of the Mean Method | 12 |
| 5 | Figure Title: Scatter plot representation of the vector containing mean weights | 13 |
| 6 | Figure Title: Variation in Accuracy across the dataset | 13 |
| 7 | Figure Title:Flow chart of the Cluster based Boosting | 14 |
| 8 | Figure Title:Flow chart of the Clustering as a Feature | 15 |
| 9 | Figure Title: Variation in Accuracy across the dataset | 16 |
| 10 | Figure Title:Flow chart of the Feature with Semi-Supervised Learning | 17 |
| 11 | Figure Title: Variation in Accuracy across the dataset | 18 |
| 12 | Figure Title:Flow chart of the Double Addition of Cluster in Semi-Supervised Learning | 19 |
| 13 | Figure Title: Variation in Accuracy across the dataset | 20 |

***Abstract*—**

***With the substantial growth in the scale of data, an increasing amount of training data is available in many Machine Learning tasks. However, it is difficult to ensure perfect labelling with a large volume of training data. Some labels can be incorrect, resulting in label noise, which could lead to deterioration in performance. Label noise is an important issue in classification, with potential negative consequences. For example, the accuracy of predictions may decrease, whereas the complexity of inferred models and the number of necessary training samples may increase. To overcome this, different techniques have been developed that tried to improve the recovery process and its accuracy which are the vital factor in the predictive analyses of Machine Learning. Many works in the literature have been devoted to the study of label noise and the development of techniques to deal with label noise. Boosting is one technique that is used to augment the accuracy of prediction in supervised learning. This paper surveys various boosting methodologies and analyses them so as to suggest a noise free environment in order to have maximum accuracy.***

**Keywords— Adaboost, Classifier, Decision stump, Semi-Supervised Learning**

# 1.INTRODUCTION

Boosting process is a Machine Learning ensemble meta-algorithm used in order to improve predictive accuracy. This iterative process primarily reduces the bias and the variance in supervised learning. But the main idea of boosting is to train weak learners sequentially, respectively trying to correct its precursor. So, it focuses more on the incorrect data and converts the weak learners into stronger ones after certain iterations. To find weak rule, base learning algorithms are applied with a different distribution and each time it is applied, it generates a new weak prediction rule. As it is an iterative process, after many iterations, the boosting algorithm combines these weak rules into a single strong prediction rule. In spite of all these, it does have some issues like the training data might contain some noise label where some instances are wrongly inserted so the boosting learning function also studies erroneously which effects the accuracy of the predictions. In boosting, overfitting occurs and the samples located in the overlapping area are generally classified irregularly. Due to the boosting process overfitting is done on the training data and the filtering of the correct data in definite functions becomes impossible as they may focus in areas not predicted well by other learners. AdaBoost or Adaptive Boosting is one of the types of boosting which is successful boosting algorithm developed for binary classification.

One of the techniques used in various decision making, grouping, pattern analysis and machine learning situations is clustering. It is used frequently by many researchers for the grouping of the unlabelled data in the required methods and in tentative data analysis to find the hidden patterns or grouping in data. Clustering is a type of unsupervised learning method, mostly used in statistical data analysis, that divides the data points into different groups so that the data points in same groups are similar to the rest of the data points in that same group and dissimilar to other data points in other groups. It helps to determine the intrinsic grouping among the unlabelled data points and it could be density-based method, hierarchical-based method, grid-based method etc. Due to the presence of noise, the classifiers are not able to predict correctlyfrom data set. So, in this paper different methods are proposed that could do the predictions correctly and provide a better accuracy and identify the noisy data on the other hand. In this paper a pre-processing technique is proposed which is supposed to remove noisy data from the raw dataset thus making is feasible for doing predictions as otherwise the predictions will be incorrect and accuracy will also decrease. This paper deals with the cluster boosting data points with various techniques.

# 2.LITERATURE SURVEY

The proposed paper L. Dee Miller et al. [2] surveyed that the boosting technique cannot handle noisy data and difficult areas where the instance’s relevant features are different from the training data. The author mentioned that boosting of incorrectly predicted data causes problems. In the proposed paper, they partitioned the training data into clusters and integrated them directly into the boosting process. They used cluster-based boosting.

McDonald et al. [6] did the empirical performance analysis of the boosting algorithm on the real sets of data with artificial class labels. They found out that Brownboost and Logitboost are proved to be less likely to overfit than Adaboost in this context and it do yield much better generalization error than Adaboost. But in real scenario it is quite difficult to estimate the class noise levels.

Brodley and Friedl [7][8] presented an empirical and analytical estimation of the precision that the elimination of noise in the context of filtering operation where training of multiple classifiers is done from the noisy data points to detect the noise. Most of the approaches filter out the bad data points but at the expense of good data.

