Reduction Strategies to Tackle Class Imbalance in Datasets

by Cv Krishnaveni

Submission date: 27-Jul-2021 04:57PM (UTC+0530)
Submission ID: 1624654303
File name: RSCI_Thesis_26_07_2021_chapters.pdf (1.03M)
Word count: 37434
Character count: 178342

Reduction Strategies to Tackle Class Imbalance in Datasets

A Thesis Submitted in Partial Fulfillment of the

Requirements for the Award of Degree of

Doctor of Philosophy

in

Computer Science

by

C.V. Krishnaveni

Reg. No. 10MCPC19

School of Computer and Information Sciences

University of Hyderabad,

(P. O.) Central University, **GachiBowli, Hyderabad – 500 046, India.**

July 28, 2021

Dedicated to my Husband Sri Tatrakallu Madhusudan (Late).

CERTIFICATE

This is to certify that the thesis entitled **Reduction Strategies to Tackle Class Imbalance in Datasets** submitted by **C.V.Krishnaveni** Reg. No. **10MCPC19**, in partial fulfillment of the requirements for award of **Doctor of Philosophy** in **Computer Science** is a bonafide work carried out by him under our supervision.

This thesis is free from plagiarism and has not been submitted previously in part or in full to this or any other University or Institution for award of any degree or diploma. Parts of this thesis have been presented in the following conferences/journals/patents

1. C.V. KrishnaVeni and T. Sobha Rani, 2011, "On the classification of imbalanced datasets", ISSN : 0976-8491(Online), ISSN : 2229-4333(Print), IJCST Vol. 2, SP 1, pp. 145-148., December 2011.

2. C. V. Krishna Veni and T. S. Rani, "Ensemble based classification using small training sets : A novel approach," 2014 IEEE Symposium on Computational Intelligence in Ensemble Learning (CIEL), Orlando, FL, USA, 2014, pp. 1-8, doi: 10.1109/CIEL.2014.7015738.

3. Chennuru V.K., Timmappareddy S.R, " MahalCUSFilter: A Hybrid Undersampling Method to Improve the Minority Classification Rate of Imbalanced Datasets. Mining Intelligence and Knowledge Exploration". MIKE 2017. Lecture Notes in Computer Science, vol 10682. Springer, Cham. https://doi.org/10.1007/978-3-319-71928-3_5.

4. C. V. K. Veni and T. S. Rani, "Quartiles based UnderSampling(QUS): A Simple and Novel Method to increase the Classification rate of positives in Imbalanced Datasets," 2017 Ninth International Conference on Advances in Pattern Recognition (ICAPR), Bangalore, 2017, pp. 1-6, doi: 10.1109/ICAPR.2017.8593202.

5. Krishna Veni, C.V. and Sobha Rani, T. "Classification of Imbalanced data sets using tiny training sets generated by employing the concept of Centroid based Grouping

(CBG).” In 2018 International Conference on Machine Learning and Data Science (ICMLDS 2018). (pp. 44-51) IEEE. ISBN: 978-1-7281-0345-7

6. Chennuru, V.K., Timmappareddy, S.R. ”Simulated annealing based undersampling (SAUS): a hybrid multi-objective optimization method to tackle class imbalance”. Appl Intell (2021). https://doi.org/10.1007/s10489-021-02369-4.

Subjects passed for fulfillment of the course work

Sl.No	Code	Name of the Subject	Result
1	CS801	Data structures & Algorithms	Pass
2	CS802	Operating System and Programming	Pass
3	AI875	Trends in Soft Computing	Pass
4	AI879	Data Mining	Pass

Dr. T. Sobha Rani
Supervisor
School of CIS,
University of Hyderabad.

Prof. Chakravarthy Bhagvati
Dean
School of CIS,
University of Hyderabad.

DECLARATION

I, C.V. Krishnaveni Reg. No. 10MCPC19 hereby declare that this thesis entitled **Reduction Strategies to Tackle Class Imbalance in Datasets** submitted by me under the supervision of Dr. T. Sobha Rani School of Computer and Information Sciences, University of Hyderabad is a bonafide research work. I also declare that it has not been submitted previously in part or in full to this University or any other University or Institution for the award of any degree or diploma.

Date:

C.V. Krishnaveni
10MCPC19

Signature of the Student

// Countersigned //

Signature of the Supervisor

Acknowledgements

I would like to express my gratitude to my supervisor Dr. T. Sobha Rani for her valuable suggestions and guidance throughout the research work. Thanks for listening to me whenever I was in trouble and thanks for her patience when discussing issues. I can never forget her support and help in this research work. Without her ongoing professional support, this thesis would not have been possible. I am extremely grateful to Prof.Chakravarthy Bhagvati, Dean of SCIS, for providing excellent computing facilities and a disciplined atmosphere for doing my research.

I would like also to take this opportunity to thank my doctoral review committee members prof. S. Bapi Raju, Prof. S.K. Udgata, Prof. S. Durga Bhavani for their insightful comments, support and advice during periodical assessments.

I would like to thank Prof. Arun Agarwal, Prof. Hrushikesha Mohanty, Prof. K. Narayana Murthy, Prof. C.R.Rao, Prof. Atul Negi, Prof. Rajeev Wankar, Prof. Alok Singh, Prof. Vineet, C. P. Nair and other teaching and non-teaching staff of SCIS.

I would like to thank M.Tech Students, co - researchers in UoH and colleagues who supported me with their moral support.

I am indebited to my daughter Sai Sudharshini, son Makarand, parents, siblings for sharing the best and the worst moments of my life. Finally, my special thanks to friends, colleagues, exclusively P. Srikala, G. Sreenivasa Rao, M. Raghava, AB Ramesh, P. Ramakrishna Babu, Siva Shankar, D. Sandhya Rani for their continuous encouragement and moral support throughout the research work.

Publications

1. C.V. KrishnaVeni and T. Sobha Rani, 2011, ”On the classification of imbalanced datasets”, ISSN : 0976-8491(Online), ISSN : 2229-4333(Print), IJCST Vol. 2, SP 1, pp. 145-148., December 2011.

2. C. V. Krishna Veni and T. S. Rani, ”Ensemble based classification using small training sets : A novel approach,” 2014 IEEE Symposium on Computational Intelligence in Ensemble Learning (CIEL), Orlando, FL, USA, 2014, pp. 1-8, doi: 10.1109/CIEL.2014.7015738.

3. Chennuru V.K., Timmappareddy S.R, ” MahalCUSFilter: A Hybrid Undersampling Method to Improve the Minority Classification Rate of Imbalanced Datasets. Mining Intelligence and Knowledge Exploration”. MIKE 2017. Lecture Notes in Computer Science, vol 10682. Springer, Cham. https://doi.org/10.1007/978-3-319-71928-3_5.

4. C.V.K. Veni and T.S. Rani, ”Quartiles based UnderSampling(QUS): A Simple and Novel Method to increase the Classification rate of positives in Imbalanced Datasets,” 2017 Ninth International Conference on Advances in Pattern Recognition (ICAPR), Bangalore, 2017, pp. 1-6, doi: 10.1109/ICAPR.2017.8593202.

5. Krishna Veni, C.V. and Sobha Rani, T. ”Classification of Imbalanced data sets using tiny training sets generated by employing the concept of Centroid based Grouping (CBG).” In 2018 International Conference on Machine Learning and Data Science (ICMLDS 2018). (pp. 44-51) IEEE. ISBN: 978-1-7281-0345-7

6. Chennuru, V.K., Timmappareddy, S.R. Simulated annealing based undersampling (SAUS): a hybrid multi-objective optimization method to tackle class imbalance. Appl Intell (2021). https://doi.org/10.1007/s10489-021-02369-4.

Abstract

Banking, retail, financial, scientific and telecommunications and various other sectors have all been using data mining technologies, for processing massive amounts of data measured in zeta bytes. While this massive amount of data is useful, datasets have to be processed effectively to perform predictive and inferential forecasts for a target population. The Class imbalance, where there are fewer instances of a class than the number of instances in other class/classes in a dataset has posed challenges to the traditional classifiers. Traditional classifiers fail to handle the imbalanced datasets due to inherent assumptions made in designing them. The distribution of classes within the dataset has a direct impact on the classifier/model performance. One of the proven practices to address this problem is to balance the classes in the training data sets. Main goals of the balancing are **increasing sensitivity**, **selecting representative samples from the majority class**, **maintaining trade-off between Majority Class and Minority Class prediction rates.**

This thesis aspires to address the inadequacies of data science models caused by class imbalance problem using data reduction strategies. In order to achieve these goals, five techniques Ensemble based Classification using Small Training sets (ECST), Centroid Based Grouping (CBG), Quartile based Under Sampling (QUS), Mahalnobis distance based Centroid based Undersamplig with Filter (MahalCUSFilter) and Simulated Annealing based Under Sampling (SAUS) are proposed here. ECST focuses on getting good sensitivity by generating small balanced training sets and using ensemble classification to produce outcomes specified in the goals. CBG generates prototypes(artificial samples) from the original training set and uses the Lp distance metric to classify test set samples in order to account for neighbourhood space. The QUS algorithm groups each negative instance with one of the five quartiles. This is how negative samples from the full negative training distribution are selected to build a balanced training set with minimal information

loss. MahalCUSFilter creates a balanced training set; the approach's originality is that it focuses on variable dependency and scale invariant characteristics, both of which are critical in multivariate dataset classification. Finally, Simulated Annealing, a metaheuristic works on selecting the best balanced training set among a large number of possible balanced training sets from an imbalanced one by selecting a set with a low Balanced Error-Rate, which is used as a cost function in each iteration. The proposed approaches in this thesis are all Reduction Strategies that effectively address the problem of class imbalance and have been empirically proven to work on par with, and in some cases better than, existing methods.

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

IDS	Imbalanced Data Sets
ECST	Ensemble based Classification using Small Training sets
CBG	Centroid Based Grouping
CUS	Centroid based Under Sampling
QUS	Quartiles based Under Sampling
SAUS	Simulated Annealing based Under Sampling
SMOTE	Synthetic Minority Oversampling TEchnique

Chapter 1

Introduction

Data has grown at a breakneck pace over the past decade. This vast amount of data offers tremendous value. Hence, it is required to increase the speed of data processing in order to produce information quickly. While this massive amount of data is useful, datasets cannot be processed effectively unless meaning can be accurately extracted from it. Banking, retail, and telecommunications have all embraced data mining technologies, which are considered the technologies of choice for processing massive amounts of data measured in zetta bytes. Some data analytics needs to go through several layers of analysis before a dataset can be moved into a database for further use by analysts in the organisation who then use it to perform predictive and inferential forecasts for a target population. In many real-world application domains, classification, a supervised machine learning algorithm has aided data analysis and prediction. When learning from imbalanced data distribution schemes, however, learning algorithms have difficulty assigning correct labels to instances, which is known as the 'class imbalance problem.'

1.1 Supervised Learning

Insights obtained from existing labelled data is used to categorize the new data in machine learning is known as Supervised Learning. That is, using labelled dataset to label(class) unlabelled data is termed as Supervised Learning. The set of labelled instances is called training set and the set of instances to which labels are to be found is called the test set. The domain, the set of possible values of an attributes can be discrete or real-valued.

If the domain of class label is real-valued, the supervised learning algorithm is considered as Regression, and if the range of class label is discrete-valued, the supervised learning algorithm is called as Classification [62, 125, 96, 66].

To obtain classification using a classifier, the given dataset is usually divided into three parts, namely training set, validation set and test set. Validation Set is used to tune the parameters of the classifier to get optimal performance and is applied to label the instances in the test set. Instead of taking separate validation set, k-fold cross validation can be applied on the training set to build a classifier. The resultant classifier is used to label the test set.

There exist several popular classifiers like Decision Trees, Naive Bayes, Lazy learners, Neural Networks etc, which learn the properties of the known data set and apply the knowledge to predict the class label of the test(unseen) data. In order to increase the classification accuracy, ensemble based classification approaches like Boosting, Bagging etc, are also proposed in the literature.

1.1.1 Models for Classification

Classification Algorithms work in two ways and are categorised as Model Based Classifiers (MBC) and Instance Based Classifiers (IBC). Model Based Classfiers build a model from the training set. Uses that model to label the test set instances. These type of classifiers are called Eager-Learners. Instance Based classifiers use the training set instances to label the test set instances. These are called Lazy-Learners. In either case, the performance of the classifier depends on the characteristics of the training set. The performance of the classifier is described by “How well the classifier labels the unseen instances correctly”. That is the number of instances in the test set that are classified correctly.

1.1.2 Impact of Data Characteristcs

The characteristics of the training set has tremendous impact on the performance of the classifier. To mention a few are size of the training set, lack of data for one class and plenty of data for another class, the curse of dimensionality, amount of overlap of

instances belonging to different classes, the amount of class separability, proportion of instances lying on and around the class boundary, density of data, noise in the data etc.

1.2 Imbalanced Datasets

In certain applications like credit card fraudulent transactions, rare disease diagnosis, spam filtering etc, number of instances available on fraudulent transactions, rare diseases, spam are much less than that of non-fraudulent transactions, non-rare disease, non-spam instances. These kind of datasets which contain very few instances of one category and many more instances of another category are termed as *Imbalanced Datasets*. Imbalance in class distributions is quite common in many real-world applications. Datasets having unequal class distributions (imbalanced datasets) require to be handled differently compared to the datasets with equal class distributions.

1.2.1 An Example

Consider the example of an automated inspection system which monitors products for defects in the products that come off a manufacturing assembly plant. It may find that the number of defective products is significantly fewer than that of non-defective products. This is a typical example of an imbalanced set. In any of such imbalanced sets, there is a disproportionate number of instances that belong to different classes. Sets with only two classes are known as binary class datasets. The class with less number of instances is designated as Positive/Minority class and the class with more number of instances is designated as Negative/Majority class. The degree of imbalance varies from one application to another. For example, a manufacturing plant operating under the six sigma principle may discover four defects in a million products shipped to their customers, while the amount of credit card frauds may be of the order of 1 in 100. Despite their infrequent occurrences, a correct classification of the rare class in these applications often has greater value than a correct classification of the majority class [125].

1.2.2 Issues with Imbalanced data

In order to illustrate the impact of misclassification of the minority samples on the performance of the classifier, consider an example of cancer disease from medical diagno-sis [83]. If the patient has cancer, then the tests show positive, it is treated as positive class. If the patient is not suffering from cancer then the test result gives negative, it is treated as negative class. If the results are correct, showing positive for cancer patient and negative for non-cancer patient means they are not misclassified. If the penalty is to be assigned to the misclassification of True Positives (TP), True Negatives (TN), that is, for correct classification there is no penalty. But giving wrong results, negative for cancer patient, False Negative (FN), positive for non-cancer patient is misclassification. It is to be noted that these both misclassifications cannot be treated as the same. False Negative may result in patient's death due to the delaly in taking treatment for cancer which is more serious than False Positive where the patient may go for another test to confirm and/or may take more care about his/her health. The cost of FN will be more than the cost of FP and the costs of TP and TN is zero [104]. It is clear that misclassification costs are not equal. They are unequal and the impact or magnitude of the cost depends upon the application and situation.

1.2.3 Traditional classifiers for Imbalanced Sets

Many research papers on imbalanced data sets commonly agree that because of this unequal class distribution, the performance of the existing classifiers tend to be biased towards the majority class. The reasons for poor performance of the existing classification algorithms on imbalanced data sets are :

- They are accuracy driven, that is, their goal is to minimize the overall error to which the minority class contributes very little.
- They assume that there is equal distribution of data for all the classes.
- They also assume that the errors coming from different classes have the same cost. [74, 131].

1.2.4 Necessity of Handling Imbalance

Classification of imbalanced datasets is fraught with several issues. In spite of that, still there is a need to handle this problem, as there are several real life situations which can result in generating an imbalanced data set. Learning is supposed to become better with more number of samples in general. But, in practice, the number of training examples used for learning may get reduced due to the costs associated with procuring the rare samples, infrequent occurrence of rare events as in imbalanced data sets. Hence, there is a need to devise methods to handle these kinds of data sets.

1.3 Techniques to handle Imbalanced Data

In literature various methods have been proposed to deal with the imbalance problem in the datasets. Cost-sensitive learning, ensemble methods are also widely implemented to address this problem. At algorithmic level, thresholds and parameters in the algorithm are adjusted in classification methods to handle the imbalance. The data set is processed at the data level to ensure that the distribution of classes is balanced.

1.3.1 Cost Sensitive Learning

In the case of Cost-Sensitive learning, a cost matrix with unequal costs, more penalty for false negatives and low penalty for false positives is used. Ensemble learning methods use subsets of the samples of the data set and several classifiers to improve the classification rates of imbalanced data sets.

1.3.2 Data Level handling

For addressing imbalanced data sets, data level sampling approaches are divided into two categories: (i) oversampling methods, and (ii) undersampling approaches.

1.3.2.1 Oversampling

Oversampling is a data preparation technique for balancing a data set by reproducing minority class examples. Upsampling is another term for it. This method has the advantage

of not causing data loss, as undersampling does. If the data set is already relatively large yet imbalanced, oversampling suffers from the disadvantage of causing overfitting and adding to the computing cost.

1.3.2.2 Undersampling

Undersampling is another flagship data pre-processing method that removes examples from the majority class to balance the data. This method is usually suitable for large scale applications, where the number of majority class examples is vast and reducing the training samples brings down the training time and storage. The drawback of undersampling method is that it may discard potentially useful information that could be important for classifiers. To alleviate imbalance in the datasets, undersampling methods reduce the size of the majority class samples in the following ways:
In Figure 1.1, Taxonomy of Class Imbalance is mentioned.

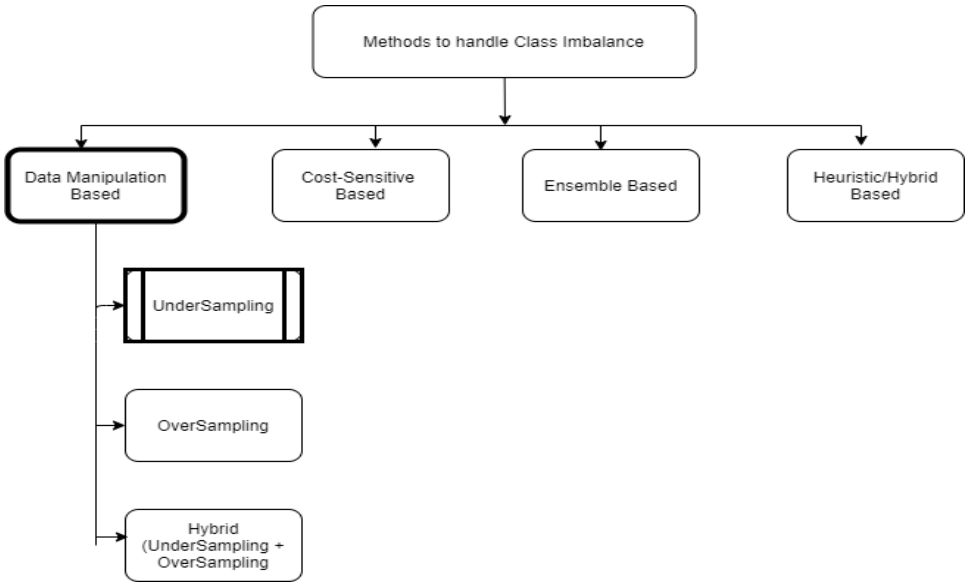

Figure 1.1: Class Imbalance Taxonomy.

1.4 Research Challenges in Class Imbalance Problem

The research challenges drawn from various popular and latest research papers related to imbalanced classification problem are discussed here.

1.4.1 Size of the Dataset

Determining required size of the dataset for training is one of the issues that needs to be tackled to obtain better classification rates. Oversampling and Undersampling are effective methods of dealing with the problem of imbalanced data sets classification. However, undersampling(downsizing) approach works better than the oversampling methods on large domains [65]. Oversampling appears to be best for small data sets [133]. Liu et al. [85], conducted experiments and show that oversampling clearly appears to be better than undersampling for local classifiers whereas some undersampling strategies outperform oversampling when employing classifiers using global learning.

Classifiers produced by sampling and using cost sensitive matrix performance are found to perform similarly [90]. By focusing exclusively on data sets with more than 10,000 examples, Weiss et al. [134] found that cost sensitive learning algorithm consistently outperforms the sampling methods. It should be noted that their focus was on using the cost information to improve the performance on the minority class. The drawbacks of this method are in deciding the cost for minority and majority classes miscalssification.

1.4.2 Class Distribution

Impact of class distribution on classification is one more factor that requires the attention of the researchers.

In situations, where the availability of minority samples is restricted, if only *equal* training examples can be selected from majority class, training will be less biased. In what proportion should the classes be represented is the question. It is shown that the naturally occurring class distribution generally performs well when classifier performance is evaluated using undifferentiated error rate. When the area under the ROC is used to evaluate classifier performance, a balanced distribution is shown to perform well. Since, neither of these choices for class distribution always generates the best performing classifier, a budget-sensitive progressive sampling algorithm is introduced for selecting training examples based on the class associated with each example.

[86] presents an empirical study which discloses that when the misclassification costs are equal, cost sensitive classifiers favour natural class distribution. When mis-classification

costs are unequal, a balanced class distribution is more favourable. Weiss and Provost [134] discuss the effect of class distribution on tree induction. They surmise that for any fixed class distribution, increasing the size of the training set always leads to improved classifier performance. The choice of class distribution may become less important as the training set size grows. But, in practice, the number of training examples used for learning will be limited due to the costs associated with procuring, preprocessing and storing the training samples and the computational costs associated with learning from them.

1.5 Problem Statement

Data Science and its constituent functional component Data Mining strive for establishment of a robust pipeline for converting real-world data generated by the business processes into actionable items. The overall aim of these notions is to evolve methodologies that can uncover salient patterns from the data and tweaking them towards a business interest. The pipeline involves data collection, cleansing, application specific algorithm selection, model implementation on the training data and evaluation using test data. However, the latter two stages are feature sensitive and demand lot of time for obtaining the expected qualitative results.

The evaluation of a data mining model is a well-addressed aspect with the support of statistical measures that can conduct meaningful trade-offs over the feature set. Bias-Variance, sensitivity-specificity, evident-hidden pattern dependencies, and tractability-complexity of the algorithm are few examples for the trade-offs. Preceding to this high level analysis, we shall understand the randomness present in the sample data compared to the test data. Thus, the size of the dataset also has a direct impact on the model performance. One of the proven practices to address this problem is to balance the training and test data sets. Important improvements that are desirable in the classification of imbalance data sets are:

- Sensitivity enhancement.
- Handling Information loss by selecting representative samples from the majority class.
- Maintaining trade-off between majority class and minority class prediction rates.

1.6 Objective

This thesis aspires to address the deficiencies of data science models caused by class imbalance problem by isolating the finer statistical issues that hamper the performance of the selected model. The finer and detailed understanding of the patterns and expressing the statistical significance and their influence on the model adds an overall advantage to the classification algorithm. Main goals of this work are:

- Less Information loss
- Computational ease
- Parameter independence
- Low Balanced Error rate, (1-$\frac{Sensitivity+Specificity}{2}$)

From the perspective of challenges discussed above, attempt is made in this work to balance the imbalanced data to improve classification accuracy. The main focus of the work is on undersampling.

Existing under sampling methods to balance the imbalanced data set either apply nearest neighbour methods or sample based methods.

- **Nearest Neighbour Methods**

  Prototype selection methods mainly are using k-NN (k-Nearest Neighbour) methods to pick the samples for training. Drawback of these methods is the complexity involved in choosing the majority class samples. It is high since selection is done based on the distances of k nearest neighbors, that is, distances of every majority sample with k nearest neighbors are to be computed, which is an arduous task. Complexity and time consumption of the method increases with the increase in number of instances or number of attributes of the dataset.

- **Sampling Based Methods**

  Clustering mechanisms are employed to get the training dataset. Once clusters are formed, this method is simple to implement but to form clusters several issues are to be addressed viz., i) Which clustering algorithm is to be used? This decision depends mostly on the size, dimension and type of the dataset. ii) How many clusters are to

be formed? This can be decided by using cluster validity indices. Again in those, if external cluster validity indices are chosen, parameters are to be supplied by the user that is, again quality of the cluster may vary depending upon the parameters. Even, if internal cluster validity indices are used, which is appropriate and why are to be known.

In order to overcome these drawbacks, methods are proposed in this work to enhance the classification accuracy of imbalanced data sets. They are ensemble based classification, MDSG, MahalCUSFilter, Quartile based undersampling, and similated annealing based methods. These methods are discussed in detail in the following chapters.

1.7 Contributions

1. Proposed an Ensemble based Classification Method using small training sets(ECST), which considers the following three point in order to get acceptable classification accuracy with small training sets.
    - Only one-third (30%) of the data set is used to represent the whole data set.
    - Using these small training sets to improve classification accuracy.
    - Considered variations in tiny training sets, such as noise or outliers.
2. Proposed a method, Centroid Based Grouping (CBG) which generates prototypes, that is synthetic samples representing the original training set and used fractional distance measure with kNN Classifier to classify test set. The points taken into consideration in this method are:
    - To make use of all the instances in the original training set without discarding even a single instance.
    - To generate a resilient training set, the mean of the group's instances is picked from each group to serve as a representative of that group. Since any noise can be removed in the process of finding the mean of the group occurrences.
    - To prepare a very tiny training set with samples that act as representatives of their respective classes.

3. Proposed a method, which selects majority class samples based on Quartiles distribution. The points taken into consideration in this method are:

    - To pick samples from majority class which spans throughout the distribution.
    - Used normalized Euclidean distance measure as the attributes of the datasets have variance.
    - This also eliminates the the issue about the number of clusters and the cluster centers to be chosen.

4. Proposed a method, MahalCUSFilter, works with the intuition that the real-world datasets are multi-variate in nature and while considering the similarity of instances with a group, that is, centroid in this research work needs to consider inter-dependencies of the attributes. Hence used Mahalanobis distance instead of Euclidean distance measure. This method handles the following issues in detail.

    - Parameter Dependence: The performance of MahalCUSFilter is independent of the settings chosen by the user, unlike cluster-based and kNN-based undersampling approaches, which are dependent on the clustering algorithm, number of clusters, and other factors.
    - Variables inter dependence: Unlike other algorithms that use Euclidean distance to find distance/similarity between instances which do not consider inter dependencies, correlations among the variables of a dataset, MahalCUSFilter uses Mahalanobis distance measure to find distance between each majority class instance with its centroid (Mean of the majority class instances) which takes into account correlation among variables of a dataset.
    - Information loss: The issue of majority class representation is handled by using a stratified sampling approach, which selects the number of samples from each group based on its size, ensuring that the samples picked are representative of the majority class as a whole.
    - Scale variant: A dataset's variables are measured in different units and have a diverse range of values. Existing algorithms, on the other hand, use Euclidean distance estimates that ignore these issues. To address this problem, the sug-

gested method employs the Mahalanobis distance, which renders the method scale-invariant.

5. Proposed a meta-heuristic method by employing Simulated Annealing to select optimal balanced sets among several possible balanced sets that can be formed from an Imbalanced Data Set. To develop this Undersampling method which is based on Simulated Annealing the following issues are taken into consideration.

   - Balanced set chosen in each iteration should have minimum Balanced Error Rate.
   - Nearest Neighbour of only misclassified majority class instance is found here unlike many popular undersampling methods which find nearest neighbours of all the samples.
   - Simulated annealing, unlike many other optimization methods such as genetic algorithms, gradient descent, hill climbing, and so on, avoids getting stuck in a local optimum.

1.8 Organization of the thesis

The thesis is organized as follows:

- chapter 2: Literature survey
- chapter 3: Ensemble of Small Training sets for Classification (ECST)
- chapter 4: Prototype generation employing the Centroid Based Grouping (CBG)
- chapter 5: Quartiles based UnderSampling(QUS)
- chapter 6: MahalCUSFilter: A Hybrid Undersampling method
- chapter 7: Hybrid Multi Objective Optimization Method (SAUS)
- chapter 8: Conclusions and future directions

Next chapter discusses about the methods proposed in the literature to handle the imbalanced data set classification.

Chapter 2

Related Work

In the real world, there are cases where the class distributions are unequal, such as oil spills observed by satellites as photographs, fraudulent credit card transactions, detection of rare diseases, and so on. In these circumstances, rare (minority) samples are low in number compared to common/normal (majority) samples. That instance, when it comes to credit card transactions, fraudulent transactions are significantly less common than regular ones. The term "*Imbalanced data set*" refers to a data set with certain characteristics. In such instances, calculating the total classification rate of the test set regardless of the class distribution would result in higher accuracy, even if all of the minority (positives) samples are misclassified. On data sets with no fatalities, this may not have a significant impact. However, in rare disease prediction, misclassifying a positive (disease) as a negative (non-disease) is deemed lethal, because it is presumed that the patient is not suffering from the disease, and hence inadequate care is not provided, and the disease may deteriorate.

To deal with such imbalances, different approaches at the data and algorithmic levels have been proposed to limit the influence of imbalance on the classification of minority instances. Major purpose of this research work is to attain a low Balanced Error rate, (1-$\frac{Sensitivity+Specificity}{2}$), by balancing the training set taken from an imbalanced data set. In doing so, the probability of an error due to misclassification can be reduced.

2.1 Methods to Handle Imbalance

In the literature, several approaches have been proposed to handle the imbalanced data sets and for the classification of imbalanced data sets. [65, 66, 99, 13, 16, 29, 131, 97, 55, 121, 104, 58, 110, 17, 74, 67, 76, 69, 109] provide a very good survey on the classification of imbalanced data sets and on various methods which can handle the imbalance problem. The latest papers [8, 75] provide a very good review of learning from class imbalance.

The techniques to handle class imbalance are mainly categorized into four classes: Data level handling techniques, Algorithmic level techniques, Cost sensitive learning methods and Ensemble methods. Methods at the data level seek to balance class distributions. Oversampling methods and Undersampling techniques are two types of data level strategies for dealing with imbalanced data sets, according to Barandela [13]. To deal with the imbalance, algorithmic level techniques strive to adjust the thresholds and parameters in classification algorithms, according to Batista [16]. According to Haibo [58], cost-sensitive learning assumes an unbalanced cost matrix with a high penalty for false negatives and a low penalty for false positives. To improve the categorization of imbalanced data sets, ensemble learning approaches use subsets of the samples of the data set and different classifiers, according to Galar [50].

2.2 Data Level Handling Techniques

To handle the problem of imbalanced data, sampling approaches are applied on the data to change the class distribution of data and make it balanced. Sampling approaches are mainly divided into two categories: Undersampling and Oversampling.

2.2.1 Undersampling

This technique removes examples from the majority class to make the data set balanced. This method is suitable for large scale applications, where the number of majority class examples is very large and reducing the training samples reduces the training time and storage required. Drawback of undersampling method is that it discards potentially useful information that could be important for classifers [74, 104].

It can be surmised that most of the methods in undersampling deal with either Exhaustive Search based approaches, Sampling based approaches or a combination of these two approaches. Undersampling methods can also be divided into Random Undersampling and Informative Undersampling. Random undersampling, removes majority instances randomly till the data set gets balanced. Because of this there is loss of useful information. Informative undersampling, chooses or discards certain majority instances based on a prespecified selection criterion to make the data set balanced. Many solutions are proposed based on informative undersampling. Informative Undersampling can be passive or active. Passive selection methods are proposed as preprocessing techniques for selecting informative samples for a classifer. In Active selection methods, informative samples are queried during the construction process of the classifier [92].

To mention briefly, popular undersampling methods include CNN, CNNTL, NCL, OSS etc. SMOTE is one the most extensively used oversampling method. Adacost, a cost-sensitive method, SMOTEBoost, AdaBoost etc. come under Ensemble based methods.

Kubat and Matwin [14] presented One Sided Selection(OSS) which is an undersampling method. OSS only removes examples from the majority class while leaving the examples from the minority class untouched. They divided majority(negative) class examples into four groups like class-label noise, Borderline examples, redundant and safe examples. The OSS algorithm works as follows: first the number of redundant negatives is reduced by creating the subset C, consistent with the training set. By definition, C, a subset of S is consistent with S, if when used by the 1-NN rule, it correctly classifies examples in S. Then the system removes those negative examples that participate at Tomek links. Borderline examples and examples suffering from the class-label noise participate at Tomek links. So, they are eliminated.

[147] describe an application of a simple kNN approach to an imbalanced data classification problem. They empirically studied the effects of undersampling on the k nearest neighbour kNN approach and five different methods of choosing negative training examples, Random Selection, selection of NearMiss examples which is done in three ways NearMiss-1, NearMiss-2, NearMiss-3 and selection of most distant examples. The NearMiss-1 selects negative examples that are close to some of the positive examples, they select negative examples whose average distances to three closest positive examples are the smallest. The NearMiss-2 selects negative examples that are close to all positive

examples. In this method, examples are selected based on their average distances to three farthest positive examples. In NearMiss-3, given number of closest negative examples for each positive example are chosen. In Selection of most distant negative examples, the negative examples whose average distances to the three closest positive examples are the farthest are chosen. They found through experiments that both kNN and C5.0 are sensitive to the percentage of negative examples selected and among the five negative example selection methods random and NearMiss-2 methods performed the best.

[92] proposed a Majority Filter-based Minority Prediction(MFMP) approach for imbalanced data sets. The goal of this approach is to achieve good prediction over minority class by avoiding unnecessary information loss from the majority class. The MFMP adopts an unsupervised learning technique for selecting samples for supervised learning. The approach works in two steps: in the first step, minority samples are clustered and majority class samples that are out of minority classification regions are identified. This improves minority prediction rate, in the second step, majority samples are randomly selected in individual clusters and this enhances majority prediction rate. Experimentally, they studied the behaviour of MFMP approach and found that it outperforms the traditional random under-sampling approach. In addition to [78, 147, 92], several other undersampling approaches are available in the literature.

