**GUIDELINES FOR THE PREPARATION OF B.Tech. MINI PROJECT REPORT**

The sequence in which the thesis material should be arranged and bound should be as follows:

1. Cover page
2. Dedication page (Optional)
3. Bonafide Certificate
4. Abstract
5. Acknowledgements
6. Table of Contents
7. List of Tables
8. List of Figures
9. List of Symbols, Abbreviations or Nomenclature (Optional)
10. Chapters
11. Appendices
12. References

**A Pre-processing Method Using Adaboost 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 Jaiswal (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 Using Adaboost 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.**

**ABSTRACT…………………………………………………………...…….. i**

**TABLE OF CONTENTS……………………………………….................... ii**

**LIST OF TABLES………………………………………………………….. iii**

**LIST OF FIGURES………………………………………………................ iv**

**CHAPTER 1 INTRODUCTION 1**

**CHAPTER 2 LITERATURE REVIEW 3**

**CHAPTER 3 METHEDOLOGIES 10**

* 1. Problem Statement…………………………………..……………... 10
  2. Proposed Work (d/m method)……………………………………… 11
  3. Mode Method……………………………………………………….
  4. Mean Method……………………………………………………….
  5. Cluster based Boosting……………………………………………..
  6. Clustering as a Feature……………………………………………..
  7. ………………...……………………………
  8. …………………...…………………………

**CHAPTER 4 RESULTS**

* 1. Mode Method………………………………………………………….. 12
  2. Mean Method………………………………………………………….. 13
  3. Clustering With AdaBoost…………………………………………….
  4. Clustering as a Feature………………………………………………...

**CHAPTER 5 CONCLUSIONS 14**

**REFERENCES 18**

**List of All Tables**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Table Title** | **Page Number** |
| 1 | Table  Title: Comparison table of accuracy obtained by mode method versus accuracy of labelled and mislabelled data | 1 |
| 2 | Table  Title: Comparison table of accuracy obtained by mean method versus accuracy of labelled and mislabelled data. | 6 |
| 3 | Table  Title: Details of the culprit points detected by our algorithm |  |
| 4 | Table  Title: Overall Global performance analysis with respect to Noisy dataset |  |
|  |  |  |

**List of All Figures**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Figure Title** | **Page Number** |
| 1 | Figure  Title: Demonstration of The Number of Culprit Points Caught | 1 |
| 2 | Figure  Title:Flow chart of the Mode Method | 6 |
| 3 | Figure  Title:Flow chart of the Mean Method |  |
| 4 | Figure  Title:Flow chart of the Cluster based Boosting |  |
| 5 | Figure  Title:Flow chart of the |  |
| 6 | Figure  Title:Flow chart of the |  |
| 7 | Figure  Title:Flow chart of the |  |
| 8 | Figure  Title: points captured by 10 mislabelled and 10 iterations |  |
| 9 | Figure  Title: points captured by 20 mislabelled and 10 iterations |  |
| 10 | Figure  Title: points captured by 30 mislabelled and 10 iterations |  |
| 11 | Figure  Title: points captured by 40 mislabelled and 10 iterations |  |
| 12 | Figure  Title:Frequency of points captured by 10 mislabelled and 10 iterations |  |
| 13 | Figure  Title:Frequency of points captured by 20 mislabelled and 10 iterations |  |
| 14 | Figure  Title:Frequency of points captured by 30 mislabelled and 10 iterations |  |
| 15 | Figure  Title:Frequency of points captured by 40 mislabelled and 10 iterations |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

***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. It adds weak models subsequently and train them using the weighted training data that’s goes on until a specific number of weak learners are being created. The AdaBoost does the predictions by calculating the weighted average of the weak classifiers.

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 correctly from 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. They showed that filtering process improved the accuracy of the classification of the data set containing the noisy 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.

Verbaeten and Van Assche [13] examined n-fold cross validation, boosting and bagging approaches and found out that n-fold cross validation and the bagging worked well. In their work boosting is used merely as a base noise filter where at each run the incorrectly predicted training instances by trained classifier gets higher weight in the succeeding run. At the end the instances having larger weights are identified as noise.

Wilson [9] proposed the idea of eliminating instances so as to improve the nearest neighbor classifier. The paper focused on k-nearest neighbor (k-NN) classifier for the selection of the instances.

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.

A. Ganatra and Y. Kosta [15] reviewed the ensemble of the classifiers obtained by the generation and the combination of the classifiers constructed using suitable machine learning methods that targeted to increase the predictive accuracy with reference to the base classifier. Boosting is an eminent process that improves any learning algorithm’s performance. The paper put forward the problems faced during the generation of accurate prediction rule.