Zhu et al. [10] proposed a multiple round noise elimination that is kind of a similar to the boosted noise filter instead of second level ensemble. Multiple rounds of ensemble improved the noisy instances as at each round the noise identified from the previous step is removed let say weight 0 for the removed instances and 1 for the rest. They used soft weighting approach where the incorrectly identified instances (noise) at a run have a chance to get correlated later unlike the heavy weighting where the incorrectly identified instances are removed.

Gamberger and Lavrac (19960 [11] developed method that handled the noise that eventually removed the inconsistent instances, like those having similar values for the features and different class labels, from the training data. They transformed them into the binary feature set and examined that by removing which set of instances may reduce the total number of literals needed to retain that the existing set of instances aren’t inconsistent.

Amitava Karmarker et al. [12] proposed a modification to Adaboost that seems to be more tolerant to the class label noise and hence enhances the prediction accuracy. Using ORBoost employed a edge scheme in the elimination of the class label noise. Adaboost however is susceptible of overfitting due to the outliers present in the data which may eventually gain more weight than the actual instances. Whereas, ORBoost distinguish the noisy instances explicitly and because of this the threshold level improves the detection performance of the outliers.

Nagarajan Natarajan et al. [16] theoretically studied the binary classification in the presence of random classification noise. They focused on the risk minimization in the presence of random noise and examined the results using unbiased estimators and weighted loss function. The classification performance improved impressively even at high noise rates. They used biased SVM and weighted logistic regression methods that wee noise tolerant evidently. The method achieved 88% accuracy when there were 44%corrupted labels. They simply suggested the use of simple weighted surrogate loss that let them gain strong empirical risk bounds.

Quinlan [18] projected that while removing noisy class labels the elimination of noise by the decision tree decreases the prediction’s precision which in turn increments the prediction accuracy. This paper synthesized the decision trees and modified in such a way that it deals well with the noisy information.

# METHEDOLOGIES

## **3.1 Problem Statement**

With the significant growth in the scale of data it is difficult to ensure a perfect labelling with a large set of training data. Some labels can be incorrect, resulting in label noise, which could lead to deterioration in the performance and also deteriorates the accuracy. In this project we used pre-processing techniques (algorithms) for the identification of the mislabelled data points and corrected it by flipping the labels so that the noisy data could be made noise-free.

**3.2 d/m Method (Existing method)**

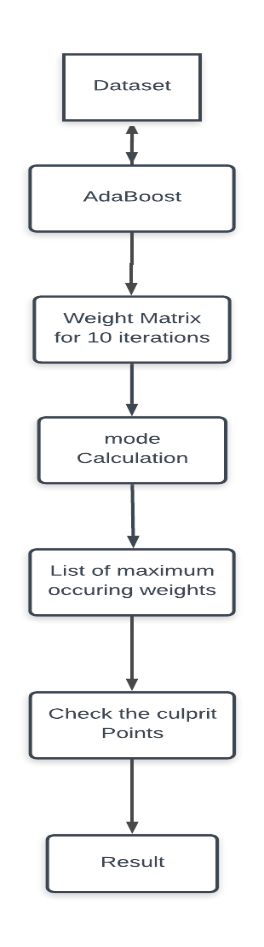
Many research works proposed different methods to deal with the class label noise. We studied different methods and one of them was d/m method using AdaBoost.For each possible value for d, they run AdaBoost on the remaining training instances. A training instance is discarded as an outlier whenever it has a weight greater than d/m where m is the number of remaining training instances. The best threshold value that gives the lowest error on the validation set is then assumed to be the optimum and the classifier constructed with that threshold is the final classifier. The classifier produced is then used to classify the test instances to get its estimated test error. Firstly, the dataset is distributed as 30% testing and 70% training set. Further the 70% set is distributed as 20% testing and 80% training set. On the 80% training dataset AdaBoost is applied. Then the Weight list obtained is used for comparison: Threshold=d/m (initially)where they took d values from 3-21 and m=length of 80% training dataset.