2.2.1.1 Popular Undersampling Techniques

Condensed Nearest Neighbor(CNN) Rule [57], the Condensed Nearest Neighbor Rule with Tomek Link (CNNTL) [16], Neighborhood Cleaning Rule (NCL) [80], One Sided Selection(OSS) [78], Tomek Link [128] etc are widely used undersampling techniques. They select majority class samples based on their distance from minority class samples using kNN classifier.

Condensed Nearest Neighbor (CNN) [57] initially places all minority class samples in D, and randomly chooses one majority class sample in 'S' from 'D'. Then 1-NN is used to classify the samples from D with respect to contents of 'S' and every misclassified sample is moved from 'D' to 'S'. The idea behind CNN method is to eliminate the majority class samples that are distant from the decision border as they are considered to be less relevant for learning.

Tomek links [128] can be used as data cleaning method which eliminates noisy and

borderline majority class samples only and not minority class samples. Consider two instances y_i and y_j that belong to different classes and are separated by a distance $d_{(i,j)}$. A pair (y_i, y_j) is a Tomek Link, if there is no sample y_l such that $d_{(i,l)} < d_{(i,j)}$ or $d_{(j,l)} < d_{(i,j)}$.

One Sided Selection(OSS) [78] applies Tomek Links followed by CNN. This method retains all the 'safe' (which do not participate in Tomek Link that is other than borderline) majority class samples and all minority class samples in the data set.

CNNTL [16] is another method similar to OSS but applies TL after CNN as TL is computationally expensive. First condensed set is formed using CNN and then TL is applied on the reduced set.

Neighborhood Cleaning rule (NCL) [80] uses Wilson's Edited Nearest Neighbor rule [136] to remove majority class samples. ENN eliminates a sample whose class label differs from the class of at least two of it's three nearest neighbors. For a two class problem NCL uses ENN in the following way: For each sample x_i in the given training set, its three nearest neighbors are found. If x_i belongs to majority class and is misclassifed by three of its nearest neighbors then x_i is removed. If x_i belongs to minority class and is misclassified by its three nearest neighbors then the three nearest neighbors which belong to majority class are removed.

Class Purity Maximization(CPM) [145] finds a pair of minority and majority samples as centers. Using these centers, it partitions all the instances into two clusters C_1 and C_2 according to their nearest centers and this process is repeated till at least one subset has class impurity less than its parent's impurity. A training set is constructed by adding all minority instances to each non-pure cluster.

Yen and Lee in [142] proposed a **cluster based undersampling techniques (SBC)**. In their approach, they cluster the entire data set and the number of majority samples to be chosen is determined by the number of minority samples in that cluster. Along with SBC, they proposed five methods namely: sampling based on clustering with NearMiss-1(SBCNM-1), sampling based on clustering with NearMiss-2 (SBCNM-2), sampling based on clustering with NearMiss-3(SBCNM-3), sampling based on clustering with Most Distance(SBCMD) and sampling based on clustering with most far(SBCMF).

Rushi et al. [87] proposed a method wherein majority class samples are clustered into 'k' clusters and select $R_i \times$size(Minority Class) number of samples from each cluster

so that the total number of selected majority samples equals the size of the minority set to balance the training set. R_i=Majority samples in $Cluster_i/Total\ majority\ samples$, $1 \leq i \leq k$ represents the number of majority class samples to be chosen is based on the ratio of the number of majority samples in each cluster to the total number of majority samples. The number of majority class samples to be chosen from ith cluster is $S_i = Total\ minority\ samples \times R_i$, $1 \leq i \leq k$.

In [117], a cluster based undersampling along with an ensemble learning is proposed. Here majority instances are clustered into k clusters where $1 \leq i \leq$ size of the minority class and $size(Minority\ Class)/k$ number of samples are selected from each cluster so that majority class samples are equal to the number of minority samples. m classifiers are trained using training sets created as described and the final result is obtained by weighted majority voting, where weight of each classifier is taken as the inverse of its error on the whole training set.

In [108], majority class samples are clustered into k clusters and k training sets are formed with each of the majority class clusters combined with all the minority class samples. Training set that gives the highest accuracy is chosen as the final training set in classification.

2.2.1.2 Latest Work on Handling Class Imbalance

Papers on imbalanced data sets [132, 148, 19, 103, 91, 14] use other types of data handling. Wang et al. [132] employ an ensemble in addition to weights and information about sample misclassification to classify imbalanced data. Zhang et al. [148] made a study of imbalanced data sets of variable imbalance ratio, size and complexity using three classifiers Naive Bayes, c4.5 and SVM. They have concluded that SVM outperforms the other two classifiers. Other cluster based methods are a cluster based one sided selection method [14], a hierarchical decomposition method based on similarity [19], diversified sensitivity-based method [103], ensembles of First Order logical Decision Trees [91], feature weighting to deal with overlap in imbalanced datasets [9], a RandomBalance method that uses ensembles of variable priors classifiers [41], ensemble method [122]. In [3], Abualigah et al. proposed Feature selection an enhanced Krill Herd algorithm for text documents.

Recently, in [102], data balancing method using neighbourhood sampling in bagging is proposed. Jinyan et al. proposed an adaptive multi-objective swarm fusion for imbalanced

data classification in [82]. Another latest work is by Fernandez et al. [48] wherein the relationship between F1 and accuracy metrics are used for multi-objective evolutionary optimization in classification tasks.

Yitian et al. proposed Pin-MMTSM method in the paper [141] which uses SVM for classification. Pin-MMTSM method works by computing two spheres using quadratic programming problem (QPP) and a linear programming problem (LPP). Majority of the majority class samples are placed in the small sphere and the large sphere pushes out most of the minority samples by enhancing the margin between two spheres.

2.2.2 Oversampling

Like undersampling, oversampling can also divided into two types. Random Oversampling and Informative Oversampling. Random Oversampling is the method which balances the class distribution by replicating the randomly chosen minority class examples. Informative Oversampling method synthetically generates minority class examples based on a pre-specified criterion. Several modifications of SMOTE [28] such as borderline-SMOTE [62], safe-level SMOTE [23], ADASYN [58] are proposed. Wenhao et al. [140] have proposed an improved oversampling algorithm. They extracted the support vectors based on Random-SMOTE algorithm and used them as the parent samples to synthesize new minority class samples to balance the data.

2.3 Cost Sensitive Learning

Cost Sensitive Learning(CSL) is another commonly used approach to handle the classification problem of imbalanced data sets. It is considered to be an algorithmic level solution.

In the cancer detection classification problem, given a dataset, the number of persons affected by cancer is usually far less than the number of persons not affected by it. Here, the two classes data distribution is unequal which says that it is imbalanced data set. By taking into consideration of this fact during the building of a classifier, the problem of classification of imbalanced data sets can be handled. The type of learning algorithm which takes misclassification cost into consideration is called Cost Sensitive Learning. It produces the classifier with minimum total cost. The advantage of this method is that no data is replicated or eliminated [95].

Let C(i,j) denote the cost of predicting an example of class i as class j. For a binary classification, misclassification costs can be presented using cost matrix. Corresponding to a confusion matrix Table 2.1, cost matrix provides the costs associated with the four outcomes of the confusion matrix [133]. Here i represents positive(minority) class and j represents negative(majority) class. C(i,i)=C(j,j)=0. It means, no cost(penalty) is associated with True Positives and True Negatives. C(i,j)=C(FN), C(j,i)=C(FP). Costs can be assumed to be constant or example dependent[44]. The goal of cost sensitive learning method is to choose a classifier with lowest total cost.

Total cost=C(FN)$\times$FN + C(FP)$\times$FP, where FN is the number of positive examples wrongly predicted as belonging to negative class, FP is the number of negative examples wrongly predicted as belonging to positive class. C(FN) and C(FP) correspond to the costs associated with False Negative and False Positive respectively. Obviously, C(FN) $>$ C(FP) to ensure that the positive misclassification is minimized.

There are many ways to implement cost sensitive learning. In [**?**], it is categorized into three types of techniques. First class of techniques apply misclassification costs to the data set as a form of data space weighting, second class applies cost-minimizing techniques to the combination schemes of ensemble methods, and the last class of techniques incorporate cost sensitive features directly into classification paradigms to fit the cost sensitive framework into these classifiers. Various ways to incorporate cost into classifiers are available in the literature [146, 120] to handle imbalanced data sets efficiently. Zheng et al. in [149] proposed a cost-sensitive hierarchical classification for imbalance classes.

As mentioned in chapter 1, cost sensitive learning handles imbalanced classification problem. Let us discuss here, how to incorporate cost into decision tree classification algorithm which is one of the most widely used and simple classifier. Cost can be incorporated into it in various ways [125, 104, 58, 44, 42, 84]. First way is that cost can be applied to adjust the decision threshold, second way is cost can be used in splitting attribute selection during decision tree construction and the other way is applying cost sensitive pruning schemes on the tree. [84] proposed a method for building and testing decision trees that minimizes total sum of the misclassification and test costs. The algorithm used by them chooses an splitting attribute that minimizes the total cost, the sum of the test cost and the misclassification cost rather than choosing an attribute that minimizes the entropy. Information gain, Gini measure are considered to be skew sensitive [33]. In [85] a new decision

tree algorithm called Class Confidence Proportion Decision Tree (CCPDT) is proposed which is robust and insensitive to size of classes and generates rules which are statistically significant.

[34] analytically and empirically demonstrate the strong skew insensitivity of Hellinger Distance and its advantages over popular alternative metrics. They arrived at a conclusion that for imbalanced data it is sufficient to use Hellinger trees with bagging without any sampling methods. [89] uses different operators of Genetic algorithms for oversampling to enlarge the ratio of positive samples and then apply clustering to the oversampled training data set as a data clearning method for both classes, removing the redundant or noisy samples. They used AUC as evaluation metric and found that their algorithm performed better.

2.4 Ensemble Methods

An ensemble method with information about sample misclassification along with weights is proposed by Wang et al. [132]. A cluster based one sided selection method for undersampling was proposed by Barella et al. [14]. Beyan et al. [19], proposed a similarity based hierarchical decomposition method to classify imbalanced data sets. A novel ensemble method to classify imbalanced datasets is proposed by Sun et al. [121]. [103], Wing et al. proposed a diversified sensitivity based undersampling method for imbalance classification. [132, 148, 19, 103, 91, 14] are some of the latest papers on imbalanced datasets. Some other latest works are [91] which uses ensembles of First Order Logical Decision Trees to handle Class imbalance problem, feature weighting is used to deal with overlap in imbalanced datasets in [9] and in [41], RandomBalance method for imbalanced data which uses ensembles of variable priors classifiers is proposed.

Latest papers on imbalanced data sets include [132, 148, 19, 103, 91, 14] etc. Wang et al. [132] use an ensemble method along with weights and information about sample misclassification to effectively classify imbalanced data. Zhang et al. [148] present an empirical analysis by conducting various experiments on imbalanced data sets of varying imbalance, size and complexity applying three popular classifiers Naive Bayes, c4.5 and SVM. Results have shown that SVM outperforms the other two classifers. Barella et al. in [14] proposed a cluster based one sided selection method for undersampling. In [19],

a similarity based hierarchical decomposition method is proposed to classify imbalanced data sets. Wing et al. [103] proposed a diversified sensitivity-based undersampling method for imbalance classification. Another latest works, [91] uses ensembles of First Order logical Decision Trees to handle the problem of imbalanced classification, [9] uses feature weighting to deal with overlap in imbalanced datasets, [41] proposed a RandomBalance method for imbalanced data which uses ensembles of variable priors classifiers. In [121], Sun et al. proposed a novel ensemble method to classify imbalanced data sets.

2.5 Heuristic Based Methods

The GP-COACH method incorporates genetic programming and rule-based fuzzy classification systems to better accommodate complex, high-dimensional problems. Using a context-free grammar, the GP-COACH will learn disjunctive normal form rules (the rules are stored as one rule per tree). It's a genetic cooperative-competitive learning approach with the population as the rule base. To maintain the diversity of the population, GP-COACH uses a token competition mechanism, which requires the rules to compete and cooperate with each other and obtains a compact set of fuzzy rules. Using non-parametric statistical tests, it has been shown that the results are accurate and interpretable [18]. IN [31], SNGEIP creates synthetic samples that are placed within a local area of the training samples and uses the union of original training samples and synthetic neighbourhoods samples to train the base classifiers. Yitian et al. proposed Pin-MMTSM method in the paper [141] which uses SVM for classification. Pin-MMTSM method works by computing two spheres using quadratic programming problem (QPP) and a linear programming problem (LPP). Majority of the majority class samples are placed in the small sphere and the large sphere pushes out most of the minority samples by enhancing the margin between two spheres.

2.6 Performance Metrics

Performance of traditional classification algorithms is evaluated by the metric accuracy which is defined as the percentage of examples that are correctly classified. This is not suitable when dealing with imbalanced data sets as the minority class has less number of

samples. In fact, misclassifying all minority samples and correctly classifying majority class samples gives a very good accuracy. Performance of a classifier is calculated based on the confusion matrix.

Table 2.1: Confusion Matrix

	Actual Positives	Actual Negatives
Predicted Positives	True Positives(TP)	False Positives(FP)
Predicted Negatives	False Negatives (FN)	True Negatives (TN)

Various measures used for describing the performance of the classifiers are listed below:

Sensitivity : Sensitivity is the percentage of Positives Correctly Classified. It denotes the accuracy of the positive class. Recall and True Positive Rate(TP_Rate), TPR are other names of Sensitivity.

$$Sensitivity = TP\_Rate = \ Recall = \frac{TP}{TP+FN}$$

Specificity: Sensitivity is the percentage of Positives Correctly Classified. It denotes the accuracy of the negative class True Negative Rate(TN_Rate), TNR are other names of Specificity.

$$Specificiy = TN\_Rate = \frac{TN}{TN+FP}$$

FalsePositiveRate: False Positive Rate is the percentage of negatives wrongly classified.

$$FP\_Rate = \frac{FP}{TP+FN}$$

FalseNegativeRate: False Negative Rate is the percentage of positives wrongly classified.

$$FN\_Rate = \frac{FN}{TN+FP}$$

Accuracy: The percentage of correctly classified instances.

$$Accuracy = \frac{(TP+TN)}{(TP+FN+TN+FP)}$$

Error_Rate: The Percentage of incorrectly classified instances.

$$Error\_Rate = \frac{(FP+FN)}{(TP+FN+TN+FP)}$$

Precision: Precision is the percentage of correctly classified positives.

$$Precisiion\ = \frac{TP}{TP+FP}$$

GMean: It is the geometric mean of Sensitivity and Specificity.

$$Gmean = \sqrt{Sensitivity \times Specificity}$$

F-Measure: It is the harmonic mean of Precision and Recall.

$$F\text{-}Measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Balanced Accuracy: It is the arithmetic mean of Sensitivity and Specificity.

$$Balanced\ Accuracy = \frac{Sensitivity + Specificity}{2}$$

$$Balanced\ Error\ Rate = 1 - BalancedAccuracy$$

AUC: The Receiver Operating Characteristic(ROC) and the Area Under ROC are the most commonly used evaluation measures for imbalanced data sets. A visual indication of the classifier superiority over another classifier overa a wide range of operating points is given by the ROC curve and the area under the ROC curve(AUC) summarizes the performance of a classifier into a single metric.

$$Area\ Under\ ROC\ Curve(AUC) = \frac{(1 + TP\_Rate - FP\_Rate)}{2}$$

AUC and Gmean are the popularly used evaluation metrics for imbalanced data sets classification. In this work also, these measures are used for comparision of the results.

2.7 Chapter Summary

This chapter discusses various existing methods available in literature to handle the problem of class imbalance. Also, mentioned the performance metrics used to evaluate the classifiers.

Chapter 3

Ensemble of Small Training sets for Classification (ECST)

Supervised learning methodology trains a model with the majority of the data (e.g. two-thirds) and then utilises the model created to label the remaining data (one-third) or uses k-fold cross-validation to classify the data. It is commonly understood that not all instances of a data set contribute equally to classification. Only a core set of instances may be required to accurately learn the characteristics of the data. Similarly, to train a classifier, not all features of the data are required. It's possible that there is some noise in those values. The term "noise" refers to events that vary from the data set's overall behaviour. These factors could have a negative impact on overall categorization rates.

However, in cases such as credit card fraud detection, earthquake data, and unusual disease data, the data for training is insufficient due to imbalance posing a barrier to general machine learning methods. These general strategies are unable to effectively adapt to changes in data distributions. Methods that can learn from limited training sets and generalise well are essential in such cases. If the classification accuracy is comparable to that obtained using the complete data set, reducing the size of the training set is always preferred. This reduction could aid in the removal of unclear occurrences from the training set. Even when the data collection is big or small, this assumes significance. In such cases, using an ensemble of classifiers to boost classification accuracy is recommended.

The purpose of this chapter is to generate a small number of core instances or a representative collection of instances that may be used to train a classifier without losing

generality.

3.1 Related Work

3.1.1 Ensemble of Classifiers

Ensemble based Learning has received enormous attention in machine learning research these days. An ensemble of classifiers classify unseen examples by voting using a set of classifiers. Main idea behind ensemble based learning is that the output is more accurate than using individual classifiers [115]. An ensemble of classifiers is constructed using different learning algorithms on either the same set of training samples or on different set of training samples obtained by processing them as in Kubat. The ensemble combines the outputs of its group of classifiers and gives an output based on a certain criterion. There are several ways of constructing the ensembles. Number of ways in which they can be constructed is mentioned in [70] and are provided below.

- Majority voting of classifiers output
- Processing training samples
- Processing the outputs of base classifiers
- Processing the attributes of the samples
- Hybrid method i.e., combined processing of training samples, attributes, merging of classifiers output etc.

Bagging and Boosting are two most commonly used ensemble methods. Bagging considers a series of n classifiers and the output is decided by majority voting of these classifiers. In Boosting, weighted majority voting is used in finalizing the output class [62]. Some works [106, 119, 40, 101, 93] give a summary of the research work and make recommendations for ensemble-based strategies that enhance classification accuracy over a single classifier. Jasmina et al. [68] employed an ensemble of AdaBoost Classifiers. Irenenensz Czarnowski proposed cluster-based instance selection algorithms [35]. The similarity coefficient, stratification strategy, and a modified approach are used to choosing instances from the clusters for training.

3.1.2 Small training Sets

The concept of learning from reduced training set sizes using methods like prototype selection [116], instance selection [64], training set selection [81] are available in the literature. In this chapter, ideas behind those methods are combined though not directly using those methods. Additionally filters are also used to improve the quality of small training sets [22].

Using small training sets, some authors [49, 15] compared the classification performance of classifiers. Sebban et al. applied prototype selection strategies for tree simplification [116]. An ensemble method for imbalanced data sets using tiny training sets is developed as an application in [127] to categorise the medications used for kinases. To cope with imbalanced classification, [91] employed ensembles of First Order logical Decision Trees, [9] used feature weighting to deal with overlap in unbalanced datasets, and [41] introduced a Random Balance approach for imbalanced data that uses ensembles of variable priors classifiers. Sun et al. introduced a novel ensemble approach for classifying imbalanced data sets [121]. In [69], numerous ensemble-based approaches to dealing with the issue of imbalanced datasets categorization have been developed. ensemble1 have offered a full analysis of the state-of-the-art ensemble based solutions for the imbalanced datasets classification problem[50].

In the literature, approaches such as prototype selection [116], instance selection [64], and training set selection [81] have been used to learn from smaller training sets. In [64], Nobert et al. compare and contrast several instance selection algorithms. Noise filters, condensation algorithms, and prototype selection algorithms were grouped into three groups. Wilson's Edited Nearest Neighbour(ENN) method starts with the original training set and removes instances that do not match the majority class of their neighbours [136]. Noise filters include Repeated ENN, AllKNN, and ENRBF. The Condensed Algorithm (CNN) begins by selecting one instance per class at random from the training set[57]. Then, using the new data, it adds each of the incorrectly identified instances from the training set to this collection. Another well-known instance-based selection method is Gates' Reduced Nearest Neighbor [53]. Salvador et al. [52] delve into more details about the taxonomy and actual examination of several prototype selection approaches. [64, 135, 6, 105, 94] describe and compare instance selection techniques. Processing the training samples and

merging the classifier outputs are included in the proposed work ensemble construction.

3.2 Motivation

The concepts behind ensemble and instance selection methods are merged, but not directly employed, in this chapter. Furthermore, filters are utilised to increase the quality of small training sets, according to [22]. Brodley et al. [21] have deleted the misclassified cases before attempting the actual learning process in order to improve the quality of training data. Classifiers are also employed as filters to exclude instances that are incorrectly categorised. There are two ways to use a classifier as a filter, according to [22]. In the first case, the same classifier serves as both a filter and a learning method. The second method is to employ one classifier for filtering and another for learning. In addition to these strategies, ensemble filters and consensus filters have been proposed in the literature to improve training accuracy.

Ensemble-based classifiers are supervised learning techniques that require the training of sets of labelled data. There is no hard and fast rule to determine the size of the training set. The number of samples should not be smaller than the number of features, and the number of samples should be large enough to characterise the problem, allowing the classifier to learn the nature of the dataset and categorise previously unknown instances. It also depends on the type of classification learning technique utilised. Whatever the case may be, it is true that not every instance contributes to classification. The time and space complexity of a training set grows as it's size grows. The aim of this work is to illustrate that it is not the quantity of the training set that improves the classifier's performance, but rather the representative samples of a given class that influence classification accuracy.

These days, learning from a simple concept is getting lot of attention. The following difficulties must be resolved in order to identify this representative set. These points demonstrate why the current strategy is being proposed:

1. Whether the data used for training is a representative of the complete data set.

2. Is it possible to get adequate training or equivalent classification accuracy with less than two-thirds of the data?

3. Are there any outliers in the training set? Outliers are events in the training set that deviate from the general characteristics of the full data set.

Three points must be considered in order to get acceptable classification accuracy with small training sets.

- Only one-third (30%) of the data set should be used to represent the whole data set.
- Using these small training sets to improve classification accuracy.
- Considering variations in tiny training sets, such as noise or outliers.

Three techniques are proposed to overcome the first issue: Divide the cases into ten bins using the *centroid* as a reference point, $3/2(min + max)$ as a reference point, and a distribution-specific binning. All of these methods use a stratified sampling methodology to create training sets, ensuring that the samples chosen are representative of the full distribution.
The second difficulty is the application of the ensemble-based weighted majority voting idea to classification.
The third problem is addressed by using four filters on the training sets. Removing outliers with the Inter Quartile Range option (included in the Weka toolbox) and removing misclassified cases with Naive Bayes, IB3, and IB5 filters are the filters employed.

3.3 Framework

Each of the afore mentioned concerns is addressed by proposing a framework that employs three different methodologies on seven different benchmark data sets.

3.3.1 Choosing representative samples for training

The first point raised in the motivation, namely whether the data chosen for training represents the complete data set or not, is addressed first. For this experimentation, binary and multi-class data sets are used. In binary datasets, the minority is considered positive while the majority is considered negative. In multi-class datasets, one class is considered positive and the others classes are considered as negative, which is known as one-versus-all.

Table 3.1: Data sets (b) stands for binary and (m) stands for multiple classes.

Data Set	Number of Features	Number of Instances (T)	Number of Positives (P)	Number of Negatives (N)	Imbalance Ratio (IR)
Pima India diabetic (b)	8	768	268	500	1.86
Wisconsin Diagnostic Breast Cancer (b)	32	569	212	357	1.684
Haberman (b)	3	306	81	225	2.68
Vowel0 (m)	13	988	90	898	9.98
LED7digit (m)	7	443	37	406	10.97
Musk2(b)	168	6598	1017	5581	5.49
Isolet5(m)	617	1559	60	1499	24.98

In this part, a framework for determining the number of training samples and how those samples are chosen to represent the whole data set is proposed (positive set and negative set separately). Benchmark data sets from the UCI machine learning repository [11] and KEEL [7] are used to test the heuristics. Only 10% to 18% of the data set is used for training, with rest of the portion being used for testing. The data sets used in this study are listed in Table 3.1.

Training sets can be generated in three ways:

Method 1: Bins are created based on the distance between the instances and the $centroid$.

Method 2: Bins are created based on the distance between instances and the reference point $3/2(min+max)$. The minimum and maximum values for each property are calculated, and 3/2 of that is used as the reference point.

Method 3: Bins are created depending on distance distribution.

The Algorithm 3.2 describes the approach for selecting representative samples from the original data using the centroid as a reference point. In the case of the reference point $3/2(min+max)$, the same process as in Algorithm 7 is followed. The method is termed Centroid based binning (CBB), Min-Max based binning (MMBB), and Distribution Specific binning (DBB) depending on the reference point used.

3.4 Algorithm

Algorithm 3.1 Construction of Training Sets for ECST

procedure CONSTRUCTION OF TRAINING SETS FOR ECST(*IDS*) ▷ IDS, an Imbalanced Data Set

Let D be a binary class dataset $\langle X, Y \rangle$*,where* $X = X_1, X_2, \ldots, X_n$*, each* X_i *is a* $m-$*dimensional Vector with* m *attributes and is associated with a label* $Y = 0,1$
N represents the size of the total dataset
N_{Min} *represents the size of Positives in the dataset*
N_{Maj} *represents the size of Negatives in the dataset*
$\frac{N_{Min}}{N_{Maj}} > 1.5$
$N_G = 10, N_G$ *Represents Number of Groups*
for $j \in m$ **do**
$Neg_{cent_j} = \sum_{i=1}^{N_{Maj}} X_{ij}/N_{Maj}$
$Pos_{cent_j} = \sum_{i=1}^{N_{Min}} X_{ij}/N_{Min}$
end for
for $i \in 1$ *to* N_{Min} **do**
$Posdist_i = (\sum_{i=1}^{N_{Min}} [X_{ij} - Pos_{cent_j}]^p)^{\frac{1}{p}}$
end for ▷ *Distance of all Positives from* $Pos_{centroid}$ *is Posdist*
for $i \in 1$ *to* N_{Maj} **do**
$Negdist_i = (\sum_{i=1}^{N_{Maj}} [X_{ij} - Neg_{cent_j}]^p)^{\frac{1}{p}}$
end for ▷ *Distance of all Negatives from* $Neg_{centroid}$ *is Negdist*
for $i \in 1$ *to* N_{Min} **do**
▷ *Distance is Normalized*
$Posdist_i = \frac{Posdist_i - min(Posdist)}{max(Posdist) - min(Posdist)}$
end for
for $i \in 1$ *to* N_{Maj} **do** ▷ *Distance is Normalized*
$Negdist_i = \frac{Negdist_i - min(Negdist)}{max(Negdist) - min(Negdist)}$
end for ▷ $max()$ *gives maximum* $-$ *value among the given input distance values*
▷ $min()$ *gives minimum* $-$ *value among the given input distance values*
$S = 30\%$ *of* D
$n = Negatives\ of\ S$
$p = Positives\ of\ S$
$Neg_{Group} = FormationofN_GGroups(N_{Maj}, Neg - dist)$
$Pos_{Group} = FormationofN_GGroups(N_{Min}, Posdist)$
for $j \in \{1\ to\ ntr(= n/p)\}$ **do**
$Neg_{Train}[j] \leftarrow StratifiedSampling(Neg_{Group}, N_Maj, n)$
$Pos_{Train}[j] \leftarrow StratifiedSampling(Pos_{Group}, N_{Min}, p)$
$Balanced - Training[j] = Neg_{Train}[j] + Pos_{Train}[j]$
end for
end procedure

Algorithm 3.2 ECST-Classification

procedure ECST-CLASSIFICATION
    CALL CONSTRUCTION OF TRAINING SETS FOR ECST(D) ▷ ntr Training Sets generated from Algorithm-1 are used to generate ntr classification models
    $Test_p$ *and* $Test_n$ *are remaining part of the dataSet*, D *are used for Testing*
    $Prediction = \sum_{i=1}^{ntr} wtr_i * outcome_i$ ▷ wtr_i *is the* $Positive - Predictive - Value(PPV) = TP/(TP+FP)$ ▷ $Outcome_i$ *is the outcome of the* i^th *Classifier*
    if $Prediction \geq Threshold$ **then**
        $test - instance\ is\ positive$
    else
        $test - instance\ is\ negative$
    end if ▷ $Threshold = PPV/2$
end procedure

function FORMATION OFN_G GROUPS($N_M, dist$) ▷ Instances belonging to N_G groups are determined.
    for $i \in \{1\ to\ N_M\}$ **do**
        for $j \in \{1\ to\ N_G\ in\ steps\ of\ 1\}$ **do**
            for $k \in \{0\ to\ 1\ in\ steps\ of\ (1/N_G)\}$ **do**
                if $(dist_i \geq k) \wedge (dist_i \leq k + (1/N_G))$ **then**
                    $Group\,j \leftarrow X_i$
                end if
            end for
        end for
    end for
    return $Group\,j$
end function

function STRATIFIEDSAMPLING($Group, N, x$)
    for $k \in \{1\ to\ N_G\ in\ steps\ of\ 1\}$ **do**
        $r_k = random(\frac{size(Group_k)}{N} x)\ \ 1 \leq i \leq k$ ▷ Select r_k *instances from* $Group_k$
    end for
    $s = \Sigma_k r_k$
return s
end function

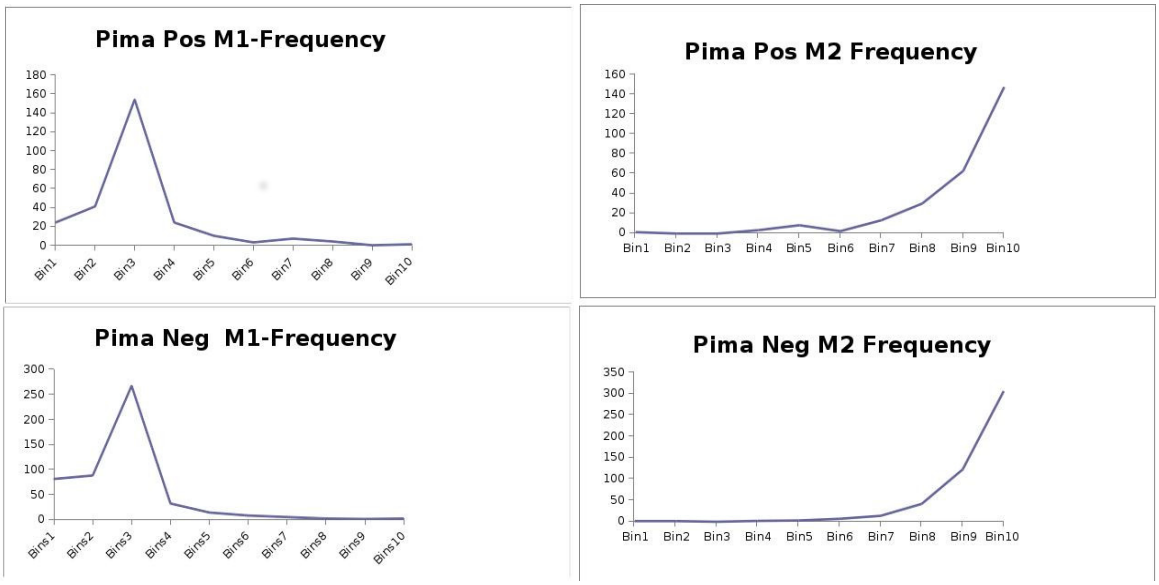

Figure 3.1: Distribution of distances for Pima using methods 1 and 2

The instances of binary classes are separated into 10 positive bins and 10 negative bins in the method described by Algorithm 3.1, which correspond to both positive and negative classes. The number 10 is arbitrary here. Adding to the discussion of bin formation: In Algorithm 3.1, construction of Traning Sets required for ECST which in turn calls functions-Algorithm 1.1 and 7 is given. In Algorithm 3.2, the process of classification with ECST is provided. The Complexity of the algorithm is O(n*c1*c2) i.e., O(n) where n is the number of majority class samples. c1 is a constant for fixed number from 0 to 1 ranging at steps $1/N_G$.

- Negative Bin1 contains negative occurrences that are between 0 and 0.1 distance from Neg_{Cent}. Negative Bin2 is made up of negative examples that are 0.1 to 0.2 distance away from Neg_{Cent} and so on. Finally, Negative Bin10 comprises negative instances that are between Neg_{Cent} and Neg_{Cent} by 0.9 to 1.00.

Figure 3.1 to Figure 3.5 show the distribution of distances using CBB and MMBB. The left side of the picture depicts the distribution of distances for positive (top) and negative (bottom) data, while the right side depicts the distribution of distances for CBB and MMBB, respectively. These distances are calculated for a single run with a high AUC value.

