

A THESIS
ON
**OPTIMAL TEXT CLASSIFICATION USING
NATURE INSPIRED ALGORITHMS**

BY

MS. ANSHU KHURANA
(2K17/PHD/CO/02)

UNDER THE SUPERVISION OF

PROF. O. P. VERMA
PROFESSOR
DEPARTMENT OF ELECTRONICS & COMMUNICATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF THE
DOCTOR OF PHILOSOPHY
IN
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



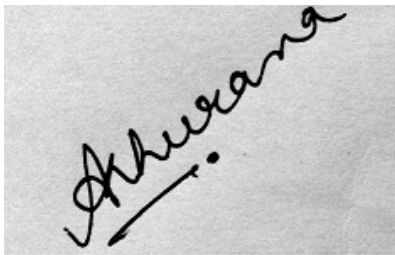
**DELHI TECHNOLOGICAL UNIVERSITY, DELHI
INDIA**

2022

DECLARATION

I here by declare that the thesis entitled “Optimal Text Classification using Nature Inspired Algorithms” submitted by Anshu Khurana, for the award of the degree of *Doctor of Philosophy* to Delhi Technological University is a record of bonafide work carried out under the supervision of Dr. O. P. Verma, Professor, Department of Electronics and Communication, Delhi Technological University, Delhi.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

A photograph of a handwritten signature in black ink on a light-colored surface. The signature reads 'Anshu Khurana' in a cursive style. There is a checkmark to the left of the first letter 'A' and a horizontal line with an arrow pointing to the right below the signature.

Anshu Khurana

Roll No. 2k17/Ph.D/CO/02

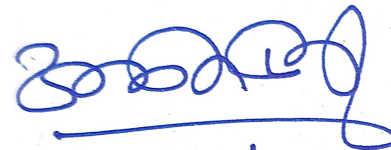
Department of Computer Science and Engineering

Delhi Technological University

Date: 12/5/2022

CERTIFICATE

This is to certify that the thesis entitled “Optimal Text Classification using Nature Inspired Algorithms” submitted by Ms. Anshu Khurana, Roll no. 2k17/Ph.D/CO/02 as a full time scholar in the Department of Computer Science and Engineering, Delhi Technological University for the award of the degree of *Doctor of Philosophy*, is a record of bonafide work carried out by her under my supervision. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.


12/5/2022

Place: Delhi

Dr. O.P. Verma

Date:

Professor

Delhi Technological University

ABSTRACT

Text classification has become a major avenue in generating valuable insights. It is being vastly used to solve real world problems by performing sentimental analysis, detecting frauds and patterns in various sectors like healthcare, e-commerce, sports etc. In Big Data, the performance of text classification can be improved by selecting relevant features and handling of imbalance problems between the distribution of classes in the dataset. In the past, the research work has mostly been done on optimizing the conventional classifiers and tuning the parameters and has deviated from the natural distribution of the data itself. There has now been a radical shift in this approach with the emergence of data science, where the focus is now on understanding the data and feature selection. This research work contributed in the optimization of text classification with four models. Firstly, different nature-inspired algorithms have been explored with various machine learning classifiers to find effective optimized model. The different nature-based techniques used for feature selection are Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Biogeography-Based Optimization (BBO). In the proposed model, feature selection was performed with BBO algorithm along with ensemble classifiers (Bagging). The selected features, after feature selection using BBO algorithm, are classified into various classes using six machine learning classifier. The experimental results are computed on eleven text classification datasets taken from UCI repository. The four different performance measures namely; Accuracy, Precision, Recall and F-measure are used to validate performance of our model with 10-fold cross-validation. Secondly, new optimization algorithm and new dataset balancing algorithm has been proposed. It handles high-dimensional dataset with new nature-based algorithm, Modified Biogeography-Based Optimization (M_BBO). The algorithm works effectively by balancing the dataset with new algorithm of Distributed Synthetic Minority Oversampling Technique (D_SMOTE). The proposed model M_BBO, performs modification in ranking of variables using feature weighting algorithm rather than randomly ranking. Two new expressions in D_SMOTE and one new expression in M_BBO are proposed. The extensive experimental results are computed out on four text classification datasets with four machine learning classifiers. The results are concluded using

three performance measures: 1) Area Under Curve (AUC), 2) G-mean and 3) F1-score. The model works for low dimensional dataset to high dimensional dataset. Thirdly, new optimized model is obtained by tuning parameters of optimization algorithm, that is Grasshopper optimization algorithm and K-Nearest Neighbor and Support Vector Machine classifiers. The tuning is performed with random search technique. The new tuned algorithm successfully provided the new optimal text classification technique. The aim of this meta-heuristic approach is to determine the minimal feature subset from all features to improve the classification performance. Five multi-class datasets are used to evaluate the performance of the model in terms of Accuracy and AUC curve. All results are computed with 10-fold-cross validation method. The evaluated results of the proposed model is compared with other algorithms, which verifies the performance of our technique. The proposed model outperformed among all the compared state-of-the-art techniques. Lastly, our new optimization approach is performed with transfer learning technique. The model aims to consider the feature vectors of both the source and target domain for training the data based on similarity of exemplar (feature) vectors of different instances, known as Instance Similarity Feature (ISF). The exemplar vectors are chosen randomly for the target datasets. Hence, to acquire relevant factual data in the knowledge base for training in our research, we worked to increase the domain separation error between source and target instances. To avoid the instability caused due to poor exemplar vector selection, the K-means clustering approach is followed after feature similarity, known as K-means Instance Similarity Feature (KISF). In order to vanquish the limitations of existing approaches, we have introduced novel optimal models with KISF with Ant Lion Optimizer (KISFA), KISF with Particle Swarm Optimization (KISFP) and KISF with Biogeography Based Optimization (KISFB). High-dimensionality can impact efficacy of the model, hence, feature selection with nature-based optimizer namely: Ant Lion Optimizer, Particle Swarm Optimization and Biogeography Based Optimization are applied. We measure the performance of the proposed models by using Support Vector Machine, Logistic Regression and Random Forest as classifier, and Accuracy and F1-score as fitness functions. Extensive experiments are performed on four datasets with 50 iterations. The proposed model is compared with eleven other techniques and our technique outperforms all other techniques in average Accuracy.

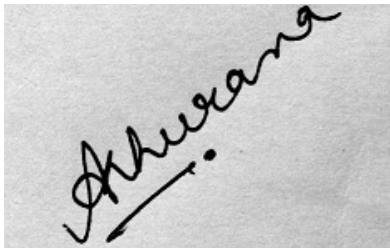
ACKNOWLEDGEMENT

With immense pleasure and deep sense of gratitude, I wish to express my sincere thanks to my supervisor **Dr. O.P. Verma**, Professor, Department of ECE, Delhi Technological University. Without his motivation and continuous encouragement, this research would not have been successfully completed. He has always been my pillar of strength.

I would like to take this opportunity to thank Head of the Department and all the faculty members of Computer Science and Engineering Department, Delhi Technological University for their encouragement and support.

I wish to extend my profound sense of gratitude to **my father Sh. Ramesh Chugh, mother Smt. Varsha Chugh, mother-in-law Smt. Shashi Khurana** for all the sacrifices they made during my research and also providing me with moral support and encouragement whenever required.

Last but not the least, I would like to thank my husband **Jatin Khurana**, my son **Parin Khurana** and my friends for their constant encouragement and moral support along with patience and understanding.

A photograph of a handwritten signature in black ink on a light-colored background. The signature is written in a cursive style and reads 'Anshu Khurana'. There is a small dot at the end of the signature and a horizontal line with an arrow pointing to the right below it.

Ms. Anshu Khurana
Department of Computer Science and Engineering
Delhi Technological University
Delhi-110042

TABLE OF CONTENTS

ABSTRACT	i
ACKNOWLEDGEMENT	iii
LIST OF FIGURES	viii
LIST OF TABLES	x
LIST OF TERMS AND ABBREVIATIONS	xii
LIST OF PUBLICATIONS	xv
1 INTRODUCTION	1
1.1 Text Classification Phases	2
1.1.1 Data Collection	2
1.1.2 Pre-Processing	2
1.1.3 Feature Extraction	3
1.1.4 Feature Selection	3
1.1.5 Building a Classifier	4
1.1.6 Performance Evaluation	4
1.2 Text Classification Techniques	4
1.2.1 Rule Based	4
1.2.2 Machine Learning	5
1.3 Text Classification Improvement	7
1.3.1 Ensemble methods	7
1.3.2 Parameter Tuning	8
1.3.3 Data Pre-processing	8
1.3.4 Meta-heuristic Methods	10
1.4 Challenges in Text classification	12
1.4.1 Problem definition	13
1.4.2 Scope and Objective of the Thesis	14

1.4.3	Contribution and Thesis Layout	16
2	LITERATURE REVIEW	18
2.1	Feature Selection Techniques	18
2.2	Ensemble Based Techniques	20
2.3	Handling Imbalanced Data Techniques	21
2.4	Parameter Tuning Techniques	24
2.5	Transfer Learning in Text Classification	25
3	OPTIMAL TEXT CLASSIFICATION WITH ENSEMBLE METH-	
	ODS	28
3.1	Introduction	28
3.2	Theoretical Foundations	29
3.2.1	Feature Selection Methods	29
3.2.2	Classification Algorithms	35
3.2.3	Performance Measure	37
3.2.4	Dataset description	39
3.2.5	Statistical Test	39
3.3	Methodology	40
3.3.1	Proposed Hybrid Model BBO-Bagging for Feature Selection and Classification	40
3.4	Result	43
3.4.1	Comparison of Different Feature Selection Techniques	43
3.4.2	Comparison of Different Hybrid Nature-Inspired Algorithms and Ensemble Classifier	44
3.4.3	Comparison with Other Techniques	54
3.4.4	Proposed Approach on a Real Dataset of Airlines	56
3.4.5	Result summary	58
3.5	Conclusion	58
4	OPTIMAL FEATURE SELECTION FOR IMBALANCED TEXT CLAS-	
	SIFICATION	60
4.1	Introduction	60
4.2	Theoretical Foundations	61

4.2.1	Modified Biogeography Based Optimization (M_BBO)	61
4.2.2	Proposed Distributed SMOTE-D_SMOTE	64
4.3	Experimental Setup	67
4.3.1	Data Sets and Classifiers	69
4.3.2	Assessment measures	70
4.4	Results and Analysis	71
4.4.1	Performance of Distributed SMOTE with Modified BBO	71
4.4.2	Statistical Analysis	80
4.5	Conclusion	81
5	FINE TUNED GRASSHOPPER OPTIMIZATION ALGORITHM FOR OPTIMAL TEXT CLASSIFICATION	83
5.1	Introduction	83
5.2	Theoretical Foundations	84
5.2.1	Grasshopper optimization	84
5.2.2	Classifiers	86
5.3	Methodology	86
5.4	Experimental Setup	90
5.4.1	Dataset	90
5.4.2	Objective functions	90
5.5	Results	90
5.6	Conclusion	94
6	OPTIMAL HETEROGENEOUS DOMAIN ADAPTATION FOR TEXT CLASSIFICATION	95
6.1	Introduction	95
6.2	The Proposed framework- KISF	97
6.2.1	Notations	97
6.2.2	Instance Similarity Features (ISF)	97
6.2.3	Heterogeneous Domain Adaptation	99
6.2.4	Optimal Feature Selection	103
6.3	Experimental Setup	106
6.3.1	Dataset	106

6.3.2	Baselines	107
6.3.3	Objective Functions	108
6.4	Result	109
6.4.1	Error reduction	114
6.4.2	Parameter Sensitivity Test	115
6.4.3	Statistical Significance	116
6.5	Conclusion	118
7	CONCLUSION AND FUTURE SCOPE	120
7.1	Summary of the work done in the thesis	120
7.2	Future work	121
	REFERENCES	121

LIST OF FIGURES

1.1	Data categorization based on similarity between them.	2
1.2	Text classification process	3
1.3	Text classification techniques	5
1.4	Ensemble model	8
1.5	Meta-heuristic algorithms	11
3.1	Feature selection with Genetic Algorithm	31
3.2	Feature selection with PSO	32
3.3	Pseudo code for BBO migration	34
3.4	Pseudocode for BBO mutation	35
3.5	BBO algorithm for feature Selection	36
3.6	Feature selection with BBO	37
3.8	Hybrid BBO-Ensemble Algorithm	41
3.7	Flow chart of the proposed method	42
3.9	Precision and Recall of tr11	45
3.10	Accuracy and F-measure of tr11	46
3.11	Precision and Recall of tr12	47
3.12	Accuracy and F-measure of tr12	48
3.13	Precision and Recall of tr21	49
3.14	Accuracy and F-measure of tr21	50
3.15	Precision and Recall of tr23	51
3.16	Accuracy and F-measure of tr23	52
3.17	Precision and Recall of airlines	56
3.18	Accuracy and F-measure of airlines	57
4.1	BBO Migration	62
4.2	Modified BBO mutation operator	63
4.3	M_BBO Feature Selection	65

4.4	Figure of instances	66
4.5	Comparison of classical SMOTE with Distributed SMOTE	67
4.6	Proposed D_SMOTE	68
5.1	Flow process of the model	88
5.2	Tuned Grasshopper Algorithm used for feature selection	89
5.3	AUC plot of Glass dataset	92
5.4	AUC plot of arrhythmia dataset	93
6.1	Flow chart of ALO as feature selection.	104
6.2	Flow process of the model.	105
6.3	Error graph of ACE and AP	114
6.4	Error graph of Spam and NER	114
6.5	sensitivity analysis for ALO	116
6.6	Sensitivity analysis for PSO	116
6.7	Sensitivity analysis for BBO	117
6.8	Comparison of Sensitivity analysis	117

LIST OF TABLES

3.1	Parameters for BBO feature selection	35
3.2	Description of multi-class dataset	39
3.3	Average Performance values in percentage for Feature selection techniques	44
3.4	Friedman Test results for feature selection techniques	44
3.5	Average performance values in percentage for feature selection techniques	44
3.6	Friedman Test results for feature selection techniques	44
3.8	Comparison of classification Accuracy	54
3.9	Comparison of classification Accuracy	54
3.7	Comparison of performance measures	55
4.1	Parameters for BBO feature selection	64
4.2	Description of multi-class dataset	70
4.3	AUC score for various oversampling methods	73
4.4	AUC score	75
4.5	G-mean score	77
4.7	Comparison of performance measure (AUC X 100) with different approaches	78
4.6	F1-score values	79
4.8	Comparison of performance measure (ACC X 100) with BERT	80
4.9	Average rank by Friedman Test results for AUC measure	81
5.1	Tuning Parameters	87
5.2	Description of multi-class dataset	90
5.3	Accuracy results for various techniques	91
5.4	Comparison of Accuracy results	94
6.1	Predictive Results summary of F1-score	97
6.2	Error reduction Table	102

6.3	F1-score	109
6.4	Accuracy	110
6.5	F1-score with feature selection	110
6.6	Accuracy with feature selection	111
6.7	Predictive Accuracy	113
6.8	Descriptive statistics	118
6.9	Data summary of ANOVA	118

LIST OF TERMS AND ABBREVIATIONS

ACO	Ant Colony Optimization
AUC	Area Under Curve
ADASYN	Adaptive Synthetic
BBO	Biogeography Based Optimization
BERT	Biderictional Encoder Representations from Transformers
CNN	Convolutional Neural Network
DT	Decision Tree
D_SMOTE	Distributed SMOTE
DE	Differential Evolution
DECOC	Diversified Error Correcting Codes
ECOC	Error Correcting Codes
ENN	Wilson Edited Nearest Neighbor
FS	Feature Selection
FP	False Positive
FN	False Negative
FPA	Flower Pollination Algorithm
FSUTL	Feature Selection for Unsupervised Transfer Learning
FA	Firefly Algorithm
FWER	Family Wide Error Rate
GA	Genetic Algorithm
GS	Grid Search
GWO	Grey Wolf Optimization
GOA	Grasshopper Optimization Algorithm
HSI	Habitat Suitability Index
ICT	Information and Communication Technology
IR	Information Retrieval
IG	Information Gain
ISF	Instance Similarity Feature

K-NN	K-Nearest Neighbor
KISFA	K-means and Ant Lion Optimization
KISFP	K-means and Particle Swarm Optimization
KISFB	K-means and Biogeography Based Optimization
LR	Logistic Regression
M_BBO	Modified Biogeography Based Optimization
ML	Machine Learning
MVO	MultiVerse Optimizer
MFO	Moth Flame Optimization
NB	Naive Baye's
OR	Odd's Ratio
OAA	One versus All
OAO	One versus One
PCA	Principial Component Analysis
PA	Progressive Alignment
PCA	Principal Component Analysis
RF	Random Forest
ROC	Receiver Operating Characteristics
RNN	Recurrent Neural Network
RAMO	Ranked Minority Oversampling
SOAP	Simple Object Access Protocol
SA	Simulated Annealing
SEO	Search Engine Optimization
SMOTE	Synthetic Minority Oversampling Technique
SVM	Support Vector Machine
SCA	Sine Cosine Algorithm
SIV	Suitability Index Variable
SMOTE-B	Borderline Synthetic Oversampling Technique
SMOTE-SL	Safe Level Synthetic Minority Oversampling Technique
TS	Tabu Search
TIT	Transfer Independently Together
TF-IDF	Term Frequency-Inverse Document Frequency

TDCN	Temporal Deep Convolutional Network
TP	True Positive
TN	True Negative
VSDM	Vector Space Model

LIST OF PUBLICATIONS

Publication in SCI/SCIE Journals:

1. Khurana, A, and Verma, O.P. (2020). Optimal Feature Selection for Imbalanced Text Classification. *IEEE Transactions on Artificial Intelligence*. doi: 10.1109/TAI.2022.3144651.
2. Khurana, A., and Verma, O. P. (2020). Novel approach with nature-inspired and ensemble techniques for optimal text classification. *Multimedia Tools and Applications*, 79(33), 23821-23848.

Publication under review in SCI/SCIE Journals:

1. Khurana, A., and Verma, O.P. Optimal Heterogeneous Domain Adaptation for Text Classification in Transfer Learning, Expert Systems and Applications.

Publication in International Conferences:

1. Khurana, A., and Verma, O. P. (2020, December). A Fine Tuned Model of Grasshopper Optimization Algorithm with Classifiers for Optimal Text Classification. *IEEE 17th India Council International Conference (INDICON)* (pp. 1-7). IEEE.
2. Khurana, A., and Verma, O. P. (2018). PSO based Optimal Text Classification using Tuned k-NN and Feature Weighting. *International Journal of Information Systems and Management Science*, 1(1). (Presented in 4th International Conference on Computers and Management, Dec 22-23 (ICCM)).

Book Chapter:

1. Khurana, A, and Verma, O.P. (2020). AviR- Autism Rehabilitation with webVR using Text Classification, chapter accepted in “Enabling Technology for Neurodevelopmental Disorders: From Diagnosis to Rehabilitation, Taylor and Francis, CRC Press.

CHAPTER 1

INTRODUCTION

In the era of digital information, when most of the content is available in textual form, there is a wide problem of classification. The area of classification has been widely explored among researchers all over the world. The classification problem is defined as if there exists a training dataset $D = \{Y_1, . . . , Y_N\}$, where the record in every dataset has some class value associated. The class value is retrieved from a labelled set of k , indexed by $\{1 . . . t\}$ discrete values. To build a classification model, there exist training data that assign the class label to every record by establishing the relationship among features. The trained classification model will assign the class to a new record [1]. A defined class label is assigned to the test instance during hard classification, and a probability value is used in case of soft classification.

Text classification is a branch of natural language processing techniques, that classifies the textual documents into predefined class labels. Manual or automatic models exist for text classification. In contrast to manual classification, which is time consuming, automated classification is fast, efficient and more accurate. The application area of text classification consist of categorizing web documents, indexing documents, labelling documents etc. For labels, text classification assumes that records are in categorical form, but, it is also possible for the labels to be continuous. The field that deals with continuous labels is called regression. The text classification model assumes that the information to be classified is in the form of text. All information about the data, and the existence or non-existence of words in the document must be classified. As shown in The focus of text classification is assigning documents like reviews, emails, social posts etc., to one or multiple class labels. As shown in Figure 1.1, the different labels can be spam, non-spam, positive review, negative review, document language, sports news. The application tasks are generally dealt with the help of two techniques namely; Information Retrieval (IR) techniques and Machine Learning (ML) algorithms. The techniques jointly work in assigning the keywords and classifying the documents to defined labels [2]. The Machine Learning technique automatically categorizes the documents, and IR illustrates the text as a feature. The text classification phases are discussed in the next section.



Figure 1.1: Data categorization based on similarity between them.

1.1 Text Classification Phases

Text classification comprises various sub-phases and every phase has its own need and importance. As shown in Fig 1.2, text classifier model consists of the following subprocess namely; a) Data Collection, b) Pre-Processing, c) Feature Extraction, d) Feature Selection, e) Building a Classifier and f) Performance Evaluation. The working of each sub-phase is described in the following subsections.

1.1.1 Data Collection

The first step of text classification is to build database that includes several type of document like .pdf, .html, .text, .doc, .jsp etc. [3] from various sources. The collected documents are trained and tested by different classifier models.