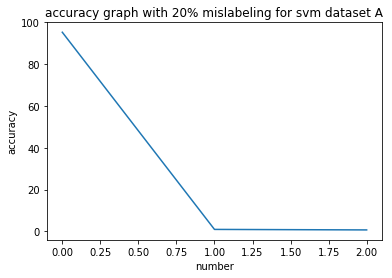
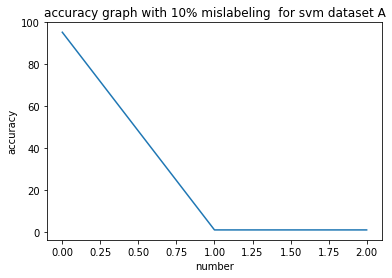


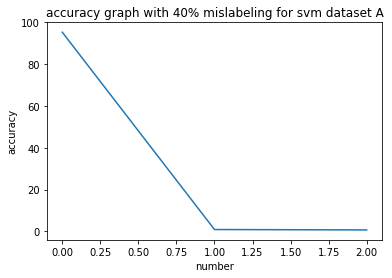
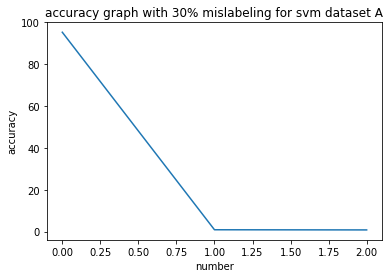
**Fig1: Demonstration of The Number of Culprit Points Caught**

**3.3 Mode Method**

We proposed two methods: mean method and mode method and evaluated the accuracies with and without labelling the data points. In mode method, we took out the mode of the weights of the last iteration and bunched out those weights which were having the highest frequency. After that we collected indices for those weights and removed those respective indices from our training data set. The method used the weight matrix for the mode calculations and obtains a list of maximum weights that are occurring, from which the culprit points are detected.

 **Fig2: Flow chart of the Mode Method**

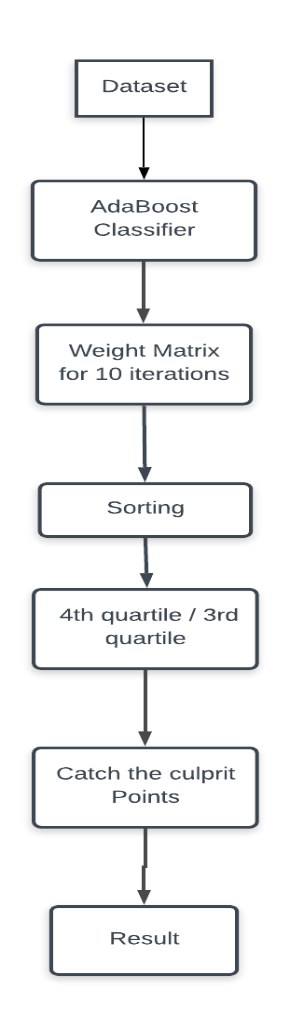




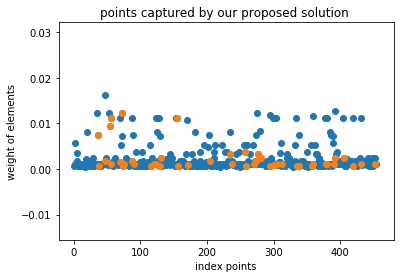
**Fig3: Variation in Accuracy across the dataset**

**3.4 Mean Method**

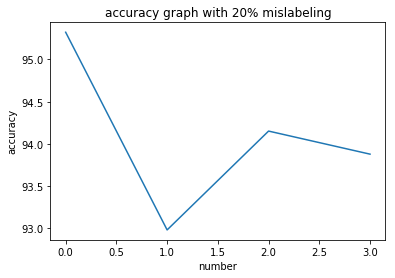
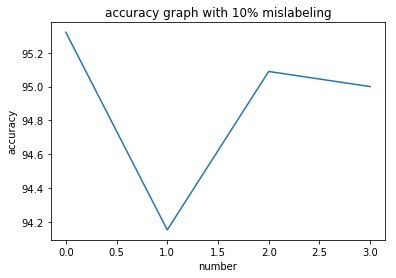
In mean method the weights obtained from all the ten iteration was collected in a vector and the mean of the weights present in this vector was taken out. Indices were attached to the respective weights and the weights were sorted accordingly. After this the fourth and third quartile data points were segregated and their respective indices i.e. the indices of the weights in third and fourth quartile were bunched. The labels of these indices from the training data set were flipped and the results for the accuracy were as follows-

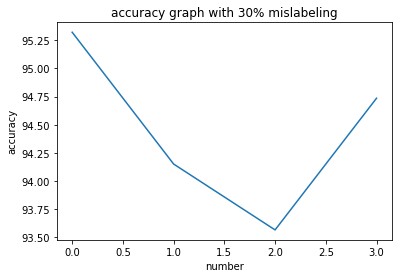
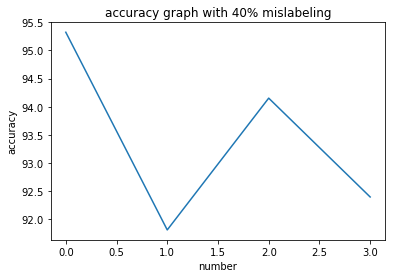