Distribution specific binning: For all data sets, the number of bins do not have to be 10. Bins can be reduced or increased depending on the number of samples. Also, the bins are chosen to be equal in this example, but they can be divided into different

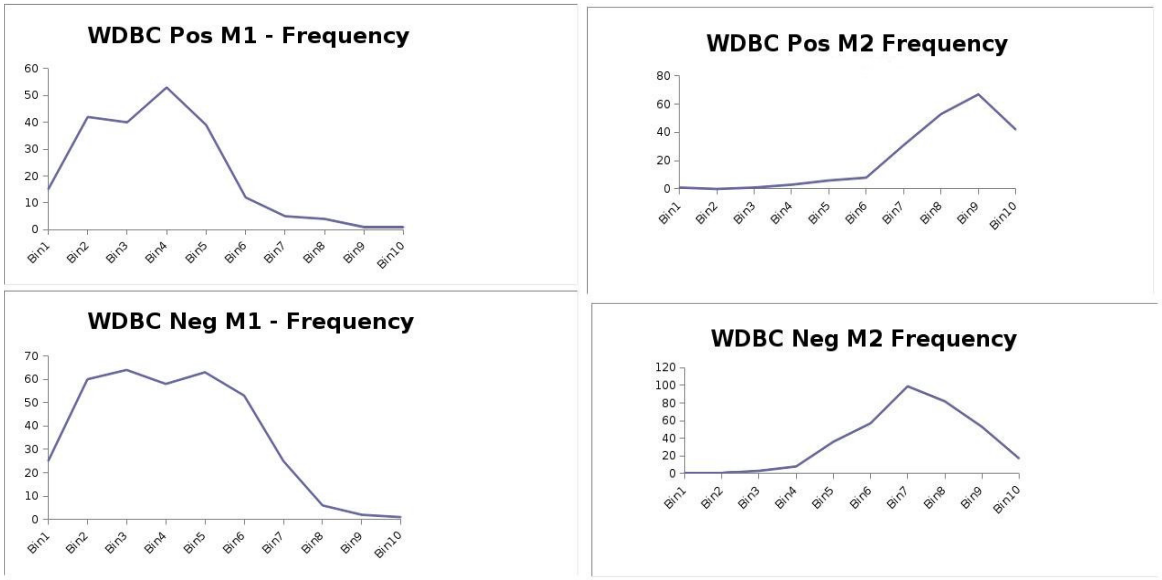

Figure 3.2: Distribution of distances for WDBC using methods 1 and 2

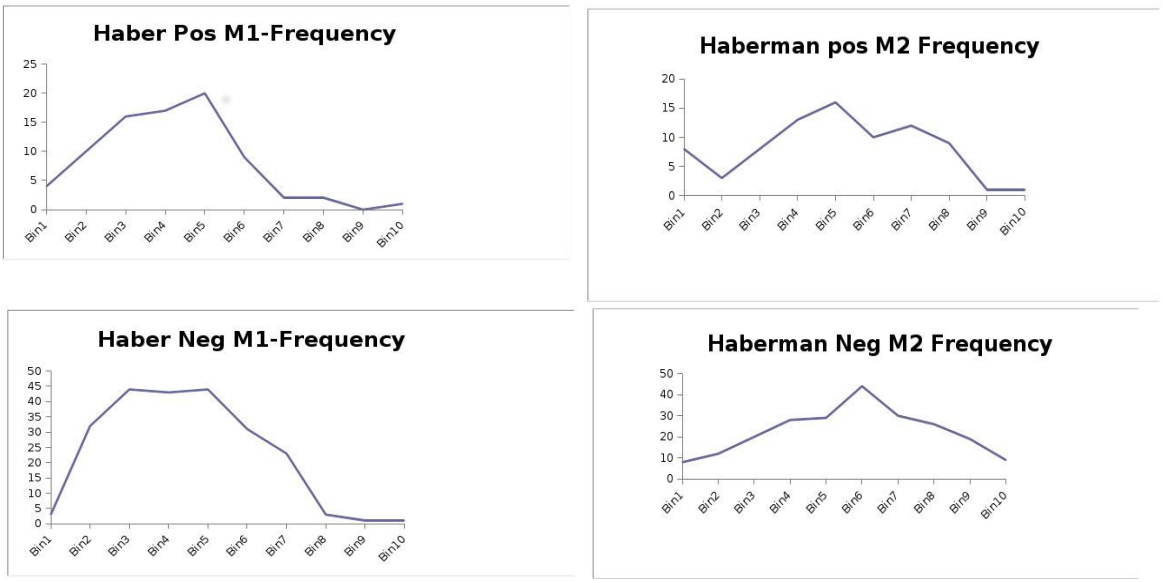

Figure 3.3: Distribution of distances for Haberman using methods 1 and 2

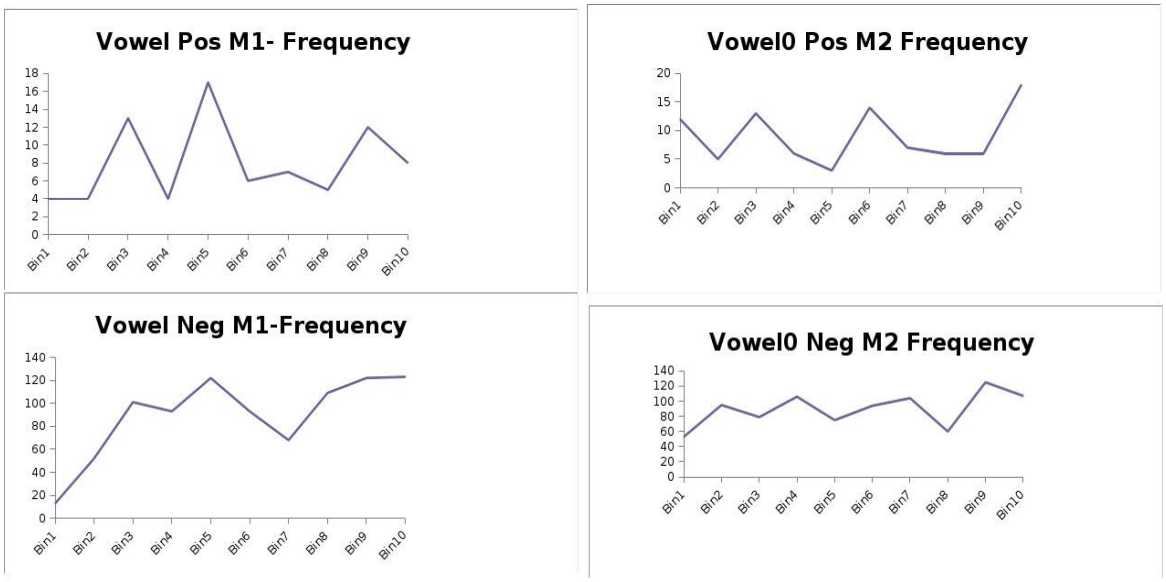

Figure 3.4: Distribution of distances for Vowel0 using methods 1 and 2

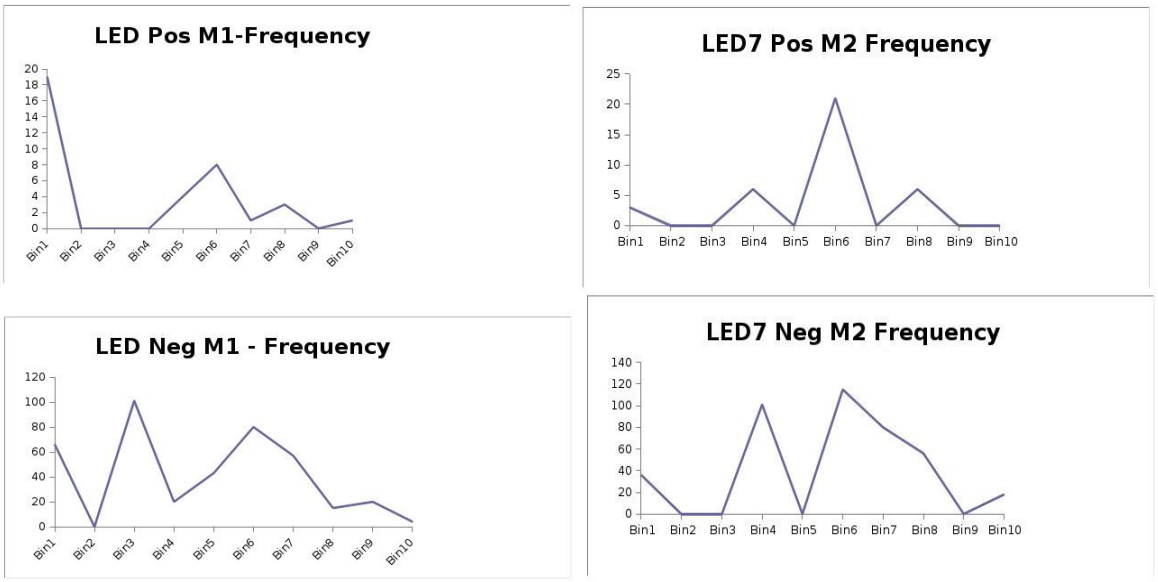

Figure 3.5: Distribution of distances for LED7 using methods 1 and 2

widths depending on the distribution, such as mean±std, mean±2×std, and so on. Here, mean±std, mean±2×std, and so on are chosen as the intervals in the distribution-specific binning approach. The interval is taken as the last interval once the maximum is surpassed.

3.4.1 Criteria to choose the number of samples

The quantity of training instances is the next issue to address. As previously stated, two-thirds of the data set, or 67%, is utilised for training and one-third of the data, or 33%, is used for testing. We want to look at it from the opposite perspective and see if we can reach classification accuracy with minimal training sets, say around 30% of the data.

The fraction of training set is chosen at 30% for implementation purpose. For example, the sum of p% and n% percentage, which represents positive and negative percentages of the complete data set, is set to be 30%. The chosen positive and negative examples for training are shown in the Table Table 3.2, with their total percentage set at 30%. It is important to note that the overall percentage of the full data set used for training is only between 10% and 18%.

3.5 Experiments and Results

Experiments are conducted on seven binary and multi-class data sets, with just 6% to 18% of the total data using for training, and the suggested three approaches are used on the training sets without any filters for noise and outlier removal. These results are compared to ada-boost and bagging ensemble techniques, as well as ENN, CNN, and RNN instance selection approaches. The three proposed techniques produce equivalent classification results to those available in the literature that use small training sets, according to empirical study.

3.5.1 Ensemble based majority voting method

The accuracy of classification must be maintained even with small training sets, which is the third issue in the motivation. The *Ensemble Method*, one of the most widely used methods for improving classification performance in the literature is used to solve this problem.

Table 3.2: Number of training sets, number of samples in each training set, total number of positive and negative samples used for training. Positive samples (Pos), Negative samples (Neg), Percentage of positives (Pos %), Percentage of negatives (Neg%), Number of positives used for training (pos-train), Number of negative samples used for training (Neg-train). Total training set varies from 10% to 18%.

Data Set	Pos	Pos%	Pos-train ($train_p$)	Neg	Neg%	Neg-train ($train_n$)	Overall Training%	Pos-test ($test_p$)	Neg-test ($test_n$)
Pima	268	10	25	500	20	100	16.27	243	400
WDBC	212	10	20	357	20	80	17.57	192	277
Haberman	81	15	12	225	15	36	15.68	79	189
LED7digit	37	20	9	406	10	36	10.15	28	370
Vowel0	90	20	18	898	10	90	10.93	78	802
Musk2	1017	10	102	5581	20	1122	18.55	1015	4459
Isolet5	60	25	15	1499	5	75	5.77	45	1424

As described in Algorithm 3.2, classification is accomplished by constructing an ensemble using ntr models while keeping $train$ as a positive subset and selecting different negative instances from negative data for each model. For classification, the Weka toolkit [138] is used. These data sets are used to build training models. The best model for each classifier with the highest G-mean is chosen based on the training data's leave-one-out cross validation. All of the models are given a common test data set consisting of remaining instances from positive and negative classes that have already been set aside. The weighted voting of the ntr classifiers on the test data is used to classify the data, with the positive predictive value (PPV) serving as the weight of each classification model.

Note: Decision trees are used as classification models in the ensemble, and the number of classifiers in an ensemble is determined by computing m/n, where m is the number of negative instances selected for training and n is the number of positive instances selected for training. The number of classifiers specify the number of negative training sets to be generated in this case. If the number of positives in a data set is low and the number of negatives is high, the same positive subset is used with different negative subsets in the ensemble classifiers. This method has been tested on data sets with imbalance ratios of over 10 such as LED7digit and VowelO.

Table 3.3: Number of training sets in an ensemble for each data set.

Data Set	Pos-train ($train_p$)	Neg-train ($train_n$)	Number of Classifiers (ntr)
Pima	25	100	4
WDBC	20	80	4
Haberman	12	36	3
LED7digit	9	36	4
Vowel0	18	90	5
Musk2	102	1122	11
Isolet5	15	75	5

3.6 Discussion

The average G-mean values for 10 runs obtained for training and test sets for all three methods (CBB, MMBB, and DBB) are reported in the Table 3.4 . According to Table 3.4, good test results are obtained utilising training sets selected from bins based on distance from $3/2(min+max)$ for Pima, WDBC, and LED7digit data sets. For the bins generated by *centroid* distance, the Haberman and VowelO results are good. Except for Haberman, both techniques produce satisfactory classification results on the test set for all other data sets. The explanation for this could be that Haberman has just three attributes and instances are chosen from bins over the centroid rather than $3/2(min+max)$, which is a border set out of the maximum distance. Because it is a multi-class data set, the one-versus-all methodology is utilised for Isolet5, which uses one class as positive data and the other 25 classes are considered as negative data to demonstrate that the method is general enough to handle a larger number of features.

3.6.1 Improving the quality of training sets by removing noise and outliers

Brodley et al. [22] have summarised that the training set's quality is increased by removing mislabeled instances before applying the chosen learning technique. On the training sets, popular filters such as Naive Bayes, IB3, and IB5 are used, according to [21]. It can be seen in Table Table 3.5 that the classifier's performance on the training sets has significantly

Table 3.4: Classification results using CBB, MMBB and DBB. Average AUC values are computed for 10 runs.

Data Set	Avg GMean Training			Avg GMean Test			Literature
	CBB	MMBB	DBB	CBB	MMBB	DBB	
Pima	0.67	0.73	0.73	0.72	**0.72**	0.69	0.71-0.76 [50]
WDBC	0.99	0.95	0.93	0.92	**0.95**	0.93	0.96-0.98 [50]
Haberman	0.73	0.44	0.61	**0.61**	0.43	0.54	0.56-0.66 [50]
LED7digit	0.72	0.85	0.85	0.79	**0.89**	0.85	0.89 [50]
VowelO	0.96	0.97	0.93	**0.96**	0.93	0.93	0.95-0.99 [50]
Musk2	0.84	0.85	0.83	0.88	**0.88**	0.87	0.9 [126]
Isolet5	0.92	0.92	0.86	0.90	**0.92**	0.87	NA [a]

[a] Not available in the literature

Table 3.5: Classification results of Experiments done with CBB and MMBB removing noise and outliers.

Data Set	Data sets	without filters		Removing outliers		NB		IB3		IB5	
		CBB	MMBB	CBB	MMBB	CBB	MMBB	CBB	MMBB	CBB	MMBB
Pima	Train Set	0.67	0.73	0.61	0.75	0.82	0.91	0.75	0.78	0.83	0.93
	Test Set	0.72	**0.72**	0.72	0.72	0.73	0.68	0.68	0.72	0.73	0.71
WDBC	Train Set	0.99	0.95	0.99	0.92	1.00	0.97	0.99	0.95	1.00	0.95
	Test Set	0.92	**0.95**	0.92	0.93	0.92	0.92	0.92	0.92	0.92	0.92
Haberman	Train Set	0.73	0.44	0.66	0.4	0.85	0.88	0.92	0.83	0.75	-
	Test Set	**0.61**	0.43	0.59	0.48	0.58	0.66	0.59	0.66	0.58	-
LED7digit	Train Set	0.72	0.85	0.72	0.85	0.84	0.88	0.97	0.91	0.98	0.91
	Test Set	0.79	**0.89**	0.79	0.89	0.8	0.89	0.83	0.75	0.83	0.75
VowelO	Train Set	0.96	0.97	0.96	0.97	0.97	0.98	0.98	0.96	0.97	0.96
	Test Set	**0.96**	0.93	0.96	0.93	0.9	0.91	0.9	0.93	0.9	0.93
Musk2	Train Set	0.84	0.85	0.79	0.81	0.97	0.98	0.90	0.93	0.95	0.81
	Test Set	0.88	**0.88**	0.83	0.72	0.77	0.77	0.82	0.85	0.78	0.72
Isolet5	Train Set	0.92	0.92	0.29	0.34	0.83	0.81	0.88	0.88	0.92	0.96
	Test Set	0.90	**0.92**	0.56	0.69	0.84	0.89	0.64	0.74	0.71	0.49

improved. That is, learning accuracy is improved to the point where little fluctuation in test set performance is noticeable. This scenario can be seen in both the CBB and MMBB approaches.

3.6.2 Analysis

Filtering appears to boost performance on the training set but not so much on the test set. Except for Haberman, classification without filtering produces good G-Mean using training sets selected from $3/2(min + max)$ bins. For Haberman, Naive Bayes filtering improves the results. MMBB produces more significant results without filters than it does

Table 3.6: Comparison of AUC Test results with other ensemble methods.

Data Set	Highest of our methods	AdaBoost	Bagging	Literature Results
Pima	0.72	0.71	0.72	0.71-0.76 [50]
WDBC	0.95	0.97	0.98	0.96-0.98 [50]
Haberman	0.61	0.55	0.48	0.56-0.66 [50]
LED7digit	0.89	0.91	0.91	0.89 [50]
VowelO	0.96	0.99	0.98	0.95-0.99 [50]
Musk2	0.88	0.97	0.94	0.9 [126]
Isolet5	0.93	0.86	0.82	NA

Table 3.7: Comparison of Test Accuracy results with instance selection methods. Accuracy values for IS-CNN, IS-ENN, IS-SNN are taken from [2].

Data Set	Highest of our methods	IS-CNN	IS-ENN	IS-SNN
Pima	0.72	0. 66	0.74	0.55
WDBC	0.93	0.94	0.96	0.68
Haberman	0.73	0.64	0.69	0.31
LED7digit	0.85	0.34	0.49	0.36
VowelO	0.92	0.96	0.96	0.90
Musk2	0.88	NA	NA	NA
Isolet5	0.89	NA	NA	NA

with filters. Filters are used to remove particular instances that contribute to the classifier's training. When filters are employed, this may result in lowering test accuracy.

When applying the Naive Bayes filter for pre-processing, MMBB produces good training and test results. According to [135], the percentage of samples kept for training in Pima using the method CNN is 36.89, with an accuracy of 0.65, and the percentage of samples retained for SNN is 42.95, with an accuracy of 67.97. However, using only 16.27% of the data set for training, an accuracy of 0.72 is obtained here in this work. In the case of WDBC, only CNN reached 0.95 accuracy with 7.09%of training set, whereas SNN reached 0.93 accuracy with 8.35%. The proposed method uses 17.57% for training and got 0.93 accuracy with the proposed strategy. For a Vowel data set of 30.05%data, CNN obtained an accuracy of 0.86, while SNN trained with 19.97% data and obtained an accuracy of 0.78. In comparison, in the research work of this chapter, needed 10.93% of the data for training and reached an accuracy of 0.92. Results employing CNN, ENN, and SNN for Musk2 and Isolet5 are not available in the literature.

There are numerous training set reduction algorithms, some of which select current examples from the data set for training and others which add new representative cases. To retain the samples in the training set, some approaches use incremental search, decremental search, or batch mode searches. Some strategies prefer to keep border points, core points, or other groups of points. The proposed method for training uses existing instances from the data set rather than creating new artificial samples. In [136], decremental reduction optimization procedure approaches employing 10 fold cross validation acquire accuracies for data sets: Pima 0.77, Vowel 0.85, whereas the suggested technique obtained an accuracy of 0.72 on Pima data sets and 0.92 on Vowel data sets with less than 30% of the total data for training.

Both the core points and the border points play a role in training in popular instance selection methods like CNN [57], SNN [112], and others. That is why, in this method, samples are chosen from the centre of the spread all the way to the fringe points. Good classification accuracy is attained in [135] utilising roughly the same percentage of samples as in this work. The suggested strategy focuses on fixing no more than 30% of the data for training while maintaining equivalent accuracy. Existing algorithms keep a sample proportion of higher than 30%. When the time required to run the proposed ensemble algorithm to other ensemble algorithms is compared, it is observed that all the methods

take about the same amount of time to execute.

3.7 Chapter Summary

To train a classifier, not all cases in a class contribute equally. Two of the presented methods show that if the chosen samples are representative, even 10% to 18% of the data set can be used to efficiently train a classifier. The primary benefit of this strategy is that it only requires calculating Euclidean distances and stratified sampling. Any random sampling method may overlook specific pockets, resulting in incorrect learning. This can be prevented by dividing the samples into bins and selecting samples proportionally to the number of samples in each bin. This also demonstrates that even when large amounts of data is unavailable, comparable conclusions can be obtained by sampling a tiny portion of the data.

The introduction of ensemble-based ideas and filters in pre-processing helped to improve learning accuracy. The deployment of the filters improves training performance greatly. However, the test scores have not risen in last step. The ability to generalise is hampered for the sake of improving training precision. Even for imbalanced data sets, the approach works fairly effectively. If the positive set is big, different positive subsets for each training set can be created as in bagging. The presented approaches are used to show that the method is also general enough to be applied to data sets with many dimensions. Musk2 and Isolet5, which have 168 and 617 features respectively, are used to demonstrate this aspect.

Chapter 4

Prototype generation employing the Centroid Based Grouping (CBG)

In the data level approach, the samples of the training sets are drawn from original data sets and are used for classification without necessitating any changes to the existing classifiers. Prototype Selection (PS) and Prototype Generation (PG) are two Prototype Reduction(PR) methodologies for shrinking the size of the training set and thus the amount of space and time required for training. Prototype Selection approaches use existing samples from the original training set to generate new prototypes, whereas Prototype Generation methods use existing samples from the original training set to generate new prototypes. The samples in the new training set in the first technique are already existing samples, whereas new synthetic samples are generated in the second method.

A synthetic sample generation method based on centroid based grouping is proposed in this chapter to address the class imbalance issue in a simple, novel and robust way.

4.1 Related Work

The major goal of the proposed study is to create a compact, robust training set that accurately represents the original training set. Unseen imbalanced test sets are classified using

the newly formed tiny training set with newly created prototypes. Prototype generation methods, class imbalance data-level methods, in particular undersampling, and prototype selection methods are all presented in the existing literature on small training sets. An overview of learning from tiny training sets is provided by papers [111, 70].

4.1.1 Prototype generation methods

The papers [129, 39, 130] provide a thorough discussion of prototype generation techniques. The four types of prototype creation methods are (i) Position Adjustment, (ii) Class Relabling, (iii) Centroid Based, and (iv) Space Splitting based on the generation mechanism. To acquire new prototype positions, the Position Adjustment method adds or subtracts some values from the prototype attribute values. This category includes DSM [54], LVQTC [137], MSE [38], AMPSO [25]. By adjusting the reduction rate in the case of Class Relabling, the generalisation accuracy of the test data is increased. This approach modifies the class labels of training set samples that are prone to being incorrect or that belong to various classes. These strategies deal with samples in the training set that are mislabeled or noisy. This category includes GENN [73], Depur [124]. With Space Splitting, the training set is separated into a few regions that will be replaced with representative samples to determine the original training set's decision bounds. These techniques act at spatial level. This group includes [30], RSP [123]. By combining a group of similar samples, these techniques create artificial prototypes. During the merging process, the average of selected subset attribute values, known as the centroid, is calculated. Although the accuracy is lost, this strategy achieves a significant reduction rate. SGA [47], MixtGauss [88], BTS3 [56], PNN [27], MCA [20], GMCA [98], ICPL [79] and so on are examples of this category.

The following are some of the most recent prototype generation approaches that involve genetic algorithms, evolutionary approaches and other techniques. Recently, Hu and Tan in [61], used particle swarm optimization to generate prototypes and provided two methods: error rank as a fitness function and the multi-objective optimization strategy as the other. Hugu et al. [46] described a genetic programming-based PG method for building extremely successful prototypes by merging multiple training samples using arithmetic operators. Hugo et al. [45] developed MOGP approaches for prototype creation in their recent study, which focuses on achieving a better trade-off between accuracy and reduction

by proposing a novel multi-objective evolutionary algorithm.

In [59], Yumilka et al. created prototypes for each similarity class using similarity relations for universe granulation. In [113], the EMOPG approach which uses a tournament to initiate a subset of training examples based on a weighted term is proposed. And then the position of initial prototypes are adjusted using the APES multiplicative evolutionary algorithm. Rash et al. [107] suggested a Similarity Based Imbalanced Classification (SIC) based on an empirical similarity function to discover patterns in the training set.

4.2 Motivation

Issues that are taken into account when creating the proposed method are:

- Information loss: Handled by considering all of the training scenarios when creating new prototypes. There is no information loss because all examples in the original training set are considered to build a new set of synthetic instances. Furthermore, the centroids created on average represent the instances, therefore there is no information loss. By dividing cases into groups based on their similarity, it is possible to consider the whole range of instances rather than taking samples from a specific location at random.

- Representative capability: Handled by taking into account all types of data based on similarity. The samples selected should be reflective of the original dataset. Based on their degree of resemblance, all the instances are separated into groups. Without leaving a single instance, all instances in each group are averaged together to represent that group. As a result, the mean of the group's instances is picked from each group to serve as a representative of that group. Since any noise can be removed in the process of finding the mean of the group occurrences, the averaging method makes the training set resilient.

On imbalanced data sets, the proposed approach is used. The premise behind this concept is to show that the prototypes developed are reflective of the training set. The influence of imbalance on the classifier performance is reduced even when the training set is very small.

The methodology of proposed CBG is a data-level class imbalance handling method that is neither oversampling nor undersampling because the size of the minority set is not raised, nor is it an undersampling method because the size of the majority class's subset is not chosen to be equal to the size of the minority class. CBG, on the other hand, generates an equal number of prototypes from both classes. It is treated as a prototype generation method since it reduces the training set size by creating new representative samples, and it can be considered a novel data level strategy to solve the problem of class imbalance because it balances the training set. MahalCUSFilter recently presented a hybrid approach of undersampling based on centroids in [32].

4.3 Framework

The proposed work's main goal is to build a small training set with low computational complexity for classifying imbalanced data sets. This attempts to demonstrate that "the *Quality* of the training set determines a classifier's performance rather than the *Quantity* of instances in the training set." The idea that samples that fall into a bin based on their similarity to the centroid display almost similar properties and that their *mean* can represent them on the whole. After generating bins, the mean of each bin is calculated instead of selecting a single instance from each bin to minimise the impact of attribute noise, if any, on classification. As a result, the newly generated smaller training set is thought to be robust.

Centroid-based classification calculates one centroid per class, that is, the mean of the attributes of the training set instances in that class. The similarity of a new test to the centroids is used to classify it. The advantage of centroid-based classification algorithms is that they are fast, as only a few similarity computations are required for a large number of classes according to Zehra [24]. Raskar et al. recently adopted Centroid based distance for signature recognition [100]. A test instance is classified by a similarity-based classifier based on the similarity between it and a collection of labelled training instances, as well as the pairwise similarities between the training examples. According to [143], the similarity-based classification does not necessarily provides direct access to the characteristics of the instances. Therefore the instance space can be any set, not necessarily a Euclidean space, as long as the similarity function for any pair of samples is well specified.

The average behaviour of each instance with regard to its class centroid is examined in order to generate a smaller training set with an equal number of prototypes from both classes with less information loss and lower computational complexity. To demonstrate this, the notion of centroid based grouping (CBG) is used to generate prototypes based on similarity from the original training set. CBG is a prototype generation method since it creates prototypes. PNN [27], one of the existing prototype generation methods, is based on this notion and employs class centroid as a prototype. It only utilises one prototype per class, but the suggested technique generates n prototypes from each class, regardless of its size. These n prototypes describe instances that differ from the class centroid. The goal is to cover the whole range of possible examples without abandoning any of them. The bins are grouped according to the criterion "How similar they are to their class centroid."

Independent of the size, dimensionality, and level of imbalance in the original training set, the proposed method generates a very small robust training set with less computational complexity and less information loss. The suggested method is unique in that it employs the concept of centroid-based categorization to create artificial instances (prototypes) for use as a training set. In addition, fractional distance measures (L0.1 and L0.5) are utilised to discover the similarity of training instances with regard to their class centroid. These measures have a broad coverage of neighbourhood space and are also well suited for multivariate, high dimensional data. The proposed technique separates the original training instances into n bins based on how similar they are to their class centroid, and then merges instances of each bin into a prototype, forming a new smaller training set, that is, $2n$ prototypes are generated for binary class dataset.

The results show that utilising a **very** small training set and a kNN classifier, the proposed method accurately classifies imbalanced, large, high-dimensional data sets, and offers comparable results to popular approaches. This means that "the $Quality$ of the training set, rather than the $Quantity$ of instances in the training set, determines the performance of a similarity-based classifier. " The suggested technique has the following key features: (i) reduced training time (ii) reduced storage to store the training set (iii) reduced information loss, and (iv) robustness to noise.

4.4 Algorithm

Proposed method's performance is evaluated using a kNN Classifier (k=1). 5 fold cross validation is carried out and the classifier's output is the average of the results acquired on five folds. The prototypes utilised are fixed to a given number n, regardless of the size of the training set. Three experiments with n=5, 10, and 20 prototypes are conducted. The algorithm 4.3 shows the steps taken in the pre-processing. The Complexity of Centroid Based Grouping Algorithm is O(n).

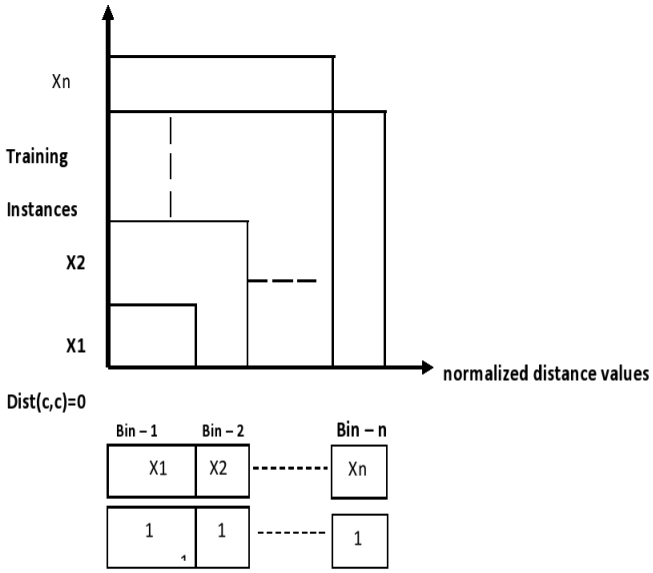

Figure 1 : The step by step process of formation of bins and creation of artificial Instances, 1 per each bin.

Figure 4.1: Centroid based grouping(CBG).

Using the suggested Centroid Based Grouping approach, Figure 4.1 depicts the process of forming bins from the original training set and creating prototypes from those bins for a single class. Both classes follow the same procedure. Origin (0,0), that is, dist(c,c), reflects the distance between the class centroid to itself as shown in the Figure 4.1. The numbers $x1$, $x2$ and so on reflect the number of instances in each bin. The size of the class is equal to the sum of $x1$ to xn. All xi instances of bin_i are merged into a single prototype, and all n prototypes per class from n bins form a new smaller training set in the case of binary classes.

If each class has five groups, all instances with a normalised distance from the centroid between 0 and 0.2 are placed in Group 1, between 0.2 and 0.4 in Group 2, 0.4 to 0.6 in Group 3, 0.6 to 0.8 in Group 4, and all instances with a normalised distance between 0.8 and 1.0 are placed in Group 5. Similarly, if the number of groups to be constructed is 10, the instances with a normalised distance from the Centroid between 0 and 0.1 are placed in Group 1, and so on, with a normalised distance between 0.9 and 1.0 in Group 10. Furthermore, for 20 Groups, instances with a distance from the Centroid of 0 to 0.05 are placed in Group 1, while instances with a distance of 0.95 to 1.0 are placed in Group 20. As a result, the normalised distance-range (interval) for each bin for groups 5, 10, and 20 is 0.2, 0.1, and 0.05, respectively.

The prototypes created using Euclidean distance measure are used by kNN. Using p=2 in Equation 4.4.1, where P and Q are d dimensional feature vectors, the Euclidean distance $s(P,Q)$ is calculated. Because of the curse of dimensionality on prototypes creation, the kNN classifier is coded with p=0.5(L0.5) and 0.1(L0.1) [26, 5]. To produce prototypes, L0.5 and L0.1 distance measurements, as well as the standard Euclidean distance measure are employed to build bins. Bins created with the L0.5 distance measure are tested with the L0.5 kNN classifier, while bins created with the L0.1 distance measure are tested with the L0.1 kNN classifier as well as the standard kNN classifier that utilises the Euclidean distance measure.

$$s(P,Q) = (\sum_{i=1}^{d} |P_i - Q_i|^p)^{\frac{1}{p}} \tag{4.4.1}$$

4.5 Experiments and Results

To assess the effectiveness of the proposed preprocessing methodology, a wide variety of attribute values and instances for the training set are chosen to account for the effects of data complexities and attribute complexities and imbalance ratios. They can be broadly classified as sets with instances (low and high) and attributes (low and high), such as a small number of features with a large number of instances, large number of features with a large number of instances, and so on. The details of these data sets can be found in Table 7.2. Except for the class attribute, all other attribute values in these data sets are

numeric. The data sets were obtained from the UCI [11] and KEEL[7] data repositories, and the tests were carried out using the KEEL [7] and weka[138] tools. The percentage of data used by CBG for training considering 5, 10, 20 sub groups per class is specified in the Table 4.2. It is clear that very small training sets are used, indicating a considerable reduction in space and time usage.

Table 4.1: Details of the data sets.

Name of the Data Set	Dimen-sionality	# Fea-tures	Size of the Data Set	Total # Instances	Imba-lance	IR
Pima	low	8	low	768	low	1.89
Haberman	low	3	low	306	low	2.78
Musk2	high	166	high	6599	low	5.49
Segment0	low	19	high	2308	low	6.02
Pageblocks0	low	10	high	5472	low	8.79
Vowel0	low	13	low	988	low	9.98
Spectrometer	high	93	high	7797	high	10.8
Scene	high	294	high	2407	high	12.6
LibrasMove	high	90	low	360	high	14
Ecoli4	low	7	low	459	high	14.3
Isolet5	high	617	high	1599	high	24.98
Yeast1289Vs7	low	8	low	947	high	30.57

4.6 Discussion

Test set accuracy, percentage of reduction in the training set, and other criteria are used to evaluate the performance of PG techniques. The number of prototypes generated for various CBG variations is fixed based on the number of subgroups employed. It may differ with other approaches. In the case of [114], the proportion of prototypes is fixed at 5% of the original data.