1.1.2 Pre-Processing

In the pre-processing step, the textual document in .doc format is represented in .csv format. The files produced after pre-processing are high-dimensional or have a high number of features [4]. The following steps are taken for pre-processing the documents:

- Removal of stop words: Words that don't change the meaning of a sentence are removed, such as "a", "an", "the".
- Tokenization: This step converts a document into a string of characters, and further splits it into tokens.
- Stemming a word: Applying the stemming algorithm that converts different word forms into similar canonical forms. This step covers the process of conflating

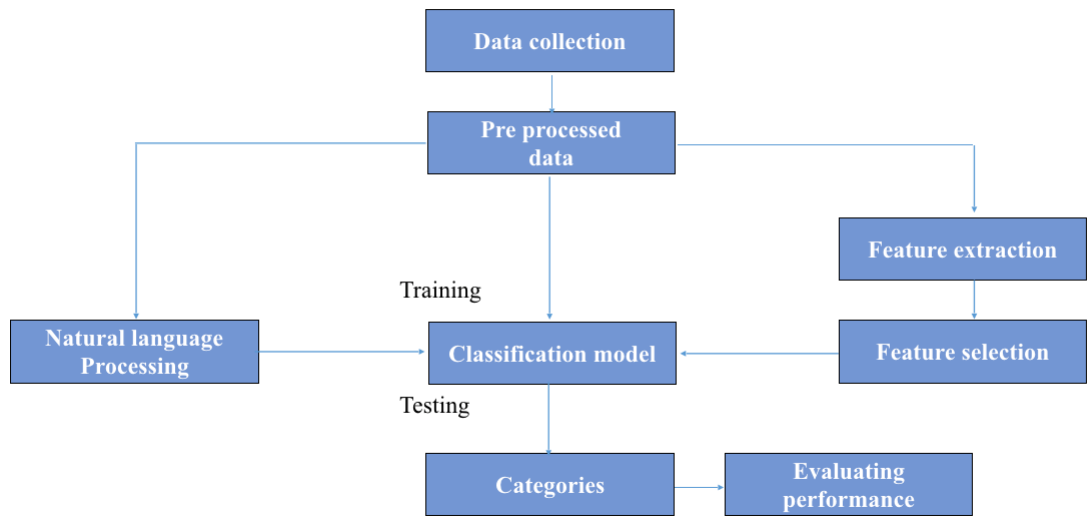


Figure 1.2: Text classification process

tokens to their root form, e.g. connection to connect, computing to compute.

1.1.3 Feature Extraction

The way documents are represented for the usage reduces complexity. In this step complete unstructured document, which is present in textual form, is converted to document vector form. The structured form of data is easy to use and handle. Vector Space Model (VSDM) is the most common way to represent the document. This model represents documents as vectors of words. It is easy to handle but brings some challenges with it too, like a high number of features, loss of actual existing relationships among the words used in the document etc. Many approaches have been introduced to overcome the challenges, namely the term weighting technique, which assigns the significant weights to each term.

1.1.4 Feature Selection

Feature extraction consists of all features of textual document. It may consist of high number of features or some undesirable features which affects the classification performance. Feature Selection (FS) selects the desired features to improve the classification accuracy. The technique uses some predefined measures to determine the important words, and keep the words with high assigned scores. This phase comes under the optimization field. In further sections, we will discuss different methods of feature selection and their working models. The FS gives us the relevant features and remove the irrelevant ones, thus improving the accuracy performance of the classification process.

The textual documents can be automatically classified into predefined labels. It can be

done by different ways, such as, unsupervised, supervised and reinforcements [4]. We will learn about these techniques in the upcoming section. From past few decades, there have been extensive research and studies regarding automatic text classification. Many approaches including the machine learning techniques such as K-Nearest Neighbor (K-NN), Naive Baye's (NB), Decision Tree (DT), Logistic Regression (LR), Support Vector Machines(SVM) etc.

1.1.5 Building a Classifier

An algorithm or classifier defines the rules to classify the facts or data automatically. The designed model is trained and tested with the help of machine learning algorithms for the purpose of classification. The classifier technique is trained on huge corpus. The main objective of the classifier model is to recognize the label or class of new arriving data, depending on the trained data. For designing different classification models, various ML classifiers have been introduced in the literature, such as Random Forest [2], Naive Baye's [2], Support Vector Machine [2] and many more to design different classification models.

1.1.6 Performance Evaluation

After building a classifier, the main goal is to evaluate the performance using different measures. The result of different classifiers is measured using Precision [2], Receiver Operating Characteristic (ROC) curve [5], Recall [2], Accuracy [2], G-mean [2] and F-measure [2].

1.2 Text Classification Techniques

The techniques used in state-of-the-art for the classification of text are Rule Based and Machine Learning. Classification of techniques is shown in Figure.1.3.

1.2.1 Rule Based

In this method the rules are manually written according to the type of problem. This method is not highly accurate as this technique is not flexible at all. Once the model is trained using these rules, it can only work for that specific problem, and any changes to the defined problem could result in misleading results. As this method is manual, it is extremely time consuming process and there could be a lot of errors in developing such model. Therefore, the need was to automate the system, that is achieved through ML.

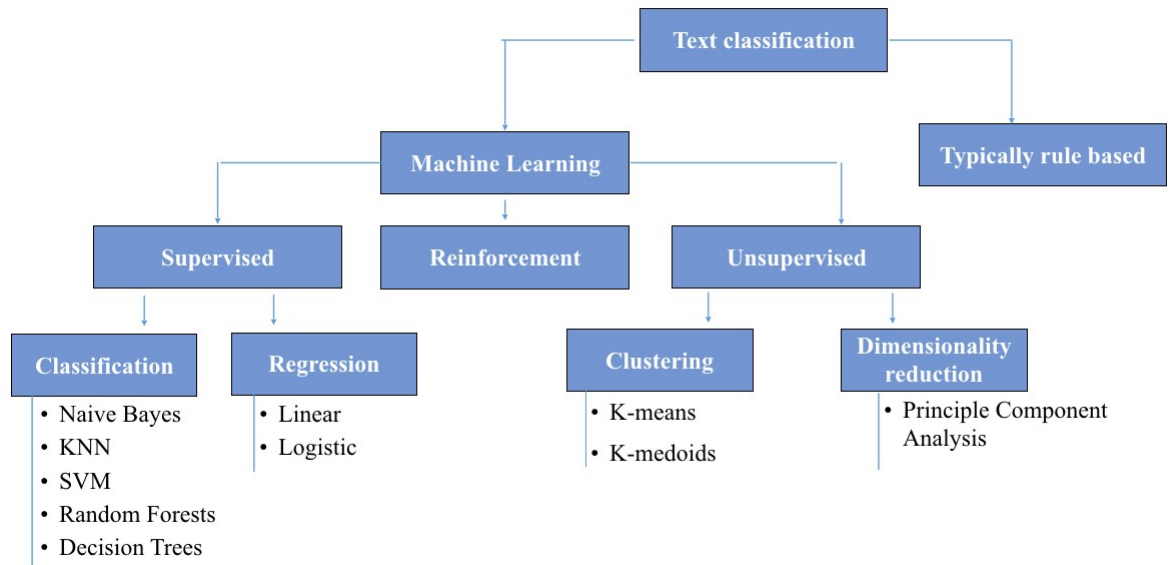


Figure 1.3: Text classification techniques

1.2.2 Machine Learning

Machine learning is a recent technique that is used for data analysis. It trains the data and predicts the new data based on the trained data. ML techniques provides precise way to train, analyze and predict the information from the trained data [6]. The various ML algorithms are introduced by the researchers, which work differently based on the behavior and working approaches. The ML tasks are categorized according to the different nature of learning approaches. It is classified into three types:

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

1.2.2.1 Supervised Learning

In this type of learning, new data can be analyzed and assigned to different categories from trained patterns of data. With the help of learned examples, it becomes easy to determine future tasks, if we have correct trained dataset. If no output trained data is available, then human intervention is required to analyze the new data based on the trained data. It needs a a lot of time and labor to work with trained data with no output data available. Hence, supervised learning algorithms are proved to be effective for wider range of applications.

Supervised machine learning algorithms are classified into two subgroups: Regression and Classification. The data that deals with continuous variable estimated by regression

techniques such as, what will be the height of the child or how many customers will stay with us, etc. Such are handled by supervised ML algorithms, which base their predicted output on the past trained examples. Some of the techniques to perform regression are:

- Linear Regression
- Ridge Regression

The classification learning deals with discrete quantities and not continuous variable, such as, does this report predict cancer? or does this image represents fruit? The simple classification deals with binary classes, where classes are assigned to two categories either true or false. Some of the known classification techniques are:

- Naive Baye's
- K-Nearest Neighbor
- Support Vector Machine
- Random Forest
- Decision Tree

1.2.2.2 Unsupervised Learning

The other type of learning is unsupervised learning. In supervised learning, we refer to the trained pattern. However, unsupervised learning is not based on trained patterns. It can be due to reason that correct values are unobservable, or impractical to obtain, or for a given task, there is no matching "correct answer". This technique can be segregated into two sub-categories:

- Clustering
- Dimensionality Reduction

1.2.2.3 Reinforcement Learning

The third type of learning is reinforcement learning, which learns from the surroundings with the help of interaction and rewards for every action performed. The natural interaction or experience with the environment is the basis of this learning mechanism. Let us understand this with an example, in winter season, if a child is sitting in a living room. The child will approach the fireplace and will feel the warmth. The child will sit around it, associate it as a positive thing and give it a positive reward (Positive reward +1). But, if the child tries to touch the fire with hand, it can burn the hand and give negative rewards (Negative reward -1). Now, it is understood that fire feels good

and warm from a distance. Interaction with the environment is the best way for human to learn. Reinforcement learning is based on optimal learning from rewards for actions. The above information discussed various text classification techniques and how they work. But are they sufficient? Above discussed techniques are not sufficient to build an accurate and precise model. To build a good model we need to optimize the classification techniques, which we will study in our next section.

1.3 Text Classification Improvement

There is an exponential growth in textual data. Text classification is a new research trend in this era. The recent studies focus on how the classification process can be optimized to achieve high performance [7]. Feature selection is one way of achieving it. The techniques which are used to improve performance of text classification are:

- Ensemble methods
- Parameter tuning
- Data pre-processing
- Meta-heuristic methods

1.3.1 Ensemble methods

The ensemble technique combines the output of several base classifiers and produces a merged output [2]. Many researchers in past have worked with ensemble techniques. A combination of various machine learning classifiers have shown improvement in classification performance. There are many techniques that have been developed to perform ensembling. The Ensemble Vote Classifier is a meta-classifier for combining similar or conceptually different machine learning classifiers for classification via majority or plurality voting, as seen in Figure. 1.4 (For simplicity, we will refer to both majority and plurality voting as majority voting.). Some of the ensemble techniques are mentioned below:

- Baye's optimal classifier
- Bagging
- Boosting
- Bucket of models
- Stacking

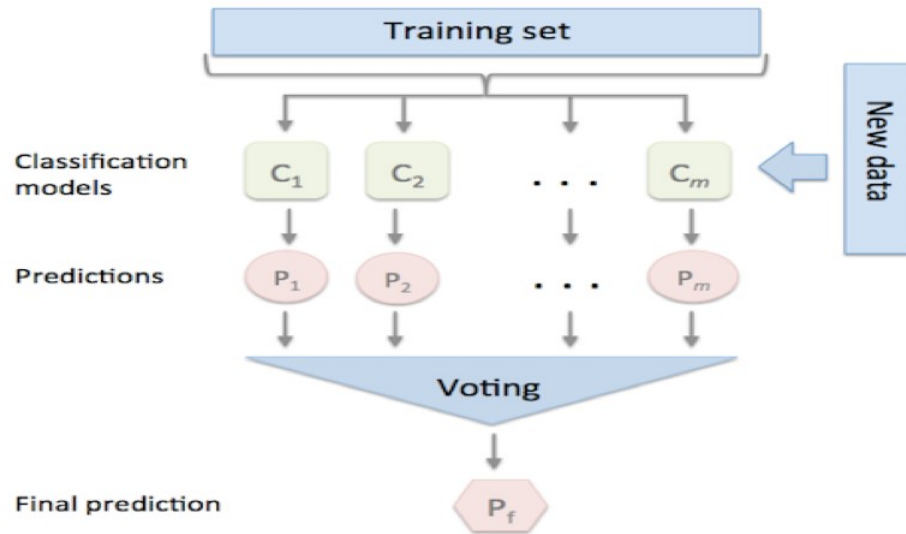


Figure 1.4: Ensemble model

1.3.2 Parameter Tuning

Different parameters have different levels of impact on the classifier. The main objective here is to improve the text classification performance of the classifier. Parameter optimization can highly increase the text classification performance [8]. There are various parameters in a classifier, and their value varies according to the problem. For some problems, some parameters are impactful and for other problems there could be other parameters that are impacting the performance of the classifier. Varied values of the parameter could give very different results. Hence, the aim is to find optimal parameters that could elevate the classifier performance.

1.3.3 Data Pre-processing

The methods we discussed, works on the classifier level, means most of the methods focused on improving the classifier model. Now, we discuss on how the type and distribution of data is important in accurate classifier performance. The pre-processing is applied at the data level. The following types of data are:

- Imbalanced data
- Noisy data
- High dimensional data

1.3.3.1 Imbalanced Data

Imbalanced data or skewed data is the type of data distribution in which majority of the data elements belong to one class and only few of them belong to the other class.

If there exists two classes, positive (+) class and negative (-) class, then, the positive class data elements are very few as compared to the data elements in the negative class. Such distribution is called skewed (imbalanced) distribution. The currently available classifiers do not handle skewed data very well. Hence, the researchers improve the working of the classifier in according to the skewed data. The other method that is used to handle skewed data is to pre-process it before giving it as an input to the classifier. With the help of pre-processing techniques imbalance of dataset can be reduced and the classification performance is improved.

The presence of skewed distribution is very common in real-world problems such as, oil spillage, web reviews etc. The major point to focus during balancing the dataset is on the minority class [9]. Many approaches have been introduced by the researchers to handle the imbalance problem. One of the most popular one is the resampling techniques.

Resampling approaches can be arranged into three categories:

- Under-sampling approach focus on creating a small subset of the original corpus by removing some majority class instances.
- Over-sampling approach creates duplicate instances of minority instances.
- Hybrids approach is a combination of both the sampling approaches.

1.3.3.2 Noisy Data

Noisy data refers to those instances or features which are irrelevant to data [9]. In imbalanced data the presence of noisy data will impact negatively on the classification performance. The current models are not able to distinguish clearly between the noisy data and usable data, which leads to a degraded performance. The classifier model gets confused in differentiating the minority instance and noise. The classifier either accepts all the instances or discards all of them, considering them as noise. In both ways the results obtained are misleading. So it is imperative to handle with the noisy and imbalanced corpus.

1.3.3.3 High-Dimensional Data

When we convert textual data into a structured form, large number of feature space becomes a big challenge [10]. The dimension refers that feature can be visually represented. The representation of the two classes and two features can be easily done in three dimensional space, but representing large number of features mathematically is difficult.

A linear separator in two dimensional space represents line

$$Xx_1 + Yx_2 = C \quad (1.1)$$

But, linear separator represents plane in three dimensional

$$Xx_1 + Yx_2 + Zx_3 = A \quad (1.2)$$

Principal Component Analysis (PCA) is widely used to identify attributes that are orthogonal to others or in other words identifies the principal components in a classification task. Similarly, impurity measures like entropy and information gain are used for dimensionality reduction. Text document is represented in a compact form with the help of keywords, and later using frequency based model, the redundant features could be removed. This reduces the dimensionality of the dataset which in turn leads to improving the classifier performance.

1.3.4 Meta-heuristic Methods

Heuristics is always defined to a specific problem. A heuristic is, for example, choosing a random element for pivoting in Quick-sort. Meta-heuristics refer to a wider range of applications, as they are problem independent. A meta-heuristic technique works like black box where the technique is not aware about the problem. Various meta-heuristic search methods such as Biogeography Based Optimization (BBO), Differential Evolution (DE), Tabu Search (TS), Genetic Programming, Genetic Algorithm (GA), Cuckoo search, Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Ant Colony Optimization (ACO) and Simulated Annealing (SA) are used to search feature subset space for selecting (sub)optimal feature set [11]. In Figure 1.5, the different sub-categories of meta-heuristic algorithms is mentioned. Nature- inspired algorithms, Evolutionary algorithms and Swarm based algorithms are some of the new fields that attract many of the researchers in recent times. They both are closely related to each other. The main goal of nature-inspired algorithms is to find the global optimum solution. The working of nature-inspired algorithms relies on two key factors: diversification and intensification, commonly known as Exploitation and Exploration, respectively [12].

Exploitation finds the local optimum value in already explored space, and exploration finds new random search solution to find the global optimum solution. The challenge with nature-inspired algorithms is to find the balance between these two. Intense exploration does not generate optimal solution and extensive exploitation lead solution to trap in local optima. Their are different ways in which each of the meta-heuristic algorithms work.

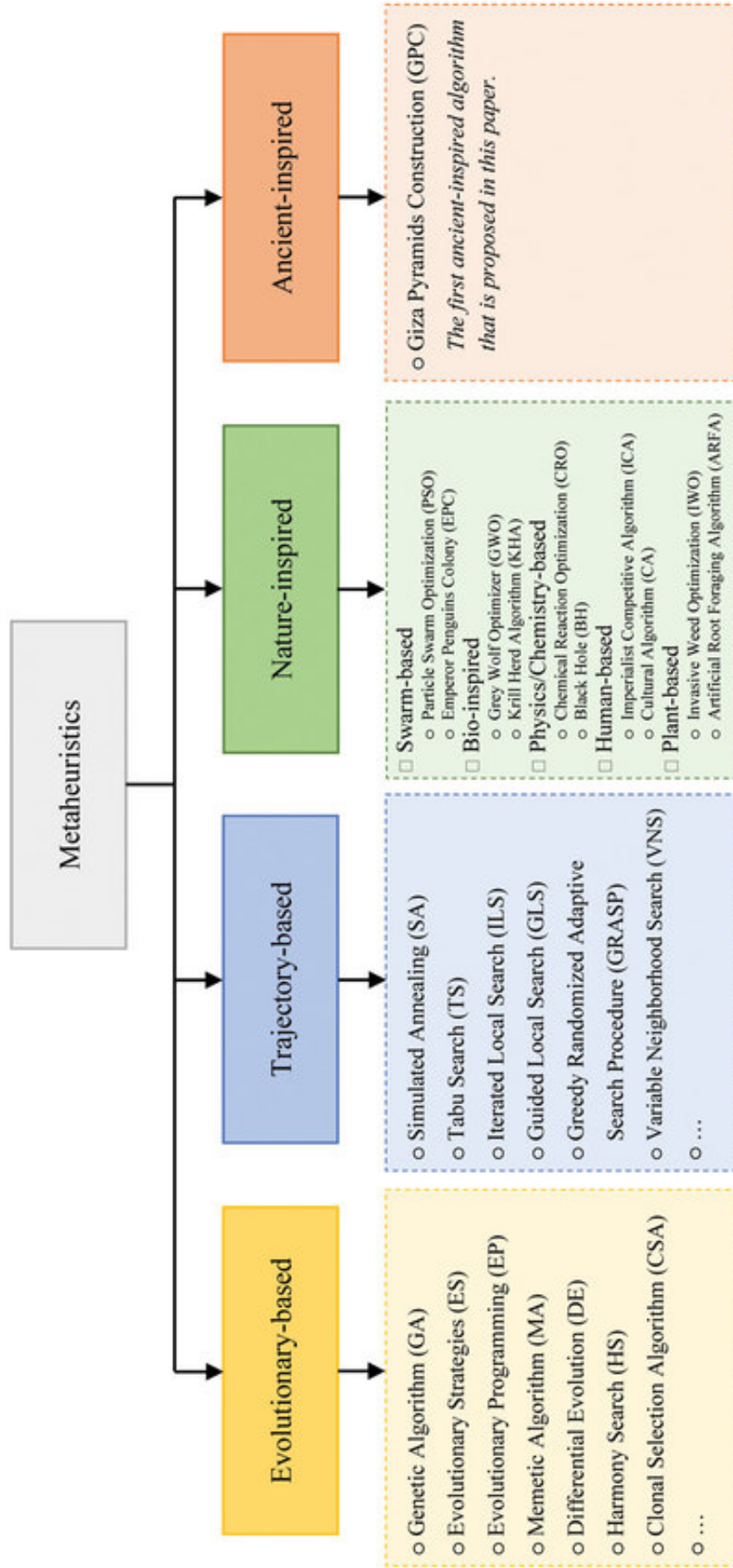


Figure 1.5: Meta-heuristic algorithms

1.4 Challenges in Text classification

- High dimensionality- Large feature space decreases the efficiency of any classification algorithm [13], hence reduces the efficacy of text classification. The presence of noise and un-useful features for dataset can decrease the efficiency to classify text into various classes. The low weight features or noise, that is features which are not essential with respect to data, decreases the efficiency of the algorithm [13], hence degrading the performance of text classification. Feature selection is a process of selecting only desired features related to the data which helps in increasing the efficiency of the model [14]. The use of meta-heuristic algorithm in the state-of-the-art literature as a feature selection technique is proved beneficial to improve classification Accuracy [15] [16] [17]. Various optimization algorithms, such as GA [3], ACO [18], Firefly algorithm [19], PSO [20], BBO [21] and many more are used as feature selection techniques.
- Class imbalance- The other issue that arises during text classification process is class imbalance [17]. Class imbalance [22] has become a major issue in determining the classification performance. Many researchers are working on this problem as imbalance causes adverse effect on the Accuracy of predicted results [23]. The existing solutions to multi-class imbalanced data is generally the extension of algorithm of binary class imbalance classifier. The common oversampling technique used is Synthetic Minority Oversampling Technique (SMOTE) and its different variations [24]. The classical SMOTE and its variant suffer from many limitations.
- Parameter selection of a classifier- The hyper-parameter optimization is the process of determining the optimal parameter value before the beginning of the training process. The main objective of hyper-parameter tuning is to generate the optimum model for a given problem. Hyper-parameter can be obtained by several ways. One way is to give parameters to an objective function to estimate the loss [25]. Another way to obtain the generalized optimization performance of the model is the use of cross-validation [26]. From past few years, focus is on applying the optimization techniques namely Grid Search (GS) [27], Baye'sian Optimization [28], gradient based optimization [29] and Random Search (RS) [26]. Feature selection algorithms emphasize on the selection of important features, hence improves model accuracy [30]. There exist many nature-inspired algorithms which are used to optimize features, namely: GA [31], PSO [12], Grey Wolf Optimization (GWO) [32], Multiverse Optimiser (MVO) [33] and many more. The rise in classification accuracy rely on the selected number of parameters needs to be tuned and regions of search space of the chosen algorithm.