**Fig4: Flow chart of the Mean Method**



**Fig5: Scatter plot representation of the vector containing mean weights**

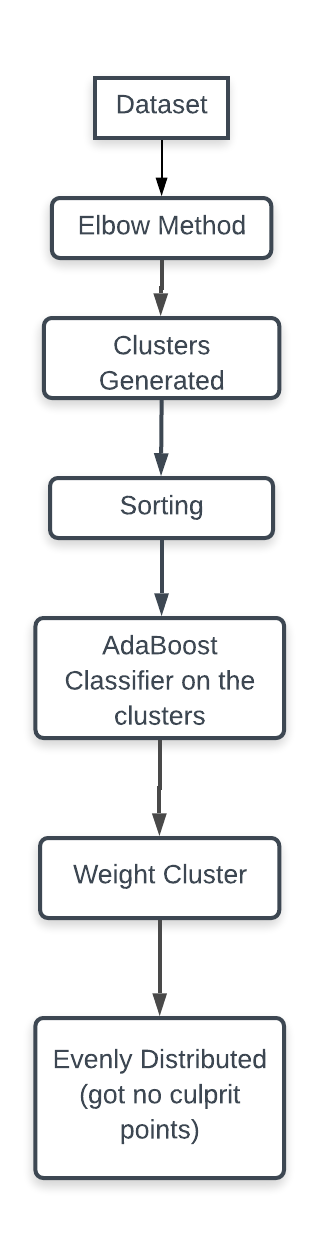




**Fig 6: Variation in Accuracy across the dataset**

**3.5 Cluster based Boosting**

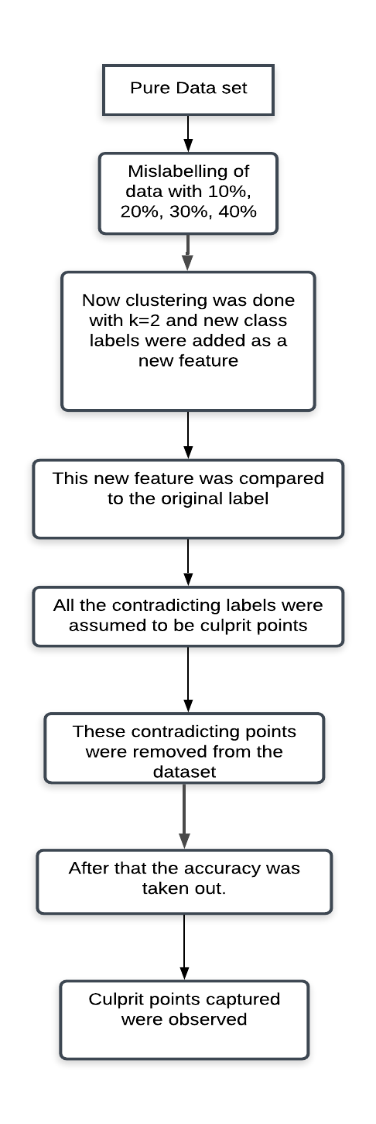
In this method k was determined for the dataset using elbow method. The entire dataset was divided into that number of clusters. Each of these cluster was passed into the adaboost. We assumed that a particular group of cluster points will show a different trend but on plotting the graph the method showed uniform distribution was obtained for the culprit as well as non culprit data points. All this led to the conclusion that adaboost is of no use for the prediction in our hypothesis.



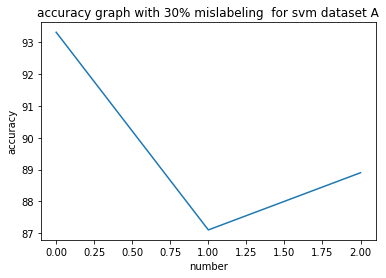
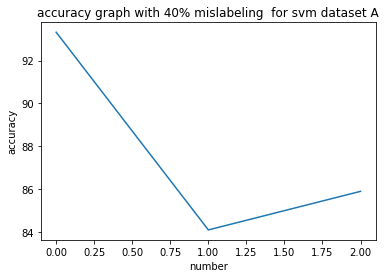
**Fig7: Flow chart of the Cluster based Boosting**

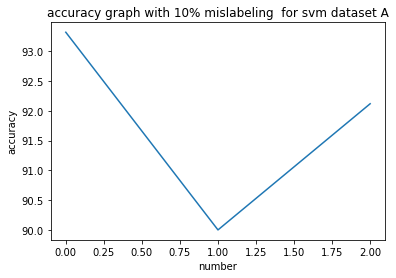
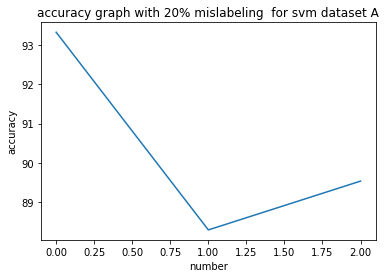
**3.6 Clustering as a feature**

In this method the pure data set was obtained and mislabelled with 10%, 20%, 30% and 40% mislabelling. After this clustering was obtained on the entire dataset with k=2. The newly created label is now compared with the old class labels. We assume that the points which contradicts in this procedure are our culprit data point. We remove these culprit points and the accuracy was obtained accordingly.