4.6.1 Comparison of the various variants of CBG method

The proposed method's performance on bins generated using Euclidean, L0.5, and L0.1 distance measures is verified using three classifiers: 1NN, L0.5NN, and L0.1NN. AUC values produced by CBG in comparison to the original training set classification are shown in the tables Table 4.3, Table 4.4, Table 4.5. No-Sampling denotes that data sets are classified

Table 4.2: Percentage of data used for training by CBG. (5,10,20 bins per class)

Data set	Size of the Training Set	% of Tr Set 5SG(10)	% of Tr Set 10SG (20)	% of Tr Set 20SG(40)
Pima	614	1.6	3.26	6.4
Haberman	245	4.08	8.16	16.32
Musk2	5279	0.18	0.38	0.72
Segment0	1846	0.54	1.08	2.0
PageBlocks0	4377	0.23	0.46	0.92
Vowel0	791	1.2	2.4	4.8
Spectrometer	6237	0.16	0.32	0.64
Scene	1925	0.52	1.04	2.08
LibrasMove	288	3.47	6.94	13.88
Ecoli4	367	2.72	5.44	10.88
Isolet5	1279	0.78	1.56	3.12
Yeast1289vs7	758	1.32	2.64	5.28

using a standard 1NN classifier without any preprocessing, that is, the test set instances are labelled with 100% of the training set.

- Data sets with low dimensions and low imbalance ratio Pima, Haberman have been classified better than using the entire training set. Vowel0 is giving lower classification result because of its complexity. Vowel0 which is actually a multiclass is converted into binary data set.

- Large data sets Musk2, Segment0, PageBlocks0 are giving acceptable results though not even 2% of the data is used for training. Increasing number of prototypes somewhat increases the performance but not fully proportionate to the size of the training set and experimentation is done on three different number of prototypes. In this case, high reduction is achieved.

- For large data sets with high dimensionality and high imbalance ratio, like Spectrometer, Scene, Isolet5, CBG is giving excellent classification results with very tiny training sets, than using entire 100% training set.

- For high dimensional small data sets with high imbalance ratio CBG variants are giving the best results compared to undersampling and other prototype generation methods.

Table 4.3: AUC Results of CBG Method using 1NN(5bins per class)

Data Set	No Sampling	Eucl-1NN	L0.5-1NN	L0.1-1NN	L0.5-0.5NN	L0.1-0.1NN
Pima	0.66	0.67	0.67	**0.69**	0.66	0.62
Haberman	**0.57**	**0.57**	0.56	0.52	**0.57**	0.52
Musk2	**0.92**	0.58	0.58	0.63	0.51	0.53
Segment0	**0.99**	0.84	0.87	0.88	0.79	0.66
PageBlocks0	**0.87**	0.71	0.71	0.72	0.62	0.66
Vowel0	**1.00**	0.89	0.85	0.84	0.84	0.73
Spectrometer	**0.86**	0.77	0.76	0.79	0.63	0.72
Scene	0.55	0.69	**0.70**	**0.70**	0.67	0.67
LibrasMove	0.85	0.89	0.90	**0.91**	0.89	0.81
Ecoli4	0.87	**0.94**	0.92	**0.94**	0.90	0.72
Isolet5	**0.94**	**0.94**	0.92	0.93	0.70	0.59
Yeast1289vs7	0.55	**0.75**	0.71	0.70	0.62	0.51

- Among the data sets used for experimentation yeast1289vs7 has the highest imbalance ratio 30.57 with 30 minority class instances and 917 majority class instances. Even this kind of data set with small number of instances and high imbalance ratio is well classified by CBG.

Friedman analysis is conducted on the variants of CBG and no-sampling. CBG is implemented by changing distance measures and classifier(i.e., kNN(k=1) with L0.1 and L0.5 distance), and is giving better results than no-sampling. For comparison with other methods, one variant of CBG is chosen by conducting Friedman test and it is evident that from tables Table 4.6, and Table 4.7, 1NN classifier on 20 bins per class using L0.1 distance measure is giving better results. So, this variant of CBG that is, L0.1-1NN-20G is taken for comparison with few other undersampling methods and prototype generation methods to test the efficacy of the proposed method.

4.6.2 Comparison with Prototype Generation Techniques

The comparison is made using the methodology described in the [114]. The approaches under consideration provides high test accuracy but may not be as good at reduction (GENN,1-NN), and approaches that provide high reduction rate may not be as good at accuracy (GENN,1-NN) (PSCSA). There is a trade-off between test accuracy and reduc-

Table 4.4: AUC Results of CBG Method using kNN(10 bins per class)

Data Set	No Sampling	Eucl-1NN	L0.5-1NN	L0.1-1NN	L0.5-0.5NN	L0.1-0.1NN
Pima	0.66	0.67	**0.68**	0.65	0.66	0.62
Haberman	0.57	0.58	**0.61**	0.5	0.57	0.52
Musk2	**0.92**	0.55	0.62	0.69	0.51	0.53
Segment0	**0.99**	0.81	0.85	0.84	0.75	0.67
PageBlocks0	**0.87**	0.71	0.73	0.76	0.62	0.66
Vowel0	**1.00**	0.89	0.87	0.85	0.84	0.73
Spectrometer	0.86	0.84	0.84	**0.87**	0.62	0.72
Scene	0.55	**0.71**	0.69	**0.71**	0.67	0.66
LibrasMove	0.85	0.91	**0.93**	**0.93**	0.89	0.80
Ecoli4	0.87	0.87	**0.93**	0.90	0.90	0.72
Isolet5	**0.94**	0.88	0.93	**0.94**	0.70	0.59
Yeast1289vs7	0.55	0.68	0.61	**0.80**	0.62	0.50

tion rates. Others may be able to do both, but it will take a long time for them to converge. Other prototype generation approaches such as centroid-based, position adjustment, space splitting, and class relabeling are also considered. Tables Table 4.8 and Table 4.9. CBG-PG present the findings. When compared to other PG approaches, the categorization can be deduced as follows. Because GENN has a low reduction rate of roughly 20%, it achieves better classification results for low IR data sets. LVQTC and MSE produce lower outcomes than MDSG. Other methods aren't fast enough to get results in under 300 seconds. PSO and AMPSO are evolutionary algorithms that require a long time to converge.

When data sets have an imbalance ratio of more than 10, CBG gives substantially better outcome than other prototype generation methods (popularly used are chosen for comparison) by employing relatively small training sets. '-' indicates that a few approaches have not generated output even after 300 seconds. It's understandable because execution and convergence take longer. The proposed method, on the other hand, can run quickly even on large data sets. The ranking achieved by the Friedman test on CBG and other prototype generation methods for which results are acquired for all data sets is specified in Table 4.11. Other prototype generating approaches have clearly been outperformed by the suggested CBG prototype generating approach.

Table 4.5: AUC Results of CBG Method using kNN(20 bins per class).

Data Set	No Sampling	Eucl-1NN	L0.5-INN	L0.1-1NN	L0.5-0.5NN	L0.1-0.1NN
Pima	0.66	0.68	0.71	**0.73**	0.64	0.54
Haberman	0.57	0.52	**0.58**	0.52	0.56	0.46
Musk2	**0.92**	0.57	0.65	0.70	0.55	0.54
Segment0	**0.99**	0.84	0.87	0.88	0.79	0.66
PageBlocks0	**0.87**	0.47	0.72	0.80	0.48	0.71
Vowel0	**1.00**	0.92	0.88	0.89	0.84	0.61
Spectrometer	0.86	**0.88**	0.86	0.87	0.86	0.86
Scene	0.55	0.69	**0.70**	**0.70**	0.67	0.67
LibrasMove	0.85	**0.94**	**0.94**	**0.94**	0.92	0.89
Ecoli4	0.87	**0.92**	**0.92**	**0.92**	0.89	0.81
Isolet5	**0.94**	**0.94**	0.93	0.93	0.69	0.55
Yeast1289vs7	0.55	**0.70**	0.68	0.68	0.68	0.52

4.6.3 Comparison with Undersampling Techniques

Tables Table 4.10, Table 4.8, Table 4.9 show the comparison of AUC results obtained by executing other undersampling methods and prototype generation methods available in KEEL with the proposed CBG method. Results of the Table 4.10 except the CBG are obtained by executing KEEL [7].

The proposed method has the following advantages: (i) faster processing, (ii) less storage, (iii) reduced information loss, and (iv) noise resistance. When the size of the training is reduced, time and storage are obviously decreased. This strategy does not discard even a single instance, thus there is little information loss, and because the mean of examples in each bin are obtained, the impact of noise or outliers on characteristics, if any, on categorization is reduced. Because the training set is balanced and the same number of prototypes are generated from both classes, regardless of class size, CBG is appropriate for the classification of imbalanced data sets.

Computational Ease is the novelty of the proposed approach, which falls in the centroid based PG category. All of the methods listed above work with practically every sample's nearest neighbours, however, the suggested method combines samples depending on how similar they are to their class average behaviour. This results in a higher reduction rate with less AUC loss.

Table 4.6: Average rankings of the algorithms (Friedman). Friedman statistic (distributed according to chi-square with 15 degrees of freedom): 90.165441. P-value computed by Friedman Test: 0.

Algorithm	Ranking
No Sampling	6.9167
Eucl-1NN-10G	7.5
L0.5-1NN-10G	5.9167
L0.1-1NN-10G	5.875
L0.5-0.5NN-10G	11.625
L0.1-L0.1NN-10G	14.2083
Eucl-1NN-20G	6.25
L0.5-1NN-20G	4.6667
L0.1-1NN-20G	**4.3333**
L0.5-L0.5-20G	10.4167
L0.1-L0.1-20G	13.0833
Eucl-1NN-5G	6.4167
L0.5-1NN-5G	7.2917
L0.1-1NN-5G	6.1667
L0.5-L0.5-5G	11.4167
L0.1-L0.1-5G	13.9167

4.7 Summary

The proposed method's main purpose is to build a small training set with low computational complexity which could be used to categorise unbalanced data sets. In order to create this training set, instances of each class are compared to their class-centroid in terms of their amount/degree/extent of similarity. The key concept here is to prevent information loss by categorizing instances based on how they behave in comparison to their typical class. Bin_1 contains instances that are more similar to the average class behaviour, bin_2 contains instances that are less similar to the class centroid, and so on, while bin_n contains instances that are more dissimilar. The importance of distance measurement in this procedure cannot be overstated. In trials, fractional distance measurements L0.5, L0.1, which are designed for high-dimensional data sets, are utilised along with Euclidean distance measures. The above-mentioned distance measures are also used in the Nearest Neighbor classifier method. Bins are constructed using the same distance measurements for these L0.1 and L0.5 NN classifiers.

The experimental results show that this method produces superior outcomes on datasets with a high imbalance ratio. Another finding is that the suggested CBG technique works

Table 4.7: Post Hoc comparison Table for $\alpha = 0.05$ (FRIEDMAN). Holm's procedure rejects those hypotheses that have an unadjusted p-value ≤ 0.005556.

i	algorithm	$z = (R_0 - R_i)/SE$	p	Holm
15	L0.1-L0.1NN-10G	5.080646	0	0.003333
14	L0.1-L0.1-5G	4.930584	0.000001	0.003571
13	L0.1-L0.1-20G	4.501838	0.000007	0.003846
12	L0.5-0.5NN-10G	3.751532	0.000176	0.004167
11	L0.5-L0.5-5G	3.644345	0.000268	0.004545
10	L0.5-L0.5-20G	3.129849	0.001749	0.005
9	Eucl-1NN-10G	1.629237	0.103263	0.005556
8	L0.5-1NN-5G	1.52205	0.127997	0.00625
7	No Sampling	1.329114	0.18381	0.007143
6	Eucl-1NN-5G	1.071866	0.28378	0.008333
5	Eucl-1NN-20G	0.986117	0.324076	0.01
4	L0.1-1NN-5G	0.943242	0.345557	0.0125
3	L0.5-1NN-10G	0.814618	0.415291	0.016667
2	L0.1-1NN-10G	0.793181	0.427672	0.025
1	L0.5-1NN-20G	0.171499	0.863832	0.05

well on datasets with high dimensionality, as well as large datasets. In cases when the imbalance ratio is more than 10, the proposed technique outperforms the original 100% training set. The same is true when comparing undersampling strategies. It is easy to see how CBG reduces the influence of exceptional cases and outliers using the mean of samples generated by Centroid Based Grouping. Training sets are created by creating an equal number of samples from both classes, regardless of their cardinality. A reduction in the size of the training set is obtained, resulting in a reduction in space and time usage. With those few created prototypes, better AUC is attained for imbalanced data sets.

Table 4.8: Comparison of AUC results of proposed methods with popular prototype generation techniques using KNN methods chosen from [114]. (- indicates output didn't obtained even after 300 seconds.)

Data set	AMPSO	GENN	LVQTC	MSE	PSCSA	PSO	**CBG**
Pima	0.66	0.67	0.68	0.67	**0.73**	0.72	**0.73**
Haberman	0.48	0.52	0.53	**0.54**	**0.54**	0.52	0.52
Musk2	-	-	**0.70**	0.61	-	-	**0.70**
Segment0	-	**0.99**	0.88	0.88	0.70	-	0.88
PageBlocks0	-	**0.87**	0.71	0.76	0.72	-	0.80
Vowel0	-	**1.00**	0.58	0.91	0.72	-	0.89
Spectrometer	-	-	0.78	0.85	-	-	**0.87**
Scene	-	-	0.52	0.49	-	-	**0.70**
LibrasMove	-	-	0.65	0.89	-	-	**0.94**
Ecoli4	0.86	0.87	0.90	**0.94**	-	-	0.92
Isolet5	-	**0.93**	0.59	0.78	0.61	-	**0.93**
Yeast1289vs7	-	0.51	0.53	0.53	0.55	-	**0.68**
Average of AUC for data sets(IR¡10)	-	-	0.68	0.728	-	-	0.753
Average of AUC for data sets(IR¿10)	-	-	0.661	0.746	-	-	0.84

Algorithm 4.3 Centroid Based Grouping

Let D be a binary class dataset $\langle X, Y \rangle$, *where* $X = X_1, X_2, \ldots, X_n$, *each* X_i *is a* $m-$ *dimensional Vector with* m *attributes and is associated with a label* $Y = 0,1$
N represents the size of the dataset
N_{Min} *represents the size of Positives in the dataset*
N_{Maj} *represents the size of Negatives in the dataset*
$\frac{N_{Min}}{N_{Maj}} > 1.5$
$N_G = 10, N_G Represents Number of Groups\ GC represents group centroid$

1: **procedure** CENTROID BASED GROUPING(ITS) ▷ ITS, an Imbalanced Training Set
2: **for** $j \in m$ **do**
3: $Neg_{cent_j} = \sum_{i=1}^{N_{Maj}} X_{ij}/N_{Maj}$
4: $Pos_{cent_j} = \sum_{i=1}^{N_{Min}} X_{ij}/N_{Min}$
5: **end for**
6: **for** $i \in 1$ *to* N_{Min} **do**
7: $Posdist_i = (\sum_{i=1}^{N_{Min}} [X_{ij} - Pos_{cent_j}]^p)^{\frac{1}{p}}$
8: **end for** ▷ *Distance of all Positives from* $Pos_{centroid}$ *is* $pos-dist$
9: **for** $i \in 1$ *to* N_{Maj} **do**
10: $Negdist_i = (\sum_{i=1}^{N_{Maj}} [X_{ij} - Neg_{cent_j}]^p)^{\frac{1}{p}}$
11: **end for** ▷ *Distance of all Negatives from* $Neg_{centroid}$ *is* $Neg-dist$
12: **for** $i \in 1$ *to* N_{Min} **do** ▷ *Distance is Normalized*
13: $Posdist_i = \frac{Posdist_i - min(Posdist)}{max(Posdist) - min(Posdist)}$
14: **end for**
15: **for** $i \in 1$ *to* N_{Maj} **do** ▷ *Distance is Normalized*
16: $Negdist_i = \frac{Negdist_i - min(Negdist)}{max(Negdist) - min(Negdist)}$
17: **end for**
18: ▷ $max()$ *gives maximum* $-$ *value among the given input distance values*
19: ▷ $min()$ *gives minimum* $-$ *value among the given input distance values* ▷ *Experimented with* $p = 0.1 and\ 0.5$
20:
21: $Pos_{Group} \leftarrow$ FORMATION OF GROUPS$(N_{Min}, Posdist)$
22:
23: $Neg_{Group} \leftarrow$ FORMATION OF GROUPS$(N_{Maj}, Negdist)$
24: **for** $l \in \{N_G\}$ **do**
25: **for** $j \in m$ **do**
26: $Neg_{GC[l]_j} = \sum_{i=1}^{Size(l)} X_{ij}/Size(l)$ ▷ *$Neg_{GC[l]}$ is the Centroid of each Negative Group l*
27: **end for**
28: **end for**
29: **for** $l \in \{N_G\}$ **do**
30: **for** $j \in m$ **do**
31: $Pos_{GC[l]_j} = \sum_{i=1}^{Size(l)} X_{ij}/Size(l)$
32: ▷ *$Pos_{GC[l]}$ is the Centroid of each Positive Group l*
33:
34: **end for**
35: **end for**
36: $BalancedTraining = \Sigma_{k=1}^{N_G} (Pos_{GC[k]}) + (Neg_{GC[k]})$
37: **end procedure**

```
1: function FORMATION OF GROUPS(N_M, dist_i)
2:     for i ∈ {1 to N_M} do
3:         for j ∈ {1 to N_G in steps of 1} do
4:             for k ∈ {0 to 1 in steps of (1/N_G)} do
5:                 if (dist_i ≥ k) ∧ (dist_i ≤ k + (1/N_G)) then
6:                     Group j ← X_i
7:                 end if
8:             end for
9:         end for
10:    end forreturn Group j
11: end function
```

Table 4.9: Comparison of AUC results of proposed methods with centroid based prototype generation techniques and one from each other category using kNN. (- indicates output didn't obtained even after 300 seconds.)

Data Set	PNN	BTS3	MCA	GMCA	ICPL	SGP	MixtGauss	Chen	DSM	GENN	**CBG**
	Centroid Based							Space Splitting	Positioning Adjustment	Class Relabelling	Proposed Method
Pima	0.70	0.65	-	0.66	-	0.54	0.68	0.65	0.66	0.67	**0.73**
Haberman	0.55	0.53	-	0.55	-	0.47	**0.58**	0.55	0.57	0.52	0.52
Musk2	-	**0.86**	-	-	-	0.75	-	**0.86**	0.84	-	0.70
Segment0	0.84	0.97	-	0.99	-	0.80	0.83	**0.99**	0.98	0.99	0.88
PageBlocks0	-	0.77	-	-	-	0.55	0.70	0.87	0.80	**0.87**	0.80
Vowel0	-	0.77	-	-	-	0.59	0.67	0.99	0.79	**1.00**	0.89
Spectrometer	-	0.76	-	-	-	0.86	-	0.86	0.72	-	**0.87**
Scene	-	0.54	-	-	-	0.57	-	0.51	0.55	-	**0.70**
LibrasMove	-	0.62	-	-	-	0.74	-	0.69	0.59	-	**0.94**
Ecoli4	**0.94**	0.86	-	0.84	-	0.85	**0.94**	0.84	0.89	0.87	0.92
Isolet5	-	0.80	-	-	-	0.91	0.91	0.91	0.84	**0.93**	**0.93**
Yeast1289vs7	-	0.52	-	-	-	0.61	0.60	0.59	0.52	0.51	**0.68**
Average of AUC for data sets(IR¡10)	-	0.758	-	-	-	0.616	-	0.813	0.773	-	0.753
Average of AUC for data sets(IR¿10)	-	0.683	-	-	-	0.756	-	0.733	0.685	-	0.84

Table 4.10: Comparison of AUC results of proposed method MDSG with **undersampling techniques** using **kNN** [7].

Data set	CNN (1968)	CNNTL (2004)	CPM (2005)	SBC (2006)	NCL (2001)	OSS (1997)	RUS (2004)	TL (1976)	**CBG**
Pima	0.67	0.64	0.64	0.71	0.72	0.66	0.72	**0.74**	0.73
Haberman	0.63	0.59	0.61	0.57	0.63	**0.64**	0.61	0.63	0.52
Musk2	**0.92**	0.88	0.78	0.50	**0.92**	0.91	0.89	**0.92**	0.70
Segment0	0.97	0.97	0.94	**0.98**	**0.98**	**0.98**	0.97	**0.98**	0.88
PageBlocks0	0.94	0.94	0.91	0.93	0.93	0.93	0.94	0.93	0.80
Vowel0	0.92	0.92	0.89	0.95	0.92	0.92	0.94	**0.97**	0.89
Spectrometer	0.81	0.79	0.83	0.62	0.85	0.81	0.85	**0.87**	**0.87**
Scene	0.60	0.58	0.58	0.50	0.60	0.57	0.63	0.58	**0.70**
LibrasMove	0.84	0.77	0.70	0.50	0.78	0.79	0.73	0.83	**0.94**
Ecoli4	0.83	0.84	0.81	0.81	0.81	0.84	0.86	0.81	**0.92**
Isolet5	0.84	0.86	0.57	0.57	0.86	0.86	0.87	0.81	**0.93**
Yeast1289vs7	0.59	0.61	0.63	0.5	0.53	0.61	0.60	0.54	**0.68**

Table 4.11: Average Rankings of the algorithms (Friedman). Friedman statistic (distributed according to chi-square with 5 degrees of freedom): 11.130952.
P-value computed by Friedman Test: 0.048845.

Algorithm	Ranking
LVQTC	4.375
MSE	3.7083
BTS3	4.25
Chen	2.6667
DSM	3.5833
CBG	**2.4167**

Chapter 5

Quartiles based UnderSampling(QUS)

The major difficulty in learning from imbalanced datasets is that the majority have a big number of training instances while the positives have a small number. Even if the classification rate of positives is greatly reduced, this may result in a pretty good performance of the classifier (minority class instances).

To choose majority class samples that may be used as a training set, either clustering or closest neighbour approaches are commonly utilised. There are certain drawbacks to these procedures. In the case of knn methods, the identification of nearest neighbours, and the distance measure to be used and so on and in clustering mechanisms the quality of the majority class samples chosen varies depending on the clustering technique, the number of clusters, and the difficulty of convergence and so on. In this chapter, a new method called Quartiles-based Under Sampling (QUS) is proposed, which is simple, unique, and effective, and can be used on dataset of any size and any number of dimensions.

The issue of class imbalance is tackled in this chapter based on the distribution of the dataset. To propose a simple undersampling method by considering less loss in data and to make the method parameter independent.

5.1 Related Work

At data level, the data set is manipulated to balance the class distribution. Data level sampling methods for handling imbalanced data sets are categorized into (i)Oversampling methods and (ii) Undersampling methods.

Latest papers on imbalanced data sets include [132, 148, 19, 103, 91, 14] etc. Wang et al. [132] use an ensemble method along with weights and information about sample misclassification to effectively classify imbalanced data. Zhang et al. [148] present empirical analysis by conducting various experiments on imbalanced data sets of varying imbalance, size and complexity applying three popular classifiers Naive Bayes, C4.5 and SVM. Results have shown that SVM outperforms the other two classifers. Barella et al. [14] have proposed a cluster based one sided selection method for undersampling. In [19], a similarity based hierarchical decomposition method is proposed to classify imbalanced data sets. Wing et al. [103] have proposed a diversified sensitivity-based undersampling method for imbalance classification. Other latest works [91] use ensembles of First Order logical Decision Trees to handle the problem of imbalanced classification, [9] uses feature weighting to deal with overlap in imbalanced datasets, [41] proposes a RandomBalance method for imbalanced data which uses ensembles of variable priors classifiers. In [121], Sun et al. proposed a novel ensemble method to classify imbalanced data sets.

5.2 Motivation

Since the clustering methods suffer from drawbacks listed earlier, a new method is proposed here to alleviate the problems with those methods. The suggested method balances the provided training set by selecting negatives from the full distribution with the least amount of information loss possible. That is, groups are established in such a way that samples are selected from the entire set of the majority class samples. Applying the stratified sampling approach, which determines the number of samples from each strata based on its size, the selected negatives operate as representatives of all the negatives in the training set.

The method's novelty is in its ability to generate groups with minimal computional complexity. The primary difference between [117, 142, 87] and the suggested technique is

that their work use clustering to generate groups, where the number of clusters, the clustering method, and the number of clusters all influence these strategies, whereas the proposed method does not use any clustering approach. In fact, the bulk of the majority class samples are separated according to how far they are from a set of reference point(s). The suggested method do not require any of these external parameters because all clustering approaches are based on the ideal number of clusters and cluster quality.

Difference between the proposed method and the other undersampling methods, namely CNN, CNNTL, NCL, OSS lies in the way of choosing the majority class instances. As all these methods choose/discard the majority class samples based on their distance from minority class sample, QUS chooses majority class samples based on their distance from the five reference points Neg_Min, Neg_Q1, Neg_Median, Neg_Q3 and Neg_Max in the case of QUS. kNN classifier is chosen to select the majority class samples in their methods, whereas the proposed method is not dependent on any classifier. This reduces lot of computation involved in finding nearest neighbors as the size of the data set increases.

5.3 Framework

Negative samples in the training set are sorted according to their distance from the reference points in this method. Then stratified sampling is used to select negatives from each group, with the number of samples selected from each group based on the group's size and the total number of negatives selected from all groups equal to the total of positives in the training set.

Let N_min be the quantity of $Minority$ class instances in the training set, and N_maj be the quantity of $Majority$ class instances. Using stratified sampling, the number of $Majority$ instances picked after group formation is equal to N_min. As a result, the training set has an equal number of minority and majority class instances. In each group $group_i$, g_i instances are picked so that the total number of $Majority$ class samples chosen are equal to the total number of $Minority$ class samples.

$$g_i = \frac{size(group_i)}{N_{maj}} N_{min} \quad 1 \leq i \leq k \tag{5.3.1}$$

$$Negative\ samples\ subset = \Sigma_i g_i \tag{5.3.2}$$

The contribution of different instances to classification changes based on their distance from the decision border. That is, internal(closer) examples have a greater influence than fringe occurrences (farther). However, more examples aid in a deeper understanding of the topic. As a result, the samples are picked with varied degree of classification influence to compensate for the left-over cases from the original training set, that is, the chosen negative samples should represent the whole negatives in the training set.

5.4 Algorithm

In statistics, quartiles are used to illustrate the distribution of a data collection. These points are used to identify outliers in machine learning. These are used as reference points, and the distance between the quartiles is employed to partition the occurrences in the training set. The idea is that all instances that are close to *min* are sorted into group1, those that are close to $Q1$ are grouped into group 2, those that are close to *median* are grouped into group 3, those that are close to $Q3$ are grouped into group 4, and those that are near to *max* are put into group 5. After that, instances equal to the size of the minority class are picked depending on the size of the groupings. The goal is to select majority class samples from throughout the distribution that have features that are closer to reference points. This is an attempt to select examples in a systematic manner rather than at random. The quartile distances are used to choose majority class samples in this strategy.Forming four groups out of samples that fall between min, Q1, etc., and Q3 to max, examples are grouped depending on how close they are to five of the quartiles. As a result, four groups are formed in QUS. The steps are shown in the 5.4 algorithm.

5.5 Experiments and Results

Pre-processing is done using QUS on the negative sets of the training sets in each fold to balance the positive and negative sets. The new balanced training sets are used to train the classifiers for each fold respectively. k-fold cross validation is used to compute efficiency

of the method. Average of the results combined together on the k test sets of the k-folds is taken as the output of the classifier. Here 5 folds are used. kNN classifier is chosen to validate the performance of the proposed undersampling method.

In order to verify the efficacy of the pre-processing technique proposed here, a varied set of data sets are chosen to take into account the effect of data complexities and attribute complexities. Broadly they can be categorized as sets with instances (low and high) and attributes (low and high), like small number of features with large number of instances, large number of features and large number of instances etc. Table 7.2 lists these data sets. Datasets considered are both binary and multi-class data sets. Multi-class data sets are made as binary by taking required class as positive and all other classes put together as negative. All the values in these data sets are numeric except the class attribute. The data sets are taken from the UCI [11], KEEL[7] data repositories and used KEEL[7], weka[138] tools to conduct the experiments. The data sets are chosen based on their number of attributes, number of instances and imbalance ratio to check the performance of the proposed method on datasets of different sizes and dimensions and also imbalance ratios.

5.5.1 Dataset

Table 5.1: Details of the data sets.

Name of the Data Set	# Features	Total # Instances	Imbalance Ratio
Ecoli4	7	459	14.3
Haberman	3	306	2.78
Iris0	4	150	2
Isolet5	617	1599	24.98
LibrasMove	90	360	14
Musk2	166	6599	5.49
NewThyroid1	5	215	5.14
Pageblocks0	10	5472	8.79
Pima	8	768	1.89
Scene	294	2407	12.6
Segment0	19	2308	6.02
Shuttlec4vsall	9	58000	5.51
Skin-Segmentation	3	245057	3.82
Spectrometer	93	7797	10.8
Vowel0	13	988	9.98
Yeast1289Vs7	8	947	30.57

The presented undersampling method QUS separates all negatives into five bins based on how close they are to the five reference points, which are *Negmin*, *NegQ*1, *Negmedian*, *NegQ*3, and *Negmax*. In the next step, a small number of negatives (*Nmin*) are picked up from each group using

stratified sampling, which implies the number of examples chosen from each group is proportional to it's size. kNN classifier is used test this method, which is then compared to various existing undersampling, oversampling, ensemble, and cost-based strategies utilising the same classifiers. Experimentation is carried out after standardising the attribute values in order to attain uniformity in the varying attribute range. Specifically, we use both regular data without standardisation experimentation and data with standardisation when providing input.

The main goal of the pre-processing strategy provided here is to increase the minority class's accuracy or prediction. It is clear from the Table 5.2 that the suggested approach QUS considerably enhances the accuracy of the *Minority* class, that is sensitivity. Except for a few data sets where there is a minor difference, all data sets show an improvement in sensitivity. The classification results obtained for several data sets using the pre-processing method QUS followed by kNN are shown in Table 5.2. Average values for the 5-fold cross validation performed on the data sets are shown in the tables. It has been discovered that the number of groups has little effect on the outcome. The proposed strategy improves the True Positive Rate (TPR) while simultaneously providing good AUC findings.

5.6 Discussion

The findings obtained by the proposed method are compared to those produced by various undersampling, oversampling, and ensemble approaches in this section.

5.6.1 Scalability

Experiments on large data sets were undertaken to determine the scalability of the approaches, including shuttlec4vsall with 58000 instances, 9 attributes, and 5.51 IR, and skin segmentation with 2,45,057 instances, 3 attributes, and IR 3.82. The AUC for both of them is 0.99 using the QUS approach. It is well established that if there are enough examples for training, the imbalance has little effect on classification accuracy. This is demonstrated by the two huge data sets mentioned above. In comparison to the suggested pre-processing method, the existing undersampling approaches take a long time. For small and medium data sets, the proposed technique and the other undersampling methods required nearly the same amount of time, however for huge data sets, the suggested technique took only a few minutes while the other undersampling approaches required hours altogether.

5.6.2 Comparison with Other Undersampling Methods

The proposed approach is compared with various undersampling approaches such as CNN, CNNTL, CPM, SBC, OSS, RUS, and TL, which are widely used in the literature. Many of these methods use a kNN classifier to choose the samples, whereas the proposed approach do not utilise any classifier at all. The findings from Table 5.3 show that the suggested technique beats current methods in a few data sets while achieving equivalent results in the remaining data sets. The proposed method has never been shown to be inferior to any of the existing popular undersampling strategies. Among the approaches utilised with kNN, Friedman test gives QUS-stand-Eucl the lowest ranking meaning that the method has best performance compared to the other methods.

The results produced for the data sets applying the alternative undersampling approaches for the kNN classifier are presented in the Table 5.3. It turns out that the proposed technique is the best among the undersampling approaches, including the currently proposed ways employing the post-hoc approach.

5.6.3 Comparison with Oversampling and Ensemble Methods

The proposed approach is compared against various class imbalance approaches available in KEEL, such as oversampling, ensemble based, algorithm based, and cost sensitive based, to see how well they perform. The tables Table 7.15, Table 7.17 demonstrate that the suggested approach is not inferior to any of the existing approaches and produces comparable results.

5.7 Summary

In this chapter, QUS, a simple and effective undersampling strategy is proposed for balancing the training set by selecting samples from the majority class that are equal to the number of samples from the minority class. In selecting majority class samples, the technique does not employ any particular clustering or classification algorithm. Majority of known undersampling approaches balance the data set using either (a) prototype selection or (b) clustering algorithms, both of which are parameter sensitive and difficult to achieve convergence whereas the proposed method is **parameter independent** and **simple**.

This method is based on distribution-specific categorization, and issues such as (i) information loss and (ii) proper representation of the majority class are also taken into consideration. The groups are established to prevent information loss by allowing samples to be selected from across the majority class. The problem of representative samples is solved by using a stratified sampling

technique, which selects the number of samples from each cluster based on its size, ensuring that the samples picked are representative of the majority class as a whole. Most undersampling approaches employ a kNN classifier to select majority class samples, but in this proposed method, majority class samples are not selected by using kNN Classifier. This reduces the computation required to find nearest neighbours as the data set grows in size. Furthermore, this approach works even for big data sets with high dimensionality.