Thus, in hyper-parameter tuning the main problem is choosing the appropriate approach.

- Domain dependent- Transfer learning is another way to improve the text classification performance with improved time complexity. Many machine learning techniques have already been developed to solve the problem of transfer learning caused due to unsupervised learning [34]. The challenge arises during transfer learning when dealing with new and different domain of data. Although, domain adaptation is a big problem [35] [36] [37] [38] [39], when we are provided with labelled source data and unlabeled target data. Various models are developed that aim to transfer the knowledge available in the source domain to classify the text in target domain [40]. The transfer learning approach is already developed and used in many other areas, namely; object recognition [41] [42], event detection [43], classification of images [39] and text categorization [44].

Sometimes, it is difficult to study and work on specific datasets and domains due to the lack of available stable data. If the problem is new and available knowledge is confined, as no historical data is present to learn about the domain [45]. Hence, to bridle the limitations of conventional existing techniques, transfer learning techniques are proposed. The basic ideology is to learn from a model that has been already worked upon to transfer the knowledge from source to the target set. However, there should be some connection/threshold similarity between the features/domains of the source and target set. In the literature, several transfer learning techniques have been developed for domain adaptation. Techniques namely; manifold embedded distribution alignment, transfer component analysis, transfer joint matching, have been proposed by researchers in [46].

1.4.1 Problem definition

Lots of researchers are working to improve text classification performance, due to its wide applicability in the real world. But, there are the some crucial loopholes in the process. So, as to achieve better text classification performance, various models are proposed in this research.

1. The application of individual machine learning algorithms for text classification has been widely used in the real-world applications. Combination of classifiers is called ensemble classifier and known to improve classification. To improve accuracy, the use of ensemble classifiers along with recent feature selection techniques is explored. Different feature selection technique namely; PCA, Information Gain (IG), correlation coefficients etc. are used but overall performance in terms of accuracy was not improved in high dimensional datasets. We analyze the

improvement in accuracy with the use of nature-inspired algorithms for feature selection in high dimensional datasets. Hence, exploring more machine learning algorithms and ensemble classifiers helped us to evaluate the performance on high dimensional datasets.

2. Other major task is to handle imbalanced dataset. Imbalanced dataset is a big problem in text classification especially when dealing with high dimensional datasets. Many techniques like undersampling, Synthetic Minority Oversampling Technique exist to balance the dataset but have one or more shortcomings which affect the classification performance.
3. There are several control parameters used in classification algorithms and evolutionary algorithms to compute results. Most of the studies used default control parameters of classifiers and nature-based optimization to solve optimization problems. Parameter tuning of classifier with optimization technique can improve accuracy.
4. Sometimes, it is difficult to study and work on specific datasets and domains due to the lack of available stable data. If the problem is new and there is confined knowledge about the data, then there is no historical data to learn about the domain. Hence, to bridle the limitations of conventional existing techniques, transfer learning techniques are proposed. Very few studies uses the transfer learning process to improve classification performance. Transfer learning process is applicable only when source and target datasets are of same datatype (homogeneous) or any relation exist between both datatypes. Work on transfer learning among heterogeneous domain datasets has been performed.

1.4.2 Scope and Objective of the Thesis

- Automatic classifying and tagging the content of real world applications such as news sites, e-commerce, content writers and bloggers. Tagging helps to identify the relatable content easily.
- Search Engine Optimization (SEO) is also an application of text classification. With the help of tagging the website can easily crawl and search the content.
- Automated emergency response is very useful during emergency times or panic situations on social sites. The authorities monitors the panic situation and can apply quick solution with the help of automated response.
- Product reviews have become an important and emerging application for online brands. Automated classification of reviews helps marketers to monitor the use of

products. The automated classifier can be used to identify the product's working, promoters, or disparagers.

- Many organizations, researchers and academicians deal with unstructured data. Classifying data into different classes or categories will help businesses or organizations to handle data easily by converting it into structured data.
- Subjectivity detection: This application will determine whether textual information contains the opinion of users or not.
- After the rise in digital blogs sentiment prediction is important for content curators. Text classification can help to classify sentiments as positive or negative.
- Aspect based sentiment summarization: This application helps create summaries of sentiments for any product. It will convert summary into scores or star ratings depending on the keywords used.
- Contrastive viewpoint summarization: This application performs contradiction tasks, where the public opinion matters, such as, working of political parties is good or bad?
- Predicting helpfulness of online comments/reviews: User comments and reviews help new users to determine the usefulness of product. Sorting reviews according to relevance will help the product and user both.

Following are the objective of thesis:

- Explored various nature-inspired algorithms for optimal text classification. Nature-inspired algorithms are used for feature selection. Machine learning algorithms including ensemble classifier is combined with feature selection techniques.
- Transfer learning among two datasets can improve classification performance. Most of the developed models for transfer learning works on homogeneous data type. But, heterogeneous domain adaptation framework is introduced with different feature selection techniques.
- Design and develop nature-based algorithm for feature selection and algorithm for balancing the dataset to improve the accuracy in text classification. Many nature-based optimization algorithms and balancing algorithms are already explored in state-of-the-art literature. The new algorithm is developed to overcome the challenges of conventional existing algorithms.

- Text classification using feature weighting techniques. Many feature selection techniques are developed, but assigning feature weight with the help of tuned optimization algorithm is a challenging task. A new technique by tuning optimization algorithm which automatically assign weights to features will be focused.

1.4.3 Contribution and Thesis Layout

- To deal with high dimensionality, a novel approach is proposed for optimal text classification based on nature-inspired algorithm and ensemble classifier. In the proposed model, feature selection is performed with Biogeography Based Optimization algorithm along with ensemble classifiers (Bagging).
- A new algorithm is introduced to balance the dataset, named Distributed SMOTE (D_SMOTE), which overcomes the problem of lack of density and reduces the formation of small disjuncts. Further, another problem handled is the large number of features or high-dimensionality. A novel feature selection technique is introduced known as modified Biogeography Based Optimization (M_BBO) to solve high-dimensionality.
- Tuned parameter help classifiers to improve classification performance using random search technique.
- A new model of transfer learning for heterogeneous domain adaptation is proposed. A vector method to represent features with the clustering technique improved the classification accuracy.

Finally, the thesis is configured in the following layout:

- **Chapter 1 : INTRODUCTION**
Chapter 1 reports the models for optimal text classification and associated challenges to achieve it. Problem definition, contribution and layout of the thesis is addressed.
- **Chapter 2: LITERATURE REVIEW**
Chapter 2 illustrates the existing models shortcomings for optimal text classification. A fleet study of handling high-dimensional techniques, techniques to handle imbalanced data and transfer learning approaches are discussed.
- **Chapter 3: OPTIMAL TEXT CLASSIFICATION WITH ENSEMBLE METHODS**
Chapter 3 presents a novel model of a nature-based optimization approach with ensemble classifier. Three nature-based optimization techniques are explored with machine learning classifiers. Results are concluded after extensive experiments and comparison with the related literature techniques.

- **Chapter 4: OPTIMAL FEATURE SELECTION FOR IMBALANCED TEXT CLASSIFICATION**

Chapter 4 illustrate a new algorithm to handle high-dimensional imbalanced dataset. Two new algorithms have been introduced with substantial experiments and comparisons with other techniques.

- **Chapter 5: FINE TUNED GRASSHOPPER OPTIMIZATION ALGORITHM FOR TEXT CLASSIFICATION**

Chapter 5 explains the parameter tuning method for the optimization algorithm to perform feature selection effectively. Two classifiers are selected for classification. Results and comparison with other models shows the effectiveness of the proposed model.

- **Chapter 6: OPTIMAL HETEROGENEOUS DOMAIN ADAPTATION FOR TEXT CLASSIFICATION**

This chapter explains the naive transfer learning approach with clustering method. The feature selection improved the transfer learning process. Finally, the result analysis and comparison with other approaches is concluded at the end.

- **Chapter 7: CONCLUSION AND FUTURE SCOPE**

In the last chapter, results of the proposed work is discussed and summarized with the future scope that can be addressed.

CHAPTER 2

LITERATURE REVIEW

Text classification is an inherent necessity of all the applications, due to growing technology and online documents. The use of textual information has been increased from past few years and especially last two years. Hence, researchers are continuously working on creating and improving text classification models from the past. Many problems arise while classifying textual documents like high-dimensionality, class-imbalance. Hence to improve the working of model the problems have been handled with the help of some optimization techniques like feature selection, using data pre-processing techniques for balancing the dataset, parameter tuning of the algorithms and use of transfer learning techniques.

2.1 Feature Selection Techniques

The text classification optimization is performed over the years with many algorithms [47]. Optimization of classification can be performed by reducing undesirable features or feature selection. Feature selection can be performed implicitly as well as explicitly in classification techniques. In [48] *et al.* have performed feature selection implicitly in genetic programming by applying changes in mutation operator. Feature selection has three approaches namely; filter approach [49] [50], wrapper approach [51] [52] [53] and embedded approach [10] [54]. The filter approach mainly depends on the selection criteria of the dataset and not on the learning algorithm [55].

The categories of wrapper approach such as PSO [56] [57], GA, ACO, DE [58][15] mainly inclined towards the design of algorithms that can be used for optimization of a large dataset. Hence, wrapper approach produces accurate results than a filter approach Zhang *et al.* [20] proposed the feature selection technique which results in cost reduction as well. They have used probability-based encoding and Pareto domination relationship. Based on the performance measure, their proposed approach yields the best results in comparison to five multi-objective algorithms. But their proposed model failed to improve the measures for low-dimensional datasets. The sparse solution produced by Pareto front was producing the worst hyper volume values. In [59], the study gives a new idea of hybrid search methodology, the advantages of cooperation of both

genetic approach and filter feature selection method. They have proposed a model with rank-based feature extraction by assigning the ranks in descending order according to their importance. The authors have used Naive Baye's and associative classification for training and testing of Arabic text dataset. Their research shows that the proposed enhanced Genetic algorithm provides better result than using classical Genetic Algorithm. But their results of Precision, Recall and F-measure shows that the proposed model didn't work for all categorization techniques. However, their model works on high-dimensional dataset and not co-related classes.

Researchers in [60], proposed a new model of feature selection with a chaotic crow search algorithm. Their proposed model works with best chaotic map. They have compared their mean fitness value with other meta-heuristic algorithms such as: Chicken Swarm Optimization [61], Gray Wolf Optimizer (GWO) [62], Sine Cosine algorithm (SCA) [63], Particle Swarm Optimization (PSO) [64], Whale Optimization algorithm (WOA) [65], Artificial Bee Colony optimization (ABC) [66], Moth Flame Optimization (MFO) [67] and Flower Pollination Algorithm (FPA) [68]. Their approach performed better than all the algorithms used for comparison. Through algorithm different chaotic maps were adopted. Selection of optimal chaotic was difficult and time-consuming process, as every dataset didn't improved result with same chaotic map.

In [58], researchers used Genetic Algorithms, Differential Evolution [69] and Simulated Annealing [70] for optimization and fine-Tuning Naive Bayesian algorithm [71] [72] for training on 53 datasets. They have also applied their model on 18 text datasets. Their multi-parent differential approach gives better accuracy with fine-tuning Naive Baye's when compared with Bernoulli NB [73] [74], Multinomial NB [75] [76] [77]. They have validated their classification model using Accuracy measures only, which is not sufficient measure for generalization. The computational time was quite high for the tuning of NB classifier. In the study[78], researchers have compared local feature selection techniques with global feature selection technique using transformation method [79]. They have performed their experiment on different applications of flat and hierarchical databases, and final computed results showed that local outperforms the global transform. Jiang *et al.* [80], presented a hybrid model of softmax regression and deep belief network. They performed feature extraction using deep belief network, and the selected features are classified using softmax regression. They have also performed parameter tuning using Broyden-Fletcher-Goldfarb Shanno algorithm. The computed results showed that fine-tuning of the hybrid algorithm performs better than the classical approach of Support Vector Machine and K-Nearest Neighbour.

In [15] Zorarpaci *et al.*, performed dimensionality reduction by using a hybrid combination of Artificial Bee Colony optimization with Differential Evolution. Their hybrid method gives improved Accuracy and run time performance. In [81], the study intro-

duced the idea of using ACO for optimum feature subset. They have used a ranking method to select relevant features. The authors have used NB, K-NN and SVM as classifiers on 15 datasets for the classification of text. Through the experiment results, authors have proved that their proposed method found better results classification performance. But, the proposed model failed with the K-NN classifier for some of the corpus. The above models proposed in [81] [80][78][15] worked only with low dimensional datasets, as performance measures fail to produce good results with high-dimensional datasets. In research [82], authors proposed model to improve the working of Convolutional Network Network (CNN), Recurrent Neural Network (RNN), BERT and transformer with the help of feature projection in orthogonal expansion. The authors in study [83] have introduced with five multiple criteria decision making techniques for feature selection. The technique implemented three classifiers on ten small datasets. The used of ranking methods illustrate the effectiveness of feature selection method. Researchers in [84] introduced feature selection with firefly algorithm for Arabic text classification. The SVM classifier is used for classification on one real dataset. In [85], authors have extracted features using supervised machine learning approach namely; SVM, NB and LR. SVM outperformed the other classifiers. It was analyzed and concluded that every classifier have their own advantages and disadvantages depending on the size of the dataset. The authors in [86] proposed escalated method for text classification by implementing representation of words as Bag-of-words combined with Term Frequency-Inverse Document Frequency (TF-IDF) and 'GloVe', a word embedding technique. The combined technique search for words which have indistinguishable meaningful semantics. The technique is compared with filter approach on four datasets.

2.2 Ensemble Based Techniques

Ensemble methods are used to improve the accuracy as it combines the output of various weak learning classifiers. In [87], the technique uses the ensemble classifiers for sentiment classification. They have used NB, Bagging and SVM with vote algorithm as base classifier for ensemble classifier. Optimization of SVM parameters is performed to improve classification accuracy, and from experimental results, it was proved that multiple classifiers shows good results.

In [88], study shows how the keyword extraction method is combined with an ensemble. All the base classifiers are compared with five ensemble methods using performance measure: Precision, Recall, F-measure and Area Under Curve (AUC). The final conclusion was made with the result shown by declaring two ensemble classifiers Bagging and Random Subspace with RF as base classifier, produces the best results. However, only statistical feature extraction method can be used with their model and results are based on accuracy measures. In [89] author proposed a model using five classifiers

namely: SVM, Rule Based classifiers, induction based classifier, general inquirer based classifier and statistics based classifier. The results shows that the use of an ensemble classifier improves the performance of text classification. Both models proposed in [87] [89] failed to improved performance on high-dimensional datasets. In the [90], the different subset of features namely; word relation based feature subset and parts of speech is extracted. The different base classifiers SVM, Maximum entropy and NB were used with ensemble learners. The results have proved the effectiveness of ensemble classifiers. Although, their results are proved only for accuracy measures.

In [91], the authors have used the feature representation scheme for sentiment classification. Empirical analysis shows that the use of set ensemble classifiers improves the classification performance. In another study [92], the five base learners are applied on three ensemble classifiers namely; NB, Maximum entropy, K-NN and SVM. In [93], the researchers proposed a Baye's model which is an average of different ensemble methods for sentiment classification. However the research presented in [91] [93] [92], ensemble classifier didn't work on large datasets and diverse features. The models presented in [93] [92] also worked on balanced datasets and failed to improve for multi-class datasets. Hence, we proposed a model that works on high dimensional datasets and analyzed results with four performance measures. The applicability on practical problems is proved by testing the proposed model on real-time dataset of an airlines. In recent study [94], researcher proposed the transfer knowledge of trained data using ensemble classifier. Author consider attack on new data as black box and predict the performance against the conventional attack.

2.3 Handling Imbalanced Data Techniques

The focus of this section is based on the related research carried out on class imbalance problem [22]. Class imbalance degrades the accuracy of classification due to more instances of one type of class [95]. To solve the imbalance problem there exist different approaches which can be categorized mainly into :-1)Data level approaches, 2)Algorithm level approaches, 3)Ensemble level approaches and 4)Cost sensitive learning [96][97]. The data level approaches are based on all the pre-processing techniques required for balancing the class distribution [98]. In data level approaches, there exist sampling techniques which performs with versatility, and also not based on the classifier chosen [99]. The sampling techniques mainly consist of two types namely: under-sampling and over-sampling [100]. In under-sampling strategy, the desired number of majority classes are removed randomly which causes the loss of useful information [101]. Some of the most under-sampling techniques used in the literature for balancing are: Wilson's Edited Nearest Neighbor rule[ENN] [102], One Sided Selection [103], Condensed Nearest Neighbor Rule [104] and Tomek Links [105] [106].

The algorithm level methods are based on cost sensitivity, and hence are not responsible to disturb the distribution of data into different classes. Higher the cost, higher the misclassification among minority classes. The cost sensitive method mainly consist of three categories: designing of suitable cost function to a given classifier, algorithm based on meta cost framework and techniques on translation theorem. Although using algorithm level methods performs better than data level method for balancing the imbalanced dataset, but constructing the predefined felicitous cost function for an imbalanced dataset is difficult [97] [107]. Ensemble methods are used to improve the accuracy as it combines the output of various weak learning classifiers. In [87], the technique uses the Ensemble classifiers for sentiment classification. They have used three machine learning algorithms Naive Baye's, Bagging and SVM with vote algorithm. Optimization of SVM parameters is performed to improve classification accuracy, and from the experimental results, it was proved that multiple classifiers show good results. Researchers in [88] shows how keyword extraction method is combined with an ensemble classifier. All the base classifiers are compared with five ensemble methods using performance measure: Precision, Recall, F-measure and AUC. A study in [108] proposed a novel approach of developing balancing methods for samples, and produces balance sets which are different. They use the obtained balanced data samples to construct high-performance classifier by collaborating multiple classifiers. In this technique there lies a SMOTE algorithm combined with Adaboost for balancing, known as SMOTEBoost [109]. The main disadvantage of using Ensemble techniques for solving imbalancing problem is that it is time consuming especially with high dimensions of data. The most emerging and efficient method for balancing the dataset are Data level methods. In the Data level method, sampling of datasets works on the distribution of data. SMOTE is the most common technique used in the literature [110]. Although many drawback was enlightened with the solutions. The main drawback of SMOTE was over-generalization as the generation of artificial new instances does not consider the nearest neighbor of the majority class, class disjuncts and class overlapping [111]. Adaptive Synthetic (ADASYN) oversampling technique was proposed by [112], to overcome the SMOTE over generalization problem by increasing the values of newly added samples.

In [113], study proposed a method to overcome the problem of those instances which lies near the border of minority samples. So, they performed interpolation(oversampling in particular area) of minority samples. The work on borderline instances was proposed by Chen *et al.* [114], where they performed collaboration of clustering and data before preprocessing techniques to determine the samples which always lies in the same cluster(class). Such type of samples are known as center samples. They classify those samples which change class cluster as border samples. So, authors apply SMOTE for minor-

ity classes in the border samples. Assigning the number of positive(minority) instances in K-nearest neighbor (K-NN) is safe-level value and safe-level SMOTE generates the samples which are closer to this value [115]. For multi-class imbalanced problem Fernandez *et al.* [116] proposed to apply decomposition algorithms, One versus All(OAA) [117] and One versus One(OAO) [118]. The study indicated the improved performance of classification. The one disadvantage of SMOTE algorithm is that it performs resampling of minority class by reconsidering the same sampling rate for all existing instances. In [119] Jiang and other researchers proposed a novel technique by introducing the use of GA with SMOTE technique known as GASMOTE. It overcomes the disadvantage of resampling of minority class by reconsidering different rates for sampling of different instances taken, and found the optimal values of sampling rate to combine the results. They compared the results in terms of F-measure and G-mean of ten datasets with classical SMOTE and Borderline SMOTE. GASMOTE proved to produce the best results among the other algorithms. Their proposed model calculated the results on low-dimensional datasets. The study [120] focused on data-oriented technique and proposed novel approach based on Mahalanobis distance for oversampling. This technique uses class mean and covariance, and generates samples for only those minority classes which are at the same distance from the considered class. They also proved that by using this technique risk of overlapping of classes decreases. This technique outperformed when compared with other over-sampling techniques(SMOTE, random oversampling, Adaptive Synthetic and Borderline SMOTE) using assessment metrics M-measure, F-measure, Precision, Recall and G-measure. But, their proposed model works with low dimensional numeric datasets. The multi-class imbalanced problem was discussed in [121] where the researchers have proposed modified K-nearest neighbor SMOTE algorithm (SMOM). Modification in the technique was to assign weights to all the nearest neighboring minority instances, and instead of randomly selecting the minority instance, instances must be selected according to weight. Thus, the problem of over-generalization was removed by selecting instances according to the weights. Experiment was carried out on 27 real-world datasets with performance measure as Precision, Recall and F-measure. Their model works with continuous dataset and not tested on nominal and ordinal attributes. Other method to solve imbalance problem is the use of Ensemble techniques [122]. The study combines the ensemble technique(dynamic selection of classifier) with preprocessing techniques used are SMOTE, Random Balance [123] and Ranked Minority Oversampling (RAMO) [124]. The preprocessing techniques used by researchers are over-sampling technique. They performed experiment on 26 multi-class datasets and compare their results with static Ensembles. The conclusion of their study was that RAMO is the best preprocessing technique when used with dynamic selection. Another novel approach for multi-class imbalance prob-

lem was solved by proposing new algorithm in [125] known as Diversified Error correcting output codes (DECOC). The main technique used in this algorithm was combining Error correcting output codes (ECOC) with Ensemble technique. The ECOC is used to balance the imbalance data. They have used 19 datasets and calculated the results with performance measures: G-mean, Area under curve, F-measure and Accuracy. They have compared results with 17 state-of-the-art algorithms and proved that DECOC gave best performance in all measures. Their model computation takes lot of time to get the results. A novel approach using ensemble technique with the use of DE for optimizing weights was proposed in [126] to solve the imbalanced problem. When the results of 12 datasets were compared on the basis of G-mean with vote-based Ensemble technique and non-ensemble technique, the proposed model proved to be the best. The study in [127] proposed a model which uses different distance measures with SMOTE algorithm. They proposed a new distance metric, Minkowski distance [128] to determine the neighbor of minority class. The various distance measure used are: Chebyshev, Euclidean and Manhattan distances. These distances are ranked with measures namely: Mutual Information, Eigenvector, Fisher Score, Correlation Score and Centrality. The main problem model faced was of over generalization, as researchers ignored the majority class while constructing the synthetic samples.