**Fig8: Flow chart of the Clustering as a Feature**

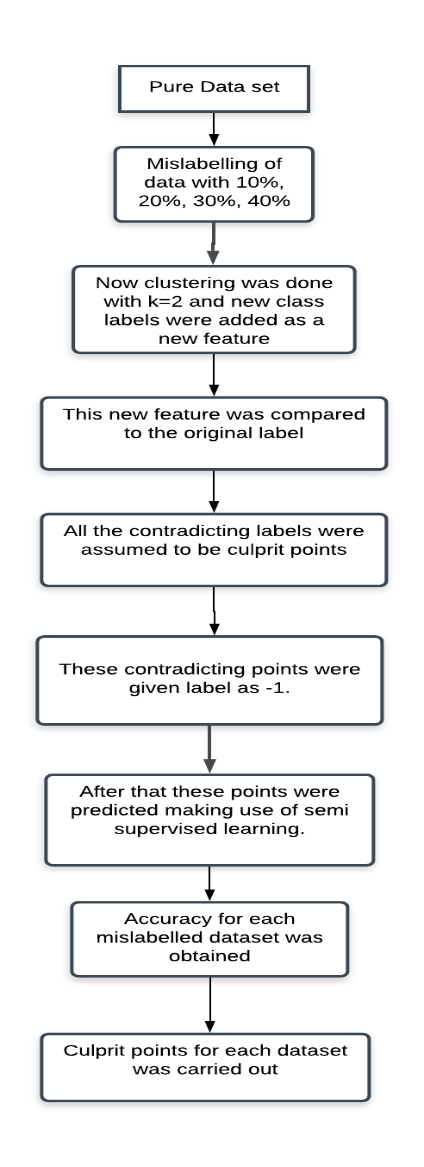




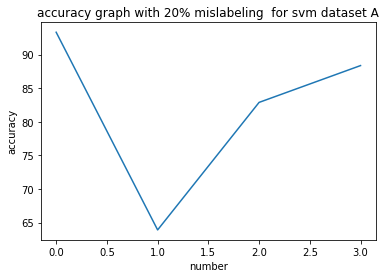
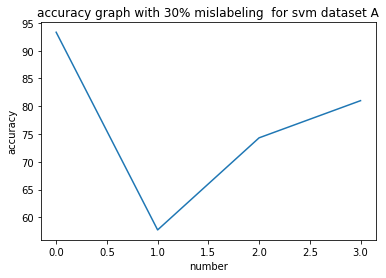
**Fig9: Variation in Accuracy across the dataset**

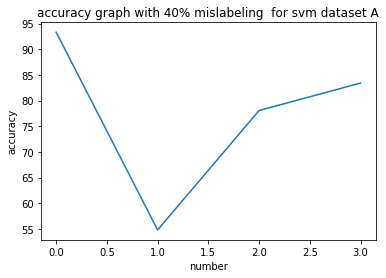
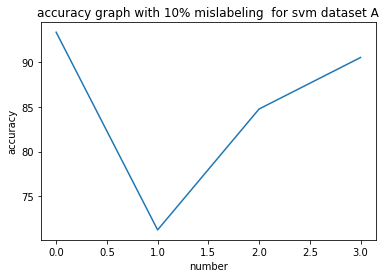
**3.7** **Clustering as a Feature with Semi-Supervised Learning**

In this method the pure data set was obtained and mislabelled with 10%, 20%, 30% and 40% mislabelling. After this clustering was obtained on the entire dataset with k=2. The newly created label is now compared with the old class labels. We assume that the points which contradicts in this procedure are our culprit data point. The labels corresponding to the obtained culprit points were removed and a model was trained with the remaining dataset.