A valuable insight is gained by this method. QUS splits the data set from a global view, as opposed to kNN or clustering algorithms. Unique to a distribution in the case of Euclidean distance-based approaches, data grouping is equivalent to a global partition of the neighbour-hood into a few groups. As a result, when compared to clustering, where a local neighbourhood determines the clusters and there is a chance of missing some disjuncts entirely, the likelihood of selecting samples from the spherical global neighbourhood is higher. All kNN-based under-sampling approaches focus on a small neighbourhood (K=1,3 or 5). This could result in a significant amount of prejudice. The theory is also supported by empirical evidence. Though other previous approaches produced better results for specific datasets, the main advantage of QUS is that it is simple to implement and hence takes less time, $O(n)$, than other undersampling methods. It is also independent of any of the input parameters and works well with data sets with large instances, large features, small instances, and small features. To our knowledge, this type of grouping used in the suggested method has never been applied in undersampling to improve the classification of imbalanced data sets before.

The experimental results show that this method produces superior outcomes on datasets with a high imbalance ratio. Another finding is that the suggested QUS technique works well on datasets with high dimensionality, as well as large datasets. In cases when the imbalance ratio is more than 10, the proposed technique outperforms the original 100% training set. The same is true when comparing undersampling methods.

Algorithm 5.4 Quartile based UnderSampling

Let D be a binary class dataset $\langle X,Y\rangle$, *where* $X = X_1,X_2,\ldots,X_n$, *each* X_i *is a* $m-$ *dimensional Vector with* m *attributes and is associated with a label* $Y = 0,1$
N represents the size of the dataset
N_{Min} *represents the size of Positives in the dataset*
N_{Maj} *represents the size of Negatives in the dataset*
$\frac{N_{Min}}{N_{Maj}} > 1.5$
$Neg_{Min} = Minimum(dist)$
$Neg_{Q1} = First\ Quartile\ of(dist)$
$Neg_{Median} = Median\ of(dist)$
$Neg_{Q3} = Third\ Quartile\ of(dist)$
$Neg_{Max} = Maximum(dist)$

procedure QUARTILE BASED UNDERSAMPLING(ITS)
  for $j \in N_{Maj}$ **do**
    $Neg_{Cent_j} = \sum_{i=1}^{N_{Maj}} X_{ij}/N_{Maj}$
  end for
  for $i \in N_{Maj}$ **do**
    $dist_i = \sqrt{\sum_{i=1}^{N_G}(X_{ij} - Neg_{cent_j})^2}$
  end for ▷ *Distance of all Negatives from* Neg_C*entroid is found*
  for $i \in 1\ to\ N_{Maj}$ **do**
    $dist_i = \frac{dist_i - min(dist)}{max(dist) - min(dist)}$
  end for ▷ $max()$ *gives maximum* $-$ *value among the given input distance values*
    ▷ $min()$ *gives minimum* $-$ *value among the given input distance values*
  for $i \in \{1\ to\ N_Maj\}$ **do**
    for $j \in \{1\ to\ 4\}$ **do**
      if $(\ dist_i \geq Neg_{Min_i}\ ) \wedge (\ dist_i \leq (Neg_{Q1_i})\ )$ **then**
        $Group_j \leftarrow X_i$
      else if $(\ dist_i \geq Neg_{Q1_i} \wedge dist_i \leq Neg_{Median_i}\ )$ **then**
        $Group_j \leftarrow X_i$
      else if $(\ dist_i \geq Neg_{Median_i} \wedge dist_i \leq (Neg_{Q3_i})\ )$ **then**
        $Group_j \leftarrow X_i$
      else if $(\ dist_i \geq Neg_{Median_i} \wedge dist_i \leq (Neg_{Q3_i})\ )$ **then**
        $Group_j \leftarrow X_i\ (\ dist_i \geq Neg_{Q3_i} \wedge dist_i \leq (Neg_{Max_i})\ )$
        $Group_j \leftarrow X_i$
      end if
    end for
  end for
  for $j \in \{1\ to\ 4\}$ **do**
    $r_k = random(\frac{size(Group_k)}{N_{Maj}} N_{Min})\ \ 1 \leq i \leq k$ ▷ Pick r_k *instances from* $Group_k$
  end for
  $N_{Maj-New} = \Sigma_k r_k$
  $BalancedTraining = N_{Min} + N_{Maj-new}$
end procedure

Table 5.2: AUC results with **kNN** classifier.Number of groups are 4 (fixed because groups are formed between reference points are min,Q1,median,Q3,max).

Data Set	Measure	No Sampling	QUS-Stand-Eucl
Ecoli4	Sensitivity	0.75	0.9
	Specificity	0.99	0.89
	AUC	0.87	**0.9**
Haberman	Sensitivity	0.34	0.54
	Specificity	0.80	0.6
	AUC	**0.57**	**0.57**
Iris0	Sensitivity	1.00	1.00
	Specificity	1.00	1.00
	AUC	1.00	**1.00**
NewThyroid1	Sensitivity	0.97	0.99
	Specificity	0.98	0.97
	AUC	0.97	**0.98**
Pima	Sensitivity	0.52	0.66
	Specificity	0.81	0.7
	AUC	0.66	**0.68**
Vowel0	Sensitivity	1.00	1.00
	Specificity	1.00	0.96
	AUC	**1.00**	0.98
Yeast1289vs7	Sensitivity	0.13	0.72
	Specificity	0.97	0.6
	AUC	0.55	**0.66**
PageBlocks0	Sensivitiy	0.76	0.9
	Specificity	0.98	0.92
	AUC	0.87	0.91
Skin-Segmentation	Sensitivity	0.99	0.99
	Specificity	0.99	0.99
	AUC	**0.99**	**0.99**
Shuttlec4vsall	Sensitivity	0.99	0.99
	Specificity	0.99	0.99
	AUC	**0.99**	**0.99**
Segment0	Sensitivity	0.99	0.99
	Specificity	0.99	0.98
	AUC	**0.99**	**0.99**
Isolet5	Sensitivity	0.88	0.99
	Specificity	0.99	0.80
	AUC	**0.94**	0.90
LibrasMove	Sensitivity	0.70	0.92
	Specificity	0.99	0.91
	AUC	0.85	**0.91**
Musk2	Sensitivity	0.87	0.93
	Specificity	0.96	0.91
	AUC	**0.92**	**0.92**
Scene	Sensitivity	0.14	0.6
	Specificity	0.95	0.67
	AUC	0.55	**0.63**
Spectrometer	Sensitivity	0.73	0.87
	Specificity	0.99	0.95
	AUC	0.86	**0.91**

Table 5.3: Comparison of AUC results of proposed methods with **other undersampling techniques** using **kNN**. (- *indicate results not obtained even after 300 seconds*)

data set	CNN (1968)	CNNTL (2004)	CPM (2005)	SBC (2006)	NCL (2001)	OSS (1997)	RUS (2004)	TL(1976)	**Q**US-Stand-Eucl
Ecoli4	0.91	0.89	0.69	0.50	0.86	0.88	**0.95**	0.87	0.9
Haberman	0.54	0.56	0.53	0.59	0.57	0.53	**0.62**	0.58	0.57
Iris0	1.00	1.00	1.00	0.50	1.00	1.00	1.00	1.00	**1.00**
Newthryroid1	0.97	0.97	0.94	0.50	**0.99**	0.96	0.96	0.97	0.98
Pima	0.64	0.65	0.62	0.67	**0.70**	0.67	0.66	**0.70**	0.68
Vowel0	0.99	0.99	0.96	0.75	**1.00**	0.99	0.97	**1.00**	0.98
Yeast1289vs7	0.56	0.65	0.59	0.50	0.59	0.63	0.62	0.55	**0.66**
PageBlocks0	0.86	0.88	0.86	0.89	0.91	0.89	**0.91**	0.89	**0.91**
Skin-segmentation	-	-	-	-	-	-	-	-	**0.99**
Shuttlec4vsall	-	-	-	-	-	-	-	-	**0.99**
Segment0	**0.99**	**0.99**	0.97	0.50	**0.99**	**0.99**	0.98	**0.99**	**0.99**
Isolet5	**0.97**	0.96	0.87	0.56	0.94	0.96	0.89	0.94	0.90
LibrasMove	0.85	0.93	0.86	0.50	0.92	0.86	**0.93**	0.85	0.91
Musk2	0.88	0.83	0.88	0.50	0.83	0.87	0.88	0.83	**0.92**
Scene	0.58	0.59	0.56	0.50	0.61	0.60	**0.65**	0.58	0.63
Spectrometer	0.91	0.90	0.90	0.67	0.90	0.88	**0.93**	0.88	0.91

Table 5.4: Comparison of AUC results of proposed method with **OverSampling techniques** using **kNN**. (- *indicate results not obtained even after 300 seconds*)

Data set	ADASYN (2008)	ADOMS (2008)	Borderline-SMOTE (2005)	ROS (2004)	SafeLevel-SMOTE (2009)	SMOTE-TL (2004)	SMOTE (2002)	**QUS-Stand-Eucl**
Ecoli4	0.90	0.91	0.89	0.87	0.87	0.92	**0.93**	0.9
Haberman	0.54	0.57	0.58	0.54	0.54	0.59	0.58	0.57
Iris0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Newthryroid1	0.97	0.97	0.97	0.97	0.97	0.97	0.95	**0.98**
Pima	0.67	0.67	0.67	0.66	0.66	**0.72**	0.66	0.68
Vowel0	0.99	1.00	1.00	1.00	1.00	0.99	0.99	0.98
Yeast1289vs7	0.60	0.58	0.59	0.55	0.55	0.63	0.60	**0.66**
PageBlocks0	0.89	**0.92**	0.91	0.87	0.86	0.91	0.91	0.91
Skin-segmentation	-	-	-	-	-	-	-	**0.99**
Shuttlec4vsall	-	-	-	-	-	-	-	0.99
Segment0	0.99	0.99	0.99	0.99	0.99	0.99	0.99	**0.99**
Isolet5	0.96	-	**0.98**	0.94	0.79	0.98	0.97	0.90
LibrasMove	-	-	-	-	-	-	0.91	0.91
Musk2	0.92	-	-	0.92	0.91	-	0.91	0.92
Scene	-	-	-	-	-	-	-	**0.63**
Spectrometer	-	-	-	-	-	-	-	**0.91**

Table 5.5: Comparison of AUC results of proposed methods with **some Ensemble Methods, Cost Sensitive and Algorithm based Methods**. (- *indicate results not obtained even after 300 seconds*.)

Data set	Balance Cascade (2009)	Easy Ensemble (2009)	AdaC2 (2007)	CSVMC (2009)	SC45CS (2002)	NNCS (2006)	**QUS-Stand-Eucl-kNN**
Ecoli4	0.84	0.85	0.92	**0.95**	0.86	0.87	0.90
Haberman	0.61	0.65	0.56	0.61	0.57	**0.62**	0.57
Iris0	0.99	0.99	0.99	1.00	0.99	1.00	1.00
Newthryroid1	0.93	0.93	0.94	0.98	0.97	0.82	**0.98**
Pima	0.69	0.73	0.70	**0.74**	0.71	0.69	0.68
Vowel0	0.94	0.94	-	0.97	0.94	0.68	**0.98**
Yeast1289vs7	0.65	0.65	0.63	-	0.67	0.51	**0.66**
PageBlocks0	0.95	**0.95**	0.88	-	0.94	0.76	0.91
Skin-segmentation	-	-	-	-	-	0.85	**0.99**
Shuttlec4vsall	-	-	-	-	-	-	**0.99**
Segment0	0.98	0.98	0.98	0.99	0.99	0.50	**0.99**
Isolet5	-	-	-	-	-	0.50	**0.90**
LibrasMove	-	-	-	-	-	0.50	**0.89**
Musk2	-	-	-	-	-	0.57	**0.92**
Scene	-	-	-	-	-	-	**0.63**
Spectrometer	-	-	-	-	-	-	**0.91**

Chapter 6

MahalCUSFilter: A Hybrid Undersampling method

Undersampling is a data-level approach that preprocesses the data set to minimise the number of the majority class instances, which is one of the approaches for dealing with the problem of class imbalance. To balance the data set, most existing undersampling methods use prototype selection or clustering algorithms. Both techniques are efficient and popular, yet they are both complicated. The disadvantage of prototype selection methods is that they must compare each majority instance with its k closest neighbours to determine which majority class instance should be selected or rejected, which is time consuming and difficult to implement for big datasets.

The nature of all real-world datasets is multivariate. As a result, a multivariate dataset distance metric should take into account not only the variances of the attributes, but also their covariances or correlations. In some cases, the Euclidean distance between two vectors is ineffective since no adjustment for variances or covariances is possible. As a result, a statistical distance, or standardised measure is used.

6.1 Related Work

Recent papers on imbalanced data sets can be found in [132, 148, 19, 103, 91, 14]. To efficiently categorise unbalanced data, Wang et al. in [132] employ an ensemble technique, weights, and information on sample misclassification. Zhang et al. [148] offer empirical study by using three common classifiers, Naive Bayes, c4.5, and SVM, to conduct numerous tests on unbalanced data sets with different imbalance, size, and complexity. SVM outperforms the other two classifiers,

according to the results. Barella et al. introduced a cluster-based one-sided selection strategy for undersampling in [14]. To categorise imbalanced data sets, [19] proposed a similarity-based hierarchical decomposition strategy. Wing et al. suggested a diversified sensitivity-based undersampling approach for imbalance classification in their paper [103].

The proposed undersampling method **Mahalanobis Centroid based Undersamping with Filter(MahalCUSFilter)** solves the concerns mentioned above: *Parameter dependence, Variables Interdependece, Scale Invariants and information loss* factors. The proposed strategy was found to enhance the minority class classification rate of all datasets with comparable overall performance for the entire dataset when used in conjunction with c4.5 and kNN classifiers. This type of grouping has not been employed in undersampling to increase the classification accuracy of imbalanced data sets, according to what has been learned from the literature.

The covariance between variables is used to calculate the Mahalanobis distance. It benefits from the use of group means and variances for each variable, as well as the inherent scale and correlation issues. In chemometrics and multivariable statistics, the Mahalanobis distance is one of the most widely used measures. It can be used to see if a sample is an outlier, if a process is under control, or if a sample belongs to a group or not. The last point mentioned above, namely whether a sample is a member of a group or not, is applicable in the suggested method.

6.2 Motivation

Mahalanobis distance was first proposed by Mahalanobis in 1936 and is often referred to as Mahalanobis distance [26]. In a Mahalanobis distance, a random variable with a higher volatility receives less weight than others. In the case of mahalanobis distance, two highly correlated variables do not contribute more than the two less correlated factors. The idea of the Mahalanobis distance measure is to employ the inverse of the covariance matrix, which has the effect of normalising all variables to the same variance and eliminating correlations Alvin. Mahalanobis distance is found using d_{Mahal} (Equation 6.2.1)

$$d_{Mahal} = \sqrt{(\overrightarrow{x} - \overrightarrow{\mu})^T S^{-1} (\overrightarrow{x} - \overrightarrow{\mu})} \tag{6.2.1}$$

where x is the instance vector, μ is the mean vector and S is the covariance matrix. In the formula Inverse of covariance matrix is used to calculate Mahalanobis distance.

Mahalanobis Centroid based UnderSampling with Filter (MahalCUSFilter) is a method for capturing majority class samples based on their resemblance to the average behaviour of all majority

class instances. To compute the distances between the reference point (mean-vector i.e., centroid) and each majority class instance in the data set, the Mahalanobis distance metric is used. This function selects majority class samples depending on their distance from their Mean-Vector(C_cent), that is, samples from closer to farther distances to reflect the distribution of majority class instances.

In the final stage, majority class instances that are close to minority class instances are filtered out, leaving only the training set. Classifying using the 1NN classifier, misclassified examples are found within the balanced training set created using the MahalCUS technique. The instances of the majority class that were misclassified are eliminated from the training set because they are risky, implying that they are likely to confuse the minority and majority classes.

By removing risky instances, this will attempt to ensure a clear boundary between minority and majority class instances. The algorithm 6.5 shows the steps taken in the pre-processing. The number of groupings is picked at random. O(n) is the complexity of MahalCUSFilter algorithm as the inner loops of the algorithm run for constant number of times.

Existing undersampling approaches have three fundamental shortcomings, which the current method addresses:

- **Parameter Dependence**: The performance of MahalCUSFilter is independent of the settings set by the user, unlike cluster-based and kNN-based undersampling approaches, which are dependent on the clustering algorithm, number of clusters, k-value, and other factors. Experimenting with different numbers of bins yielded little variance in the results.
- **Variables Inter dependence**: Unlike other algorithms that use euclidean distance to find distance/similarity between instances which do not consider inter dependencies, correlations among the variables of a dataset, MahalCUSFilter uses Mahalanobis Distance measure to find distance between each majority class instance with its centroid (Mean of the Majority class instances) which takes into account correlation among variables of a dataset.
- **Information loss**: The issue of majority class representation is handled by using a stratified sampling approach, which selects the number of samples from each group based on its size, ensuring that the samples picked are representative of the majority class as a whole.
- **Scale variants**: A dataset's variables are measured in different units and have a diverse range of values. Existing algorithms, on the other hand, use euclidean distance estimates that ignore these issues. To address this problem, the suggested method employs the Mahalanobis distance, which renders the method Scale-Invariant.

This chapter discusses the method's novel properties in comparison to other popular and recent techniques. The primary difference between Parinaz,Lee,Rushi and the proposed technique is that the former uses clustering to construct groups, whilst the latter does not. Instead, the majority class samples are divided depending on their distance from the reference point.

6.3 Framework

The classifiers are trained using the new balanced training sets obtained after pre-processing with MahalCUSFilter. The experimentation employs five-fold cross-validation. The classifier's output is the average of the results acquired on five test sets. To test the proposed method's performance, two classifiers, C4.5 and kNN are chosen.

MahalCUSFilter separates all instances of the majority class into m bins based on their distance from their centroid(Neg_cent). The number of negatives picked in the second phase (N_Min) is determined by stratified sampling, which means that the number of examples chosen from each bin is determined by the size of the group and the total number of majority class instances chosen from all groups is equal to the number of minority class examples in the training set. C4.5 and kNN classifiers are used to test this method, which is then compared against other existing undersampling strategies using the same classifiers.

Let N_{Min} and N_{Maj} represent the training set's $Minority$ and $Majority$ class instances, respectively. To balance the minority and majority instances, N_{Min} majority examples are selected based on stratified sampling once the groups are formed. In each group, r_i examples are picked so that the total number of $Majority$ class samples chosen equals the total number of $Minority$ class samples.

$$r_i = \frac{size(group_i)}{N_{Maj}} N_{Min} \quad 1 \leq i \leq k \tag{6.3.1}$$

$$Negative\ samples\ chosen = \Sigma_i r_i \tag{6.3.2}$$

6.4 Algorithm

6.5 Experiments and Results

6.5.1 Details of the Datasets

In the experiments, binary class data sets are employed. Multi-class data sets are transformed to binary class by assigning the required class to the minority class and the remaining classes to

the majority class. To test the performance of the proposed approaches on small, medium, and large numbers of attributes, instances, and imbalance ratio, the data sets are chosen based on their number of attributes, number of instances, and imbalance ratio. The data sets are provided in Table 7.2. Except for the class attribute, all of the values in these data sets are numeric. The data sets were obtained from the UCI [11], KEEL[7] data repositories, and the tests were carried out using the KEEL[7], WEKA[138] tools.

Table 6.1: Details of the data sets.

Name of the Data Set	# Features	Total # Instances	Imbalance Ratio
Ecoli4	7	459	14.3
Haberman	3	306	2.78
Iris0	4	150	2
LibrasMove	90	360	14
NewThyroid1	5	215	5.14
Pima	8	768	1.89
Scene	294	2407	12.6
Spectrometer	93	7797	10.8
Yeast1289Vs7	8	947	30.57

6.6 Discussion

The proposed MahalCUSFilter method is compared to various undersampling methods in the literature, such as **CNN, CNNTL, CPM, OSS, TL** which are common among undersampling methods in the literature. The proposed method differs from the others in the manner in which the majority class instances are picked. They select and discard majority class samples based on their distance from minority class samples, whereas MahalCUSFilter selects majority class samples based on their distance from the reference point Centroid(Neg_Cent). They choose majority class samples using a kNN classifier but the proposed approaches do not, lowering the computation required to find nearest neighbours as the size of the data set grows. Table 6.2, show that the suggested technique outperforms current methods in a few data sets while achieving equivalent results in the remaining data sets. In no case, the proposed approach has been shown to be inferior in classifying minority class cases to any of the existing popular undersampling methods.

6.7 Summary

MahalCUSFilter is a hybrid undersampling method to balance the training set by selecting samples from the majority class equal to the number of samples from the minority class. To balance the

Table 6.2: Comparison of Sensitivity, GMean and Balanced Accuracy results with **c4.5** classifier with Unprocessed Original training set, MahalCUSFilter and other popular undersampling methods.

Data Set	Measure	Original	**M**ahalCUSFilter	CNN (1968)	CNNTL (2004)	CPM (2005)	NCL (2001)	OSS (1997)	TL (1976)
Ecoli4	Sensitivity	0.56	0.94	0.75	0.85	0.70	0.65	0.80	0.65
	GMean	0.75	0.87	0.83	0.85	0.81	0.80	0.84	0.80
	Balanced-Accuracy	0.77	0.87	0.83	0.85	0.81	0.81	0.84	0.81
Haberman	Sensitivity	0.40	0.48	0.54	0.72	0.46	0.74	0.56	0.46
	GMean	0.57	0.57	0.62	0.58	0.59	0.62	0.63	0.61
	Balanced-Accuracy	0.62	0.60	0.63	0.60	0.61	0.63	0.64	0.63
Iris0	Sensitivity	0.98	0.98	0.94	0.94	0.80	0.98	0.94	0.98
	GMean	0.99	0.99	0.97	0.97	0.69	0.99	0.97	0.99
	Balanced-Accuracy	0.99	0.99	0.97	0.97	0.70	0.99	0.97	0.99
NewThyroid1	Sensitivity	0.91	0.91	0.94	0.97	0.94	0.91	0.94	0.86
	GMean	0.95	0.93	0.94	0.92	0.81	0.94	0.94	0.92
	Balanced-Accuracy	0.95	0.93	0.94	0.92	0.82	0.94	0.94	0.92
Pima	Sensitivity	0.62	0.77	0.76	0.89	0.51	0.85	0.78	0.69
	GMean	0.70	0.74	0.72	0.62	0.65	0.70	0.66	0.71
	Balanced-Accuracy	0.71	0.74	0.72	0.66	0.67	0.72	0.67	0.71
Spectrometer	Sensitivity	0.74	0.83	0.84	0.89	0.82	0.76	0.84	0.78
	GMean	0.85	0.84	0.81	0.79	0.83	0.85	0.81	0.87
	Balanced-Accuracy	0.86	0.84	0.81	0.80	0.83	0.86	0.81	0.88
Yeast1289vs7	Sensitivity	0.24	0.57	0.20	0.27	0.27	0.07	0.23	0.10
	GMean	0.42	0.51	0.45	0.51	0.52	0.26	0.48	0.31
	Balanced-Accuracy	0.62	0.59	0.60	0.62	0.63	0.53	0.61	0.54
LibrasMove	Sensitivity	0.63	0.88	0.88	0.79	0.58	0.58	0.79	0.67
	GMean	0.78	0.79	0.84	0.78	0.70	0.75	0.79	0.81
	Balanced-Accuracy	0.80	0.80	0.84	0.78	0.71	0.78	0.79	0.83
Scene	Sensitivity	0.23	0.61	0.47	0.55	0.35	0.28	0.41	0.24
	GMean	0.47	0.61	0.59	0.59	0.54	0.51	0.30	0.47
	Balanced-Accuracy	0.59	0.61	0.61	0.59	0.50	0.60	0.32	0.59

data set, most known undersampling approaches use either (a) prototype selection or (b) clustering algorithms, both of which are parameter dependent and difficult to achieve convergence.

In this chapter, **Scale-Invariant**, **Variables-Correlation Inherent** and **Parameter Independent** algorithm, MahalCUSFilter is proposed. It focuses on issues such as (i) information loss and (ii) proper representation of the majority class are also taken into account. Furthermore, even for high-dimensional and big data sets, the method is straightforward to implement and effective.

MahalCUSFilter splits the data set from a global perspective, unlike kNN or clustering algorithms. Specific to a distribution in the case of Euclidean distance-based approaches, data grouping is a form of circular neighbourhood. As a result, when compared to clustering, where a local neighbourhood determines the clusters and there is a risk of missing some disjuncts entirely, the likelihood of selecting samples from the globular neighbourhood is higher. All kNN-based undersampling approaches consider a relatively small neighbourhood (K=1,3 or 5). This could lead to

Table 6.3: Comparison of Sensitivity, GMean and Balanced Accuracy results with **kNN(k=1)** classifier with Unprocessed Original training set, MahalCUSFilter and other popular undersampling methods.

Data Set	Measure	Original	**M**ahalCUSFilter	CNN (1968)	CNNTL (2004)	CPM (2005)	NCL (2001)	OSS (1997)	TL (1976)
Ecoli4	Sensitivity	0.69	1.00	0.85	0.85	0.40	0.75	0.80	0.75
	GMean	0.82	0.89	0.91	0.90	0.63	0.85	0.88	0.86
	Balanced-Accuracy	0.84	0.89	0.91	0.90	0.69	0.86	0.88	0.87
Haberman	Sensitivity	0.50	0.53	0.46	0.69	0.36	0.68	0.53	0.51
	GMean	0.53	0.54	0.54	0.54	0.50	0.56	0.53	0.58
	Balanced-Accuracy	0.53	0.55	0.54	0.56	0.53	0.57	0.53	0.58
Iris0	Sensitivity	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	GMean	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Balanced-Accuracy	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
NewThyroid1	Sensitivity	0.97	1.00	0.97	1.00	0.91	1.00	0.97	0.97
	GMean	0.98	0.99	0.97	0.98	0.95	0.99	0.97	0.98
	Balanced-Accuracy	0.98	0.99	0.97	0.98	0.95	0.99	0.97	0.98
Pima	Sensitivity	0.53	0.69	0.64	0.89	0.54	0.83	0.84	0.74
	GMean	0.65	0.70	0.65	0.56	0.64	0.70	0.68	0.73
	Balanced-Accuracy	0.67	0.70	0.65	0.62	0.65	0.71	0.70	0.73
Spectrometer	Sensitivity	0.74	0.85	0.87	0.89	0.84	0.82	0.80	0.78
	GMean	0.85	0.90	0.91	0.90	0.91	0.90	0.88	0.88
	Balanced-Accuracy	0.86	0.91	0.92	0.90	0.91	0.90	0.88	0.88
Yeast1289vs7	Sensitivity	0.14	0.70	0.33	0.67	0.27	0.23	0.47	0.13
	GMean	0.32	0.66	0.51	0.65	0.50	0.47	0.61	0.36
	Balanced-Accuracy	0.95	0.65	0.56	0.65	0.60	0.59	0.64	0.55
LibrasMove	Sensitivity	0.71	0.67	0.75	0.92	0.75	0.88	0.75	0.71
	GMean	0.84	0.80	0.85	0.93	0.86	0.92	0.85	0.84
	Balanced-Accuracy	0.85	0.81	0.86	0.93	0.87	0.92	0.86	0.85
Scene	Sensitivity	0.17	0.60	0.37	0.63	0.25	0.34	0.44	0.24
	GMean	0.41	0.59	0.54	0.59	0.47	0.55	0.58	0.47
	Balanced-Accuracy	0.56	0.59	0.58	0.59	0.56	0.61	0.60	0.58

a lot of bias. Furthermore, when computing the distance between each sample and its class-mean, correlation among variables is taken into account (centroid). In the case of multi-variable datasets, this is really desirable.

Empirical evidence backs up the notion. MahalCUSFilter significantly improves minority class classification rate on all datasets, compared to unprocessed original imbalanced datasets. When only a few datasets were compared, other well-known undersampling approaches produced better results. However, such techniques have a larger time and space complexity than MahalCUSFilter.

The following are some of the benefits of the proposed technique: It is variables-correlation intrinsic, taking into account inter dependencies of variables in a dataset, which is extremely important when working with multi-variate datasets. It is independent of any input parameters; it works with data sets with large instances, large features, small instances, and small features; and it uses Mahalanobis distance measure to balance the training set, unlike existing undersampling methods

that use euclidean distance to find the distance/similarity in the process of selecting or discording majority class instances to balance the training set. The type of grouping used in the suggested methodologies hasn't been used in undersampling to improve the classification of imbalanced data sets yet, according to the literature.

Algorithm 6.5 Mahalanobis Centroid based UnderSampling with Filter

Let D be a binary class dataset $\langle X, Y \rangle$, where $X = X_1, X_2, \ldots, X_n$, each X_i is a $m-$ dimensional Vector with m *attributes and is associated with a label $Y = 0,1$*
N represents the size of the dataset
N_{Min} represents the size of Positives in the dataset
N_{Maj} represents the size of Negatives in the dataset
$\frac{N_{Min}}{N_{Maj}} > 1.5$
$N_G = 10$, N_G Represents Number of Groups

procedure MAHALCUSFILTER(ITS) ▷ ITS, *anImbalancedTrainingSet*
  for $j \in N_{Maj}$ **do**
    $Neg_{cent_j} = \sum_{i=1}^{N_{Maj}} X_{ij}/N_{Maj}$
  end for
  $DistMahal = \sqrt{(\overrightarrow{X} - \overrightarrow{\mu})^T S^{-1} (\overrightarrow{X} - \overrightarrow{\mu})}$ ▷ μ *is* Neg_{Cent}, S *is CoVariance matrix of Negatives*
  $DistMahal_i = \frac{DistMahal_i - min(DistMahal)}{max(DistMahal) - min(DistMahal)}$ ▷ *DistMahal is Normalized* ▷ *max() gives maximum − value among the given input distance values*
  ▷ *min() gives minimum − value among the given input distance values*
  for $i \in \{1\ to\ N_Maj\}$ **do**
    for $j \in \{1\ to\ N_G\ in\ steps\ of\ 1\}$ **do**
      for $k \in \{0\ to\ 1\ in\ steps\ of\ (1/\mathrm{N}_G)\}$ **do**
        if $(DistMahal_i \geq \mathrm{k}) \wedge (DistMahal_i \leq k + (1/N_G))$ **then**
          $Group\,j \leftarrow X_i$
        end if
      end for
    end for
  end for
  for $k \in \{1\ to\ N_G\ in\ steps\ of\ 1\}$ **do**
    $r_k = \frac{size(Groupk)}{N_{Maj}} * N_{Min}$
    $group_k = random(Groupk, r_k)$ ▷ *Randomly Pick* r_k *instances from Groupk*
  end for
  $N_{Maj-New} = \Sigma_{k=1}^{N_G} group_k$
  $New - Training = N_{Min} + N_{Maj-new}$
  $Balanced - Training = NewTraining - MCI$ ▷ *MCI are misclassified instances*
end procedure

Chapter 7

Hybrid Multi Objective Optimization Method (SAUS)

The classification process is hugely affected by the training data employed for training. The problem is determining the appropriate training set. Another key challenge is determining which collection of cases comprises a training set that allows the classifier to generalise well. Traditional classifiers work under the assumption that the training sets' classifications are evenly distributed. As a result, the cost of misclassification is the same for all classes, and accuracy is used to evaluate the classifier's performance, taking into consideration both classes' correct classification rate equally. [65, 66]. Traditional classifiers suffer from biased classification towards the majority class, resulting in a low minority class prediction rate, making learning from imbalanced datasets a difficult topic in machine learning research. The reasons for this poor performance have been recognised as the fundamental assumptions of equal class distribution and accuracy-driven evaluation. Furthermore, false negatives are penalised more severely than false positives. To address this problem, a straightforward logical answer is to create a balanced training set from the imbalanced one. For a given imbalanced set, however, numerous such balanced training sets can be produced, from which an ideal balanced training set must be obtained. This is a computationally hard problem with a high likelihood of local-optimal maxima and minima.

Meta-heuristics can be used to obtain such balanced set. In the case of meta-heuristic approaches, a candidate solution is improved iteratively using a provided quality metric. While meta-heuristics do not guarantee an optimal solution, they do aid in the development of near-optimal solutions. Meta-heuristics look for candidate solutions in very wide spaces and usually make no or few assumptions about the optimization problem, according to [4].

7.1 Related Work

Zhi et al. presented the SNGEIP technique [31], with the main goal of producing diversity by using sample generation to create distinct training sets for different base classifiers while also regulating the quantity of generated samples to balance the class distribution.

Using over-sampling and instance selection strategies to learn from imbalanced data, Ireneusz et al., [36] presented a hybrid methodology. Pawel combined the Random Subspace approach and stochastic oversampling to solve the problem of imbalanced data categorization in [77].

The paper [12] describes a multi-objective optimization technique based on simulated annealing that adds the idea of archive to achieve trade-off solutions to challenges. The article by[10] discusses several principles and variants of algorithms on multi-objective Simulated Annealing.

7.2 Motivation

To balance the imbalanced data sets, the most of under-sampling approaches use either (i)Exhaustive Search Methods or (ii) Sampling Methods. The first set of techniques has two flaws. The first is that, because they are based on nearest neighbour approaches, determining nearest neighbours takes time. The second is that as the number of instances or attributes in the data set grows, the time it takes to compute k nearest neighbours for each of the majority class samples grows. The second group of approaches have some limitations as well, such as cluster formation. This necessitates decision about clustering methodology, the number of clusters to be produced, and whether or not external or internal validity indices should be used. Simulated annealing is employed in this study to address these concerns.