2.4 Parameter Tuning Techniques

Text classification can be binary classification [1] used in various task namely: spam filtering, opinion mining, information retrieval and many more. Multi-class classification is used where classification of more than two classes are required [129]. Tuning the parameters of the algorithms used for classification can improve the classification performance [130]. The main aim of hyper-parameter optimization is to control the parameters to improve the text classification performance. There are so many ways to tune parameters. First approach followed many years back was trying the different sets of values using hit and trial method [131] to obtain the acceptable parameters values. The second approach followed is grid search [132]. Grid search is a slow optimization algorithm which generates results after the several parameter combination. The third approach is the use of meta-heuristic algorithms, which are based on biological phenomena [133], [134]. Some used meta-heuristic algorithms in state-of-the-art are GA [135], GWO [136], Multi Verse Optimizer (MVO) [137], [33], ACO [138]. In [27], authors applied grid search to K-NN classifier for tuning the parameters with BM25 similarity. They proved that using BM25 similarity measure approach is a fast tuning method and proved it by comparing with other conventional approaches. The study in [139], proposed a technique of performing parameter tuning and feature selection simultaneously using GOA on SVM model. The authors compared their proposed

model with other techniques and proved the performance in terms of classification accuracy. In [136] Ibrahim *et al.*, proposed approach of using grasshopper optimizer for feature selection and parameter tuning for SVM classifier. The authors have compared the model accuracy with other meta-heuristics feature selection technique and proved the model is yielding better efficiency. The research in [130], shows the generalized model for optimizing the parameters of SVM classifier. The authors have tested their model on 15 medical diagnosis dataset. They have made the conclusion that estimation of distribution algorithms are best for the parameter tuning of SVM classifier. In [135], the authors have proposed GA-SVM model. They have used GA for feature weighting and parameters tuning for SVM. The authors proved the model efficiency by comparing their proposed technique with other feature weighting techniques. In research [2], authors proposed novel hybrid technique of nature-inspired and ensemble classifiers. They have used BBO algorithm for feature selection technique and ensemble classifier for classifying the text. The authors have compared the classification performance with other state-of-the-art techniques and found that proposed technique outperformed among all the techniques.

2.5 Transfer Learning in Text Classification

The crucial point in unsupervised transfer learning is that it arises to find the similar feature space for the source as well as target domain. In some research like [140] [141] [142] proposed model that converts source and target into common feature space in supervised learning. Many transfer learning techniques have been already proposed using features, relations between domain parameters and instances [40]. There are mainly two classes on which transfer learning can be classified, namely: homogeneous transfer learning and heterogeneous transfer learning [143]. Homogeneous transfer learning exists when the features of source and target dataset are similar. Heterogeneous transfer learning deals with different feature space in source and target domains. However, one problem arises if there is less or no similarity between the source domain and target domain, known as negative transfer. So, there are further categorization of transfer learning categories into instance-based approach, feature-based approach and parameter-based approach [144].

Many studies have worked on different approaches of transfer learning. In [145] TrAdaBoost is used for the adjustment of weights for increasing the similarity to the instances in target domain. Pardoe and Stone [146] proposed extended technique of TrAdaBoost called ExBoost.R2 and TrAdaBoost.R2 which deals with instance-based regression. The study [147], proposed algorithm known as Bi-weighting domain adaptation (BIN) which performs text categorizations for cross-language. It adjusts the feature spaces of both domains into one coordinate space.

In feature-based transfer learning, the main aim is to map the feature between the source and the target domain. In [148], researchers worked on subspace learning and proposed a new criterion for distance measuring between the distribution problem and Gradient-based approach is used for optimization. Pan *et al.* [149], proposed a method to learn the latent space for the source and target domain. They have also solve the optimization problem using Eigen decomposition.

In the metric transfer learning, Zhang *et al.* [44] proposed a new model which exploits the correlations between source and target domain. In [150], the metric transfer learning framework is proposed in which weight of instances are adjusted to normalize the distribution of data between different domains. They have used Mahalanobis distances for weight adjustment. Many researchers also worked for Cross-Domain adaptation like in [151], the authors proposed softly associative transfer learning algorithm (sa-TL) by combining two non-negative matrix obtained from features. Their proposed method performs well with binary classes. In study [152], character level convolutional network is used for transfer learning, with fine tuning of layers of network known as Temporal Deep Convolutional Network (TDCN). They performed transfer learning on four datasets, and proved that their model outperformed the other techniques. Their model performed well for a semantically similar domain and not on cross-domain. In study, Li *et al.* [153] proposed a generalized model for domain adaptation. They re-weighted the pivot features and focus on decreasing the outliers weights and name the technique as Transfer Independently Together (TIT). The main goal is to convert a feature into geometric graph vertices. Their model worked for text categorization, image classification, and text-to-speech recognition. Their technique outperformed with the other proposed models, but they only analyzed SVM classifiers for their experiments. The researchers in [154] proposed a method which optimizes features and distribute divergence using single objective function. They used a method of Progressive Alignment (PA) based on the learning of a new feature space that can be transferred using dictionary sharing. The ability of model was proved experimentally by performing best on image classification, text-to-image recognition and text categorization datasets, when compared to other state-of-the-art techniques. The other models proposed by researchers in [155] [156] accord novel approach of transfer learning in images. Their proposed models worked on the evaluation of distance loss and adaptation of features respectively. The efficacy of both the models was proved on five benchmark datasets of images. In [157], the researchers worked on similarity features of multiple instances using exemplar as features. They have trained their model using various metrics between two instances and use of multiple kernels are proposed for the visual object detection. The researchers in [46] introduce a novel technique using PSO as Feature Selection for Unsupervised Transfer Learning on image dataset (FSUTL-PSO), referring as FP technique in further

sections. They have used a fitness function to select an appropriate feature and used only K-NN classifier to calculate the fitness value. The proposed method is based on the manual parameter settings and is a very time-consuming process.

To summarize the above all techniques have been used for improving text classification performance. All the methods require extensive experiments. To measure the performance objective functions Precision, Recall, Accuracy, F-measure and G-means has been used widely.

CHAPTER 3

OPTIMAL TEXT CLASSIFICATION WITH ENSEMBLE METHODS

Large feature space decreases the efficiency of the algorithm [13], hence reduces the efficacy of text classification. The presence of noise and un-useful features for dataset can decrease the efficiency to classify text into various classes. Feature selection mainly selects the desired and important features which provides necessary related information [14], hence increases the efficiency of text classification.

3.1 Introduction

As there is a rise in research areas related to data mining, the advancement in Information Communication and Technology (ICT), gives opportunities to all users to access the information quickly and at a faster rate. Due to increase in the demand of text documents, there is a simultaneously increase in the number of text documents, as the availability and accessibility of the digital information are saved and organized in the forge of text [80]. The application area of text classification is widely falling in the era of text analysis, and according to our survey , many machine learning supervised algorithms are applied for text classification such as Naive Baye's [81], Decision Tree [13], K- nearest neighbor [81], Random Forest [13], Support Vector Machine [87] [158] [81], and ensemble classifiers [88]. There are many related issues that arises for text classification majorly due to high-dimensional feature set [17]. Large feature space decreases the efficiency of the algorithm [13], hence reduces the efficacy of text classification. The presence of noise and un-useful features for dataset can decrease the efficiency to classify text into various classes. Feature selection mainly selects the desired and important features which provides necessary related information [14], hence increases the efficiency of text classification. Feature selection techniques are broadly classified into filter approach and wrapper approach [159]. To extract only desirable features, the filter approach and the wrapper approach have many feature selection techniques. Some of the filter techniques used are Information Gain (IG), Poisson distribution, improved Gini Index (GINI), Odd's Ratio(OR), Chi-Square, Binomial Hypothesis testing,

and many other techniques [90]. The proposed nature-based optimization, used for feature selection, is categorized into the wrapper approach. Many evolutionary optimization algorithms such as: Genetic Algorithm (GA) [3], Firefly algorithm (FA) [19], Particle Swarm Optimization (PSO) [20], Ant Colony Optimization (ACO) [18], and Biogeography Based Optimization (BBO) [21] has been applied in literature for feature selection. Hybrid approaches are also used for feature selection [59]. The hybrid approach produces the best performance among all approaches, as it combines the wrapper approach and filter approach. In [21], Biogeography Based optimization solved the problem of best selection of sensor for aircraft engine. The BBO outperformed when compared to other nature optimization techniques such as GA, PSO and Differential Evolution. In many classification problems other than text classification, BBO is used for optimization and have proved best among others [160] [161] [162] [163]. Hence, in our proposed method, we have chosen the BBO algorithm for feature selection.

To the best of our knowledge, the BBO is not applied for the text classification. In this work, we proposed a hybrid for feature selection with ensemble classifier and the best results of BBO in other area inspired us to select BBO to construct our proposed approach. The main idea of this research is to propose a novel technique, the hybrid of nature-based optimization technique (BBO) for feature selection with ensemble classifier. The extracted features are stored in an array and passed for training to six Machine learning algorithms namely: Naive Baye's (NB), K-Nearest Neighbor (K-NN), Support Vector Machine (SVM), Random Forest (RF), Decision Tree (DT) and ensemble classifier.

3.2 Theoretical Foundations

Our approach of feature selection with BBO algorithm in combination with the ensemble classifier is compared with similar state-of-the-art-algorithms. We have compared the feature selection technique by implementing GA, PSO and BBO. The selected classifiers for training and testing are NB, K-NN, SVM, RF, DT and ensemble(bagging). To validate the technique we have chosen performance measures namely; Precision, Recall, Accuracy and F-measure. All these performance values are calculated with every model using ten-fold cross-validation. In this work, we have created 35 models of every dataset. Experimental results are validated using statistical test, i.e. Friedman test. All the algorithms implemented are briefly described in the following subsection and the detailed working of proposed technique is described in the next section.

3.2.1 Feature Selection Methods

There are various existing feature selection methods [164] which extract only and important features from the data, as discussed earlier. These techniques helps to improve

the accuracy of text classification. In this study, we have used the wrapper approach. Wrapper approach mainly consists of nature- inspired algorithms that can be used for feature selection. The comparison has been performed on all the selected datasets of text. The computation has been done on a large number of features by performing feature selection methods.

3.2.1.1 Genetic Algorithm

Genetic Algorithm(GA) [165] [166] is used as feature selection technique, where all features are considered as input. By using feature selection approach, the main goal is to improvise the general performance using optimization. By following the procedure of Genetic Algorithm for feature selection, the main work is to produce the best feature subset based on the fitness function. GA starts with random initialization of population. At every generation, new individuals are selected depending on the value of the fitness function. In this study, we have taken rank based fitness selection. After fitness assignment, the selection operator chooses the individual that recombines with the other individuals to form next generation. Roulette wheel selection is performed in which individuals are selected randomly with the help of the spinning movement of a wheel. Finally, the two parents were randomly taken for the crossover operation, which is responsible for reproducing the new fittest subsets known as children of parents. But the working of mutation operator depends on a subset (single) which randomly adjust some preferred values of features for the survival of the fittest. The flowchart of feature selection approach is shown in Figure 3.1.

3.2.1.2 Particle Swarm Optimization

The other nature-inspired algorithm is Particle Swarm Optimization [167]. It chooses random particle and the velocity associated with it [57]. Position at generation t is denoted as $X_i(t)=(x_{i,1}(t),x_{i,2}(t)\dots x_{i,d}(t))$, where $x_{i,d}(t)$ is the position of i^{th} particle in correspondence to the d^{th} dimension and velocity as $V_i(t)=(V_{i,1}(t),V_{i,2}(t)\dots V_{i,d}(t))$, where $V_{i,d}$ is the velocity of i^{th} particle with respect to the d^{th} dimension . Then, at generation $t+1$ the updated position and updated velocity is described by the Equation 3.2 and Equation 3.1 respectively.

$$V_{id}(t+1) = w \times V_{id}(t) + c_1 \times rand_1 \times (P_{best}(t) - X_{id}(t)) + c_2 rand_2 \times (G_{best} - X_{id}(t)), \quad (3.1)$$

$$X_i(t+1) = x_i(t) + v_i(t+1) \quad (3.2)$$

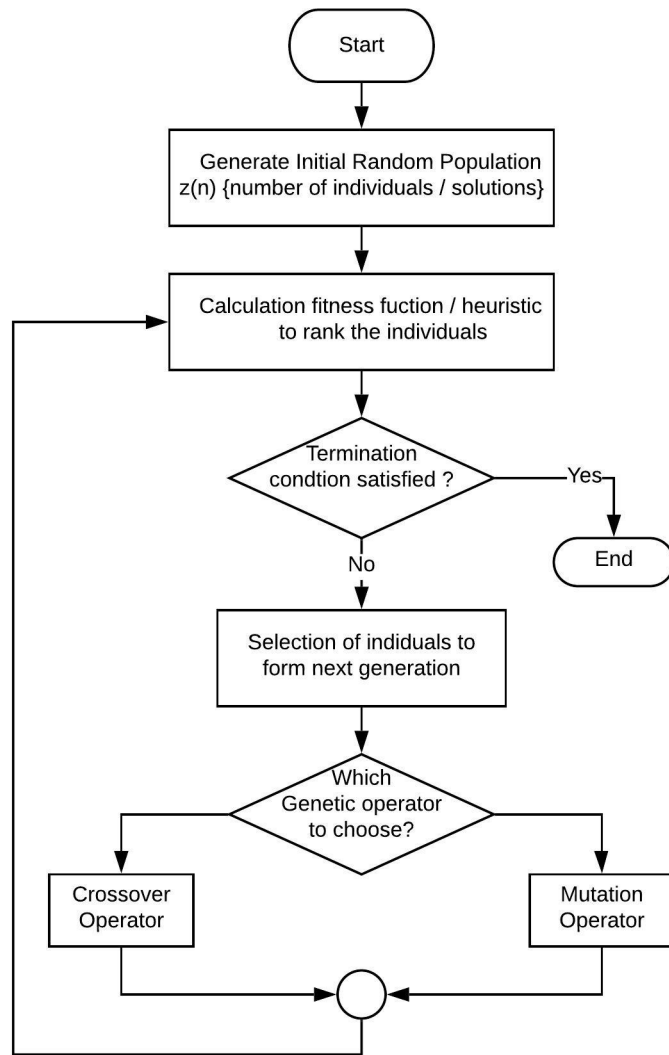


Figure 3.1: Feature selection with Genetic Algorithm

where $P_{best}(t)$ represents the best position of individual i in d dimensional space until generation t and G_{best} , which represents the best position of the group in d dimension until generation t . The non-negative velocity coefficients, c_1 is the cognitive parameter and c_2 is the social parameter, w is an inertia weight, $rand_1$ and $rand_2$ are two random values range between 0 and 1. The flowchart of PSO as feature selection is described in Figure 3.2.

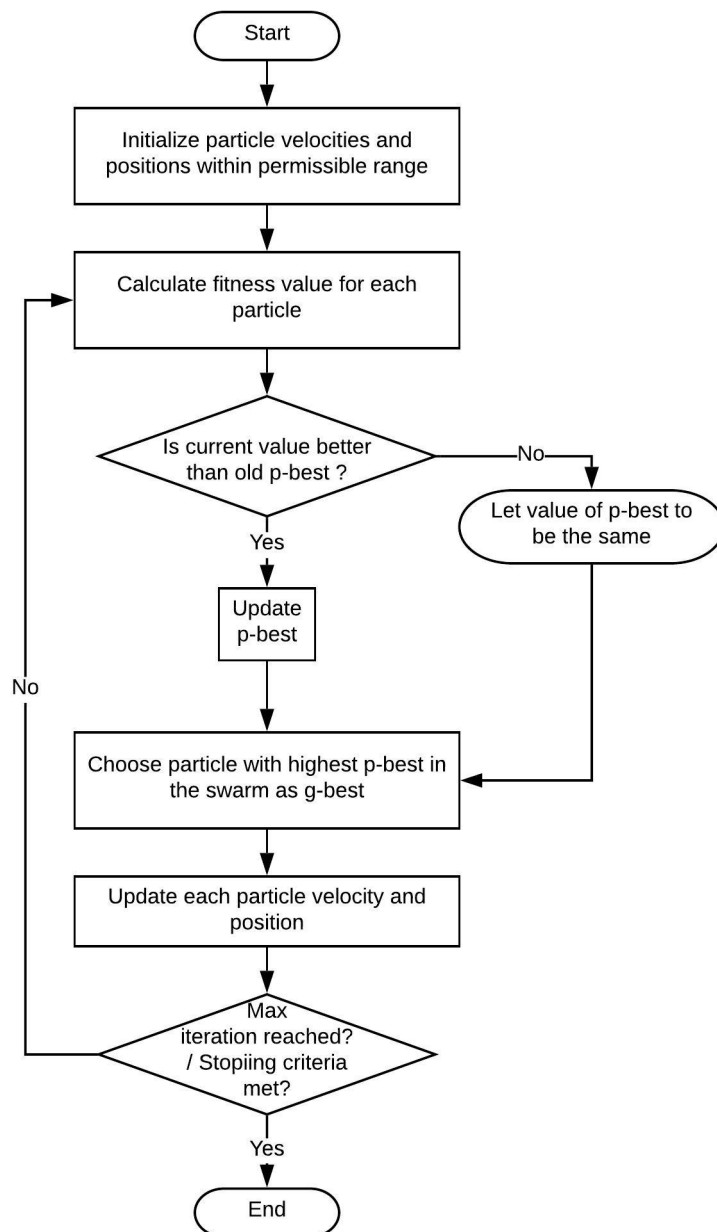


Figure 3.2: Feature selection with PSO

3.2.1.3 Biogeography Based Optimization

Biogeography Based Optimization is a nature-inspired evolutionary algorithm which mainly finds a solution to the problem of global optimization. The convergence of algorithm depends on immigration(of species) and emigration (of species) that reside on different islands according to the various factors considered by the species for a more friendly life. It gives the candidate a solution which can be referred as "habitat" and known as Habitat Suitability Index (HSI). The various factors like temperature, rainfall etc. on which migration on the island by species depends is Suitability Index

Variables (SIV). HSI is k-dimension vector from which initial population is randomly generated. The improved suitability increases the count of species that emigrates raising the HSI and decreasing the immigration count. The habitats with high HSI refers as good solutions and they are densely populated. The habitat with low HSI refers as poor solutions and they are sparsely populated. There is some information sharing from high HSI to low HSI to control the emigration rate. In BBO, mutation and migration [21] [168] of species is calculated using SIV vector, H(A). By considering the values of SIV, an optimal solution is obtained. Information sharing among the island is possible with migration. Islands with a high value of HSI have low immigration rate due to less number of resources and high population. The immigration rate (λ), which signifies the living of species on other island and emigration rate μ , signifies the leaving of species from the native island are consider as objective functions. When there exist zero species on any island then only maximum immigration, I is possible. When habitat supports the maximum number of species then rate of immigration starts falling to zero, due to lack of resources and high population, species start moving to other islands, and also emigration reaches its maximum possible value E. Alternatively, the rate of emigration is zero when there is no existence of any single species on the island. Variation in immigration is due to factors like climate change, resource scarcity, etc.

In BBO, λ_k is the probability of replacing a given independent candidate chosen from K-th candidate solution. After performing random selection with roulette wheel and independent variable candidate (which is chosen) has to be replaced, the probability of emigrating candidate solution (selected) is proportional to the emigration probability, which is performed using roulette wheel selection is given in Equation 3.3.

$$Prob(x_k), \text{ selected for emigration} = \frac{\mu_k}{\sum_{j=1}^A \mu_j} \quad (3.3)$$

where, $k=1\dots n$ are the number of species,

and $j=1,2\dots A$, A is presented as the number of candidate solutions in the population.