# 3.Methodologies

## **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 of subsequent predictions. In this project we have 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 Proposed Work**

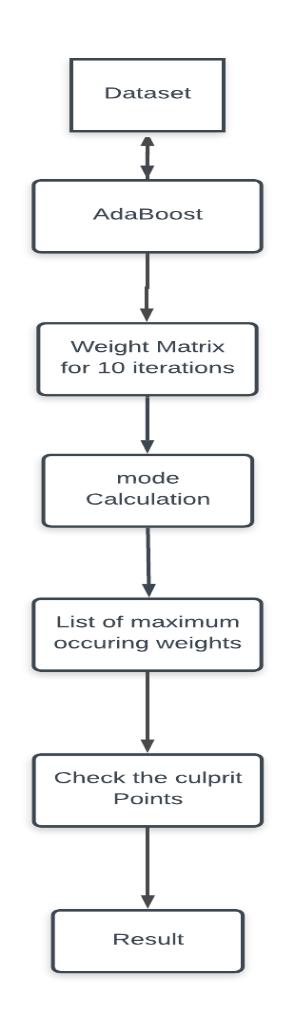
Many research works proposed different methods to deal with the class label noise. We studied different methods and one of them was d/m method using AdaBoost.For each possible value for d, they run AdaBoost on the remaining training instances. A training instance is discarded as an outlier whenever it 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.

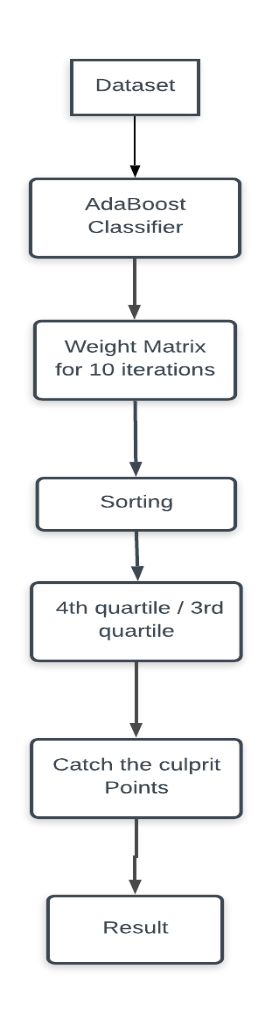
 **Fig2: Flow chart of the Mode Method**

|  |  |  |  |
| --- | --- | --- | --- |
| **Classifier** | **Accuracy Without Mislabelling** | **Accuracy with Mislabelling (10%)** | **Accuracy by Mode method (10%)** |
| **SVM** | 0.9561403508 | 0.9122807017 | 0.97368421052631 |
| **Knn** | 0.9649122807 | 0.9561403508771 | 0.81578947368421 |
| **Decision Tree** | 0.9473684210 | 0.8070175438596 | 0.61403508771929 |
| **Gaussian NB** | 0.9561403505 | 0.9473684210526 | 0.92982456140350 |

**Table1:**  **Comparison table of accuracy obtained by mode method versus accuracy of labelled and mislabelled data.**

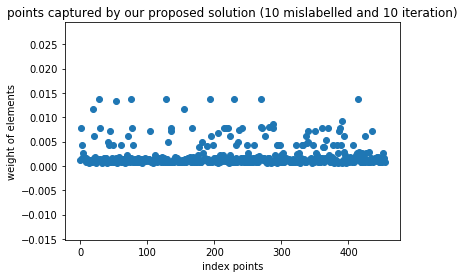
**3.4 Mean Method**

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-

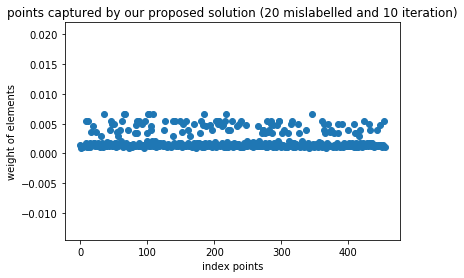


**Fig3: Flow chart of the Mean Method**

Scatter plot representation of the vector containing mean weights:



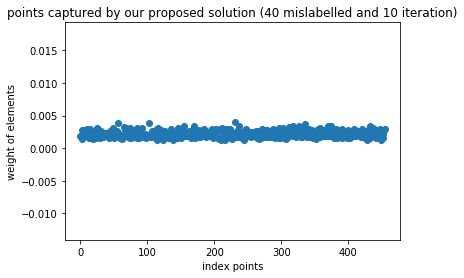
**Fig8: points captured by 10 mislabelled and 10 iterations**