7.2.1 Simulated Annealing: A General Approach

Simulated annealing is based on the physical annealing process, in which metals are melted at high temperatures and then cooled gradually until they achieve a stable condition [71]. Simulated annealing attempts to settle into a final state with the least amount of energy. The functional form that captures this is the energy level, which is analogous to valley descent. Physical entities typically shift from high to low energy levels, hence valley descent is a natural result. However, there is a chance of a probable transition to a higher energy level.

This probability is given by the function given in (Equation 7.2.1), where ΔE is the positive change in the energy level, T is the temperature, and K is Boltzmann's constant.

$$P = e^{-\Delta E/kT} \tag{7.2.1}$$

where ΔE is the positive change in the energy level, T is the temperature, and K is Boltzmann's constant.

As a result, the likelihood of a large uphill motion during annealing is smaller than the probability of a small downhill motion. In addition, as the temperature drops, the likelihood of an upward move reduces. As a result, such movements are more likely in the beginning of the process when the temperature is high, and they become less likely as the temperature drops to lower levels. This approach can be described as allowing downhill moves at any time. Large upward moves are permitted in the early phases of the process, but as the process proceeds, only relatively tiny upward moves are permitted until the process converges to a local minimum configuration[71].

7.2.2 Simulated annealing for finding a best solution

To find a good solution to an optimization problem, simulated annealing is performed. Simulated annealing can be applied in circumstances where an objective function must be maximised or minimised. Simulated annealing, unlike many other optimization methods such as genetic algorithms, gradient descent, hill climbing, and so on, avoids getting stuck in a local optimum. This method produces a solution that is closer to the optimal answer than any other method, while it is not the greatest. Typically, an optimization algorithm obtains the optimum solution by producing a random initial solution and then searching the neighbourhood. If an adjacent solution is better than the present one, the existing one is updated or kept. However, this may make it to get stuck in a less-than-ideal spot, such as the local maximum/minimum. Simulated annealing infuses an appropriate amount of randomness into objects to allow them to escape local optimums early in the process without straying too far from the solution later on. This enables it to locate a potential solution regardless of its starting place.

7.3 Framework

To reduce the rate of misclassification in the event of unbalanced data sets, a balanced training set must be chosen from the original data set. Simulated annealing is presented to achieve optimised subset selection consisting of majority examples equal to the number of minority examples.

Instead of using the error rate as the objective (cost) function, the multi-objective optimization function is Balanced Error Rate. In the event of imbalanced data sets, accuracy alone may not yield accurate conclusions because it reflects bias towards the majority class, even if zero minority class cases are accurately identified. As a result, the Balanced Error Rate is utilised because it is an effective indicator for imbalanced data issues since it reflects the impact of the imbalance, that is, it considers both majority and minority classification rates when evaluating performance.

The Balanced Error Rate is used as the cost function in this chapter to achieve multi-objective optimization of both minority class and majority class classification rates. The cost function Balanced Error Rate= (1-Balanced Accuracy) must be *minimized*, as specified in the algorithm. Balanced Accuracy is a useful metric for assessing the performance of binary class data sets because it is calculated as the average of the proportion of true classifications for each class independently. Parameters of Simulated Annealing used for experimentation are given in the Table 7.1. These values are picked from the literature [71]. Simulated Annealing based UnderSampling Algorithm (SAUS) is given in Algorithm 7.6 and diagrammatically represented in the Figure 7.1 and the complexity of SAUS is $O(k*n2)$.

In Figure 7.7, SAUS process is depicted. The process is initialised using initial values shown in Table 7.1. Figure 7.2 shows the initial training set. To balance the training set, N_p negatives from the majority set are chosen for training, where N_p is the number of positives in the training set, as illustrated in Figure 7.3. $current_sol$ is the name of the first balanced training set, which consists of all positives and chosen negatives. The cost of $current_{sol}$ that is $current_{cost}$, is calculated using (Equation 7.3.1)

$$BalancedErrorRate = (1 - \frac{Sensitivity + Specificity}{2}) \tag{7.3.1}$$

As seen in the images, all of the misclassified majority class samples are replaced with their nearest neighbour majority class samples determined by neighbor($current_{sol}$) as shown in figures Figure 7.4 and Figure 7.5. This forms new solution, new_{sol} depicted in Figure 7.6. The cost of new solution, $cost(new_{sol})$ is calculated using (Equation 7.3.1). If the $cost(new_{sol})$, that is Balanced Error Rate of new solution is less than $current_{cost}$ then $current_{sol}$ is replaced by the new_{sol}. Otherwise, accep_prob is calculated using (Equation 7.3.2).

$$accep\_prob = e^{(new_{cost} - current_{cost})/temp} \tag{7.3.2}$$

If accep_prob is greater than a random number, the current solution is replaced with the new solution, despite the fact that the new solution's cost is higher. It is important to keep in mind that

the expense must be kept to a minimum. This phase allows you to accept even a terrible solution, allowing the process to move forward without being slowed down by a local minimum. These procedures are continued until the temperature meets the user-defined minimum temperature, as stated in Table 7.1.

Table 7.1: SAUS Parameters, Description and Values [71]

Parameter	Description	Value
min_temp	final temperature	0.00001
alpha	Cooling Rate	0.9
Max_ite	Maximum number of iteratons	100
temp	initial temperature	1.0

7.4 Algorithm

Algorithm 7.6 Simulated Annealing based UnderSampling(SAUS)

An Imbalanced Data Set A Balanced Data Set Initialize min_temp=0.00001, temp=1.0, alpha=0.9
To balance the training set, select N_p negatives from the majority set for training, where N_p is the amount of positives in the training set. $current_sol$ is the first balanced training set consisting of all positives and chosen negatives.
$current_{cost} \leftarrow cost(current_{sol})$

while (temp $>$ min_temp) **do**

while (i $\leq$ Max_ite) **do** $new_{sol} \leftarrow neighbor(current_{sol})$ $new_{cost} \leftarrow cost(new_{sol})$ accep_prob $\leftarrow e^{(new_{cost}-current_{cost})/temp}$

if $current_{cost} < new_{cost}$ **then**

if accep_prob $>$ random() **then** $current_{sol} \leftarrow new_{sol}$ i $\leftarrow$ i+1 temp $\leftarrow$ temp$\times$alpha **return** $current_{sol}$ and $current_{cost}$

Algorithm 7.7 cost(Solution set)

Solution set Cost of the solution Use the solution set as the training set for the classification. **return** $(1-(Sensitivity+Specificity)/2)$;

Algorithm 7.8 neighbour(Solution set)

Solution set neighbour of the current solution set

for (each misclassifed majority class instance) **do** neighbour $\leftarrow$ nearest-neighbour(misclassified majority class instance) misclassified majority class instance $\leftarrow$ neighbour New Solution $\leftarrow$ all minority class instances + correctly classified majority class instances + nearest neighbours of misclassified majority class instances

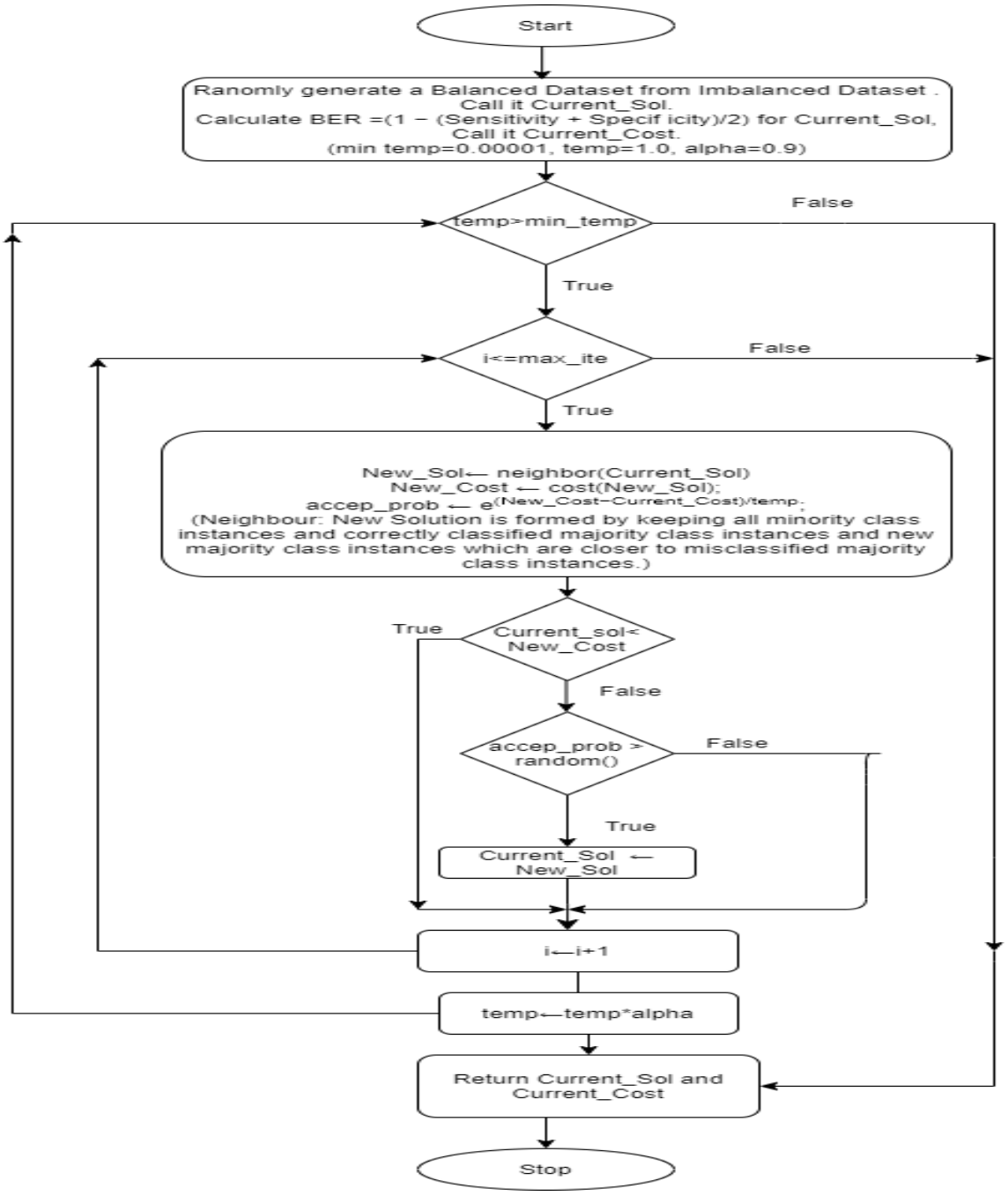

Figure 7.1: Process of Simulated Annealing based UnderSampling

7.5 Experiments and Results

This section contains information about the data sets that are used as well as the evaluation criteria that are employed. A diverse group of data sets are chosen to verify the efficacy of the suggested methodology. These are selected to account for the impact of data complexities, attribute complexity, and imbalance ratios. They can be classified as sets with instances (low and high) and attributes in general (low and high). That is, a high number of features with a small number of instances, a small number of features with a huge number of instances, and so on. These data sets are listed in Table 7.2. Binary and multi-class data sets are both taken into account. Multi-class data sets are converted to binary by making the needed class positive/minority and the remaining classes negative/majority. The data sets are detailed in Table 7.2. Except for the class attribute, all of the values in these data sets are numeric. The data sets were obtained from the UCI [11], KEEL[7] data repositories, respectively.

KEEL[7], Weka[138] tools are used to conduct the experiments. Parameter Values used for kNN Classifier in the SAUS Experiment are given in Table 7.3. 5-fold cross-validation is used to compute the performance metrics. Sensitivity and AUC are calculated before and after applying the proposed method SAUS and the values are presented in tables Table 7.4, Table 7.5 respectively. It can be observed that sensitivity has improved for all data sets, particularly for pima, yeast1, vehicle 1,2, and 3, ecoli1, yeast3, ecoli3, yeast2vs4, yeast05679vs4, glass016vs2, glass2, yeast1vs7, and so on. At the same time, it's worth noting that the AUC values haven't changed all that much, indicating that the specificity part hasn't been compromised.

Table 7.2: Details of the data sets considered for experimentation.

Name of the Data Set	Imbalance Ratio	Number of Features	Number of Instances	Class pos;neg	%pos;%neg
glass1	1.82	9	214	build-win-non_float-proc; remainder	35.51,64.49
wisconsin	1.86	9	683	malignant; benign	35.00,65.00
pima	1.90	8	768	tested_positive; tested_negative	34.84,66.16
iris0	2.00	4	150	Iris-Setosa; remainder	33.33,66.67
glass0	2.06	9	214	build-win-float-proc; remainder	32.71, 67.29
yeast1	2.46	8	1484	nuc; remainder	28.91,71.09
vehicle1	2.52	18	846	Saab; remainder	28.37, 71.63
vehicle2	2.52	18	846	Bus; remainder	28.37, 71.63
vehicle3	2.52	18	846	Opel; remainder	28.37, 71.63
glass0123vs456	3.19	9	214	non-window glass; remainder	23.83,76.17
vehicle0	3.23	18	846	Ven; remainder	23.64,76.36
ecoli1	3.36	7	336	im; remainder	22.92,77.08
new-thyroid2	4.92	5	215	hypo; remainder	16.89, 83.11
new-thryoid1	5.14	5	215	hyper; remainder	16.28, 83.72
ecoli2	5.46	7	336	pp; remainder	15.48, 84.52
segment0	6.01	19	2308	brickface; remainder	14.26,85.74
glass6	6.38	9	214	headlamps; remainder	13.55,86.45
yeast3	8.11	8	1484	m3; remainder	10.98, 89.02
ecoli3	8.19	7	336	imU; remainder	10.88, 89.11
page-blocks0	8.77	10	5472	remainder; text	10.23, 89.77
yeast2vs4	9.08	8	514	cyt; me2	9.92, 90.08
yeast05679vs4	9.35	8	528	me2;mit,me3,exc,vac,erl	9.66,9.034
vowel0	10.10	13	988	hid; remainder	9.01, 90.99
glass016vs2	10.29	9	192	ve-win-float-proc; build-win-float-proc, build-win-non_float-proc, headlamps	8.89, 91.11
glass2	10.39	9	214	ve-win-float-proc; remainder	8.78,91.22
ecoli4	13.84	7	336	om; remainder	6.74, 93.26
yeast1vs7	13.87	8	459	nuc; vac	6.72, 93.28
shuttle0vs4	13.87	9	1829	Rad Flow; Bypass	6.72, 93.28
glass4	15.47	9	214	containers; remainder	6.07,93.93
page-blocks13vs4	15.85	10	472	graphic; hori.line.picture	5.93, 94.07
abalone9vs18	16.68	8	731	18;9	5.65, 94.25
glass016vs5	19.44	9	184	tableware; build-win-float-proc, build-win-non_float-proc headlamps	4.89,95.11
shuttle2vs4	20.5	9	129	Fpv Open; Bypass	4.65, 95.35
yeast1458vs7	22.10	8	693	vac;nuc,me2,me3,pox	4.33,95.67
glass5	22.81	9	214	tableware; remainder	4.20,95.80
yeast2vs8	23.10	8	482	pox; cyt	4.15, 95.85
yeast4	28.41	8	1484	me2; remainder	3.43, 96.57
yeast5	32.78	8	1484	me1; remainder	2.96, 97.04
ecoli0137vs26	39.15	7	281	pp,imL; cp,im,imU,imS	2.49,97.51
yeast6	39.15	8	1484	exc;remainder	2.49, 97.51
abalone19	128.87	8	4174	19;remainder	0.77, 99.23

Table 7.3: Parameter Values used for kNN Classifier in the SAUS Experiment.

For kNN Classifier	
Parameter Descriptor	Value
k Value	1
Distance Function	Euclidean

Table 7.4: Sensitivity with **kNN** classifier.

	kNN Classifier	
Data Set	No Sampling	SAUS
glass1	0.67±0.12	**0.72 ±0.07**
wisconsin	0.93±0.01	**0.95 ±0.01**
pima	0.53±0.03	**0.67 ±0.02**
iris0	1±0	**1 ±0**
glass0	0.81± 0.10	**0.86 ±0.11**
yeast1	0.49±0.02	**0.66 ±0.04**
vehicle1	0.42±0.03	**0.68 ±0.09**
vehicle2	0.92±0.01	**0.96±0.02**
vehicle3	0.5±0.06	**0.74 ±0.06**
glass0123vs456	0.86±0.05	**0.92 ±0.04**
vehicle0	0.88±0.05	**0.93 ±0.04**
ecoli1	0.68±0.12	**0.86 ±0.10**
new-thyroid2	0.97±0.06	**1 ±0**
new-thyroid1	0.97±0.06	**0.97 ±0.06**
ecoli2	0.85±0.13	**0.93 ±0.07**
segment0	0.99±0.01	**1 ±0.06**
glass6	0.75±0.16	**0.82 ±0.18**
yeast3	0.67±0.03	**0.88 ±0.03**
ecoli3	0.54±0.06	**0.89 ±0.11**
page-blocks0	0.77±0.03	**0.9 ±0.01**
yeast2vs4	0.72±0.18	**0.86 ±0.11**
yeast05679vs4	0.41±0.07	**0.78 ±0.11**
vowel0	1±0	**1 ±0**
glass016vs2	0.23±0.27	**0.72 ±0.18**
glass2	0.28±0.29	**0.6 ±0.30**
ecoli4	0.99±0.01	**0.99 ±0.01**
yeast1vs7	0.33±0.11	**0.77 ±0.19**
shuttle0vs4	0.99±0.01	**0.99 ±0.01**
glass4	0.67±0.20	**0.83 ±0.23**
page-blocks13vs4	0.96±0.08	**1 ± 0**
abalone9vs18	0.29±0.34	**0.62 ±0.18**
glass016vs5	0.96±0.08	**1 ±0**
shuttle2vs4	0.9±0.22	**1 ±0**
yeast1458vs7	0.17±0.16	**0.67±0.26**
glass5	0.8±0.44	**1 ±0**
yeast2vs8	0.55±0.20	**0.75 ±0.25**
yeast4	0.35±0.13	**0.82 ±0.14**
yeast5	0.7±0.10	**1 ±0**
ecoli0137vs26	0.7±0.44	**0.9 ±0.22**
yeast6	0.51±0.21	**0.86 ±0.14**
abalone19	0±0	**0.7 ±0.27**

Table 7.5: AUC with **kNN classifier.**

Data Set	kNN Classifier	
	No Sampling	SAUS
glass1	0.76±0.03	**0.78 ±0.07**
wisconsin	0.95±0.09	**0.96 ±0.01**
pima	0.67±0.02	**0.69 ±0.02**
iris0	1±0	**1±0**
glass0	0.79±0.06	**0.84 ±0.06**
yeast1	0.65±0.01	**0.66 ±0.02**
vehicle1	0.54±0.04	**0.54 ±0.02**
vehicle2	0.92±0.02	**0.94±0.01**
vehicle3	0.67±0.02	**0.72±0.02**
glass0123vs456	0.91±0.02	**0.93 ±0.03**
vehicle0	0.91±0.02	**0.92 ±0.02**
ecoli1	0.8±0.06	**0.85±0.04**
new-thyroid2	0.98±0.02	**0.98 ±0.01**
new-thyroid1	**0.98±0.02**	0.97 ± 0.02
ecoli2	0.87±0.02	**0.91 ±0.07**
segment0	**1±0**	0.99± 0.02
glass6	0.87±0.08	**0.89 ±0.10**
yeast3	0.81±0.01	**0.87 ±0.02**
ecoli3	0.74±0.02	**0.87±0.04**
page-blocks0	0.88±0.01	**0.91 ±0.01**
yeast2vs4	0.85±0.08	**0.87 ±0.05**
yeast05679vs4	0.68±0.04	**0.76±0.06**
vowel0	**1±0**	0.98 ±0.01
glass016vs2	0.58±0.15	**0.69±0.12**
glass2	0.58±0.16	**0.6 ±0.16**
ecoli4	0.87±0.08	**0.95±0.01**
yeast1vs7	0.64±0.06	**0.71 ±0.09**
shuttle0vs4	**1±0**	**1 ±0**
glass4	0.82±0.09	**0.86 ±0.11**
page-blocks13vs4	**0.98±0.04**	0.96±0.02
abalone9vs18	0.63±0.17	**0.67 ±0.11**
glass016vs5	0.84±0.22	**0.93 ±0.02**
shuttle2vs4	0.95±0.11	**0.94 ±0.13**
yeast1458vs7	0.57±0.07	**0.64 ±0.13**
glass5	**0.89±0.22**	0.88 ± 0.02
yeast2vs8	0.77±0.10	**0.79 ± 0.09**
yeast4	0.67±0.07	**0.8 ±0.06**
yeast5	0.85±0.05	**0.96±0.01**
ecoli0137vs26	0.84±0.22	**0.86±0.10**
yeast6	0.75±0.10	**0.83 ±0.04**
abalone19	0.5±0.01	**0.68 ±0.12**

7.6 Discussion

Simulated Annealing is a framework that is not dependent on a specific situation. Metaheuristics can provide a good solution without compromising the computational time for optimization problems. The ability of simulated annealing to achieve a global optimal solution despite landing in a local minimum is one of its advantages. It do not require more computer resources and can deliver acceptable results in a fair amount of time. The proposed method has the advantage of not requiring any of the criteria such as cluster creation or locating nearest neighbours to all of the majority class samples. Instead, this method creates a balanced training set that is close to optimal by selecting a subset of majority class samples in each iteration and determining if the chosen majority class samples, combined with all minority class samples, give the lowest balanced error rate. Figures

Figure 7.8 and Figure 7.9 provide an example of data set ecoli1 before and after SAUS application, respectively. t-sne [1] was used to capture this visualisation. This shows a clear discriminating barrier, which aids in the classification of minorities. This pattern may be seen in the vast majority of data sets.

7.6.1 Sensitivity and AUC Results

It is clear from the SAUS experimental results in Tables Table 7.4 and Table 7.5 that sensitivity and AUC for practically all datasets have improved after using SAUS. The accuracy of the class of interest, namely the Minority Class Classification Rate, has improved significantly.

7.6.1.1 Small Data sets

The results in Table 7.4 demonstrate that for data sets with low density and fewer instances, such as glass1, glass0, glass0123vs456, new-thyroid2, new-thyroid1, glass6, glass016vs2,glass2, glass4, glass016vs5, shuttle2vs4, glass5, and ecoli0137vs26, the classifier's sensitivity and overall performance have improved.

7.6.1.2 Large data sets

The developed framework SAUS also improves outcomes for datasets with more instances, such as yeast1, segment0, yeast3, pageblocks0, shuttle0vs4, yeast4, yeast5, yeast6, and abalone19.

7.6.1.3 Data sets with low imbalance ratio

Experiments are carried out on datasets ranging from 1.82 to 128.87 in terms of imbalance ratio. SAUS enhanced the sensitivity of all datasets as compared to the unprocessed original training sets. SAUS has no effect on the classifier's overall performance on these datasets.

7.6.1.4 Data sets with high imbalance ratio

Lack of samples is a problem for datasets with a high imbalance ratio. For example, glass5 and shuttle2vs4 have imbalance ratios of 22.81 and 20.5, with 214 and 129 positive examples, respectively. SAUS improves sensitivity in these circumstances as well, but because to the loss of information in undersampling, there is little improvement in total performance. However, because Simulated Annealing do not get stuck in a local optimum and instead seeks to reach the global optimum, the AUC has not decreased considerably after SAUS undersampling.

7.6.1.5 Performance of SAUS on Phishing data set

Web service is one of the most important Internet communications software services. One of the most common security dangers to web services on the Internet is web phishing. By impersonating a reputable company, web phishing collects personal information such as usernames, passwords, and credit card numbers. This results in the leaking of information, which may cause harm to the users, according to [144].

[37, 43, 72] present a survey on Phishing research. The performance of the algorithm presented in the study [63] is 84% sensitivity and 97% percent specificity, but the approach proposed in this study, SAUS, obtained 93% sensitivity and 97% specificity.

7.6.2 Comparison with Latest Method(SNGEIP) Results

Table 7.6 provides a comparison of findings with the most recent study on imbalanced datasets categorization [31]. Even though it is not precisely an undersampling method, this method is superior. The proposed technique SAUS has got equivalent results with the recent article on unbalanced datasets classification, as shown in Table 7.6. Oversampling and ensembles were utilised in SNGEIP [31], whereas SAUS just employed undersampling and produced comparable results.

Table 7.6: AUC of SAUS kNN classifier and SNGEIP [31].

	Comparison with latest paper	
Data Set	SNGEIP	SAUS
pima	0.75	0.69
iris0	0.98	1.00
glass0	0.81	0.84
yeast1	0.71	0.66
vehicle2	0.97	0.94
vehicle3	0.77	0.72
glass0123vs456	0.94	0.93
vehicle0	0.95	0.92
ecoli1	0.89	0.85
new-thyroid2	0.95	0.98
new-thyroid1	0.95	0.97
ecoli2	0.91	0.91
glass6	0.91	0.89
yeast3	0.93	0.87
ecoli3	0.88	0.87
yeast2vs4	0.90	0.87
vowel0	0.99	0.98
glass016vs2	0.70	0.69
glass2	0.74	0.6
ecoli4	0.90	0.95
shuttle0vs4	1.00	1.00
glass4	0.93	0.86
abalone9vs18	0.60	0.67
glass016vs5	0.96	0.93
shuttle2vs4	1.00	0.94
Average	**0.88**	**0.8612**

7.6.3 Data Complexity Measures

Datacomplexity measurements of datasets are taken into account to further analyse how the suggested strategy performs on datasets with diverse properties in addition to class imbalance. These parameters are used to describe the efficacy of the proposed [60] approach. For analysing the influence of SAUS before and after data selection, four measures are identified: Fraction of points on Class Boundary(N1), Ratio of average intra/inter class closest neighbour distance(N2), Error rate of 1NN Classifier(N3), and Fraction of points with related adherence subsets retained(T1).

7.6.3.1 Definitions

Fraction of points on Class Boundary(N1): The percent of points on the boundary over the total number of points in the data set is used to calculate N1. This metric is calculated using the Minimum Spanning Tree idea (MST). The data set's points are all linked to their nearest neighbours. The number of points in the MST that are connected to the opposite class by an edge is then counted. These points are thought to be near the class boundary, according to [118].

Ratio of average intra/inter class nearest neighbor distance(N2): This method compares the intra-class dispersion to the inter-class separability. The average distance to intra-class nearest neighbour divided by the average distance to inter-class nearest neighbour is the ratio. Smaller values indicate data that is more discriminating.[118].

Error rate of 1NN Classifier(N3): This method uses the leaving-one-out method to determine the nearest neighbour classifier's error rate.[118].

Fraction of points with associated adherence subsets retained (T1): This metric counts how many samples are required to cover each class, with each sample being centred at a training point and enlarged to its maximum size before reaching a point from another class. Samples that are fully redundant in the interior of other samples are eliminated. The total number of points is then used to normalise the count. Instead of a boundary description, this provides an inner description [118].

KEEL Tool [7] was used to calculate all of the measurements stated above. Table 7.7 provides a more detailed discussion of these data complexity measures. The complexity of a dataset has a significant impact on categorization accuracy. As a result, the intricacies of the training sets before and after SAUS are discovered. The results are provided in Table 7.8.

7.6.3.2 N1 Results

(N1) is the percent of points on the Class Boundary, as defined in Table 7.7. The bulk of the points were close to the class boundary, as indicated by the high value of the measure.

Table 7.7: Description of Data Complexity Measures.

DC Measure	Description	Range	Analysis
N1	Fraction of points on Class Boundary	$[0, 1]$	Large Values of the measure indicate that the majority points lay closely to the class boundary
N2	Ratio of average intra/inter class nearest neighbor distance. This measure compares the within-class spread to the size of the gap between classes.	$[0, infinity]$	Low values suggests that the examples of the same class lay closely in the feature space. Large ones indicate that examples of the same class are disperse.
N3	Error rate of 1NN Classifier	$[0, 1]$	Low value of this metric indicates that there is a large gap in the class boundary.
T1	Fraction of points with associated adherence subsets retained	$[0, 1]$	Small values of this measure indicates that the instances which compose the dataset are highly grouped and the boundaries are clearly defined.

As demonstrated in Table 7.8, the datasets glass1, pima, glass0, yeast1, vehicle1 have substantial values of N1 in the unprocessed (imbalanced) training set. That suggests there are more instances near to the class boundary, deceiving the classifier into properly identifying unseen occurrences. The proposed method has chosen balanced training sets in each iteration, lowering the cost function, that is (1-Balanced Accuracy). Internally, it selects training sets with instances that have a smaller number of occurrences near the class boundary.

7.6.3.3 N2 Results

N2 is a metric that relates the dispersion within classes to the magnitude of the gap between them. Low values imply that examples of the same class are clustered together, whereas high values indicate that they are spread. In each iteration, SAUS has chosen a random majority of cases; no extra processing is done to increase or decrease the spread of the magnitude of the gap between classes.

However, while the N2 value increased in several datasets, such as Wisconsin, glass0213vs456, ecoli2, lass6, yeast2vs4, glass2, yeast1458vs7, yeast2vs8, yeast4, yeast5, yeast6, the sensitivity and AUC of the datasets rose after applying SAUS. It's worth noting that SAUS had no effect on the spread of similar-class instances. In these datasets, other dataset factors affect the classifier's performance more than this measure.

7.6.3.4 N3 Results

The N3 data complexity measure represents the error rate of a 1NN classifier. This metric's low value indicates a significant gap in the class boundary. A larger gap in the class boundary increases the classifier's performance. The classifier utilised to evaluate SAUS's performance in this paper is 1NN. SAUS has found balanced training sets with a big gap in the class border, as indicated by the lower values of N3 in the Table 7.8.

7.6.3.5 T1 Results

T1, which indicates the percent of points with related adherence subsets retained, is another data complexity measure calculated on these datasets. This metric was reduced in virtually all datasets after applying SAUS, indicating that the instances that make up the dataset are well clustered and the borders are well defined. This is because, while SAUS chooses the bulk of cases at random, it focuses on balanced sets, which boost the classifier's performance. Internally, this means selecting a balanced set with instances that properly define the boundaries.

Table 7.8: Comparison of Data Complexity Measures.

Data Set	Training Set	N1	N2	N3	T1
glass1	Original	0.5476	0.3938	0.3095	0.6190
	SAUS	0.0819	0.3373	0.0409	0.3032
wisconsin	Original	0.0735	0.2037	0.0367	0.1176
	SAUS	0.0235	0.2302	0.0157	0.0602
pima	Original	0.4901	0.4588	0.3071	0.7124
	SAUS	0.1116	0.3771	0.0534	0.3953
iris0	Original	0.0667	0.1380	0	0.0667
	SAUS	0.025	0.1155	0	0.025
glass0	Original	0.4285	0.3681	0.2619	0.5714
	SAUS	0.0892	0.3057	0.0446	0.2678
yeast1	Original	0.4560	0.4417	0.3243	0.6216
	SAUS	0.0537	0.3263	0.0247	0.2427
vehicle1	Original	0.4319	0.4209	0.2366	0.7041
	SAUS	0.2068	0.4036	0.1293	0.5948
vehicle2	Original	0.1893	0.3445	0.0828	0.4378
	SAUS	0.06	0.3198	0.0228	0.3571

vehicle3	Original	0.4319	0.4248	0.3076	0.6745
	SAUS	0.1970	0.3945	0.0911	0.6323
glass0123vs456	Original	0.0714	0.2272	0.0714	0.2142
	SAUS	0.0731	0.2582	0.0243	0.2317
vehicle0	Original	0.1834	0.3013	0. 0887	0.4319
	SAUS	0.0562	0.2962	0.0187	0.2906
ecoli1	Original	0.2835	0.3301	0.1641	0.2985
	SAUS	0.0645	0.3007	0.0322	0.1451
new-thyroid2	Original	0.1162	0.2667	0.0697	0.1627
	SAUS	0.0357	0.2360	0	0.0892
new-thyroid1	Original	0.0930	0.2734	0.0232	0.1162
	SAUS	0.0357	0.2460	0	0.0357
ecoli2	Original	0.1641	0.2726	0.0895	0.2238
	SAUS	0.1428	0.3345	0.0357	0.2619
segment0	Original	0.0260	0.1557	0.0086	0.0911
	SAUS	0.0114	0.1776	0.0019	0.0684
glass6	Original	0.0476	0.1750	0	0.1428
	SAUS	0.2083	0.3206	0.0833	0.2291
yeast3	Original	0.0979	0.3250	0.0540	0.2398
	SAUS	0.0496	0.3107	0.0267	0.1793
ecoli3	Original	0.1641	0.2568	0.1343	0.2388
	SAUS	0.1071	0.3238	0.0357	0.25
yeast2vs4	Original	0.0980	0.3313	0.0784	0.1862
	SAUS	0.0609	0.3536	0.0365	0.2560
yeast0567vs4	Original	0.1904	0.3656	0.1047	0.3714
	SAUS	0.0853	0.3474	0.0365	0.2560
vowel0	Original	0.1065	0.2327	0.0355	0.1522
	SAUS	0.0972	0.2317	0.0208	0.2013
glass016vs2	Original	0.1842	0.3974	0.1315	0.2894
	SAUS	0.2142	0.3896	0.0357	0.5357
glass2	Original	0.1904	0.3356	0.0714	0.2380
	SAUS	0.25	0.3750	0.0714	0.4642
ecoli4	Original	0.0746	0.2170	0.0447	0.1194

	SAUS	0.0937	0.2967	0.0312	0.1562
yeast1vs7	Original	0.1648	0.3739	0.1208	0.2857
	SAUS	0.1041	0.3640	0.0416	0.4583
shuttle0vs4	Original	0.0109	0.0536	0	0.0136
	SAUS	0.0101	0.0746	0	0.0202
glass4	Original	0.0952	0.2627	0.0238	0.1667
	SAUS	0.1818	0.2987	0.0454	0.4090
yeast1458vs7	Original	0.1086	0.3597	0.0724	0.1667
	SAUS	0.3541	0.4032	0.1458	0.625
yeast2vs8	Original	0.1354	0.3343	0.0833	0.1667
	SAUS	0.2812	0.3771	0.125	0.5625
yeast4	Original	0.0743	0.2988	0.0506	0.1216
	SAUS	0.0731	0.3527	0.0365	0.3170
yeast5	Original	0.037	0.2138	0.0270	0.0608
	SAUS	0.0555	0.2938	0.0277	0.1667
yeast6	Original	0.0405	0.2646	0.0236	0.0979
	SAUS	0.1428	0.3590	0.0714	0.375

7.6.3.6 Comparison with Other Methods

The proposed technique is compared to several widely used undersampling methods in the literature, such as **CNN, CNNTL, CPM, SBC, OSS, RUS, and TL**. Table 7.9 contains parameter values for alternative undersampling algorithms. The default values in KEEL are as follows. On the partitions generated to evaluate the SAUS algorithm, we ran these algorithms accessible in KEEL. They haven't been re-implemented. Many of these methods use a kNN classifier to choose the samples, however the suggested method do not employ a classifier at all. Table 7.10 demonstrates that the suggested method outperforms current methods in a few data sets while achieving equivalent results in the remaining data sets. The proposed method has never been shown to be inferior to any of the existing popular undersampling strategies.