In BBO, migration is calculated using probability among species with consideration of suitability factors. Due to some low suitability factors on an island, the species tend to migrate or move to any other or new island. The migration process modifies the habitat of island. Migration uses probability to change a habitat H_k . Then based on immigration rate λ , the probability H_k is modified proportionally, and changes in λ is done with respect to (probability) $H_t \propto \mu$, emigration rate. Migration of species is mapped into habitat vector H(A), immigration rate and emigration rate is represented in Equation 3.4.

$$\lambda_p = I(1 - \frac{p}{k})\mu_p = E(\frac{p}{k}) \quad (3.4)$$

where, I is considered as the maximum possible immigration rate, E is considered as the maximum possible emigration rate, p is the number of species of p-th individual and k is the maximum number of species. Pseudocode for the migration is described in Figure 3.3 and migration equation is represented in Equation 3.5. Mutation is a proba-

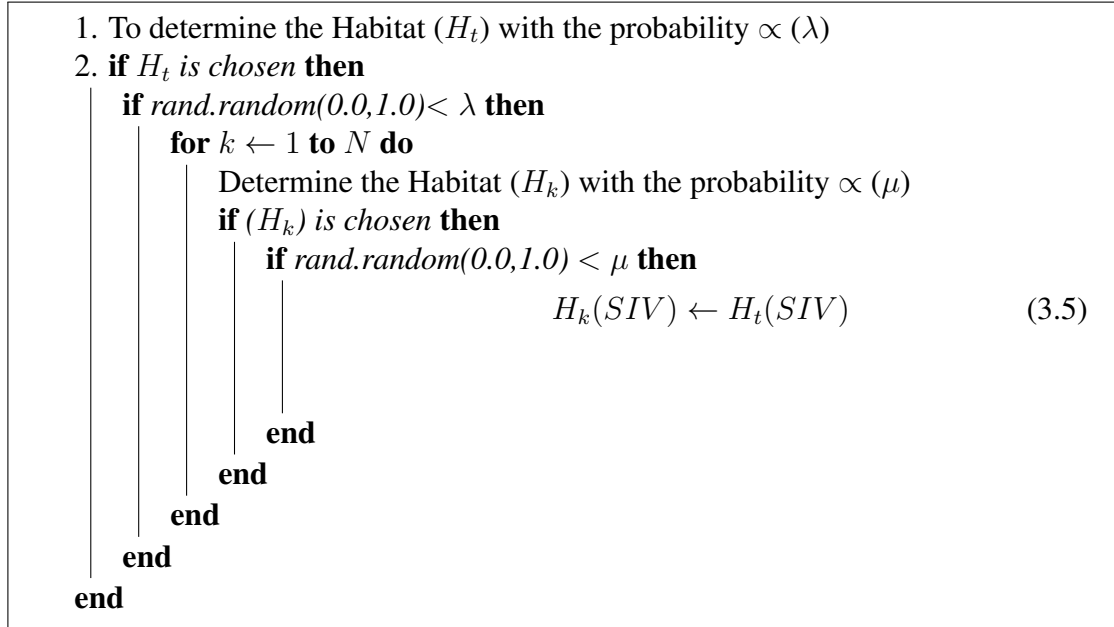


Figure 3.3: Pseudo code for BBO migration

bility operation that is responsible for variations in habitat SIV's. The mutation rate is calculated in Equation 3.6.

$$\mu(s) = \mu_{max}(1 - P_i)/P_{max} \quad (3.6)$$

where μ is a mutation operator, μ_{max} is user-defined parameter, P_i is the probability of ith species and P_{max} is probability of the maximum number of species. Mutation is likely to increase the melange among the population. Pseudocode for mutation is described in Figure 3.4.

3.2.1.4 BBO as feature selection

The working of BBO algorithm for feature selection is explained in Figure 3.5. Flowchart of the BBO as feature selection is shown in Figure 3.6. Parameters considered for BBO algorithm are described in Table 3.1.

```

for  $y=1$  to  $m$  do
    The parameters  $\lambda$  and  $\mu$  are used to calculate probability  $P_i$ 
    Select Habitat  $H_t(y)$ (SIV) with the probability  $\propto P_i$ 
    if  $H_t(y) == selected$  then
        if  $rand.random(0.0,1.0) < \mu$  then
            Replace  $H_t$ (SIV) with randomly produced SIV
        end
    end
end

```

Figure 3.4: Pseudocode for BBO mutation

Table 3.1: Parameters for BBO feature selection

Parameters	Symbols
Number of habitats, habitat[]	T
The feature set	SIV[]
Maximum number of iterations	K
Immigration	λ []
Emigration	μ []
Maximum immigration	$\lambda = 500$
Maximum emigration	$\mu = 600$
Array to store crucial/redundant SIV factor	gb[]

3.2.2 Classification Algorithms

To evaluate the classification performance, we have chosen supervised machine learning classifiers. The six best performing machine classifiers used for text classification in state-of-the-art are considered. The six classifiers we have chosen are Naive Baye's [169], K-Nearest neighbor [167], Support Vector Machine [170] [87], Random Forest [171], Decision Tree [13] and ensemble classifier [87].

```

1. Initialize BBO parameters
2. Generate initial population
2.1 Random population in each habitat ( habitat [ ] ):  $H_1, H_2, \dots, H_T$ 
2.2 Each habitat index is associated with SIV values(feature)
2.3 Initial immigration rate of each habitat =  $\lambda [ ]$ 
2.4 Initial emigration rate of each habitat =  $\mu [ ]$ 
2.5 Each index of Habitat is associated to a feasible answer for the given problem.
3. Calculate the fitness of each habitat(solution) of the population.
3.1 It is based on the classification rate of the evolved subset of features.
3.2 Calculate HSI value of each habitat of the population: hab_index [ ]
3.3 Rank habitats based on their HSI value.
Set iteration variable K: depicts the maximum number of iterations and k is the variable which is gradually increased.
while  $k < T$  do
    Calculate emigration rate ( $\mu_j$ ) and immigration rate ( $\lambda_i$ ) for each habitat(solution) of the population. Here,  $j = 1, 2, \dots, T$  and  $i = 1, 2, \dots, T$ 
     $H_j$  is selected where selection criterion is based on emigration rate ( $\mu_j$ ) for the emigration
     $H_i$  is selected where selection criterion is based on immigration rate ( $\lambda_i$ ) for the immigration
    To Perform migration operation
    To Perform mutation operation on  $H_i$ 
    Produce new population by replacing previous (old)  $H_i$  from previous population with new  $H_i$ 
    Re-calculate the values for habitats, compute their corresponding HSI values
    Increment k(the iteration variable)
end
4. Get fittest habitats based on the threshold value.
5. Ranking is performed second time with respect to population
// The SIV value that causes the change in hab_index [ ], is a crucial feature (and the redundant features are removed)
for  $k < N$  do
    if  $SIV[k] == crucial\ value$  then
        gb[k]=1
    else gb[k]=0
    end
end
6. Based on gb[]=1 the crucial feature are kept and the redundant features are removed
7. Dataset with reduced features is stored in SIV[] (Each index of hab_index [] corresponds to the list of features stored in SIV[])

```

Figure 3.5: BBO algorithm for feature Selection

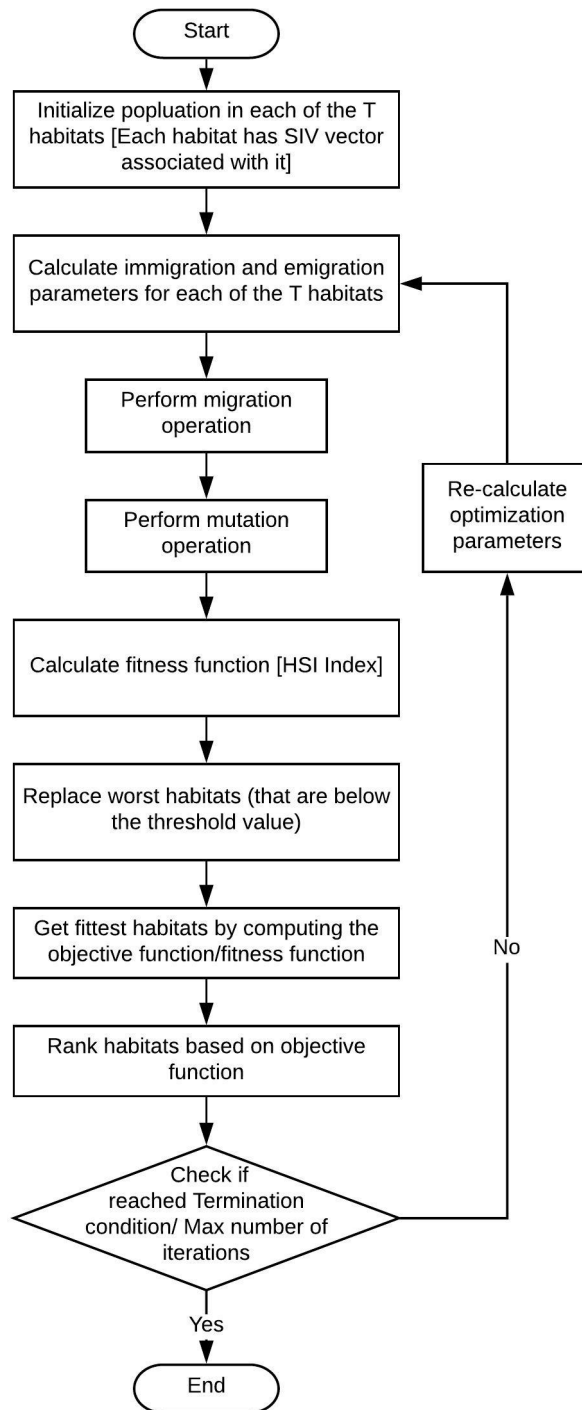


Figure 3.6: Feature selection with BBO

3.2.3 Performance Measure

To evaluate the performance of our hybrid classification model, we have considered four measures to verify the model behavior. The four performance measures chosen are Precision, Recall, Accuracy and F-measure. In many studies [58] [159] [14], only

Accuracy is the main performance measure to evaluate the model but choosing Precision, Recall and F-measure for multi-class classification verifies the model Accuracy. We have computed all performance measures with the help of confusion matrix consist of following instances:

1. If an instance is predicted as positive and also lies in the positive class in its correct actual state it is known as True Positive(TP).
2. If an instance is predicted as negative but lies in the positive class in its correct actual state it is known as False Negative (FN).
3. If an instance is predicted as positive but belongs in the negative class in its correct actual state it is known as False Positive (FP).
4. If an instance is predicted as negative and also belongs in the negative class in its correct actual state it is known as True Negative (TN).

- The performance measures are briefly described in the following subsections.

3.2.3.1 Accuracy

It is calculated by taking the ratio of instances that are correctly predicted and the entire number of instances (both correctly predicted and incorrectly predicted) [58][14][13]. Accuracy formula is represented in Equation 3.7.

$$Accuracy : \frac{TrueNegative + TruePositive}{TrueNegative + TruePositive + FalseNegative + FalsePositive} \quad (3.7)$$

3.2.3.2 Precision

Precision [81] is calculated by taking the ratio of true positive instances and the entire instances which are predicated as positive. The formula for Precision is shown in Figure 3.8.

$$Precision : \frac{TruePositive}{TruePositive + FalsePositive} \quad (3.8)$$

3.2.3.3 Recall

It is calculated by taking the ratio of true positive instances and the entire instances which are positive in their correct actual state [52]. Recall is formalized in Equation 3.9.

$$Recall : \frac{TruePositive}{FalseNegative + TruePositive} \quad (3.9)$$

3.2.3.4 F-measure

F-measure is given by the harmonic mean of Precision and Recall [13] [17]. The formula for F-measure is represented in Equation 3.10.

$$F - measure : \frac{2 \times Precision \times Recall}{Recall + Precision} \quad (3.10)$$

3.2.4 Dataset description

We have considered ten text classification dataset from UCI repository [58] and one real-time dataset of airlines from MOA repository [172]. Description of all dataset is given in Table 3.2.

Table 3.2: Description of multi-class dataset

Dataset	instances	Classes	Features
tr11	414	9	6430
tr12	313	8	5805
tr21	336	6	7903
tr23	204	6	5833
tr31	927	7	10129
tr41	878	10	7455
tr45	690	10	8262
oh0	1003	10	3183
oh10	1050	10	3239
oh15	914	10	3101

3.2.5 Statistical Test

The use of a statistical test evaluates the techniques and validates the model performance. We have chosen a Friedman test for the evaluation of performance measure and checked whether there is any significant difference when compared to other techniques. The Friedman test [11] is a non-parametric test and it is preferred over the parametric test as for parametric test we have to consider some data assumptions. We have computed results with degree of freedom as 3, significant value ($\alpha=0.05$) and the obtained p-value is 0.001. Higher rank indicate the significance of that particular technique. With the help of rank, we can find out the significant difference between the techniques. The following hypothesis is considered:-

Null Hypothesis:- There exists no significant difference between the algorithms GA, PSO, BBO and without optimization technique.

Alternate hypothesis:- There is a significant difference between the algorithms GA, PSO, BBO and without optimizer techniques.

3.3 Methodology

For extensive experiment ten text classification datasets are collected from the UCI repository and one real-time dataset from MOA. For a given dataset with features $s_1, s_2, s_3, \dots, s_n$, we used a feature subset selection technique. For feature selection, we have used GA, PSO and BBO and compared the performance measures. To completely validate the results we have used 10-fold-cross validation. During implementation of model, firstly we calculated all performance measures without using any feature selection technique on all datasets. Then we performed feature selection on all datasets with using GA, PSO and BBO to check the improved performance. By comparing the performance values of our proposed algorithm using BBO for the feature selection with GA and PSO, we concluded that BBO produces significantly better results. We selected the essential features by applying feature selection techniques individually and then pass the new dataset with selected features to classifiers. In total, 20 models were applied to all datasets to find the best model for the optimal solution.

3.3.1 Proposed Hybrid Model BBO-Bagging for Feature Selection and Classification

Ensemble techniques give promising research models in many areas especially in machine learning and pattern recognition [88]. The major goal of ensemble classifier is to combine the decision of predictions of several weak classifiers (base learner). Combination of classifiers enhances the robustness and Accuracy of a model than an individual classifier. There are mainly two categories of ensemble classifiers namely: averaging methods and boosting methods. In averaging methods, several predictive models are made independently and final results are concluded after averaging the results. Bagging and forest of randomized trees are examples of averaging methods. In boosting method, base learners are implemented sequentially with a goal to combine several weak models to produce strong ensemble classifier. We have selected Bagging as an ensemble classifier for our model as Bagging produces better results with strong and complex models compared to boosting technique. Bagging is also known as bootstrap aggregating technique [88][87] which combines the individual classifiers trained on different training sets to build a highly improved predictive model. When a weak learning classifier is under the training phase, the size of every new sample is equal to the size of original training set. Simple randomized sampling with replacement is deployed for producing new training sets. The aggregation of considered individual classifiers is performed to produce the predictive results with the technique of majority voting.

In the proposed approach, feature selection is performed using BBO as explained earlier in section 3.2.1.3 and final subset of features are stored in array SIV. Each index of `hab_index` is associated to the list of features stored in SIV. Initialization of parameters is performed for every base learner classifier. Initially, D is taken as an empty set of classifiers and C is the number of classifiers to be train. Bootstrap samples $S_1, S_2 \dots S_C$, which we have to train, are selected from `Habitat[]`. A classification model, D_k is build utilizing bootstrap samples by adding the current classifier to D. Now, aggregation of results of individual classifiers is performed with majority voting. Class with the highest number of votes is assigned to the test instance. Proposed hybrid approach of BBO and Bagging has improved the performance values further. Ensemble classifier was also combined and compared with GA and PSO. Separate 15 new models were designed for ensemble technique. Flow chart of the proposed methodology is shown in Figure 3.7. Algorithm for the Proposed model is elaborated in Figure 3.8.

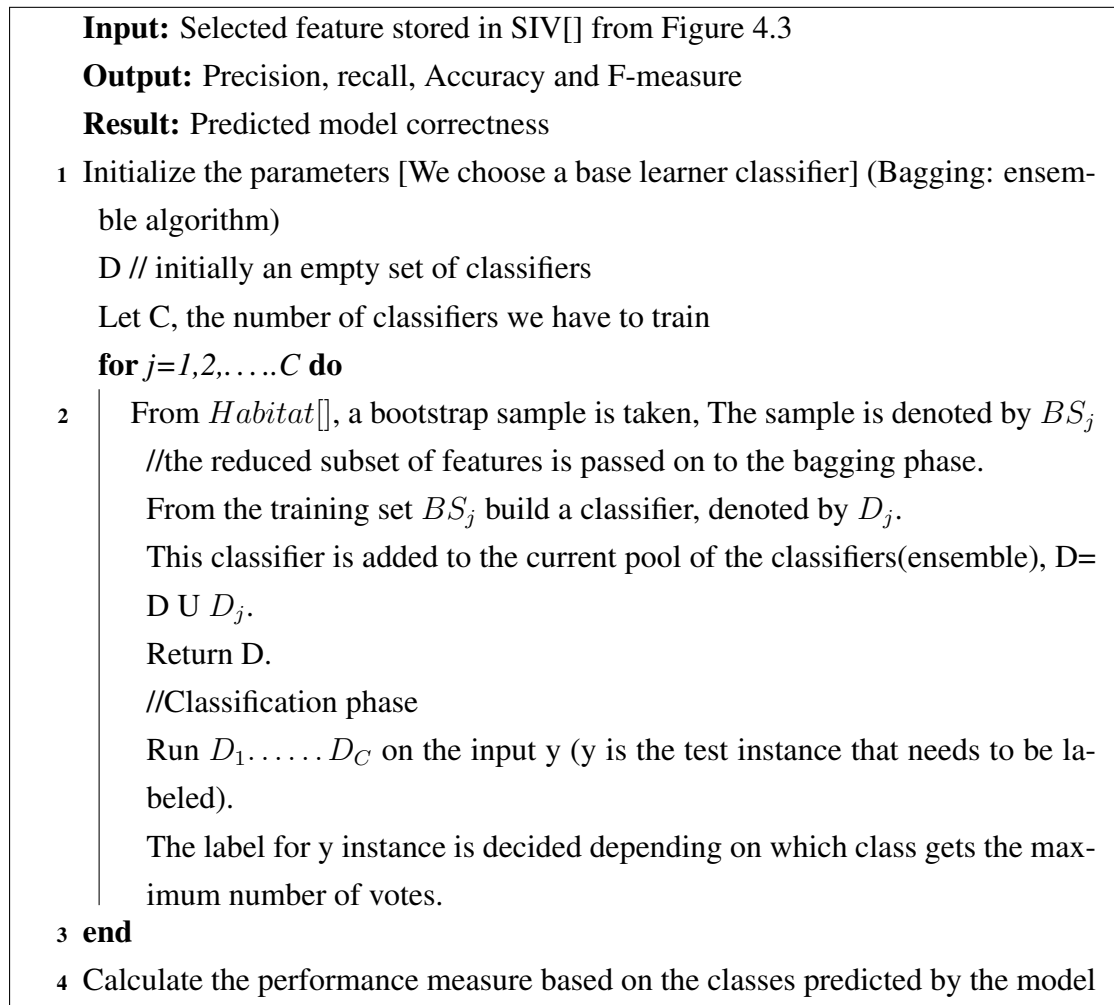


Figure 3.8: Hybrid BBO-Ensemble Algorithm

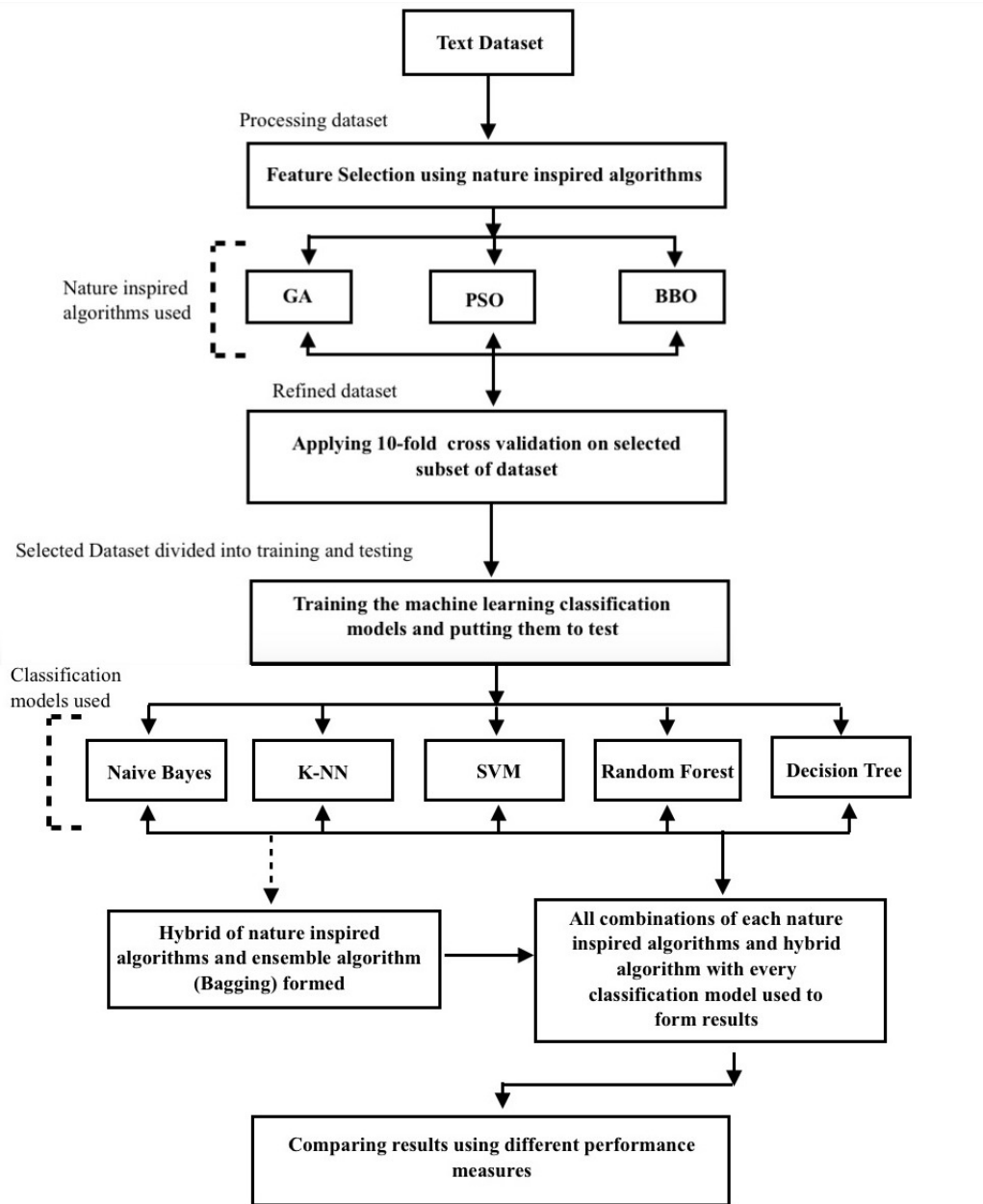


Figure 3.7: Flow chart of the proposed method

3.4 Result

The implementation of the proposed model is performed on Windows 10 64-bit OS, Intel® Core™ i5-5200U CPU @2.20GHz processor. Python3 is selected as a simulation tool for data analysis, hence providing user-friendly environment. The standard libraries of python namely: numpy (used for objects creation and arrays), pandas (for memory management), Scikit-learn or sklearn which have supervised and unsupervised machine learning algorithms, matplotlib for generating graphs were included. In our method GA, PSO and BBO are used for feature selection. For the proposed model, parameter [number of generations] is set to 100, since no improvement in results was shown after 100 number of iterations.