**Fig9: points captured by 20 mislabelled and 10 iterations**

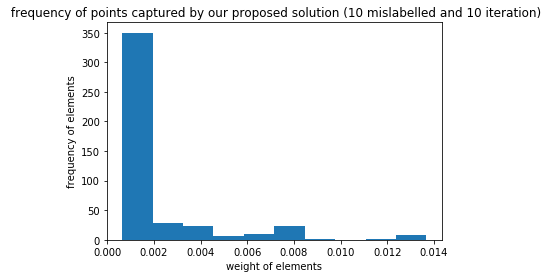


**Fig10: points captured by 30 mislabelled and 10 iterations**

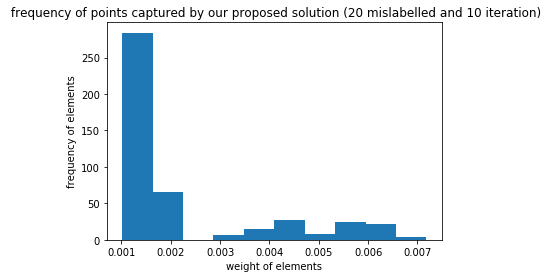


**Fig11: points captured by 40 mislabelled and 10 iterations**

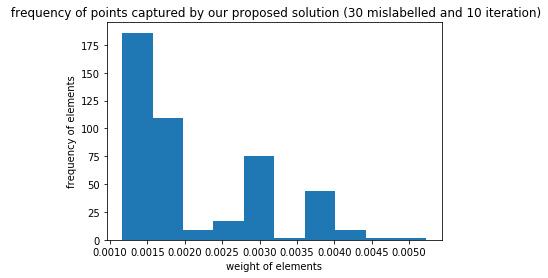
Histogram Representation of the vector containing mean weights:



**Fig12: Frequency of points captured by 10 mislabelled and 10 iterations**



**Fig13: Frequency of points captured by 20 mislabelled and 10 iterations**



**Fig14: Frequency of points captured by 30 mislabelled and 10 iterations**



**Fig15: Frequency of points captured by 40 mislabelled and 10 iterations**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S No.** | **Accuracy without mislabeling** | **Accuracy with mislabeling** | **Accuracy by flipping 3rd quartile** | **Accuracy by flipping 4th quartile** |
| **SVM** | 0.95614035 | 0.9122807017 | 0.8333333333 | 0.9649122807 |
| **Knn** | 0.96491228 | 0.9561403508 | 0.7982456140 | 0.9473684210 |
| **Decision Tree** | 0.94736842 | 0.8070175438 | 0.6929824561 | 0.8333333333 |
| **Gaussian NB** | 0.95614035 | 0.9473684210 | 0.9649122807 | 0.9561403508 |

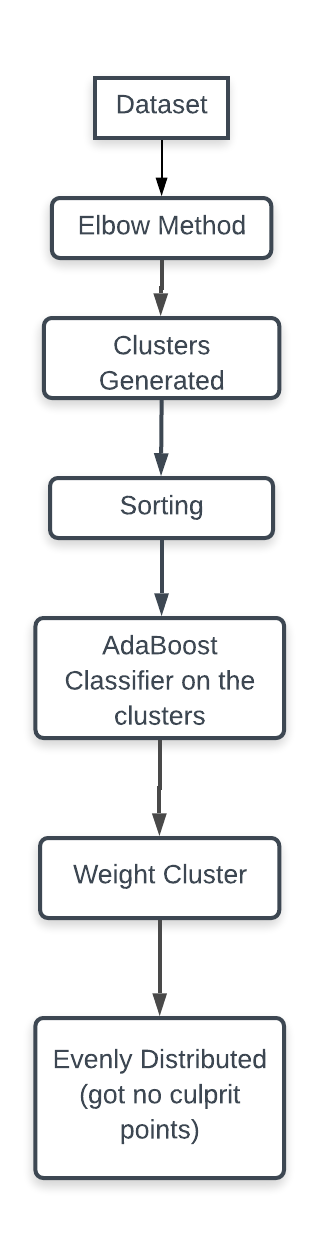
**Table2: Comparison table of accuracy obtained by mean method versus accuracy of labelled and mislabelled data.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S No.** | **10%-10**  **Iterations** | **20%-10**  **Iterations** | **30%-10**  **Iterations** | **40%-10**  **Iterations** |
| **No. of detected elements in 3rd quartile** | 6 | 12 | 12 | 24 |
| **No. of detected elements in 4th quartile** | 15 | 23 | 28 | 51 |
| **Total no of mislabeled data points** | 45 | 91 | 136 | 182 |
| **Data points present in 3rd quartile** | 114 | 114 | 114 | 114 |
| **Data points present in 4th quartile** | 55 | 55 | 55 | 55 |

**Table-3 Details of the culprit points detected by our algorithm**

**3.5 Cluster based Boosting**

Another method proposed is clustering with Adaboost. In this method cluster is created and using elbow method number of clusters are found out. Further, Adaboost is applied to the individual cluster and the weight list obtained is visualized. Strong SVM classifier is used in this method. But no culprit point got detected hence this didn’t go right.