Table 7.9: Parameter Values of other Undersampling Methods used in Experiment.

Other Undersampling Methods		
Method	Parameter Description	Value
CNN	seed	1
	Number of Neighbors	5
CNNTL	seed	1
CPM	seed	1
NCL	seed	0
	Number of Neighbors	5
OSS	seed	1
	Number of Neighbors	5

Table 7.10: Comparison of AUC with Other UnderSampling Method.

data set	CNN	CNNTL	CPM	NCL	OSS	SAUS
glass1	0.78	0.69	0.73	0.74	0.75	**0.78**
wisconsin	0.95	0.96	0.89	0.96	0.96	**0.97**
pima	0.64	0.65	0.62	**0.70**	0.67	0.69
iris0	1.0	1.0	1.0	1.0	1.0	1.0
glass0	0.77	0.74	0.74	0.82	0.80	**0.84**
yeast1	0.61	0.63	0.60	**0.68**	0.65	0.66
vehicle2	0.92	0.89	0.92	**0.94**	0.92	**0.94**
vehicle1	0.63	0.66	0.61	**0.70**	0.67	0.68
vehicle3	0.68	0.71	0.67	0.70	0.69	**0.72**
glass0123vs456	0.91	0.89	0.90	0.95	0.90	**0.93**
vehicle0	0.91	0.90	0.87	0.92	0.92	**0.92**
ecoli1	0.79	0.82	0.74	0.87	0.83	**0.88**
new-thyroid2	0.96	0.96	0.94	**0.98**	0.96	**0.98**
new-thyroid1	0.96	0.97	0.94	**0.98**	0.96	**0.98**
ecoli2	0.83	0.77	0.88	0.90	0.85	**0.91**
segment0	0.99	0.98	0.97	0.99	0.99	**1.0**
glass6	0.85	0.80	0.85	0.88	0.83	**0.89**
yeast3	0.81	0.84	0.78	0.86	0.85	**0.87**
ecoli3	0.76	0.80	0.69	0.81	0.81	**0.87**
page-blocks0	0.86	0.88	0.86	0.90	0.88	**0.91**
yeast2vs4	0.81	0.84	0.78	0.88	0.86	**0.87**
yeast05679vs4	0.68	0.73	0.71	0.73	0.71	**0.76**
vowel0	0.99	0.99	0.96	0.97	0.99	**1.0**
glass016vs2	0.56	0.62	0.60	0.67	0.66	**0.69**
glass2	0.65	0.68	0.63	0.65	0.65	**0.6**
ecoli4	0.90	0.89	0.69	0.70	0.87	**0.95**
yeast1vs7	0.67	0.67	0.65	0.69	0.68	**0.71**
shuttle0vs4	1.0	1.0	1.0	1.0	1.0	**1.0**
glass4	0.82	**0.88**	0.72	0.70	**0.88**	0.86
page-blocks13vs4	0.97	0.97	0.89	0.90	0.95	**0.98**
abalone9vs18	0.64	0.62	0.63	0.64	0.59	**0.67**
glass016vs5	0.83	0.87	0.88	0.87	0.88	**0.93**
shuttle2vs4	0.95	0.94	0.95	0.94	0.95	**0.95**
yeast1458vs7	0.58	0.62	0.61	0.62	0.60	**0.64**
glass5	0.93	**0.96**	0.89	0.89	0.92	0.89
yeast2vs8	0.79	0.77	0.72	0.75	0.78	**0.79**
yeast4	0.67	0.77	0.71	0.73	0.71	**0.8**
yeast5	0.85	0.91	0.83	0.85	0.89	**0.96**
ecoli0137vs26	0.84	0.78	0.83	0.84	0.74	**0.86**
yeast6	0.71	0.77	0.79	0.78	0.77	**0.83**
abalone19	0.47	0.50	0.54	0.51	0.52	**0.68**
Average	0.8029	0.8126	0.7856	0.8192	0.8168	**0.8497**
Average for IR>10	0.78	0.8005	0.7642	0.7737	0.7910	**0.8310**

7.6.4 Friedman test

The Friedman test compares three or more matched or paired groups and is a non-parametric test. Each matched set (each row) is first ranked from low to high by the Friedman test. Each row is given its own ranking. The ranks in each group are then added up (column). The p value will be minimal if the sums are considerably diverse. Compare the p-value to the significance threshold to see if the differences between the medians are statistically significant and to evaluate the null hypothesis. According to the null hypothesis, all population medians are equal. A significance level of 0.05 (abbreviated as $alpha$ or alpha) is usually sufficient. A significance level of 0.05 represents a 5% chance of finding that there is a difference when there isn't one.

Because non-parametric tests are commonly employed, the Friedman test is used to assess which approach is the best among the undersampling approaches, including the newly presented approach [51]. It turns out that the strategy proposed is the most effective. It is the lowest ranked (best). In the case of a non-parametric test, the data does not have to originate from a specific distribution. Table 7.11 shows the average ranks attained by each procedure in the Friedman test.

Algorithm	Ranking
CNN-kNN	4.1585
CNNTL-kNN	3.878
CPM-kNN	4.9268
NCL-kNN	2.8902
OSS-kNN	3.5122
SAUS-kNN	**1.6341**

Table 7.11: Average rankings of the algorithms (Friedman). Friedman statistic (distributed according to chi-square with 5 degrees of freedom): 75.74216.
P-value computed by Friedman Test: 0.

Table 7.12 shows the p-values obtained by using **Post-hoc comparison** methods to the Friedman procedure findings. The null hypothesis is rejected if the p-value is less than or equal to the significance level, indicating that not all group medians are equal. There is insufficient evidence to reject the null hypothesis that the group medians are all equal if the p-value is greater than the significance level.

i	algorithm	$z = (R_0 - R_i)/SE$	p
5	CPM-kNN	7.968798	0
4	CNN-kNN	6.109412	0
3	CNNTL-kNN	5.430588	0
2	OSS-kNN	4.545166	0.000005
1	NCL-kNN	3.039949	0.002366

Table 7.12: Post-Hoc comparison Table for $\alpha = 0.05$ (FRIEDMAN)

Adjusted P-values obtained through the application of the post-hoc methods (Friedman) are shown in Table 7.13.

i	algorithm	unadjusted p
1	CPM-kNN	0
2	CNN-kNN	0
3	CNNTL-kNN	0
4	OSS-kNN	0.000005
5	NCL-kNN	0.002366

Table 7.13: Adjusted p-values (FRIEDMAN)

The AUC values produced using the kNN Classifier and popular UnderSampling methods CNN, CNNTL, CPM, NCL, OSS, and Simulated Annealing based UnderSampling (SAUS) are visually depicted in Figure 7.10. As shown in the Figure 7.10, the suggested approach SAUS outperforms other widely used Under Sampling methods.

Table 7.14: Comparison of AUC results of proposed methods with **other undersampling techniques** using **C4.5**.(- *indicate results not obtained even after 300 seconds*)

Data set	CNN (1968)	CNNTL (2004)	CPM (2005)	SBC (2006)	NCL (2001)	OSS (1997)	RUS (2004)	TL (1976)	**SAUS**
Ecoli4	0.83	0.84	0.81	0.81	0.81	0.84	**0.86**	0.81	0.84
Haberman	0.63	0.59	0.61	0.57	0.63	**0.64**	0.61	0.63	0.66
Isolet5	0.84	0.86	0.57	0.57	0.86	0.86	0.87	0.81	0.89
LibrasMove	**0.84**	0.77	0.70	0.50	0.78	0.79	0.73	0.83	0.80
Newthryroid1	0.93	0.92	0.82	**0.94**	**0.94**	**0.94**	0.91	0.92	0.95
Spectrometer	0.81	0.79	0.83	0.62	0.85	0.81	0.85	**0.87**	0.85
Vowel0	0.92	0.92	0.89	0.95	0.92	0.92	0.94	**0.97**	0.94
Yeast1289vs7	0.59	0.61	**0.63**	0.5	0.53	0.61	0.60	0.54	0.66

7.6.5 Comparison with Oversampling and Ensemble Methods

To know the performance of the proposed methods with respect to other methods of class imbalance viz., oversampling, ensemble based, algorithm based and cost-sensitive based, it is compared with each of them available in KEEL. Results shown in the Table 7.17 prove that the proposed method is not inferior to any of the existing methods and is giving comparable results.

Post-hoc tests decide which groups are significantly different from each other, based upon the mean rank differences of the groups. Post hoc comparison (Friedman) is done to compare proposed SAUS with other popular undersampling methods. P-values obtained by applying post hoc methods over the results of Friedman procedure are shown in Table 7.19.

Adjusted P-values obtained through the application of the post hoc methods (Friedman).

Table 7.15: Comparison of AUC results of proposed method with **OverSampling techniques** using **kNN**. (- *indicate results not obtained even after 300 seconds*)

Data set	ADASYN (2008)	ADOMS (2008)	Borderline-SMOTE (2005)	ROS (2004)	SafeLevel-SMOTE (2009)	SMOTE-TL (2004)	SMOTE (2002)	**SAUS**
Ecoli4	0.90	0.91	0.89	0.87	0.87	0.92	**0.93**	0.91
Haberman	0.54	0.57	0.58	0.54	0.54	0.59	0.58	0.58
Isolet5	0.96	-	**0.98**	0.94	0.79	0.98	0.97	0.91
LibrasMove	-	-	-	-	-	-	0.91	0.93
Newthryroid1	0.97	0.97	0.97	0.97	0.97	0.97	0.95	0.94
Spectrometer	-	-	-	-	-	-	-	0.93
Vowel0	0.99	1.00	1.00	1.00	1.00	0.99	0.99	0.98
Yeast1289vs7	0.60	0.58	0.59	0.55	0.55	0.63	0.60	0.64

Table 7.16: Comparison of AUC results of proposed methods with **other OverSampling techniques** using **C4.5**. (- *indicate results not obtained even after 300 seconds*)

Data set	ADASYN (2008)	ADOMS (2008)	Borderline-SMOTE (2005)	ROS (2004)	SafeLevel-SMOTE (2009)	SMOTE-TL (2004)	SMOTE (2002)	SAUS
Ecoli4	0.87	0.90	0.84	0.84	0.89	0.87	**0.95**	0.84
Haberman	0.63	0.59	0.61	0.55	0.63	0.59	0.63	0.66
Isolet5	0.82	-	0.88	0.84	-	0.89	0.86	0.89
LibrasMove	0.76	0.87	0.89	0.82	0.83	0.83	**0.85**	0.80
Newthryroid1	0.95	0.95	0.96	0.96	0.95	0.95	0.93	0.95
Spectrometer	**0.90**	0.87	0.83	0.85	0.86	0.86	-	0.85
Vowel0	0.96	0.98	0.97	0.95	0.95	**0.98**	0.97	0.94
Yeast1289vs7	**0.68**	0.66	0.54	0.66	0.64	0.58	0.67	0.66

7.7 Summary

Conventional classifiers are error rate/accuracy driven, which means that they evaluate the classifier's performance based on an equal distribution of classes. A Simulated Annealing-based Under Sampling (SAUS) method is presented to resolve these difficulties. Simulated annealing is a prominent meta-heuristic search strategy that uses a novel cost function in terms of Balanced Error Rate to construct a novel cost function. While analysing the solution at each iteration in the subsampling process, this cost function strikes a balance between Sensitivity and Specificity measures and is also free of the local trap.

Table 7.17: Comparison of AUC results of proposed methods with **some Ensemble Methods, Cost Sensitive and Algorithm based methods**. (- *indicate results not obtained even after 300 seconds*)

Data set	Balance Cascade (2009)	Easy Ensemble (2009)	AdaC2 (2007)	CSVMCS (2009)	C45CS (2002)	NNCS (2006)	**SAUS-kNN**	**SAUS-c4.5**
Ecoli4	0.84	0.85	0.92	**0.95**	0.86	0.87	0.91	0.84
Haberman	0.61	0.65	0.56	0.61	0.57	**0.62**	0.58	0.66
Isolet5	-	-	-	-	-	0.50	0.91	0.89
LibrasMove	-	-	-	-	-	0.50	0.93	0.80
Newthryroid1	0.93	0.93	0.94	0.98	0.97	0.82	0.94	0.95
Spectrometer	-	-	-	-	-	-	0.93	0.85
Vowel0	0.94	0.94	-	0.97	0.94	0.68	0.98	0.94
Yeast1289vs7	0.65	0.65	0.63	-	0.67	0.51	0.64	0.66

Algorithm	Ranking
CNN-c4.5	3.6098
CNNTL-c4.5	3.6585
CPM-c4.5	5.5366
NCL-c4.5	2.9268
OSS-c4.5	3.3902
SAUS-c4.5	1.878

Table 7.18: Average Rankings of the algorithms (Friedman).

i	algorithm	$z = (R_0 - R_i)/SE$	p	Holm
5	CPM-c4.5	8.85422	0	0.01
4	CNNTL-c4.5	4.309054	0.000016	0.0125
3	CNN-c4.5	4.190997	0.000028	0.016667
2	OSS-c4.5	3.659744	0.000252	0.025
1	NCL-c4.5	2.53821	0.011142	0.05

Table 7.19: Post Hoc comparison Table for $\alpha = 0.05$ (FRIEDMAN)

i	algorithm	unadjusted p	p_{Holm}
1	CPM-c4.5	0	0
2	CNNTL-c4.5	0.000016	0.000066
3	CNN-c4.5	0.000028	0.000083
4	OSS-c4.5	0.000252	0.000505
5	NCL-c4.5	0.011142	0.011142

Table 7.20: Adjusted p-values (FRIEDMAN) (I)

i	algorithm	unadjusted p
1	CPM-c4.5	0
2	CNNTL-c4.5	0.000016
3	CNN-c4.5	0.000028
4	OSS-c4.5	0.000252
5	NCL-c4.5	0.011142

Table 7.21: Adjusted p-values (FRIEDMAN) (II)

In comparison to unprocessed imbalanced training sets and other contemporary techniques, the experimental findings show a significant increase in sensitivity. This methodology also minimises the trade-off between sensitivity and specificity, resulting in an overall improvement in AUC values. This research will be expanded to address the issue of multi-class imbalance in the future. It can also be used in conjunction with dimensionality reduction. This method is more economical in terms of complexity and yields better sensitivity values than previous undersampling approaches. SAUS's experimental results show that the average Sensitivity measure on the test set has improved from 0.68 to 0.86, demonstrating its efficacy in addressing the dataset's imbalance issue. The results of the Area Under the ROC Curve (AUC) show that SAUS outperforms numerous prominent undersampling approaches. SAUS is on par with cutting-edge solutions to the problem of class disparity.

The insights obtained from the proposed method is that, SAUS works for high imbalance ratio, high dimensional datasets and not suitable for large size datasets, as it takes more time to converge. Hence, it is recommended to use the proposed Simulated Annealing based Under Sampling method after applying any prototype reduction methods available in the literature. Major contributions of this work are:

(i) Simulated Annealing is used in a novel approach to improve the True Positive Rate and overall performance of the classifier. Using either kNN or clustering algorithms, it is shown to overcome the disadvantages of classic undersampling methods.

(ii) Simulated Annealing, unlike the other metaheuristics, is not susceptible to local minima/maxima, resulting in a near-optimal solution. The proposed Simulated Annealing based Under Sampling (SAUS) method outperforms numerous prominent and recent Under Sampling methods such as CNN, CNNTL, OSS, and others, increasing the average AUC value from 0.80% to 0.84%. Its performance is comparable to that of the most recent approach, SNGEIP [31]. In comparison to original data sets meeting the chosen aim, the proposed technique improves the True Positive Rate by 18%. AUC values suggest that the specificity is not compromised. The proposed technique has a lower time complexity than the other common approaches with which it is compared.

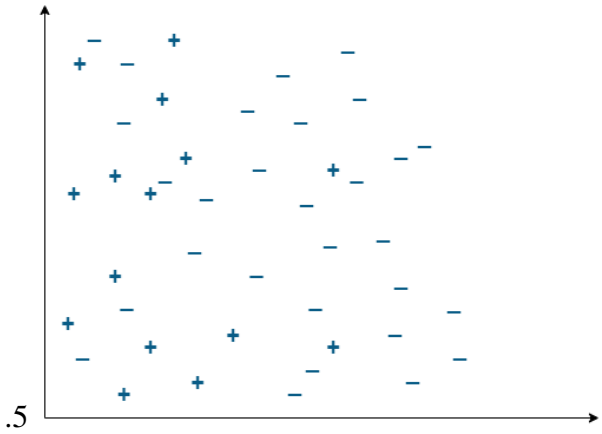

Figure 7.2: Imbalanced Training Set

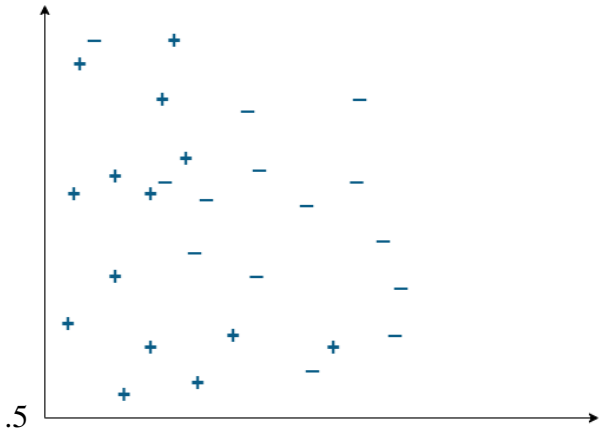

Figure 7.3: Randomly chosen Initial balanced training set, $current_{sol}$.

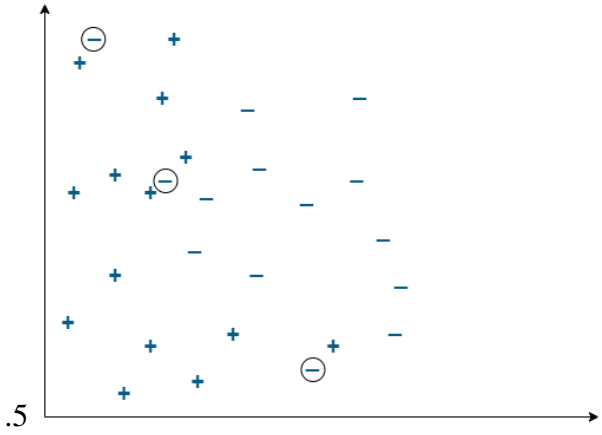

Figure 7.4: Misclassified Majority class Instances in the balanced training set are represented by ◯.

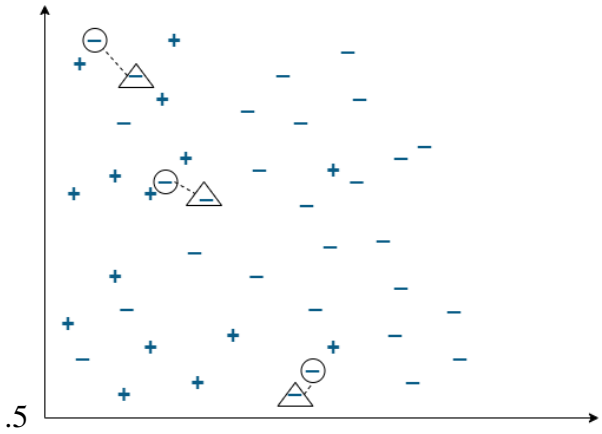

Figure 7.5: Choosing nearest majority class samples(represented by △) of misclassified

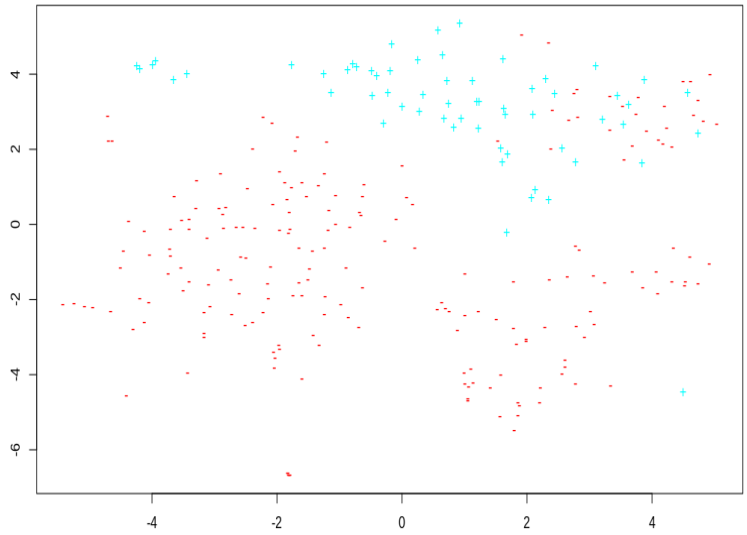

Figure 7.8: Before applying SAUS.

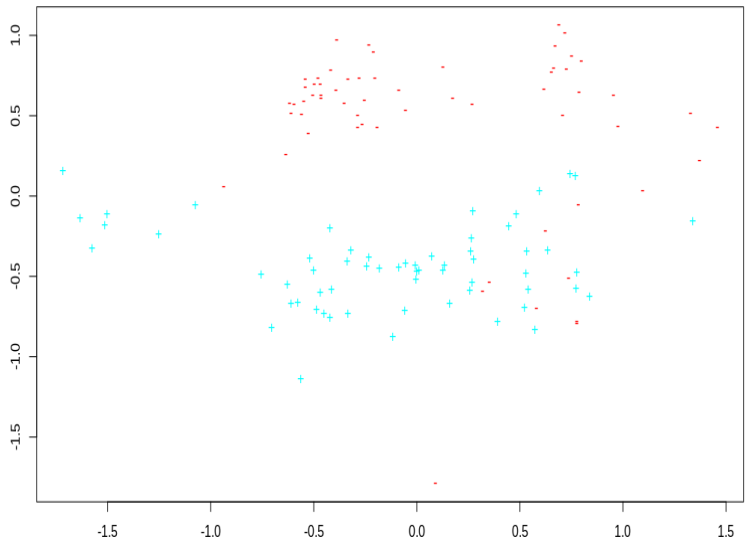

Figure 7.9: After the application of SAUS
Example of ecoli1 data set before and after applying SAUS. '+' denotes the positive class and '-' denotes a negative class sample.

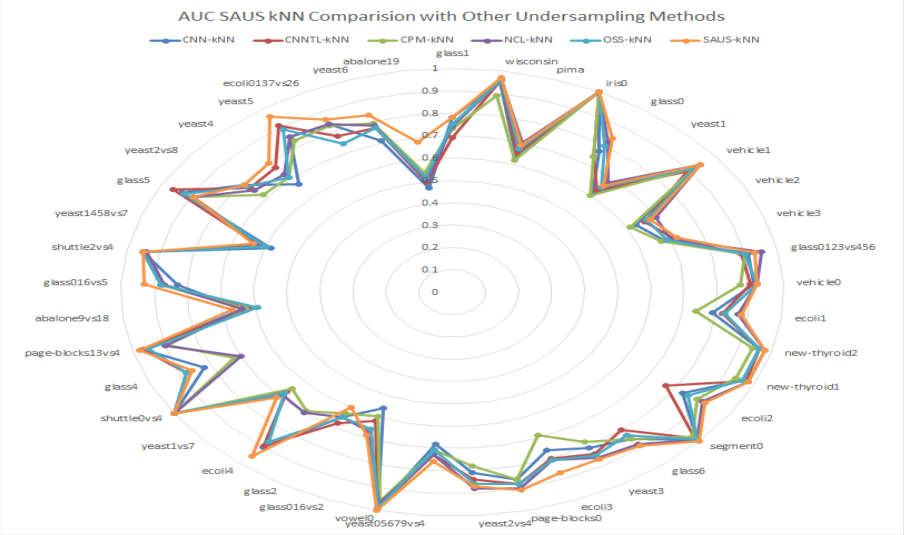

Figure 7.10: SAUS AUC kNN comparision with other undersampling methods

Chapter 8

Conclusions and Future Scope

8.1 Conclusions

This thesis attempts to tackle the issue of classifying datasets that have imbalanced information, where a majority class of data outnumbers a minority class of data in the two-class classification problem. Classification methods such as k-Nearest Neighbor, Decision Tree, Neural Networks, and Naive Bayes suffer when applied on imbalanced datasets. As a solution to the problem, this thesis presents five solutions: Ensemble-based prototype generation, Quartiles-based distribution, Mahalanobis distance based undersampling, Simulated annealing, and Centroid-based groupping.

”No Free Lunch Theorem” as described by Wolpert [139], states that no single model can be the most effective for all the problems. However, the objectives of the proposed work achieved are:

- Increasing Sensitivity
- Parameter Independence
- Information Loss
- Maintaining trade-off between Majority Class and Minority Class prediction rate

Additionally,

- Scale Invariant
- Variable-Independence

Objective 1: Increasing Sensitivity By reducing the impact of majority class on the classifier, Sensitivity is improved in all the proposed methods.

1. ECST: In this approach, positives and negatives are chosen from the bins using stratified sampling to balance the training sets. Hence, Sensitivity is improved.

2. CBG: Equal Number of prototypes are generated from both the classes to reduce the impact of negatives on classification and to increase Sensitivity.

3. QUS: Balanced training set is constructed by selecting negatives equal to the number of positives from groups formed based on Quartiles. Hence, Sensitivity is increased as the size of negatives is reduced thereby reducing it's impact on classification.

4. MahalCUSFilter: Sensitivity is increased as the negatives influence on classifier is reduced as their size is taken equal to the size of positives by forming Centroid based groups. Again stratified sampling is used to achieve this.

5. SAUS: Balanced Training Set is generated in each iteration of Simulated annealing approach which reduces the impact of size of negatives on classification thereby increasing Sensitivity.

Objective 2: Parameter Independence Unlike clustering algorithms, where the number of clusters and validation mechanism to be used to get appropriate clusters, parameter independence is achieved by the proposed methods.

1. ECST: It is found that the number of bins are not impacting the sensitivity rates. Hence there is not much dependence on the parameter values.

2. CBG: In this prototypes are generated from the centroid based groups. These are independent of any parameters as in Clustering algorithms.

3. QUS: Number of groups based on each instance distance from quartile reference point is fixed. This is purely parameter-independent.

4. MahalCUSFilter: Number of groups formed based on centroid is not affecting the classifier. Default number of groups taken is 10, here. It is not much dependent on the number of groups(parameters) unlike Clustering algorithms.

5. SAUS: The default values which are set to parameters for Simulated Annealing based Undersampling approach are not affecting the classifier performance, they impact only on complexity.

Objective 3: Information Loss

Selecting representative samples from the majority class from the entire distribution of the samples without missing the disjuncts is important to enhance the classifier performance.

1. ECST: Instances selected from Majority class are not chosen randomly. They are chosen based on their distance from reference point and bins are formed. Hence, they are representative of the entire negatives distribution. Information loss is restricted through this.

2. CBG: The prototypes generated from groups formed by first partitioning the majority class instances into groups based on their similarity with their average behaviour(Centroid). No single negative instance is discarded in this method. This also eliminates the outlier effect on the classifcation process.

3. QUS: Groups are formed as per Quartiles as reference points. Hence, all the negatives are partitioned based on their distance from quartiles. Hence, the formed training set resembles the original training set. Here also, the information loss is restricted.

4. MahalCUSFilter: The majority class instances are chosen using stratified sampling from the groups formed by their Mahalanobis distance from their centroid. There by taking care of information loss.

5. SAUS: The concept of choosing an optimal Balanced Set among several possible solutions i.e., several possible balanced sets from the given imbalanced dataset leads to less information loss and gives good classification performance.

Objective 4: Trade-off Between Majority Class and Minority Class prediction rates Using evaluation measures like G-Mean, AUC, and Balanced-Error Rate in all the proposed methods, the process of creating Balanced Training Sets from an Imbalanced set .

Objective5: Scalability All the methods proposed here are scalable. They all are of $O(n)$ complexity.

Two more Objectives Additionally, the following two more objectives **Scale Invariant** and **Variable-Independence** are achieved in MahalCUSFilter by using Mahalanobis distance measure.

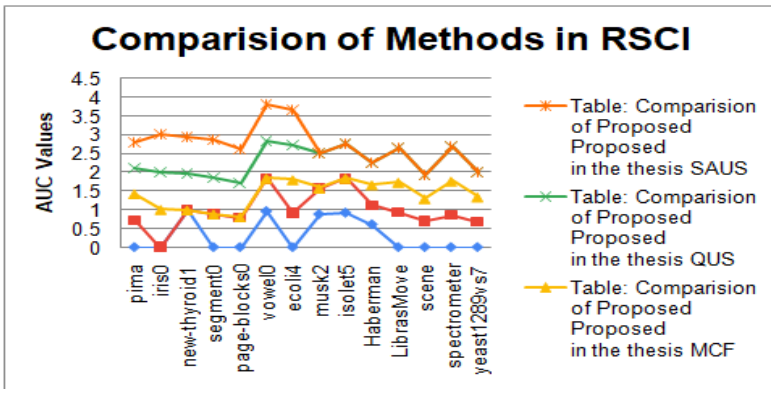

Figure 8.1: (Comparision of Proposed Methods in RSCI-Thesis)

The methods proposed in this thesis are compared with respect their AUC values. Figure Figure 8.1 shows the comparison plot. All the methods are giving good results, however it is clear that SAUS is predominant giving better results than other methods. Reason for this could be that SAUS is based on metaheuristics and a better balanced set is obtained through this.

8.2 Future Scope

The findings of this study's research and experiments suggest that there are many more possibilities to pursue in order to find more solutions to the problem of class imbalance. This section outlines the most important future research directions.

- Instead of limiting the training set to 30% of the dataset, the first contribution, ECST might be extended to include balancing the training set.
- The second proposed approach, CBG, is a prototype-generated method for classifying datasets that are unbalanced. This research can be used to solve the classification challenge for attribute-noise datasets.
- The third contribution, QUS, which dealt with quartiles, can be improved by using the Mahalonobis distance measure to account for the dataset's interdependencies of variables.
- In the fourth proposal, MahalCUSFilter, the use of a filter as a last step actually causes the problem of overfitting. As a result, it's performance can deteriorate. As a future development of this study, it could be used with oversampling methodologies to improve the overall classification rate's performance.
- Simulated Annealing based on Fifth Contribution Any of the resampling methods available in the literature can be extended by combining with the pretreatment procedures of undersampling.

This thesis work can be expanded with Principal Component Analysis and collaborate with Dimensionality Reduction techniques, among other things. It's still a mystery which method works best for a particular imbalanced dataset with asymmetric misclassification costs. Researchers proposed a variety of models for addressing the issue of class disparity from various perspectives. However, undersampling, oversampling, cost-sensitive learning, and other techniques all have benefits and drawbacks, and there is no clear winner. As a result, more research is needed to determine what works best for a given dataset and to examine the impact of factors like data size on the solution to the problem of class imbalances. Furthermore,

- All the proposed methods can be extended with combination of Feature Reduction methods.
- These methods can be extended for multiclass imbalanced datasets
- Can be effectively implemented in association with Ensemble approaches.
- Can be extended for Big Data

Bibliography

[1] https://lvdmaaten.github.io/tsne/.

[2] http://sci2s.urg.es/pstax.

[3] Laith Abualigah. *Feature Selection and Enhanced Krill Herd Algorithm for Text Document Clustering*. 12 2018.

[4] A. Adewole, K. Otubamowo, and T. Egunjobi. A comparative study of simulated annealing and genetic algorithm for solving the travelling salesman problem. *International Journal of Applied Information Systems*, 4:6–12, 2012.

[5] Charu Aggarwal, Alexander Hinneburg, and Daniel Keim. On the surprising behavior of distance metric in high-dimensional space. *First publ. in: Database theory, ICDT 200, 8th International Conference, London, UK, January 4 - 6, 2001 / Jan Van den Bussche ... (eds.). Berlin: Springer, 2001, pp. 420-434 (=Lecture notes in computer science ; 1973)*, 02 2002.

[6] W. Aha, Dennis Kibler, and Marc Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 01 1991.

[7] Jesús Alcalá-Fdez, Luciano Sanchez, Salvador Garcia, Maria Jose del Jesus, Sebastian Ventura, Josep Maria Garrell, Jose Otero, Cristóbal Romero, Jaume Bacardit, Victor M Rivas, et al. Keel: a software tool to assess evolutionary algorithms for data mining problems. *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, 13(3):307–318, 2009.