**Fig10: Flow chart of the Clustering as a Feature with Semi-Supervised Learning**

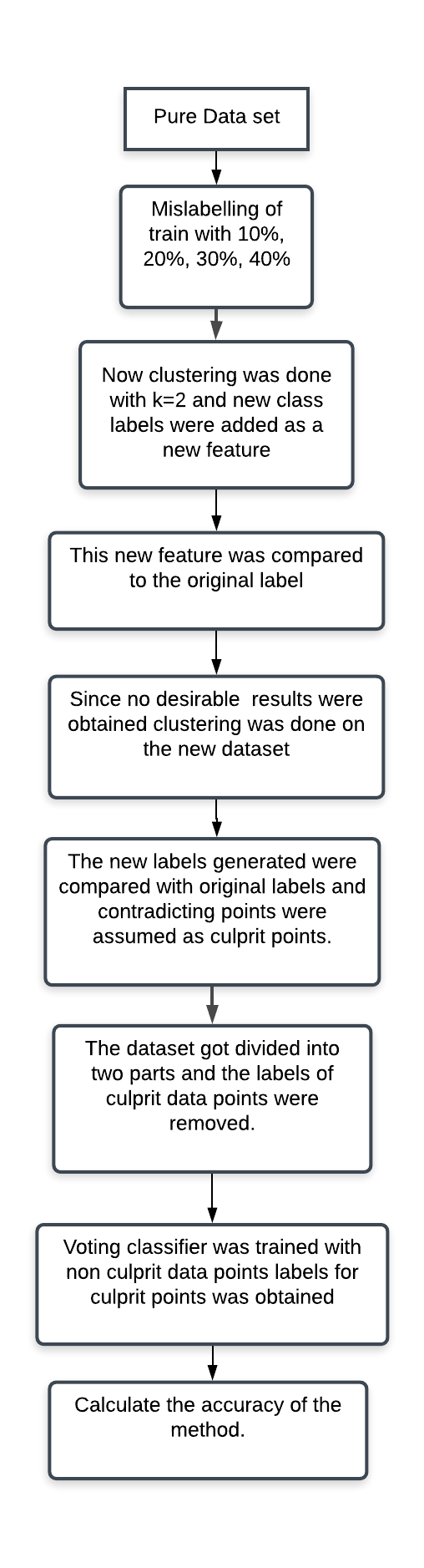




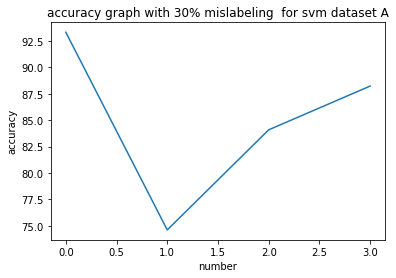
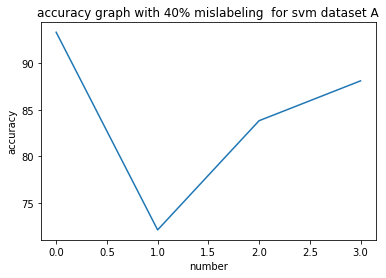
**Fig 11: Variation in Accuracy across the dataset**

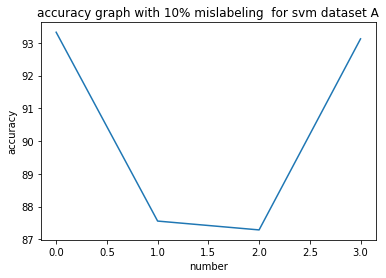
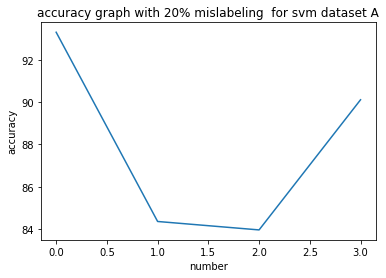
**3.8 Double addition of Cluster as a feature with semi supervised learning**

In this method the pure data set is taken and mislabelling in the percentage of 10%, 20%, 30% and 40% are introduced. After this clustering is done in the dataset with k=2. This newly created feature is compared with original label and poor results were obtained with this. After this these labels were added as feature and again clustering with k=2 was done. These labels were compared with original labels and culprit points were decided according to the assumption that those labels which did not match with the previous labels are culprit points. Thus, the dataset gets divided into two parts with the result that one part contains the entire culprit point and the other part is devoid of any culprit point. The labels of the data points which are assumed to be culprit points are removed and a model is created by training it on the non-culprit data set. The predictions of the labels of the culprit data points is made using this method. In this way though we are not able to detect culprit points we are able to minimise its effect with the result that it gives a minimum of removal of 50% culprit points and a maximum of removal of 70% culprit points.