The primary objective of our model is to improve the performance of text classification. The NB, K-NN, SVM, RF, DT and Bagging classifiers are used for classification purpose after feature selection. Figure 3.9 - Figure 3.16, shows the comparison of experimental values of all performance measures trained and computed on six ML techniques on four datasets with each feature selection technique and where no feature selection techniques are used. There could be observed significant improvement after using feature selection approach when compared with values obtained without feature selection approach. Our approach of using BBO has proven to produce the best results in all dataset with the rise in values of performance measure from 15-50% To further validate the experimental results, we applied the Friedman test in Minitab statistical analysis software. The results are compared at a 95% confidence level. Besides, the p-values ($p < 0.001$) indicates that the given technique is significantly better than the other technique available in the literature.

3.4.1 Comparison of Different Feature Selection Techniques

The average values of performance metric is given in Table 3.3 for all the datasets. Average values for all the individual classifiers were computed with feature selection techniques. After computing the empirical results, the statistical analysis shows the order of rank for all the three feature selection algorithms based on average performance values is $BBO > PSO > GA$.

According to the hypothesis taken in section 3.2.5, null hypothesis rejected, the results are shown in Table 3.4. From the results, it can be shown that there is significant difference in techniques used for selection i.e. GA vs. PSO vs. BBO (with $\alpha=0.01$). Feature selection with BBO produce the best results for all the classification techniques. The high convergence rate of BBO and ability to retain previous solutions is one of the reasons behind its superior performance.

Table 3.3: Average Performance values in percentage for Feature selection techniques

optimizer	Average Precision	Average Recall	Average Accuracy	Average F-measure
GA	61.6	44.71	70	51.81
PSO	62.25	51.97	69.69	56.65
BBO	71.21	61.19	80.05	65.82

Table 3.4: Friedman Test results for feature selection techniques

Technique	Precision(Ranks)	Recall(Ranks)	Accuracy(Ranks)	F-measure(Ranks)
without optimizer	1.59	1.18	1.26	1.18
GA	1.99	2.2	2.38	2.14
PSO	2.52	2.62	2.42	2.68
BBO	3.9	4	3.94	4

3.4.2 Comparison of Different Hybrid Nature-Inspired Algorithms and Ensemble Classifier

Table 3.5 shows the average performance values of all the datasets for bagging technique with all feature selection techniques. The conclusion can be drawn that the hybrid model of BBO-ensemble is the better model than GA-ensemble and PSO-ensemble.

To verify the technique Friedman test is performed and results obtained are given in Table 3.6. From experimental results and Friedman test results, it can be concluded that BBO-bagging > PSO-bagging > GA-bagging. After comparing the values in Table 3.3 and Table 3.5, it can be seen that the proposed model stands out with a rise of approximately 20% in all the performance measures. This is possible because of combination of the best converging algorithm, BBO with ensemble classifier (combines advantages of individual classifier), results in an overall superior model with greater advantages and fewer weaknesses.

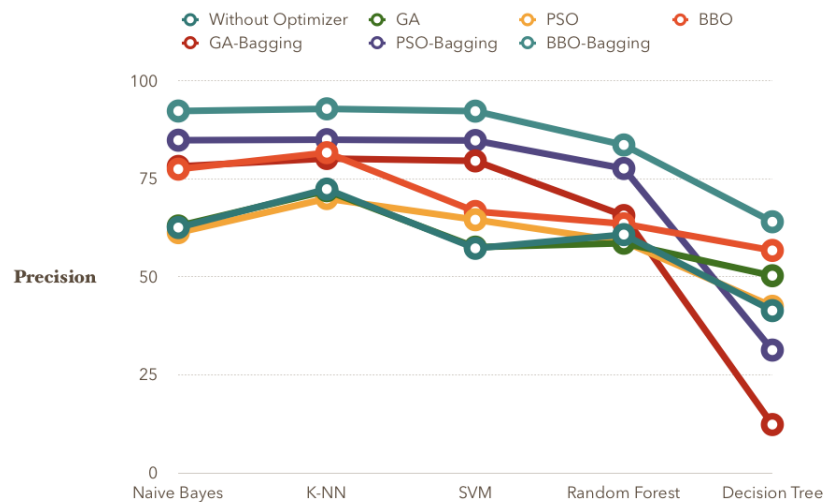
Table 3.5: Average performance values in percentage for feature selection techniques

optimizer	Average Precision	Average Recall	Average Accuracy	Average F-measure
GA-bagging	69.23	52.61	70.55	59.79
PSO-bagging	73.55	59.33	75.35	65.68
BBO-bagging	83.87	70.67	85.16	76.71

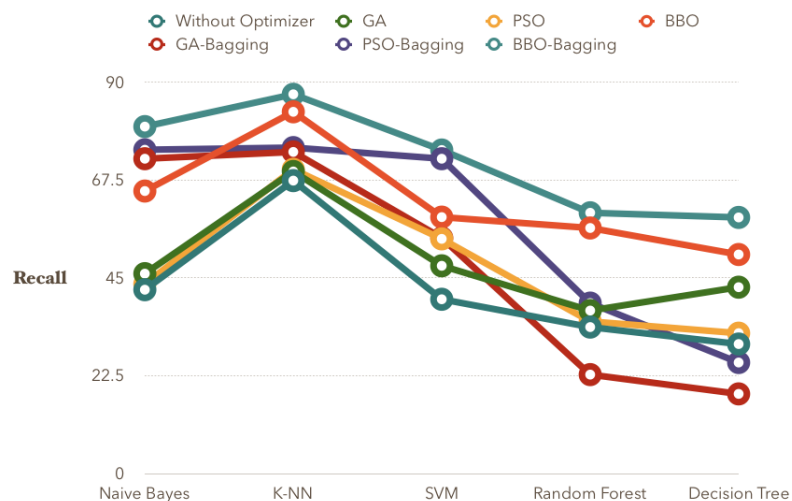
Table 3.6: Friedman Test results for feature selection techniques

Technique	Precision(Ranks)	Recall(Ranks)	Accuracy(Ranks)	F-measure(Ranks)
GA-Bagging	1.1	1.14	1.16	1.08
PSO-Bagging	1.94	1.88	1.86	1.92
BBO-Bagging	2.96	2.98	2.98	3

Figure 3.9, Figure 3.10, Figure 3.11, Figure 3.12, Figure 3.13, Figure 3.14, Figure 3.15, Figure 3.16, Figure 3.17 and Figure 3.18 shows the classification performance measure results in terms of Precision, Recall, Accuracy and F-measure. From the results it can be concluded that our proposed technique is best for optimal classification. To represent the value obtained by each technique with respective classifier we have used different colors. Colors chosen are dark sea-green (without optimizer), green (GA), yellow (PSO), orange (BBO), red (GA-Bagging), blue (PSO-Bagging) and turquoise (BBO-Bagging).

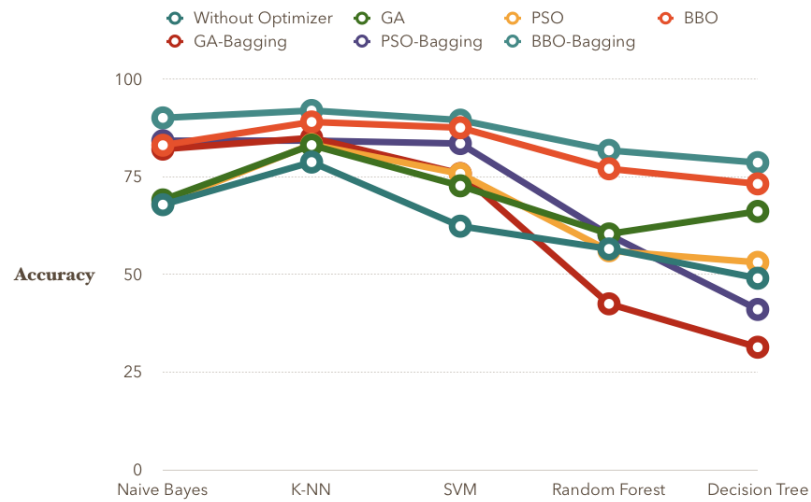


(a) Precision

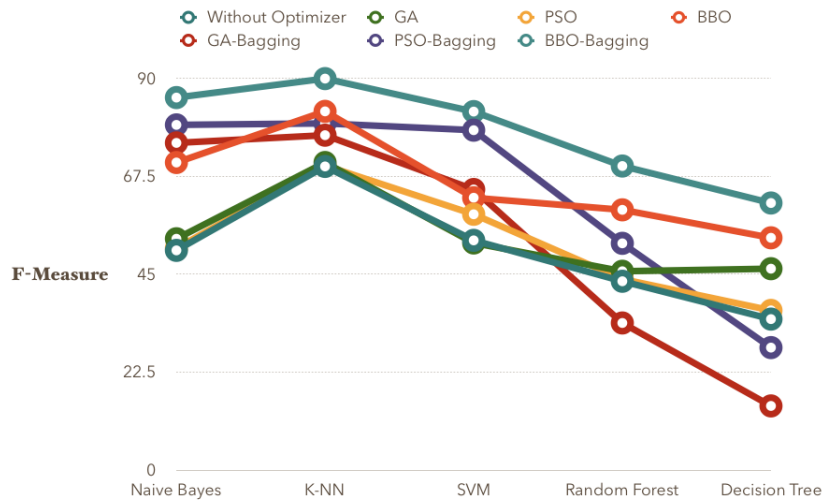


(b) Recall

Figure 3.9: Precision and Recall of tr11



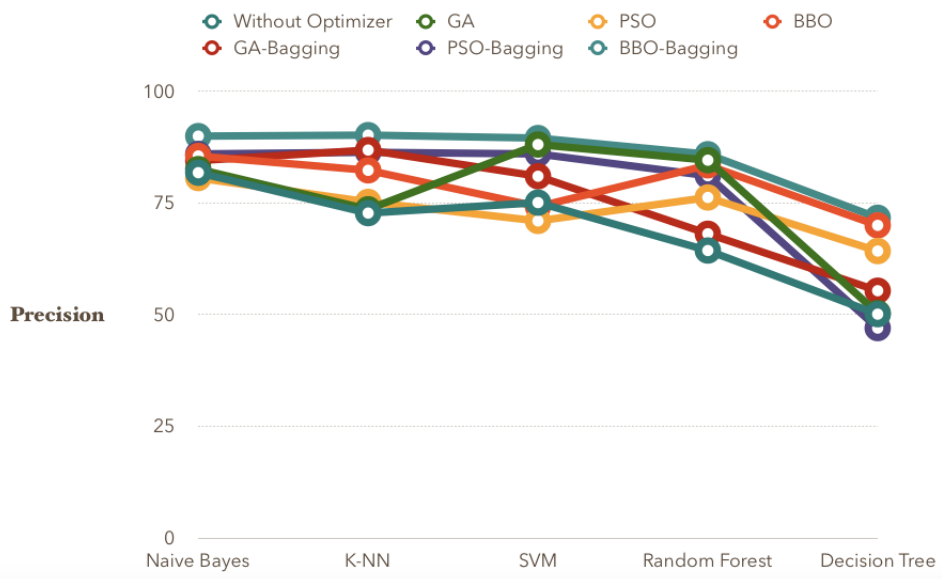
(a) Accuracy



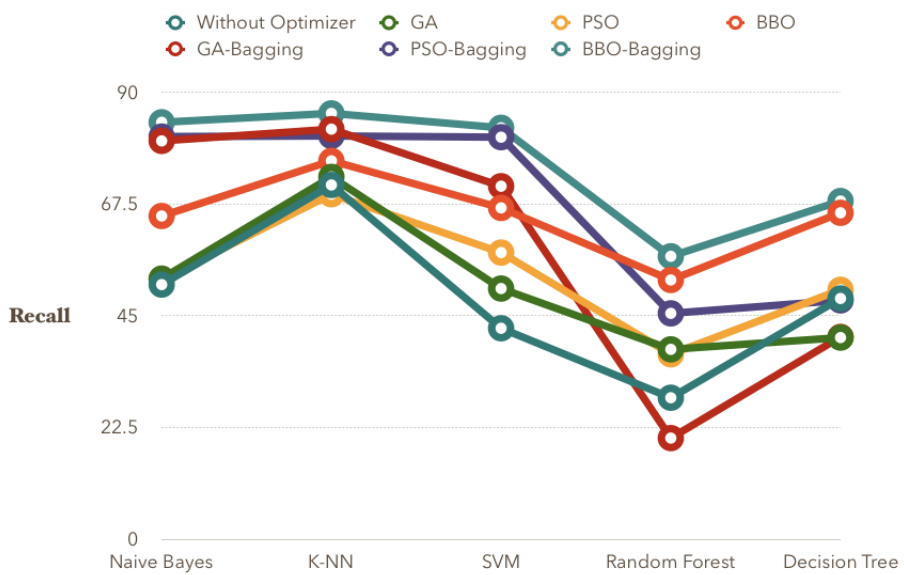
(b) F-measure

Figure 3.10: Accuracy and F-measure of tr11

Figure 3.9 and Figure 3.10 represents all four performance measures for the dataset tr11. It can be analyzed that our model outperformed with a rise in all measures ranging from 15% to 30%. Best values for Precision, Recall, Accuracy and F-measure are 92.89, 87.26, 92 and 89.99 respectively. Our model with base classifier K-NN performed best while worst results were obtained with decision tree (as base classifier for Bagging). Base classifiers as NB and SVM for Bagging also obtained near values to K-NN.

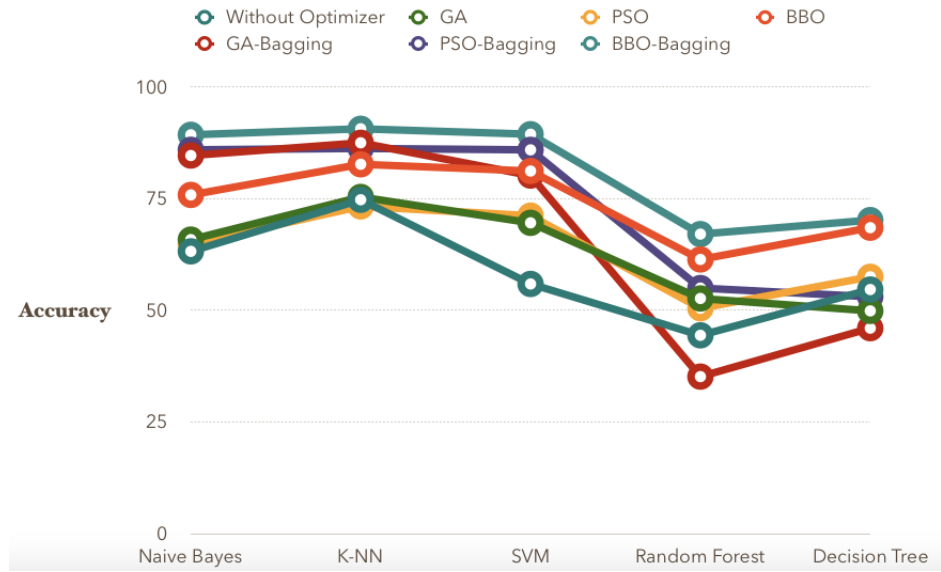


(a) Precision

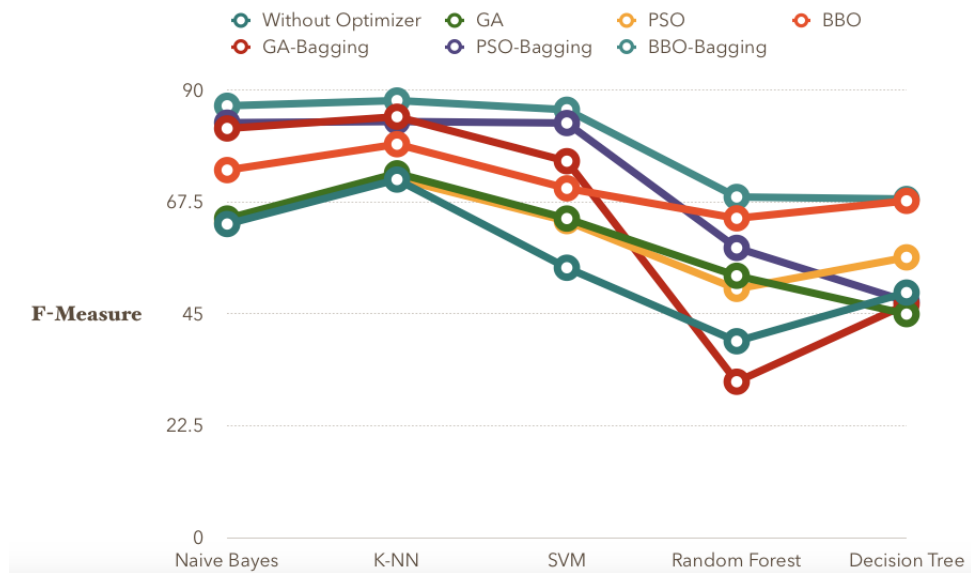


(b) Recall

Figure 3.11: Precision and Recall of tr12



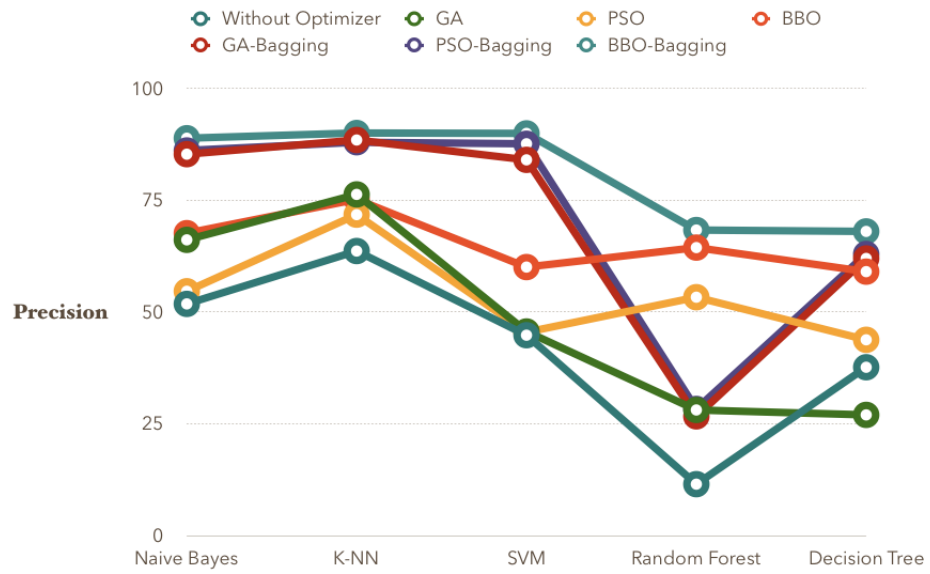
(a) Accuracy



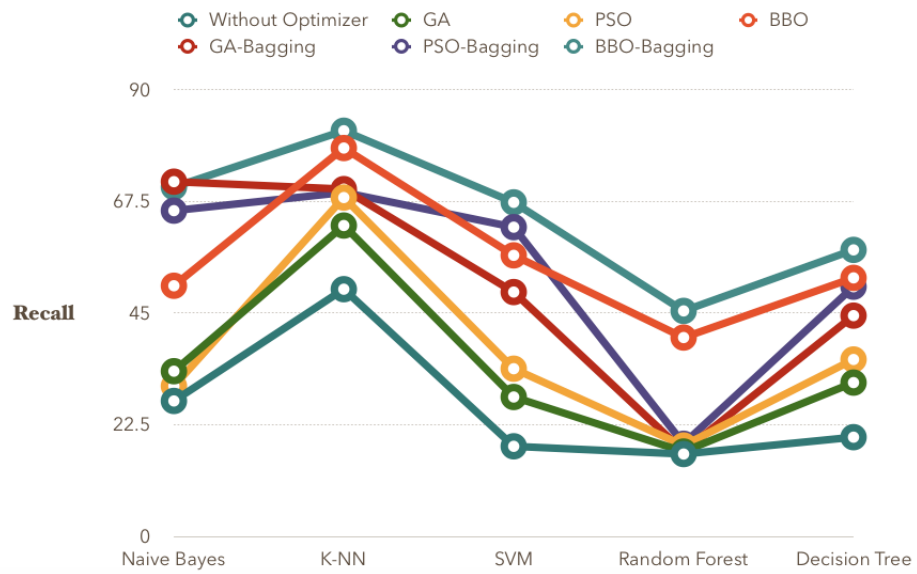
(b) F-measure

Figure 3.12: Accuracy and F-measure of tr12

Figure 3.11 and Figure 3.12 shows the results for dataset tr12. Our model outperformed in all the measures with rise of 15% to 35%. Best results are obtained with K-NN as a base classifier with values 90.21, 85.8, 90.67, and 87.95 for Precision, Recall, Accuracy and F-measure respectively. SVM and NB also performed well near to K-NN for all performance measures. The worst base classifier for this dataset is random forest despite the high precision value of 85.93, as it fails to perform for all other measures.