[8] Haseeb Ali, Mohd Salleh, Rd Saedudin, Kashif Hussain, and Muhammad Mushtaq. Imbalance class problems in data mining: A review. *Indonesian Journal of Electrical Engineering and Computer Science*, 14, 03 2019.

[9] Saleh Alshomrani, Abdullah Bawakid, Seong-O Shim, Alberto Fernández, and Francisco Herrera. A proposal for evolutionary fuzzy systems using feature weighting: dealing with overlapping in imbalanced datasets. *Knowledge-Based Systems*, 73:1–17, 2015.

[10] Khalil Amine. Multiobjective simulated annealing: Principles and algorithm variants. *Advances in Operations Research*, 2019.

[11] Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

[12] Sanghamitra Bandyopadhyay, Sriparna Saha, Ujjwal Maulik, and Kalyanmoy Deb. A simulated annealing-based multiobjective optimization algorithm: Amosa. *IEEE Transactions on Evolutionary Computation*, 12(3):269–283, 2008.

[13] Ricardo Barandela, José Salvador Sánchez, Vicente Garcıa, and Edgar Rangel. Strategies for learning 382 in class imbalance problems. *Pattern Recognition*, 36(3):849–851, 2003.

[14] Victor Hugo Barella, Eduardo de Paula Costa, André Carlos Ponce de Leon Carvalho, et al. Clusteross: a new undersampling method for imbalanced learning. In *Brazilian Conference on Intelligent Systems, 3th; Encontro Nacional de Inteligência Artificial e Computacional, 11th*. Universidade de São Paulo-USP, 2014.

[15] Ajay Basavanhally, Scott Doyle, and Anant Madabhushi. Predicting classifier performance with a small training set: Applications to computer-aided diagnosis and prognosis. In *2010 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 229–232, 2010.

[16] Gustavo EAPA Batista, Ronaldo C Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM Sigkdd Explorations Newsletter*, 6(1):20–29, 2004.

[17] Mohamed Bekkar and Taklit Akrouf Alitouche. Imbalanced data learning approaches review. *International Journal of Data Mining & Knowledge Management Process*, 3(4):15, 2013.

[18] F.J. Berlanga, A.J. Rivera, M.J. del Jesus, and F. Herrera. Gp-coach: Genetic programming-based learning of compact and accurate fuzzy rule-based classification systems for high-dimensional problems. *Information Sciences*, 180(8):1183–1200, 2010.

[19] Cigdem Beyan and Robert Fisher. Classifying imbalanced data sets using similarity based hierarchical decomposition. *Pattern Recognition*, 48(5):1653–1672, 2015.

[20] J.C. Bezdek, T.R. Reichherzer, G.S. Lim, and Y. Attikiouzel. Multiple-prototype classifier design. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 28(1):67–79, 1998.

[21] C. Brodley and M. Friedl. Identifying and eliminating mislabeled training instances. In *AAAI/IAAI, Vol. 1*, 1996.

[22] C. E. Brodley and M. A. Friedl. Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, 11:131–167, Aug 1999.

[23] Chumphol Bunkhumpornpat, Krung Sinapiromsaran, and Chidchanok Lursinsap. Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In Thanaruk Theeramunkong, Boonserm Kijsirikul, Nick Cercone, and Tu-Bao Ho, editors, *Advances in Knowledge Discovery and Data Mining*, pages 475–482, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[24] Zehra Cataltepe and Eser Aygun. An improvement of centroid-based classification algorithm for text classification. In *2007 IEEE 23rd International Conference on Data Engineering Workshop*, pages 952–956, 2007.

[25] Alejandro Cervantes, InÉs MarÍa Galvan, and Pedro Isasi. Ampso: A new particle swarm method for nearest neighborhood classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(5):1082–1091, 2009.

[26] S. Cha. Comprehensive survey on distance/similarity measures between probability density functions. 2007.

[27] Chin-Liang Chang. Finding prototypes for nearest neighbor classifiers. *IEEE Transactions on Computers*, C-23(11):1179–1184, 1974.

[28] Nitesh Chawla, Kevin Bowyer, Lawrence Hall, and W. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res. (JAIR)*, 16:321–357, 06 2002.

[29] Nitesh V Chawla. Data mining for imbalanced datasets: An overview. In *Data mining and knowledge discovery handbook*, pages 875–886. Springer, 2009.

[30] C. H. Chen and Adam Jóundefinedwik. A sample set condensation algorithm for the class sensitive artificial neural network. *Pattern Recogn. Lett.*, 17(8):819–823, July 1996.

[31] Zhi Chen, Tao Lin, Xin Xia, Hongyan Xu, and Sha Ding. A synthetic neighborhood generation based ensemble learning for the imbalanced data classification. *Applied Intelligence*, 48, 08 2018.

[32] Venkata Krishnaveni Chennuru and Sobha Rani Timmappareddy. Mahalcusfilter: A hybrid undersampling method to improve the minority classification rate of imbalanced datasets. In Ashish Ghosh, Rajarshi Pal, and Rajendra Prasath, editors, *Mining Intelligence and Knowledge Exploration*, pages 43–53. Springer International Publishing, 2017.

[33] David Cieslak and Nitesh Chawla. Learning decision trees for unbalanced data. pages 241–256, 09 2008.

[34] David Cieslak, T. Hoens, Nitesh Chawla, and W. Kegelmeyer. Hellinger distance decision trees are robust and skew-insensitive. *Data Min. Knowl. Discov.*, 24:136–158, 01 2012.

[35] Ireneusz Czarnowski. Cluster-based instance selection for machine classification. *Knowl. Inf. Syst.*, 30:113–133, 01 2012.

[36] Ireneusz Czarnowski and Piotr Jedrzejowicz. *An Approach to Imbalanced Data Classification Based on Instance Selection and Over-Sampling*, pages 601–610. 08 2019.

[37] Avisha Das, Shahryar Baki, Ayman El Aassal, Rakesh Verma, and Arthur Dunbar. Sok: A comprehensive reexamination of phishing research from the security perspective, 2019.

[38] Christine Decaestecker. Finding prototypes for nearest neighbour classification by means of gradient descent and deterministic annealing. *Pattern Recognition*, 30(2):281–288, 1997.

[39] J. Derrac, I. Triguero, S. García, and F. Herrera. Survey of new approaches on prototype selection and generation. 2011.

[40] Thomas G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems*, pages 1–15, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[41] José F Díez-Pastor, Juan J Rodríguez, César García-Osorio, and Ludmila I Kuncheva. Random balance: ensembles of variable priors classifiers for imbalanced data. *Knowledge-Based Systems*, 85:96–111, 2015.

[42] Chris Drummond and Robert Holte. Exploiting the cost (in)sensitivity of decision tree splitting criteria. *ICML*, 05 2000.

[43] Ayman El Aassal, Shahryar Baki, Avisha Das, and Rakesh M. Verma. An in-depth benchmarking and evaluation of phishing detection research for security needs. *IEEE Access*, 8:22170–22192, 2020.

[44] Charles Elkan. The foundations of cost-sensitive learning. *Proceedings of the Seventeenth International Conference on Artificial Intelligence: 4-10 August 2001; Seattle*, 1, 05 2001.

[45] H. J. Escalante, Maribel Marin-Castro, A. Morales-Reyes, Mario Graff, Alejandro Rosales-Pérez, M. Montes y Gómez, C. A. R. García, and J. A. Gonzalez. Mopg: a multi-objective evolutionary algorithm for prototype generation. *Pattern Analysis and Applications*, 20:33–47, 2015.

[46] Hugo Jair Escalante, Mario Graff, and Alicia Morales-Reyes. Pggp: Prototype generation via genetic programming. *Applied Soft Computing*, 40:569–580, 2016.

[47] Hatem A. Fayed, Sherif R. Hashem, and Amir F. Atiya. Self-generating prototypes for pattern classification. *Pattern Recognition*, 40(5):1498–1509, 2007.

[48] Juan Carlos Fernández, Mariano Carbonero, Pedro Antonio Gutiérrez, and Cesar Martínez. Multi-objective evolutionary optimization using the relationship between f1 and accuracy metrics in classification tasks. *Applied Intelligence*, 49, 09 2019.

[49] George Forman and Ira Cohen. Learning from little: Comparison of classifiers given little training. In Jean-François Boulicaut, Floriana Esposito, Fosca Giannotti, and Dino Pedreschi, editors, *Knowledge Discovery in Databases: PKDD 2004*, pages 161–172, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[50] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4):463–484, 2012.

[51] S. García, Alicia D. Benítez, F. Herrera, and A. Fernández. Statistical comparisons by means of non-parametric tests : A case study on genetic based machine learning. 2007.

[52] Salvador Garcia, Joaquin Derrac, Jose Cano, and Francisco Herrera. Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):417–435, 2012.

[53] G. Gates. The reduced nearest neighbor rule (corresp.). *IEEE Trans. Inf. Theory*, 18:431–433, 1972.

[54] S. Geva and J. Sitte. Adaptive nearest neighbor pattern classification. *IEEE Transactions on Neural Networks*, 2(2):318–322, 1991.

[55] Xinjian Guo, Yilong Yin, Cailing Dong, Gongping Yang, and Guangtong Zhou. On the class imbalance problem. In *Natural Computation, 2008. ICNC'08. Fourth International Conference on*, volume 4, pages 192–201. IEEE, 2008.

[56] Y. Hamamoto, S. Uchimura, and S. Tomita. A bootstrap technique for nearest neighbor classifier design. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1):73–79, 1997.

[57] Peter Hart. The condensed nearest neighbor rule (corresp.). *IEEE transactions on information theory*, 14(3):515–516, 1968.

[58] Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328, 2008.

[59] Yumilka B. Fernandez Hernandez, Rafael Bello, Yaima Filiberto, Mabel Frias Dominguez, Lenniet Coello Blanco, and Y. Mota. An approach for prototype generation based on similarity relations for problems of classification. *Computación y Sistemas*, 19, 2015.

[60] T. K. Ho and M. Basu. Complexity measures of supervised classification problems. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(3):289–300, March 2002.

[61] Weiwei Hu and Ying Tan. Prototype generation using multiobjective particle swarm optimization for nearest neighbor classification. *IEEE Transactions on Cybernetics*, 46(12):2719–2731, 2016.

[62] M. Kamber J. Han and J. Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc.,, 3rd edition edition, 2011.

[63] Kahksha Jalal and Sameena Naaz. Detection of phishing website using machine learning approach. 04 2019.

[64] Norbert Jankowski and Marek Grochowski. Comparison of instances selection algorithms i. algorithms survey. In Leszek Rutkowski, Jörg H. Siekmann, Ryszard Tadeusiewicz, and

Lotfi A. Zadeh, editors, *Artificial Intelligence and Soft Computing - ICAISC 2004*, pages 598–603, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[65] Nathalie Japkowicz et al. Learning from imbalanced data sets: a comparison of various strategies. In *AAAI workshop on learning from imbalanced data sets*, volume 68, pages 10–15. Menlo Park, CA, 2000.

[66] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

[67] S Jayasree and A Alice Gavya. Addressing imbalance problem in the class–a survey. *International Journal of Application or Innovation in Engineering & Management*, 3(9), 2014.

[68] I.Jahorina J.D. Novakovic, A.Veijovic. Adaboost as classifier ensemble in classification problems. 13:616–620, 2014.

[69] Taeho Jo and Nathalie Japkowicz. Class imbalances versus small disjuncts. *ACM Sigkdd Explorations Newsletter*, 6(1):40–49, 2004.

[70] Anna Jurek-Loughrey, Yaxin Bi, Shengli Wu, and Chris Nugent. A survey of commonly used ensemble-based classification techniques. *The Knowledge Engineering Review*, 29:551–581, 11 2013.

[71] B. Nair Kevin Knight, Elaine Rich. *Artificial Intelligence (3e)*. Tata Mcgrahill, 2017.

[72] Mahmoud Khonji, Youssef Iraqi, and Andrew Jones. Phishing detection: A literature survey. *IEEE Communications Surveys Tutorials*, 15(4):2091–2121, 2013.

[73] J. Koplowitz and T. A. Brown. On the relation of performance to editing in nearest neighbor rules. *Pattern Recognit.*, 13:251–255, 1981.

[74] Sotiris Kotsiantis, D. Kanellopoulos, and P. Pintelas. Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*, 30:25–36, 11 2005.

[75] B. Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5:221–232, 2016.

[76] CV KrishnaVeni and T Sobha Rani. On the classification of imbalanced datasets. *IJCST*, 2(SP1):145–148, 2011.

[77] Pawel Ksieniewicz. Combining random subspace approach with smote oversampling for imbalanced data classification. In *Hybrid Artificial Intelligent Systems*, pages 660–673, Cham, 2019. Springer International Publishing.

[78] Miroslav Kubat, Stan Matwin, et al. Addressing the curse of imbalanced training sets: one-sided selection. In *ICML*, volume 97, pages 179–186. Nashville, USA, 1997.

[79] Wai Lam, Chi-Kin Keung, and Danyu Liu. Discovering useful concept prototypes for classification based on filtering and abstraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1075–1090, 2002.

[80] Jorma Laurikkala. Improving identification of difficult small classes by balancing class distribution. *Artificial Intelligence in Medicine*, pages 63–66, 2001.

[81] Dalong Li and Steven Simske. Training set compression by incremental clustering. *ORG JOURNAL OF PATTERN RECOGNITION RESEARCH*, 1:56–64, 01 2011.

[82] Jinyan(Leo) Li, Simon Fong, Raymond Wong, and Victor Chu. Adaptive multi-objective swarm fusion for imbalanced data classification. *Information Fusion*, 39, 03 2017.

[83] Charles Ling and Victor Sheng. Cost-sensitive learning and the class imbalance problem. *Encyclopedia of Machine Learning*, 01 2010.

[84] Charles X. Ling, Qiang Yang, Jianning Wang, and Shichao Zhang. Decision trees with minimal costs. In *In Proceedings of the Twenty-First International Conference on Machine Learning*, pages 4–8. Morgan Kaufmann, 2004.

[85] Wei Liu, Sanjay Chawla, David Cieslak, and Nitesh Chawla. A robust decision tree algorithm for imbalanced data sets. pages 766–777, 04 2010.

[86] Xu-ying Liu and Zhi-hua Zhou. The influence of class imbalance on cost-sensitive learning: An empirical study. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 970–974, 2006.

[87] Mr Rushi Longadge, Ms Snehlata S Dongre, and Latesh Malik. Multi-cluster based approach for skewed data in data mining. *Journal of Computer Engineering (IOSR-JCE)*, 12(6):66–73, 2013.

[88] M. Lozano, J. M. Sotoca, J. S. Sánchez, F. Pla, E. Pekalska, and R. P. W. Duin. Experimental study on prototype optimisation algorithms for prototype-based classification in vector spaces. *Pattern Recognition*, 39(10):1827–1838, January 2006.

[89] S. Maheshwari, Jitendra Agrawal, and S. Sharma. A new approach for classification of highly imbalanced datasets using evolutionary algorithms. 2011.

[90] Marcus Maloof. Learning when data sets are imbalanced and when costs are unequal and unknown. *Analysis*, 21, 07 2003.

[91] M Manjula and T Seeniselvi. Ensembles of first order logical decision trees for imbalanced classification problems.

[92] T. Maruthi Padmaja, P. Radha Krishna, and Raju S. Bapi. Majority filter-based minority prediction (mfmp): An approach for unbalanced datasets. In *TENCON 2008 - 2008 IEEE Region 10 Conference*, pages 1–6, 2008.

[93] Pyry Matikainen, Rahul Sukthankar, and Martial Hebert. Classifier ensemble recommendation. In Andrea Fusiello, Vittorio Murino, and Rita Cucchiara, editors, *Computer Vision – ECCV 2012. Workshops and Demonstrations*, pages 209–218, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[94] Maciej A Mazurowski, Jordan M Malof, and Georgia D Tourassi. Comparative analysis of instance selection algorithms for instance-based classifiers in the context of medical decision support. *Physics in Medicine and Biology*, 56(2):473–489, dec 2010.

[95] Kate McCarthy, Bibi Zabar, and Gary Weiss. Does cost-sensitive learning beat sampling for classifying rare classes? *Learning*, 01 2005.

[96] Tom Mitchell. *Machine Learning*. McGraw Hill, 1997.

[97] RA Mollineda, R Alejo, and JM Sotoca. The class imbalance problem in pattern classification and learning. In *II Congreso Español de Informática (CEDI 2007). ISBN*, pages 978–84, 2007.

[98] R.A. Mollineda, F.J. Ferri, and E. Vidal. A merge-based condensing strategy for multiple prototype classifiers. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 32(5):662–668, 2002.

[99] Maria Carolina Monard and Gustavo EAPA Batista. Learmng with skewed class distrihutions. *Advances in Logic, Artificial Intelligence, and Robotics: LAPTEC*, 85(2002):173, 2002.

[100] Mr. KHARADE SACHIN Mr. RASKAR RAHUL BHAUSAHEB, Prof. SANDEEP KUMAR. Centroidal distance based offline signature recognition system using global and local

features. *IPASJ INTERNATIONAL JOURNAL OF COMPUTER SCIENCE(IIJCS)*, 3:36–42, 2015.

[101] M.Sewell. Ensemble learning, 2008.

[102] Krystyna Napierala and Jerzy Stefanowski. Types of minority class examples and their influence on learning classifiers from imbalanced data. *Journal of Intelligent Information Systems*, 46, 07 2015.

[103] Wing WY Ng, Junjie Hu, Daniel S Yeung, Shaohua Yin, and Fabio Roli. Diversified sensitivity-based undersampling for imbalance classification problems. *IEEE transactions on cybernetics*, 45(11):2402–2412, 2015.

[104] Giang Nguyen, Abdesselam Bouzerdoum, and Son Phung. *Learning Pattern Classification Tasks with Imbalanced Data Sets*. 10 2009.

[105] José Olvera-López, Jesús Carrasco-Ochoa, José Francisco Martínez-Trinidad, and Josef Kittler. A review of instance selection methods. *Artif. Intell. Rev.*, 34:133–143, 08 2010.

[106] David Opitz and Richard Maclin. Popular ensemble methods: An empirical study. 11, 12 1999.

[107] Arash Pourhabib. Empirical similarity for absent data generation in imbalanced classification. *Advances in Information and Communication*, page 1010–1030, Feb 2019.

[108] M Mostafizur Rahman and D Davis. Cluster based under-sampling for unbalanced cardiovascular data. In *Proceedings of the World Congress on Engineering*, volume 3, pages 3–5, 2013.

[109] M Mostafizur Rahman and DN Davis. Addressing the class imbalance problem in medical datasets. *International Journal of Machine Learning and Computing*, 3(2):224, 2013.

[110] D Ramyachitra and P Manikandan. Imbalanced dataset classification and solutions: a review. *International Journal of Computing and Business Research (IJCBR)*, 5(4), 2014.

[111] S.J. Raudys and A.K. Jain. Small sample size effects in statistical pattern recognition: recommendations for practitioners. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):252–264, 1991.

[112] G. Ritter, H. Woodruff, S. Lowry, and T. Isenhour. An algorithm for a selective nearest neighbor decision rule (corresp.). *IEEE Transactions on Information Theory*, 21(6):665–669, 1975.

[113] Alejandro Rosales-Pérez, Hugo Jair Escalante, Carlos Coello, Jesus Gonzalez, and Carlos Alberto Reyes-Garcia. An evolutionary multi-objective approach for prototype generation. pages 1100–1107, 07 2014.

[114] Alejandro Rosales-Pérez, Hugo Jair Escalante, Carlos A. Coello Coello, Jesus A. Gonzalez, and Carlos A. Reyes-Garcia. An evolutionary multi-objective approach for prototype generation. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 1100–1107, 2014.

[115] Nidhi H. Ruparel, Nitin M. Shahane, and Devyani P. Bhamare. Article: Learning from small data set to build classification model: A survey. *IJCA Proceedings on International Conference on Recent Trends in Engineering and Technology 2013*, ICRTET(4):23–26, May 2013. Full text available.

[116] Marc Sebban, Richard Nock, Jean-Hugues Chauchat, and Ricco Rakotomalala. Impact of learning set quality and size on decision tree performances. *Int. J. Comput. Syst. Signal*, 1:85–105, 01 2000.

[117] Parinaz Sobhani, Herna Viktor, and Stan Matwin. Learning from imbalanced data using ensemble methods and cluster-based undersampling. In *International Workshop on New Frontiers in Mining Complex Patterns*, pages 69–83. Springer, 2014.

[118] José Sotoca, José Sánchez, and R Mollineda. A review of data complexity measures and their applicability to pattern classification problems. *Actas del III Taller Nacional de Mineria de Datos y Aprendizaje*, 01 2005.

[119] K. Mishra S.Site, Sahna. A review of ensemble technique for improving majority voting for classifier. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3:177–180, 2013.

[120] Yanmin Sun, Mohamed S. Kamel, Andrew Wong, and Yang Wang. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition*, 40:3358–3378, 12 2007.

[121] Zhongbin Sun, Qinbao Song, Xiaoyan Zhu, Heli Sun, Baowen Xu, and Yuming Zhou. A novel ensemble method for classifying imbalanced data. *Pattern Recognition*, 48(5):1623–1637, 2015.

[122] Zhongbin Sun, Qinbao Song, Xiaoyan Zhu, Heli Sun, Baowen Xu, and Yuming Zhou. A novel ensemble method for classifying imbalanced data. *Pattern Recognition*, 48(5):1623–1637, 2015.

[123] José Sánchez. High training set size reduction by space partitioning and prototype abstraction. *Pattern Recognition*, 37:1561–1564, 01 2004.

[124] J.S. Sánchez, R. Barandela, A.I. Marqués, R. Alejo, and J. Badenas. Analysis of new techniques to obtain quality training sets. *Pattern Recognition Letters*, 24(7):1015–1022, 2003.

[125] Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, and Vipin Kumar. *Introduction to Data Mining*. Pearson Education, 2005.

[126] David Tax and Robert Duin. Learning curves for the analysis of multiple instance classifiers. volume 5342, pages 724–733, 12 2008.

[127] Sobha Ti and P.V. Soujanya. An ensemble method using small training sets for imbalanced data sets: Application to drugs used for kinases. pages 516–521, 08 2013.

[128] Ivan Tomek. Two modifications of cnn. *IEEE Trans. Systems, Man and Cybernetics*, 6:769–772, 1976.

[129] I. Triguero, J. Derrac, S. García, and F. Herrera. Prototype generation for nearest neighbor classification : Survey of methods 1 prototype generation for nearest neighbor classification : Survey of methods. 2011.

[130] Isaac Triguero, Joaquín Derrac, Salvador Garcia, and Francisco Herrera. A taxonomy and experimental study on prototype generation for nearest neighbor classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(1):86–100, 2012.

[131] Sofia Visa and Anca Ralescu. Issues in mining imbalanced data sets - a review paper. *Proc. 16th Midwest Artificial Intelligence and Cognitive Science Conference*, 01 2005.

[132] C Wang, L Hu, M Guo, X Liu, and Q Zou. imdc: an ensemble learning method for imbalanced classification with mirna data. *Genetics and Molecular Research*, 14(1):123–133, 2015.

[133] Gary Weiss, Kate McCarthy, and Bibi Zabar. Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs? pages 35–41, 01 2007.

[134] Gary Weiss and Foster Provost. Learning when training data are costly: The effect of class distribution on tree induction. *J. Artif. Intell. Res. (JAIR)*, 19:315–354, 07 2003.

[135] D. Wilson and Tony Martinez. Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38:257–286, 01 2000.

[136] Dennis L. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-2(3):408–421, 1972.

[137] Odorico R. Learning Vector Quantization with Training Count (LVQTC). *Neural Netw.*, pages 1083–1088, 1997.

[138] Ian H Witten, Eibe Frank, Leonard E Trigg, Mark A Hall, Geoffrey Holmes, and Sally Jo Cunningham. Weka: Practical machine learning tools and techniques with java implementations. 1999.

[139] D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[140] Wenhao Xie, Gongqian Liang, Zhonghui Dong, Baoyu Tan, and Baosheng Zhang. An improved oversampling algorithm based on the samples' selection strategy for classifying imbalanced data. *Mathematical Problems in Engineering*, 2019:1–13, 05 2019.

[141] Yitian Xu, Qian Wang, Xinying Pang, and Ying Tian. Maximum margin of twin spheres machine with pinball loss for imbalanced data classification. *Applied Intelligence*, 48:1–12, 01 2018.

[142] Show-Jane Yen and Yue-Shi Lee. Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Systems with Applications*, 36(3):5718–5727, 2009.

[143] M.R. Gupta A. Rahimi L. Cazzanti Y.H.Chen, E.K. Garcia. Similarity -based classification : Concepts and algorithms. *Journal of Machine Learning Research*, pages 747–776, 2009.

[144] Ping yi, Yuxiang Guan, Futai Zou, Yao Yao, Wei Wang, and Ting Zhu. Web phishing detection using a deep learning framework. *Wireless Communications and Mobile Computing*, 2018:1–9, 09 2018.

[145] Kihoon Yoon and Stephen Kwek. An unsupervised learning approach to resolving the data imbalanced issue in supervised learning problems in functional genomics. In *Fifth International Conference on Hybrid Intelligent Systems (HIS'05)*, pages 6 pp.–, 2005.

[146] B. Zadrozny, J. Langford, and N. Abe. Cost-sensitive learning by cost-proportionate example weighting. In *Third IEEE International Conference on Data Mining*, pages 435–442, 2003.

[147] J. Zhang and I. Mani. KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction. In *Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Datasets*, 2003.

[148] Shu Zhang, Samira Sadaoui, and Malek Mouhoub. An empirical analysis of imbalanced data classification. *Computer and Information Science*, 8(1):151, 2015.

[149] Weijie Zheng and Hong Zhao. Cost-sensitive hierarchical classification for imbalance classes. *Applied Intelligence*, 50, 08 2020.

Reduction Strategies to Tackle Class Imbalance in Datasets

ORIGINALITY REPORT

33%	10%	29%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

Overall Similarity : 33-(10+5+5+4+4) = 5%.

1	Venkata Krishnaveni Chennuru, Sobha Rani Timmapparedy. "Simulated annealing based undersampling (SAUS): a hybrid multi-objective optimization method to tackle class imbalance", Applied Intelligence, 2021 Publication This publication belongs to my student	10%
2	www.ijcst.com Internet Source This publication belongs to my student.	5%
3	C.V. Veni, T. Rani. "Ensemble based classification using small training sets : A novel approach", 2014 IEEE Symposium on Computational Intelligence in Ensemble Learning (CIEL), 2014 Publication This publication belongs to my student,	5%
4	"Mining Intelligence and Knowledge Exploration", Springer Science and Business Media LLC, 2017 Publication This publication belongs to my student.	4%
5	C.V. Krishna Veni, T. Sobha Rani. "Quartiles based UnderSampling(QUS): A Simple and Novel Method to increase the Classification This publication belongs to my student.	4%

T. SOBHA RANI
Associate Professor
School of Computer & Information Sciences
University of Hyderabad,
Hyderabad-500 046.

rate of positives in Imbalanced Datasets", 2017 Ninth International Conference on Advances in Pattern Recognition (ICAPR), 2017
Publication

6	sci2s.ugr.es Internet Source	1 %
7	www.springerprofessional.de Internet Source	<1 %
8	Submitted to KYUNG HEE UNIVERSITY Student Paper	<1 %
9	Submitted to The Hong Kong Polytechnic University Student Paper	<1 %
10	Venkata Krishnaveni Chennuru, Sobha Rani Timmapparedy. "Chapter 5 MahalCUSFilter: A Hybrid Undersampling Method to Improve the Minority Classification Rate of Imbalanced Datasets", Springer Science and Business Media LLC, 2017 Publication	<1 %
11	Lecture Notes in Computer Science, 2006. Publication	<1 %
12	link.springer.com Internet Source	<1 %
13	tutorsonspot.com Internet Source	<1 %

14	Lecture Notes in Computer Science, 2002. Publication	<1 %
15	Submitted to Liverpool John Moores University Student Paper	<1 %
16	Wibowo Adi, Kosuke Sekiyama. "One double-stranded DNA probes as classifier of multi targeting strand", 2014 International Symposium on Micro-NanoMechatronics and Human Science (MHS), 2014 Publication	<1 %
17	Submitted to Indian Institute of Science, Bangalore Student Paper	<1 %
18	dar.aucegypt.edu Internet Source	<1 %
19	Lee, Yen-Hsien, Paul Jen-Hwa Hu, Tsang-Hsiang Cheng, Te-Chia Huang, and Wei-Yao Chuang. "A preclustering-based ensemble learning technique for acute appendicitis diagnoses", Artificial Intelligence in Medicine, 2013. Publication	<1 %
20	www.ijetae.com Internet Source	<1 %

21	Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C. Prati, Bartosz Krawczyk, Francisco Herrera. "Learning from Imbalanced Data Sets", Springer Science and Business Media LLC, 2018 Publication	<1 %
22	www.tdx.cat Internet Source	<1 %
23	"Artificial Neural Networks – ICANN 2006", Springer Science and Business Media LLC, 2006 Publication	<1 %
24	mafiadoc.com Internet Source	<1 %
25	"Hybrid Artificial Intelligent Systems", Springer Science and Business Media LLC, 2018 Publication	<1 %
26	krchowdhary.com Internet Source	<1 %
27	150.214.191.180 Internet Source	<1 %
28	www.jcomsec.org Internet Source	<1 %
29	Zhang, Zhiwang, Guangxia Gao, and Yingjie Tian. "Multi-kernel multi-criteria optimization classifier with fuzzification and penalty factors	<1 %

for predicting biological activity", Knowledge-Based Systems, 2015.
Publication

30	Submitted to Bournemouth University Student Paper	<1 %
31	Submitted to Tilburg University Student Paper	<1 %
32	"Uncertainty Management with Fuzzy and Rough Sets", Springer Science and Business Media LLC, 2019 Publication	<1 %
33	repository.tudelft.nl Internet Source	<1 %
34	Maryam Amir Haeri, Katharina Anna Zweig. "The Crucial Role of Sensitive Attributes in Fair Classification", 2020 IEEE Symposium Series on Computational Intelligence (SSCI), 2020 Publication	<1 %
35	Submitted to University of Stellenbosch, South Africa Student Paper	<1 %
36	biomedical-engineering-online.biomedcentral.com Internet Source	<1 %
37	hal-insu.archives-ouvertes.fr Internet Source	<1 %

38	oro.open.ac.uk Internet Source	<1 %
39	"Evolutionary Multi-Criterion Optimization", Springer Science and Business Media LLC, 2021 Publication	<1 %
40	"New Trends in Computational Vision and Bio-inspired Computing", Springer Science and Business Media LLC, 2020 Publication	<1 %
41	Cano, José-Ramón. "Analysis of data complexity measures for classification", Expert Systems with Applications, 2013. Publication	<1 %
42	Frédéric Galliano. "Variations of the Mid - IR Aromatic Features inside and among Galaxies", The Astrophysical Journal, 05/20/2008 Publication	<1 %
43	MaciÃ , NÃºria, Ester BernadÃ³-Mansilla, Albert Orriols-Puig, and Tin Kam Ho. "Learner excellence biased by data set selection: A case for data characterisation and artificial data sets", Pattern Recognition, 2013. Publication	<1 %
44	Shan Suthaharan. "Machine Learning Models and Algorithms for Big Data Classification",	<1 %

Springer Science and Business Media LLC, 2016
Publication

45	T. Maruthi Padmaja, P. Radha Krishna, Raju S. Bapi. "Majority filter-based minority prediction (MFMP): An approach for unbalanced datasets", TENCON 2008 - 2008 IEEE Region 10 Conference, 2008 Publication	<1 %
46	digitalcommons.njit.edu Internet Source	<1 %
47	"Computational Science – ICCS 2021", Springer Science and Business Media LLC, 2021 Publication	<1 %
48	Bay, S.D.. "Nearest neighbor classification from multiple feature subsets", Intelligent Data Analysis, 199909 Publication	<1 %
49	Lecture Notes in Computer Science, 2011. Publication	<1 %
50	docplayer.org Internet Source	<1 %
51	journals.plos.org Internet Source	<1 %
52	solon.cma.univie.ac.at Internet Source	<1 %

53	Submitted to Auckland University of Technology Student Paper	<1 %
54	Bart Baesens, Véronique Van Vlasselaer, Wouter Verbeke. "Predictive Analytics for Fraud Detection", Wiley, 2015 Publication	<1 %
55	Gabriela Oliveira Biondi, Ronaldo Cristiano Prati. "Setting Parameters for Support Vector Machines using Transfer Learning", Journal of Intelligent & Robotic Systems, 2015 Publication	<1 %
56	www.wip.opticsinfobase.org Internet Source	<1 %
57	"Advances in Natural Computation", Springer Science and Business Media LLC, 2005 Publication	<1 %
58	"Imbalanced Learning", Wiley, 2013 Publication	<1 %
59	Haonan Tong, Shihai Wang, Guangling Li. "Credibility Based Imbalance Boosting Method for Software Defect Proneness Prediction", Applied Sciences, 2020 Publication	<1 %
60	Saeed Zeraatkar, Fatemeh Afsari. "Interval–valued fuzzy and intuitionistic fuzzy–KNN for	<1 %

imbalanced data classification", Expert Systems with Applications, 2021
Publication

61	Submitted to Symbiosis International University Student Paper	<1%
62	Yifeng Zheng, Guohe Li, Wenjie Zhang. "A New Efficient Algorithm Based on Multi-Classifiers Model for Classification", International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2020 Publication	<1%
63	tel.archives-ouvertes.fr Internet Source	<1%

Exclude quotes	On	Exclude matches	< 14 words
Exclude bibliography	On		