**Fig12: Flow chart of the Double addition of Cluster as a feature with semi supervised learning**





**Fig 13: Variation in Accuracy across the dataset**

**4.RESULTS**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dataset** | **Total Observations** | **% noise injected (#)** | **Methods** | **# observations in culprit** | **Culprit**  **dataset** | | **Cleaned dataset** | | **Efficiency with dataset (%)** | | | |
|  |  |  |  |  | **N N** | **P N** | **N P** | **P P** | **Pure** | **Noisy** | **Cleaned** |
| **Breast Cancer** | 715 | 10 (45) | M1.1 | 114 | 22 | 92 | 23 | 578 | 95.32 | 94.15 | 95.08 |
| M1.2 | 57 | 15 | 42 | 30 | 628 | 95.32 | 94.15 | 95 |
| M2 | 145 | 11 | 134 | 34 | 536 | 95.32 | 95 | 95 |
| M3 | 74 | 42 | 32 | 3 | 638 | 93 | 90 | 92.12 |
| M4 | 74 | 41 | 33 | 4 | 637 | 93.50 | 71.25 | 84.75 |
| M5 |  |  |  |  |  |  |  |  |
| 20 (90) | M1.1 | 114 | 54 | 60 | 36 | 565 | 95.32 | 92.98 | 94.15 |
| M1.2 | 57 | 30 | 27 | 60 | 598 | 95.32 | 92.98 | 93.87 |
| M2 | 136 | 34 | 102 | 56 | 523 | 95.32 | 94.3 | 72 |
| M3 | 149 | 73 | 76 | 17 | 549 | 93 | 89.1 | 90.1 |
| M4 | 149 | 73 | 76 | 17 | 549 | 93.50 | 63.90 | 88.36 |
| M5 |  |  |  |  |  |  |  |  |
| 30 (135) | M1.1 | 114 | 65 | 49 | 70 | 531 | 95.32 | 94.15 | 93.56 |
| M1.2 | 57 | 23 | 34 | 112 | 546 | 95.32 | 94.15 | 94.73 |
| M2 | 118 | 38 | 80 | 97 | 500 | 95.32 | 87 | 79 |
| M3 | 224 | 112 | 112 | 23 | 468 | 93 | 87.1 | 88.9 |
| M4 | 224 | 101 | 123 | 34 | 457 | 93.50 | 57.75 | 81.01 |
| M5 |  |  |  |  |  |  |  |  |
| 40 (180) | M1.1 | 114 | 67 | 47 | 113 | 488 | 95.32 | 91.81 | 94.15 |
| M1.2 | 57 | 27 | 30 | 153 | 505 | 95.32 | 91.81 | 92.39 |
| M2 | 130 | 19 | 111 | 161 | 424 | 95.32 | 83 | 65 |
| M3 | 299 | 99 | 200 | 81 | 335 | 93 | 84.1 | 85.9 |
| M4 | 299 | 150 | 149 | 30 | 386 | 93.50 | 54.81 | 83.42 |
| M5 |  |  |  |  |  |  |  |  |
| **Mushroom** | 800 | 10 (45) | M1.1 | 114 | 35 | 79 | 10 | 676 | 91 | 97.15 | 97.08 |
| M1.2 | 57 | 17 | 40 | 28 | 715 | 91 | 97.15 | 96.00 |
| M2 | 145 | 14 | 131 | 31 | 624 | 91 | 90 | 89 |
| M3 | 74 | 42 | 32 | 3 | 723 | 90 | 91 | 91.78 |
| M4 | 74 | 41 | 33 | 4 | 722 | 90.70 | 85.6 | 91.2 |
| M5 |  |  |  |  |  |  |  |  |
| 20 (90) | M1.1 | 114 | 50 | 64 | 40 | 646 | 91 | 94.98 | 95.34 |
| M1.2 | 57 | 24 | 33 | 66 | 677 | 91 | 94.98 | 96.45 |
| M2 | 136 | 30 | 106 | 60 | 604 | 91 | 93.76 | 64.25 |
| M3 | 149 | 73 | 76 | 17 | 634 | 90 | 87.1 | 88 |
| M4 | 149 | 73 | 76 | 17 | 634 | 90.70 | 83.4 | 89 |
| M5 |  |  |  |  |  |  |  |  |
| 30 (135) | M1.1 | 114 | 67 | 47 | 68 | 618 | 91 | 93.15 | 95.34 |
| M1.2 | 57 | 27 | 30 | 108 | 635 | 91 | 93.15 | 95.67 |
| M2 | 118 | 34 | 84 | 101 | 581 | 91 | 81 | 72 |
| M3 | 224 | 112 | 112 | 23 | 553 | 90 | 87.36 | 88.7 |
| M4 | 224 | 101 | 123 | 34 | 542 | 90.70 | 72.3 | 82.3 |
| M5 |  |  |  |  |  |  |  |  |
| 40 (180) | M1.1 | 114 | 68 | 46 | 112 | 574 | 91 | 92.81 | 94.65 |
| M1.2 | 57 | 29 | 28 | 151 | 592 | 91 | 92.81 | 94.57 |
| M2 | 130 | 18 | 112 | 162 | 508 | 91 | 77.36 | 67 |
| M3 | 299 | 99 | 200 | 81 | 420 | 90 | 85.36 | 84.57 |
| M4 | 299 | 150 | 149 | 30 | 471 | 90.70 | 79 | 80 |
| M5 |  |  |  |  |  |  |  |  |