**Fig4: Flow chart of the Cluster based Boosting**

**3.6 Clustering as a feature**

Later we also proposed another method where cluster is used as a feature. Firstly, the cluster is created and by using elbow method number of clusters are found out. The cluster label is added as a feature in the data set. Now, strong classifier SVM is applied and contradict points are obtained which are further compared with the actual noisy label and the classifier predicted label. The culprit points were obtained but they were less in number.

**Fig5: Flow chart of the**

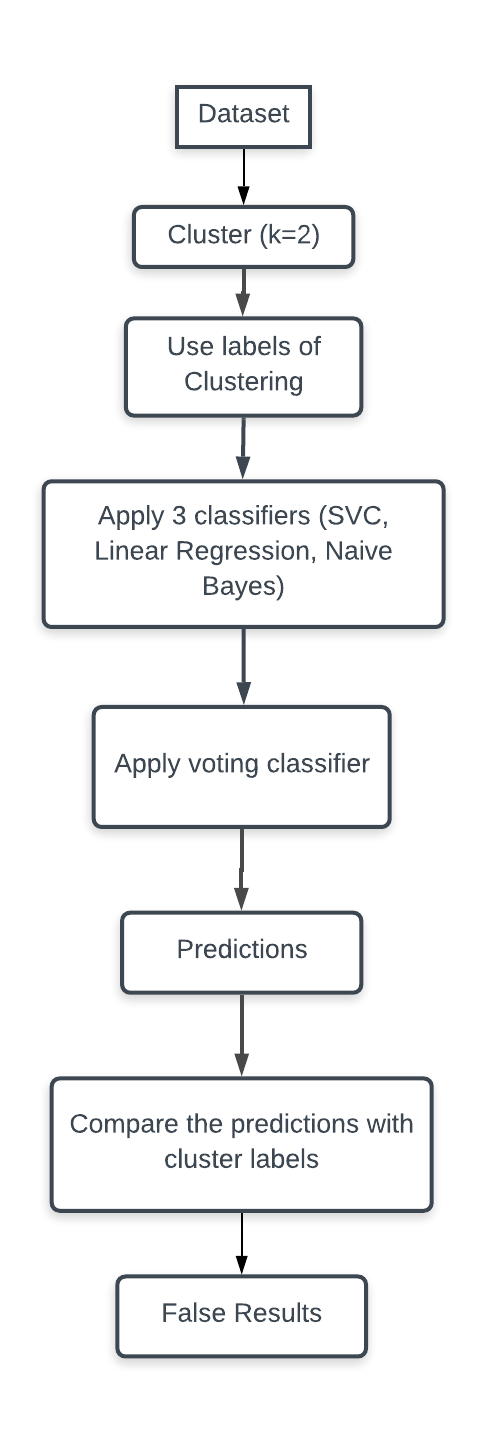
**3.7**

Another method creates cluster (k) and are added as a feature. In this method four classifiers are used: Adaboost, SVM, Logistic Regression and Ensemble Classifier (voting classifier). // not explained to me properly