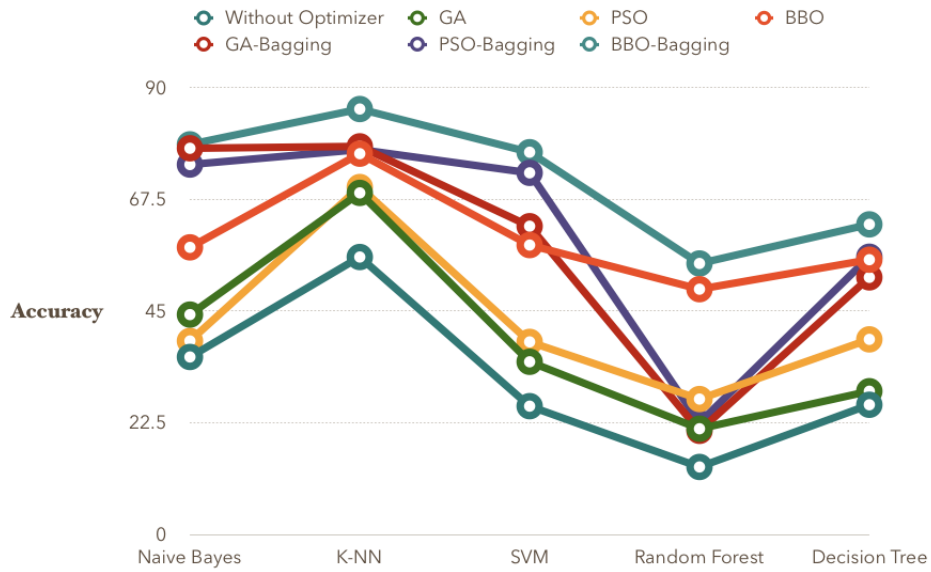


(a) Precision

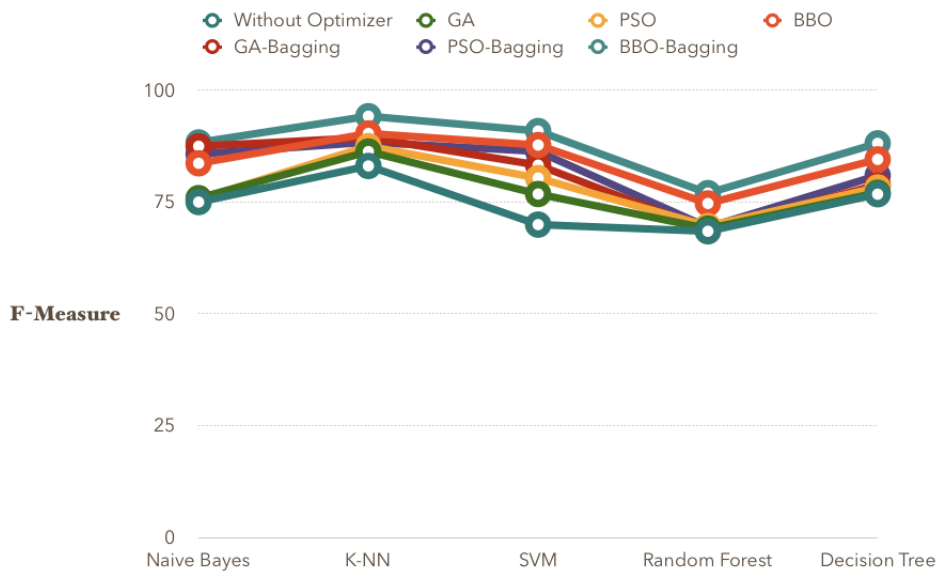


(b) Recall

Figure 3.13: Precision and Recall of tr21



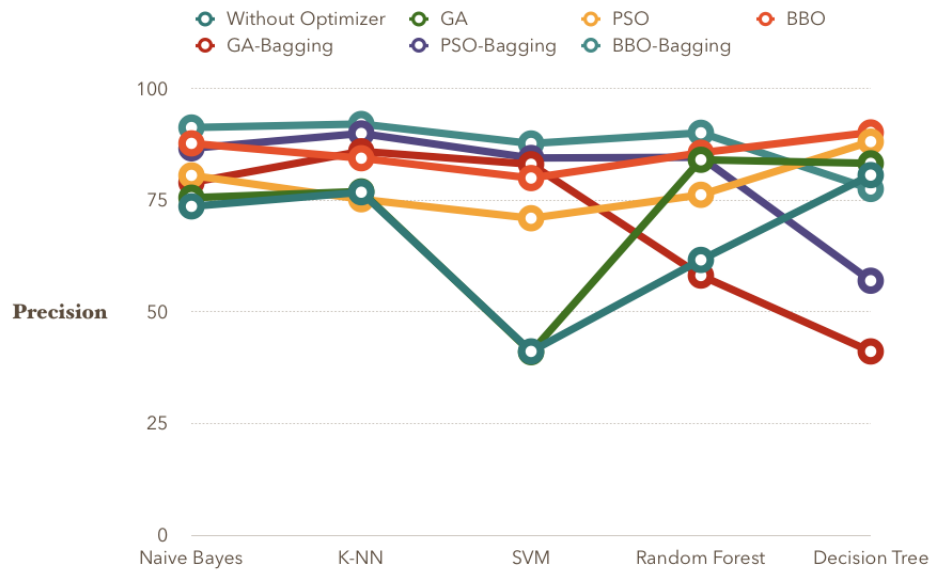
(a) Accuracy



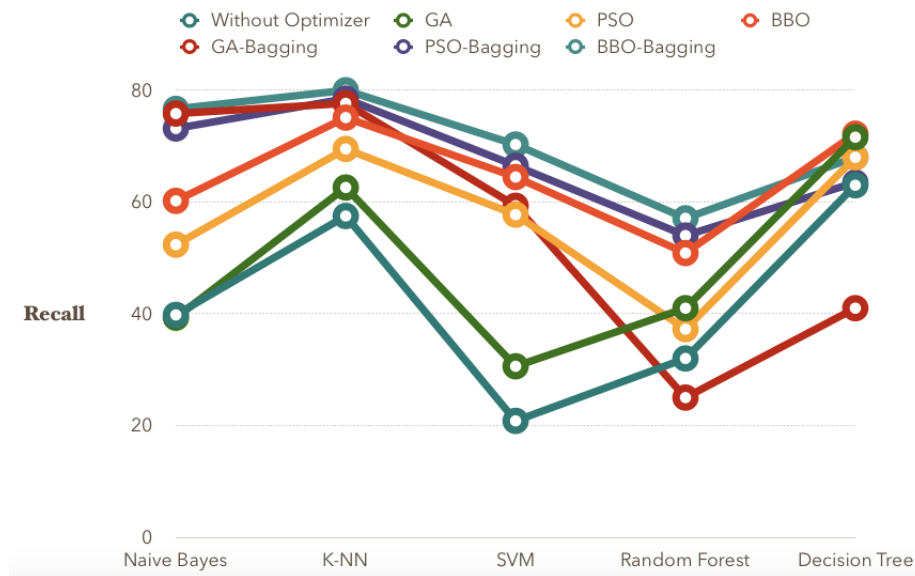
(b) F-measure

Figure 3.14: Accuracy and F-measure of tr21

Figure 3.13 and Figure 3.14 depicts the results for dataset tr21. The performance of all measures rises with our model from 10% to 40% for Precision with best value of 90.07. Rise in Recall from 15% to 40% with best value of 81.77. There is a significant rise of Accuracy and F-measure values from 10% to 20% with best values of 94.21 and 85.72 respectively. All best values are obtained using K-NN as base classifier. SVM didn't perform well for all measures for this dataset.

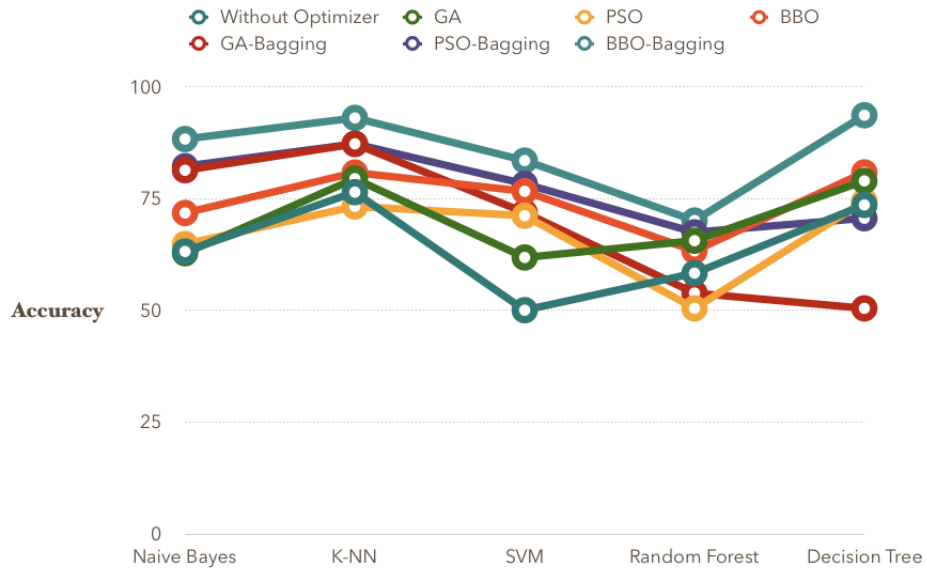


(a) Precision

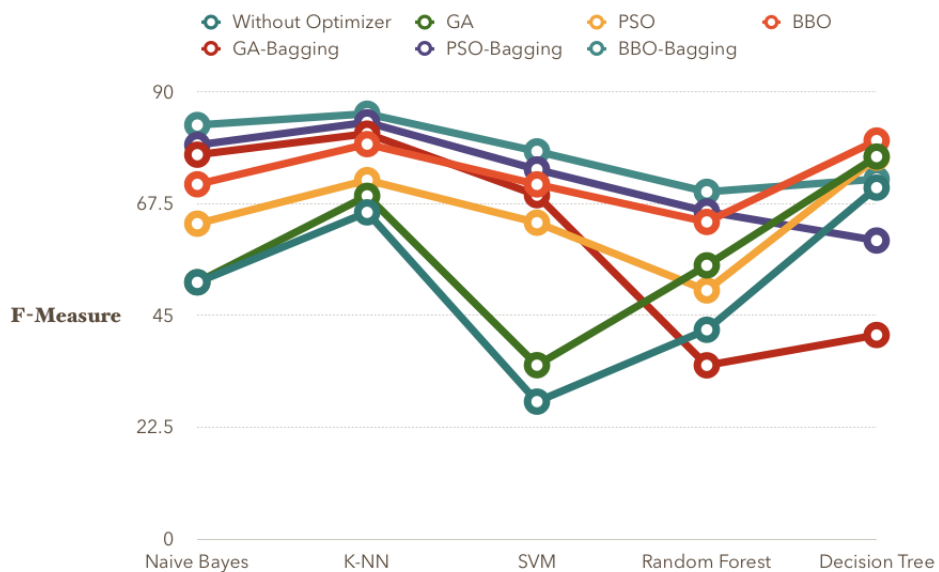


(b) Recall

Figure 3.15: Precision and Recall of tr23



(a) Accuracy



(b) F-measure

Figure 3.16: Accuracy and F-measure of tr23

Figure 3.15 and Figure 3.16 shows the results of all performance measures for dataset tr23. Our model outperformed with base classifier K-NN when compared with other techniques. The best value obtained for each performance measure is 92.11, 80, 93.09 and 85.63 for Precision, Recall, Accuracy and F-measure respectively. NB as a base classifier for Bagging in our model obtained near values to best values. There is a

rise of 10% to 40% for each performance measure.

The other datasets also performed best with our model. Dataset tr31, tr41 and tr45 performed best with our model (base classifier as K-NN or NB). Oh10 and oh15 showed best results with our model having SVM as base classifier. Oh0 obtained best values when BBO-Bagging combined with base classifiers K-NN or SVM. Hence, after analysis, we recommend using SVM as base classifier to Bagging for low dimension datasets while K-NN or NB works better for high-dimensional dataset as a base classifier to Bagging for our proposed model. Hence, discussion and analysis of results for all datasets are as follows. On comparing the performance measure values in tr11, tr21, tr31, oh10, oh15, there is a high improvement in all measures when feature selection is done using BBO, although there is improvement in results when feature selection is performed with GA and PSO, but significant improvement was shown when feature selection is performed with BBO. In tr12, BBO produces higher performance measures, BBO shows best results with all classifiers except SVM for the precision measure. In tr23, BBO is best with all classifiers except with RF for Accuracy measure. In tr41, tr45 and oh10 precision is not improved when feature selection is performed using GA and compared to without feature selection approach, but feature selection with BBO shows great improvement in results of all performance measure.

In oh0, improvement is shown when values are compared with feature selection and without feature selection. Feature selection with BBO produces better results with all performance measures improvement from 5% to 20% except the precision value of SVM. Our model is a hybrid of ensemble technique-Bagging with BBO. The hybrid model is used to test the impact on classification performance, and measure values proves its optimal nature. All performance measures are computed and compared with each other when feature selection is done using GA, PSO and BBO. BBO in all performance measures shown best results with each dataset.

After performing experimental results, the proposed model, where feature selection was performed using BBO, produced best results with ensemble classifiers with the base learner as K-NN except for one dataset oh10 and oh15. There is a rise in accuracy from 5% to 50%, a rise in Precision from 5% to 20%, a rise in recall from 5% to 25% and rise in F-measure from 5% to 30%. In dataset tr31, only precision values (Bagging with k-NN) are not good when compared to values of Bagging with Naive Bayes. In oh10, Bagging with SVM gives better results than any other classifier. For the high value of accuracy and low value of F-measure, there is a class imbalance problem which will be considered for future work.

3.4.3 Comparison with Other Techniques

In [58], authors have used GA, Simulated Annealing, Differential Evolution(DE) and multi-parent mutation and crossover for optimization and tuning of Naive Bayes on 11 text classification datasets. Their experimental results show that their approach of multi-parent mutation and crossover with Naive Bayes is the best approach among other optimization algorithms used for tuning. Our proposed model of optimal text classification has shown better results in all performance measures when compared to other proposed models. The comparison of results is presented in Table 3.7.

In [80], the study has proposed a novel approach of a hybrid model on deep belief network and softmax regression. They introduced an approach of deep belief network for dimensionality reduction and softmax regression for classification. They have tested their model on two datasets i.e. Reuters-21,578 and 20-Newsgroups. We have also implemented our model on these two datasets to compare the classification accuracy and results proved that our model performed better as shown in Table 3.8. The study [173], proposes four term weighting methods and calculate the missing terms of weight. The best performance of classification was shown by SVM classifier. They have used Reuters-21,578 and 20-Newsgroups. We compare accuracy of their approach with our approach and our proposed hybrid approach outperforms their approach as shown in Table 3.8.

In [60], researchers propose a novel approach of the chaotic crow search algorithm to overcome the problem of low convergence rate and trapping in local optima. They have used 20 datasets and we have compared results of our approach on four clinical datasets. Our approach produced much better results than their proposed approach as shown in Table 3.9.

Table 3.8: Comparison of classification Accuracy

Dataset	our approach	jiang[80]	Sabbah[173]
20-Newsgroups	92.57	85.57	55
Reuters-21,578	92.11	86.80	88

Table 3.9: Comparison of classification Accuracy

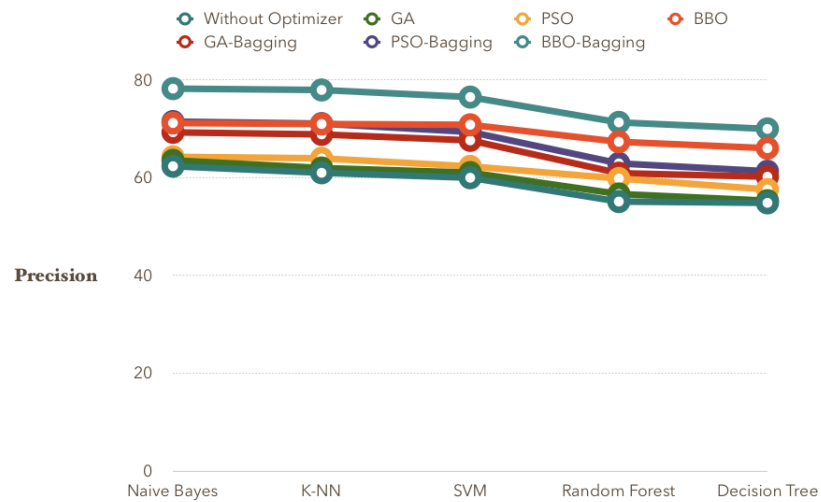
Dataset	our approach	Sayed[60]
Parkinson's Disease detection dataset(PDD)	94.25	90.78
Single proton emission computed tomography(Spect)	86.2	81.5
Stalog(Heart)	82.6	78.84
Indian Liver patient dataset	89.71	71.68

Table 3.7: Comparison of performance measures

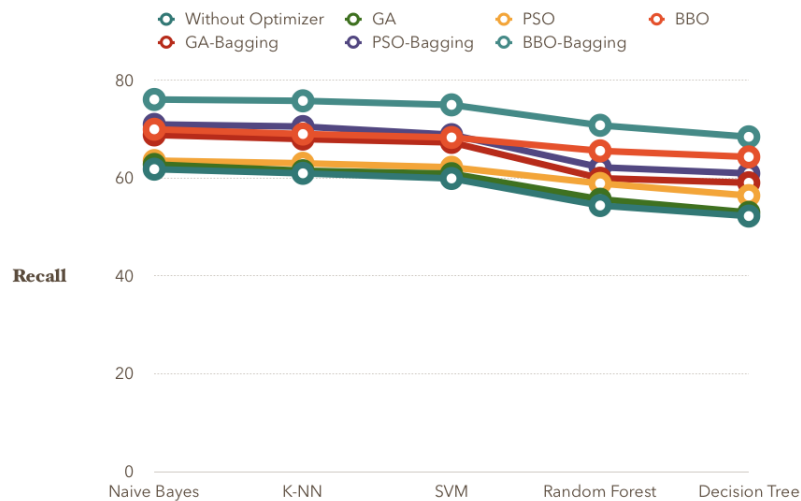
Dataset	Precision	Precision	Recall	Recall	Accuracy	Accuracy	F-measure	F-measure
	our approach	Diab[58]	our approach	Diab[58]	our approach	Diab[58]	our approach	Diab[58]
tr11	92.89	85.5	87.26	82.3	92	87.93	89.99	83
tr21	90.07	86.2	81.77	77.3	94.21	84.19	85.72	81
tr31	89.4	83.2	83.23	79	96.89	92.77	83.88	76
tr41	94.33	89.1	83.2	80.3	92.42	91.46	88.42	83.5
tr45	94.33	89	83.13	80.1	92.42	88.70	88.42	85

3.4.4 Proposed Approach on a Real Dataset of Airlines

We have performed the testing of our proposed model on a real dataset of airlines [172]. The task of correctly predicting the labels and reducing the feature dimensionality leads to optimal performance for the text classification. Airlines dataset have 539,384 records. The objective was to predict whether the flight is delayed or not, in accordance with other attributes. The result of our proposed model on this dataset is shown in Figure 3.17 and Figure 3.18.

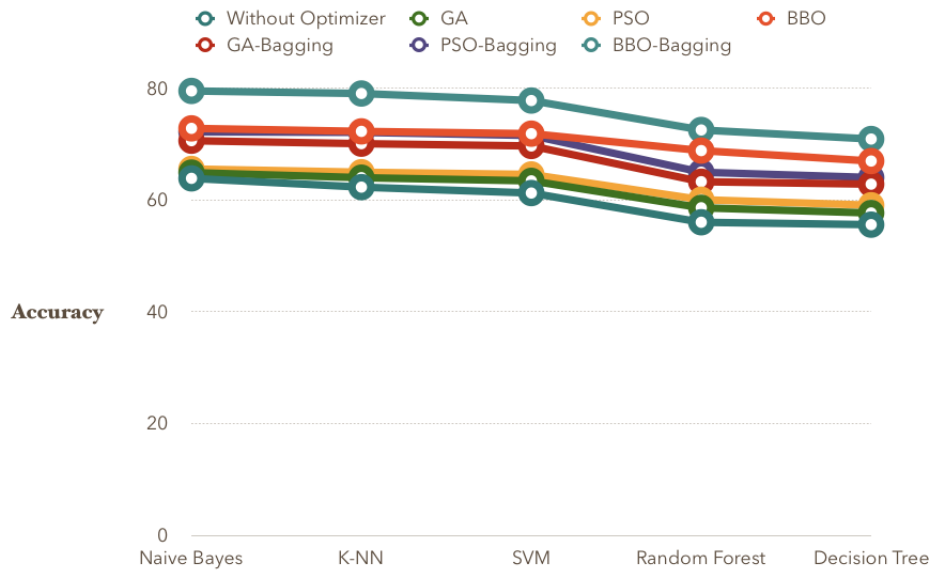


(a) Precision



(b) Recall

Figure 3.17: Precision and Recall of airlines



(a) Accuracy



(b) F-measure

Figure 3.18: Accuracy and F-measure of airlines

The comparison is done with the results obtained from state-of-the-art algorithms of feature selection techniques (with individual classifiers) with our proposed model. It can be observed that the proposed model produces the best result classification performance among all other classification models. After comparing the results, we can conclude that our model gives best results for optimal text classification. The best val-

ues which we obtained are with base classifier NB. The obtained values are 79.96, 76.1, 79.55 and 77.16 for Precision, Recall, Accuracy and F-measure respectively. K-NN also showed almost near values to NB. Hence, our model can be applied for large datasets as well.

3.4.5 Result summary

- 1) Our proposed model performed feature selection with BBO and produces the better classification results when used with machine learning classifiers. All performance values rise from of 5% to 20%.
- 2) In proposed model, we use an ensemble classifier with a feature selection approach using BBO. The model produces better results as compared to individual classifier.
- 3) When experimental values were compared, we found the best results were produced using Bagging with base classifier as K-NN or NB for high dimensional datasets and SVM for low dimensional datasets.
- 4) The ensemble classifier-bagging is paired with other five machine learning classifiers. This results in the enhancement of the classifiers performance in terms of Accuracy, Precision, Recall and F-measure.
- 5) The feature selection method and different classifiers, individual and ensemble, have a different impact on classification results.
- 6) The application on real dataset of airlines also signifies that the proposed model can be to more real-world practical problems.
- 7) All results are validated using 10-fold cross-validation.

The model proposed in this chapter improved the classification results. But, it was analyzed that the classification results suffered for imbalanced dataset.