**Table 1:** **Overall Global performance analysis with respect to Noisy dataset:M1.1-Mean method 3rd quartile, M1.2- Mean method 4th quartile, M2- Mode method, M3- Cluster as features, M4- Clustering as a Feature with Semi-Supervised Learning, M5- Double addition of Cluster as a feature with semi supervised learning**

# 5.CONCLUSION

Thus we conclude from the implementation of our project that adaboost gives a uniform distribution of weight and cannot be used to detect culprit data point solely on the basis of weight. We came up with a new concept of semi supervised learning and used to reduce the effect of culprit data points.

# 6.ACKNOWLEDGMENT

The authors would like to thank Dr. Vivek Tiwari for his backend support, for the review and discussion of this project at every step.

**7. REFERENCES**

[1]. D. Frossyniotis, A. Likas, and A. Stafylopatis, “A clustering method based on boosting,” Pattern Recog. Lett., vol. 25, pp. 641–654, 2004.  
[2]. L. Dee Miller and Leen- Kiat Soh, Member, IEEE “Cluster-Based Boosting”.  
[3]. C. L. Blake, C. Mertz. UCI Repository of Machine Learning Databases, 1998.  
[4]. C. E. Brodley, M. A. Friedl. Identifying mislabelled training data. In: proceedings of Thirteenth National Conference on Artificial Intelligence, pp. 799-805, 1996.  
[5]. Amitava Karmaker, Stephen Kwek A Boosting Approach to remove Class Label Noise. In: Proceedings of the Fifth International Conference on Hybrid Intelligent Systems (HIS’05) 0-7695-2457-5/05 2005 IEEE.  
[6]. R. A. McDonald, D. J. Hand, I. A. Eckley. An Empirical Comparison of Three Boosting Algorithms on Real Data Sets with Artificial Class Noise. Multiple Classifier Systems, 35-44, 2003.  
[7]. C. E. Brodley, M. A. Friedl. Identifying and eliminating mislabeled training instances. Journal of Artificial Intelligence Research, vol 11, 131-167, 1999.  
[8]. C. E. Brodley, M. A. Friedl. Identifying mislabeled training data. In: proceedings of Thirteenth National Conference on Artificial Intelligence, pp. 799-805, 1996.  
[9]. Wilson, D. (1972). Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans. on Systems, Man and Cybernetics, 2, 408{421.  
[10]. X. Zhu, X. Wu, and Q. Chen. Eliminating class noise in large datasets. In *Proc. 20th Int. Conf. Machine Learning*, pages 920.927, Washington, DC, 2003.  
[11]. Gamberger, D., Lavrac, N., & Dzeroski, S. (1996). Noise elimination in inductive concept learning: A case study in medical diagnosis. In Seventh International Workshop on Algorithmic Learning Theory, pp. 199{212. Springer.  
[12].Karmaker,Amitava,Kwek,Stephen (2006) “A boosting Approach to remove class label noise” hybrid intelligent systems in ensembles :[International Journal of Hybrid Intelligent Systems](https://content.iospress.com/journals/international-journal-of-hybrid-intelligent-systems), vol. 3, no. 3, pp. 169-177, 2006  
[13]. S. Verbaeten and A. Van Assche. Ensemble methods for noise elimination in classi\_cation problems. In T. Windeatt and F. Roli, editors, *Multiple Classi\_er Systems: Fourth International Workshop,* volume 2709 of *Lecture Notes in Computer Science*, pages 317.325, 2003.  
[14]. J. R. Quinlan. Induction of decision trees. Machine Learning, 1(1): pp. 81-106, 1986.  
[15]. A.Ganatra and Y. Kosta, “Comprehensive evolution and evaluation of boosting,” Int. J. Comput. Theory Eng., vol. 2, pp. 931–936, 2010.  
[16].Nagarajan Natarajan, Inderjit S. Dhillon, Pradeep Ravikumar (2013)”Learning with Noisy Labels” [Advances in Neural Information Processing Systems 26 (NIPS 2013)](http://papers.nips.cc/book/advances-in-neural-information-processing-systems-26-2013)