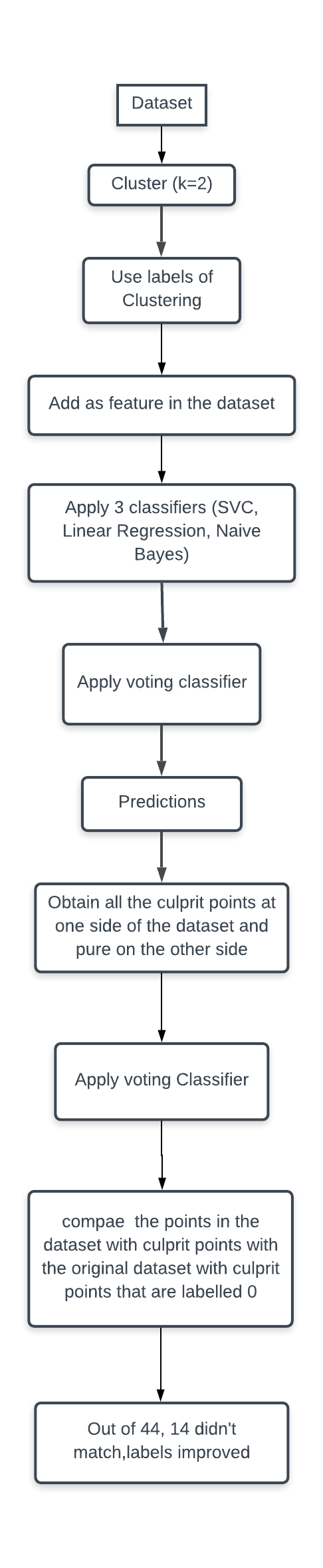
**Fig6: Flow chart of the**

**3.8**

This next method creates cluster where k=2 (binary dataset). Labels are obtained and are checked with the actual noisy label. The label is added as feature in the original dataset and again clustering (k=2) is done. Labels are obtained and are compared with the actual culprit points. All the points are mostly obtained but the contradict points were also high. So, we applied voting classifier and detected the indices of all the contradict points and applied voting classifier to them. The predicted labels are compared with the actual labels that were free from the noisy labels. The obtained accuracy of catching the culprit was almost 70% and in the worst case its 50%.



But the Result obtained were wrong so,



**Fig7: Flow chart of the**

**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 | 1345 | 10 (134) | M1 |  | 46 | 68 |  |  |  |  |  |
| M2 |  |  |  |  |  |  |  |  |
| M3 |  |  |  |  |  |  |  |  |
| M4 |  |  |  |  |  |  |  |  |
| M5 |  |  |  |  |  |  |  |  |
| M6 |  |  |  |  |  |  |  |  |
| 20 (268) | M1 |  |  |  |  |  |  |  |  |
| M2 |  |  |  |  |  |  |  |  |
| M3 |  |  |  |  |  |  |  |  |
| M4 |  |  |  |  |  |  |  |  |
| M5 |  |  |  |  |  |  |  |  |
| M6 |  |  |  |  |  |  |  |  |
| 30 (268) | M1 |  |  |  |  |  |  |  |  |
| M2 |  |  |  |  |  |  |  |  |
| M3 |  |  |  |  |  |  |  |  |
| M4 |  |  |  |  |  |  |  |  |
| M5 |  |  |  |  |  |  |  |  |
| M6 |  |  |  |  |  |  |  |  |
| 40 (268) | M1 |  |  |  |  |  |  |  |  |
| M2 |  |  |  |  |  |  |  |  |
| M3 |  |  |  |  |  |  |  |  |
| M4 |  |  |  |  |  |  |  |  |
| M5 |  |  |  |  |  |  |  |  |
| M6 |  |  |  |  |  |  |  |  |
| Mushroom | 2500 | 10 (250) | M1 |  |  |  |  |  |  |  |  |
| M2 |  |  |  |  |  |  |  |  |
| M3 |  |  |  |  |  |  |  |  |
| M4 |  |  |  |  |  |  |  |  |
| M5 |  |  |  |  |  |  |  |  |
| M6 |  |  |  |  |  |  |  |  |
| 20 (268) | M1 |  |  |  |  |  |  |  |  |
| M2 |  |  |  |  |  |  |  |  |
| M3 |  |  |  |  |  |  |  |  |
| M4 |  |  |  |  |  |  |  |  |
| M5 |  |  |  |  |  |  |  |  |
| M6 |  |  |  |  |  |  |  |  |
| 30 (268) | M1 |  |  |  |  |  |  |  |  |
| M2 |  |  |  |  |  |  |  |  |
| M3 |  |  |  |  |  |  |  |  |
| M4 |  |  |  |  |  |  |  |  |
| M5 |  |  |  |  |  |  |  |  |
| M6 |  |  |  |  |  |  |  |  |
| 40 (268) | M1 |  |  |  |  |  |  |  |  |
| M2 |  |  |  |  |  |  |  |  |
| M3 |  |  |  |  |  |  |  |  |
| M4 |  |  |  |  |  |  |  |  |
| M5 |  |  |  |  |  |  |  |  |
| M6 |  |  |  |  |  |  |  |  |

**Table 4:** **Overall Global performance analysis with respect to Noisy dataset**

# 5.Conclusion

Adaboost is an efficient classifier which makes use of weak classifier to do predictions. It can be efficiently used to remove noisy class labels from the dataset by making use of weight assignment concept. Thus, an efficient pre-processing technique can be built up making use of adaboost algorithm.

# 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)