3.5 Conclusion

In this chapter, we have explored the novel approach of nature-based optimization as feature selection with an ensemble classifier for optimal text classification. The proposed model of BBO and ensemble classifier is also compared with other individual models with state-of-the-art algorithms available in the literature on all datasets. Based on the obtained results, the analysis shows that the performance of BBO as feature selection technique is better than GA and PSO. Performance of text classification is measured in terms of Accuracy, Precision, Recall and F-measure. This model is also verified on real-time dataset of airlines. Our model can be used in practical problems for optimization in text classification. The model is highly competitive optimal text classification method as compared to other feature selection algorithms.

The optimal ensemble model is proven to produce best results among individual classifier. But, in some datasets the improvement is not shown or not producing good higher

accuracy. It is due to class imbalance which will be solved by our next proposed model, explained in our next chapter.

Publication

The work discussed in this chapter is published in:

Khurana, A., and Verma, O. P. (2020). Novel approach with nature-inspired and ensemble techniques for optimal text classification. *Multimedia Tools and Applications*, 79(33), 23821-23848.

CHAPTER 4

OPTIMAL FEATURE SELECTION FOR IMBALANCED TEXT CLASSIFICATION

Text classification has become a major avenue in generating valuable insights. It is being vastly used to solve real world problems by performing sentimental analysis, detecting frauds and patterns in various sectors like healthcare, e-commerce, sports etc. In Big Data, the performance of text classification can be improved by selecting relevant features and handling of imbalance problems between the distribution of classes in the dataset. Class imbalance has become a major issue in determining the classification performance, hence many researchers are working on this problem as imbalance causes adverse effect on the correct predicted results.

4.1 Introduction

In this new era there is an increase in textual information [80]. Text classification is widely used for data analysis and prediction. Many machine learning algorithms are applied in different application areas of text classification such as NB [58], [81], K-NN [81], SVM [87], [158], [81], RF [13], DT [13] and Ensemble classifiers [88]. Some real-world applications in text classification are disease prediction [174], Text categorization [175] and fraud detection [176].

During text classification process, there are many related issues that arise specifically are class imbalance and high dimensional dataset [17]. Class imbalance [22] has become a major issue in determining the classification performance, hence many researchers are working on this problem as imbalance causes adverse effect on the correct predicted results [23]. Many techniques have been developed for balancing the binary imbalanced dataset [174]. But however, the main challenge which is faced by researchers is imbalanced data classification for multi-class dataset [125]. The existing solutions to multi-class imbalanced data is generally the extension of algorithm of binary class imbalance classifier. The construction of Decision Trees [177] and Ensemble based classifiers also proved to give better results in multi-class [178]. The common oversampling technique used is Synthetic Minority Oversampling Technique (SMOTE)

and its different variations [24]. The classical SMOTE and its variant suffer with a problem of small disjuncts which lead to less training data, hence lack of information. In our research, we overcome the problem of lack of information caused by lack of density or class disjuncts.

The other problem that leads to the classification performance is high-dimensionality. The low weight features or noise, that is features which are not essential with respect to data, decreases the efficiency of the algorithm [13], hence degrading the performance of text classification. Feature selection is a process of selecting only desired features related to the data which helps in increasing the efficiency of the model [14]. The various feature selection techniques are discussed in the section 3.1. The new model for language representation was proposed by [179] known as BERT (Bidirectional Encoder Representations from Transformers). It is widely used for many artificial intelligence real world applications [180]. But, main limitation of using BERT arises when dealing with low dimensional dataset.

The proposed technique for balancing the dataset by overcoming the problem of class disjuncts or lack of density is referred as Distributed SMOTE (D_SMOTE). For reducing the problem of high-dimensionality, a novel Modified Biogeography Based Optimization approach (M_BBO) is proposed in this paper. The proposed modification is based on changing the selection procedure of Suitability Index Variable (SIV) in the mutation process. The training is performed using four machine learning algorithms namely: NB, RF, LR and SVM. The performance measures used to validate the model are: AUC, G-mean and F1-score. We compared our model with other techniques, and found that our model for optimal text classification is the most suitable among all. The proposed model is also compared with the other state-of-the-art models available in the literature and the new technique BERT.

4.2 Theoretical Foundations

The classical BBO algorithm is already discussed in Section 3.2.1.3. We will now discuss the new modified Biogeography Based Optimization.

4.2.1 Modified Biogeography Based Optimization (M_BBO)

BBO [21] is a population-based nature inspired algorithm which focused on solution of a problem of global optimization. In the literature, BBO is used as feature selection technique in [2] [181], [182], and proved as best feature selection technique. Hence, in our proposed model, we chose to improve mutation operator in BBO and named technique as modified BBO (M_BBO). M_BBO is used for feature selection to reduce the high dimensionality and hence, improved classification performance. For K-th candidate solution x_k , λ_k is the immigration rate. After performing random selection with

roulette wheel and independent variable candidate (which is chosen) has to be replaced, then the probability of emigrating candidate solution (selected) is proportional to the emigration rate, μ_k , which is performed using roulette wheel selection, and expressed in Equation 4.1. It is in continuation to Equation 3.3.

$$Prob(x_i) \text{ selected for emigration} = \frac{\mu_i}{\sum_{k=1}^A \mu_k} \quad (4.1)$$

where, $i=1,2\dots A$, A is presented as the number of candidate solutions in the population. We have proposed modification in mutation operator of BBO for the improvement in selection of candidate's solution, and migration operator works similar as in classical BBO. Migration uses probability among species. Due to some low suitability factors on islands like lack of resources, species tend to migrate to habitat with low HSI. The migration process modifies the habitat of the island. The immigration probability is denoted by H_t which is directly proportional to λ , the immigration rate. The probability of emigration habitat is H_k , where $H_k \propto \mu$, emigration rate. Pseudocode for migration is shown in Figure 4.1 and migration is shown by Equation 4.2.

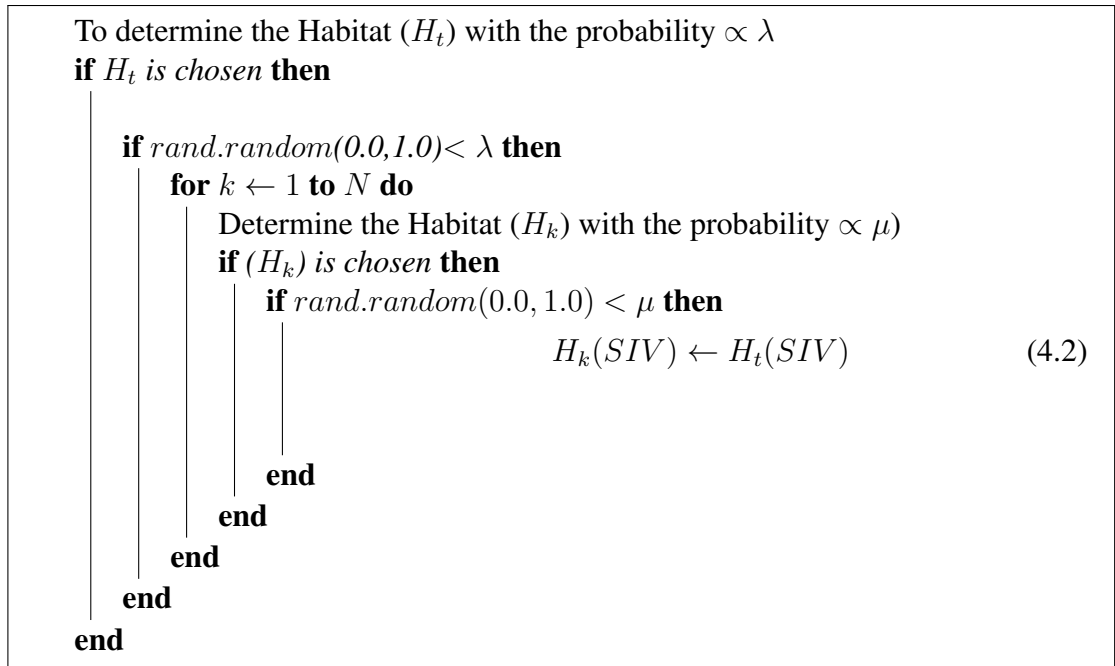


Figure 4.1: BBO Migration

In classical mutation process, BBO calculates the probability of individual to create variations in population. This operator is less effective in terms of difference between actual habitat and resultant habitat. This operator is not frequent and intensifies the solution of search space. In this operator, one habitat is chosen based on mutation probability. Afterward, one index is chosen randomly of this habitat, and put one

new Suitability Index Variable/element between $[0, 2^n]$ (where n is the total number of sequences) in place of the previous element. The SIV's of the i th habitat H_i can be randomly modified by the mutation operator according to the habitat's priori probability P_i . The mutation probability μ_i of the i th habitat H_i is expressed in Equation 4.3.

$$\mu_i = \mu_{max} \times (1 - P_i/P_{max}) \quad (4.3)$$

where μ_{max} is a user-defined parameter and $P_{max} = \max(P_i)$ and $i = 1, 2, \dots, n$. In modified BBO mutation operator, in place of random generation, the SIV feature set is ranked using Recursive Feature Elimination (RFE) method and the lowest rank SIV is mutated with higher rank SIV. RFE [183] is the effective method for recognising the higher rank and lower rank. To revamp the search space and handle the outliers, so that exploration can be controlled, the deciding variable adapts the boundary rank value. It can be lower rank value or higher rank value. This modified mutation process helps in achieving higher fitness for habitat overall. The proposed Mean Ranking method is expressed in Equation 4.4.

$$SIV'_i(j) = (SIV_l(j) + SIV_H(j))/2 \quad (4.4)$$

where, SIV_l = updated SIV with the lowest rank for j th individual solution, SIV_H = feature with highest weightage, SIV'_i = new feature formed with weighted mean of SIV_l and SIV_H . Mutation follows the pseudocode described in Figure 4.2 and modified mutation Equation 4.5.

In this work, we have proposed and implemented M.BBO as feature selection algo-

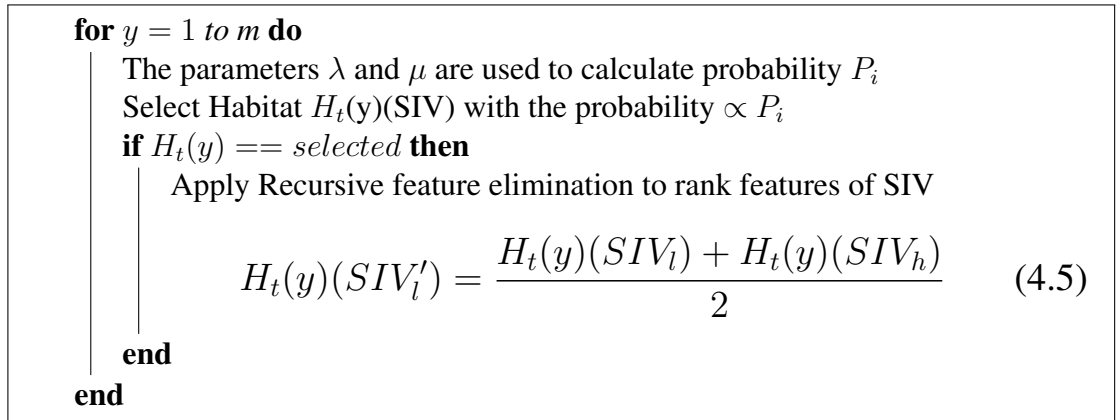


Figure 4.2: Modified BBO mutation operator

arithm. We have considered parameters of an algorithm mentioned in Table 4.1.

Table 4.1: Parameters for BBO feature selection

Parameters	Symbols
Number of habitats, habitat[]	T
The feature set	SIV[]
Maximum number of iterations	K
Immigration	λ []
Emigration	μ []
Maximum immigration	$\lambda = 500$
Maximum emigration	$\mu = 600$
Array to store crucial/redundant SIV factor	gb[]

The working of modified BBO (M_BBO) algorithm for feature selection is explained in Figure 4.3.

4.2.2 Proposed Distributed SMOTE-D_SMOTE

The original SMOTE technique works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line. The need to remove the problem of lack of data by oversampling the minority class instances by SMOTE method leads to the problem of small disjuncts and badly distributed data (with lack of density already a problem). To overcome this problem, we propose a modified distributed SMOTE (D_SMOTE) technique, wherein the synthetic data is generated outside the line joining two present minority class instances. This helps in better distribution of minority class data point and also overcomes the problem of small disjuncts, which in turn, improves the performance of the classifier. The classical SMOTE plots the synthetic sample using K-NN algorithm in between the two minority class instances which causes less instance spatial exploration, hence causing lack of density problem. The other similar technique of oversampling, known as Density-Based Synthetic Minority Over-sampling Technique (DBSMOTE) [184] works on creating the artificial sample near the cluster define by DBSCAN. Their synthetic samples are densely populated near centroid and sparse at far locations. Hence, their technique suffered from small disjunct problem, and our proposed technique overcame this limitation. So, in D_SMOTE synthetic point is generated outwards from the seed samples using Equation 4.6 and Equation 4.7:

$$X_{new} = X_{lower} - (X_{higher} - X_{lower}) \times \delta \quad (4.6)$$

$$X_{new}' = X_{higher} + (X_{higher} - X_{lower}) \times \delta \quad (4.7)$$

Input: habitat[], λ , μ

Output: Optimized feature Set

1. Initialize parameters
2. Generate initial population
 - 2.1 Randomize each habitat population (habitat []): H_1, H_2, \dots, H_T
 - 2.2 Link each habitat index to SIV values(feature)
 - 2.3 λ [], initial immigration rate of each habitat
 - 2.4 μ [], initial emigration rate of each habitat
 - 2.5 Assign index to habitat associated to a feasible solution
3. Calculate fitness of each habitat (solution)
 - 3.1 Calculate using classification rate with new subset of features.
 - 3.2 Compute each habitat HSI value: hab_index []
 - 3.3 Rank habitats using calculated HSI value.
4. Set iteration variable x: depicts the maximum number of iterations and is gradually increased.
5. **while** $x < T$ **do**
 - Calculate emigration rate (μ_j) and immigration rate (λ_i) for each habitat.
 - Here, $j = 1, 2, \dots, T$ and $i = 1, 2, \dots, T$
 - H_j is selected where selection criterion based on emigration rate (μ_j).
 - H_i is selected where selection criterion based on immigration rate (λ_i)
 - Perform migration operation
 - Perform mutation operation on H_i
 - Produce new population by replacing previous (old) H_i with new H_i
 - Re-calculate habitat values, compute corresponding HSI values
 - Increment x
- end**
6. Evolve fittest habitats based on threshold value.
7. Perform ranking second time with respect to population
 - // SIV value that causes the change in hab_index [], is a crucial feature (removed redundant features),
 - for** $x < N$ **do**
 - if** $SIV[x] == crucial\ value$ **then**
 - gb[x]=1;
 - else gb[x]=0 ;
 - end**
 - end**
8. Based on gb[]=1, keep crucial feature and remove redundant features
9. Dataset with reduced features is stored in SIV[]

Figure 4.3: M_BBO Feature Selection

where, X_{higher} is upper coordinates, X_{lower} is lower coordinates of chosen minority samples, δ is a random number between $[0,1]$. X_{new} and X_{new}' are two new coordinates respectively on both the sides of chosen minority samples. For the better distribution of synthetic points we plot the points at varied angle. Consider, $X_{new} = (x_{-1}, x_{-2}, x_{-3}, \dots, x_{-r}, \dots, x_{-n})$ where X_{new} is feature vector in n-dimensions, n corresponds to number of independent features/dimensions. To plot the new synthetic data point of minority class in a different direction to its parent data points, we randomly select a direction for the feature say $x_{-r} = X_{new}$ in direction r ($r \leq n$). Hence, new synthetic features are shown in Equation 4.8.

$$X = x_{-1}, x_{-2}, x_{-3}, \dots, x'_{-r}, \dots, x_{-n} \quad (4.8)$$

where, $x'_{-r} = x_{-r} \cdot \cos \theta$, and $\cos \theta \neq 1$, where, θ is random angle between $[0^\circ, 360^\circ]$ and r varies from 1 to n.

Hence, with the help of above equations we oversample the synthetic points far from the cluster centroid for better learning. The problem of lack of density of instances that needs to be trained and small disjuncts created due to oversampling is removed as uniform distribution is focused for locating synthetic samples. Figure 4.4(a) represents

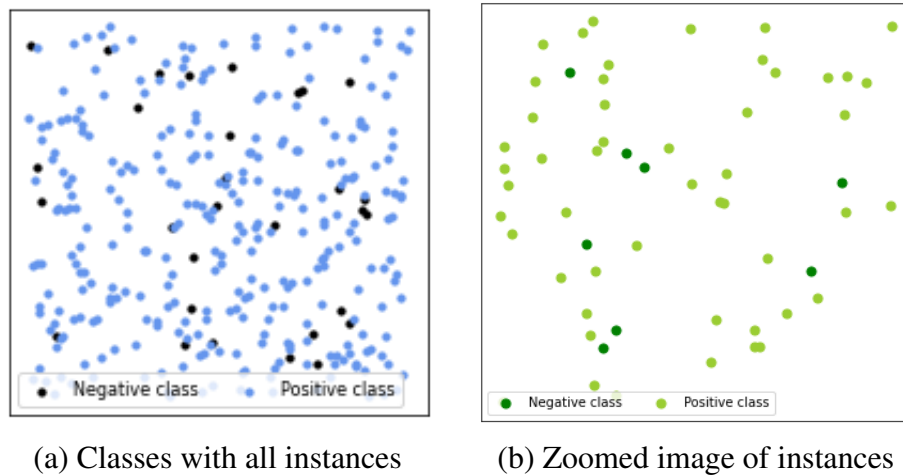


Figure 4.4: Figure of instances

the instances of the classes of dataset and Figure 4.4(b) exhibit zoomed image of some instances. Figure 4.5(a) is showing that how classical SMOTE algorithm works and generates synthetic samples. It is prominent in Figure 4.5(b) that synthetic samples of minority class lie in a single line, shown in red colour. Figure 4.5(c) shows the synthetic samples that are plotted for different instances for the complete dataset. It can be noted that small clusters are formed near to each other minority samples, which is a disadvantage of SMOTE algorithm.

Our proposed algorithm is represented in Figure 4.5(d) where synthetic samples to be

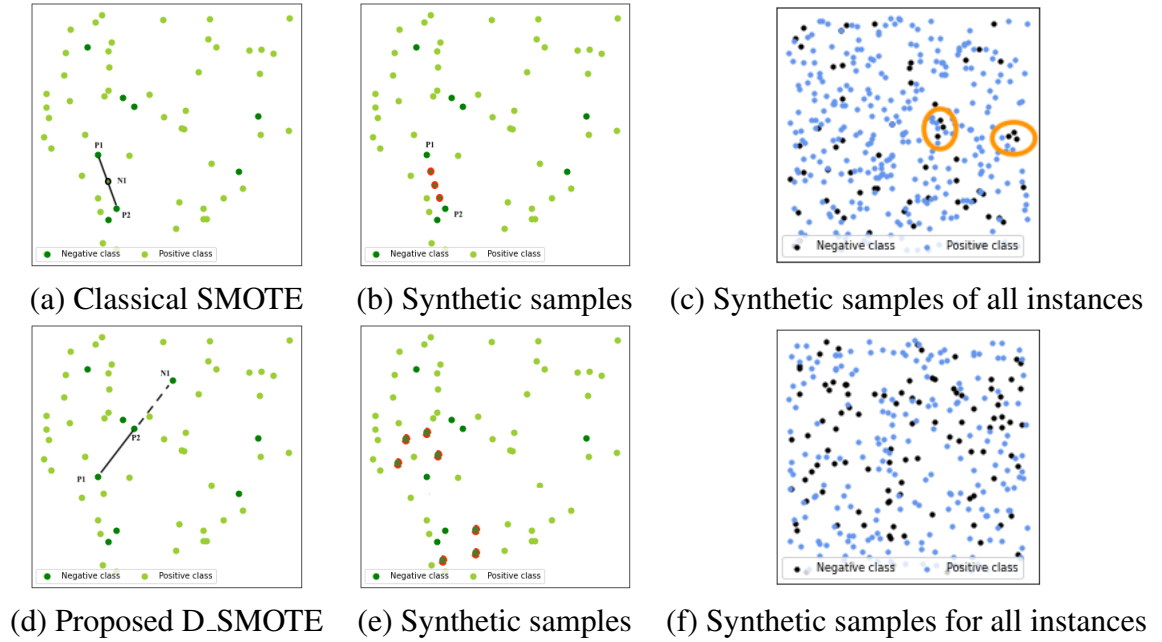


Figure 4.5: Comparison of classical SMOTE with Distributed SMOTE

generated are located outside the two co-ordinates of minority samples, shown by dotted line. The two samples, that are considered, are shown by a solid blue line and the new minority sample that is plotted is shown by a dotted blue line. Figure 4.5(e) shows the generated minority samples and it can be clearly seen that better distribution will lower the number of small clusters or small disjuncts. The new samples are plotted on the opposite side of considered minority samples using different coordinates, to overcome the problem of lack of density and small disjuncts by increasing the exploration area and not leading to form clusters near minority samples. Hence, representation in Figure 4.5(f) shows the elimination of small disjuncts in the complete dataset. The representation in figures shows working of D_SMOTE on binary classes, similarly it works for minority instances in multiple classes. Algorithm for the proposed D_SMOTE is described in Figure 4.6.

4.3 Experimental Setup

The proposed method is analyzed and compared on 17 multi-class datasets. The five techniques were compared for validating the working of the proposed models namely: No random oversampling (No ROS), SMOTE, BBO+SMOTE, M_BBO+SMOTE and M_BBO+D_SMOTE on four datasets. The four classifiers used for classification are: NB, RF, LR and SVM. The three performance measures namely; AUC, G-mean and F1-score are used to analyze the classification performance. The 10-fold cross validation and 50 runs are performed on each model. After the analysis of results, our approach outperformed the other techniques and can be used for balancing the dataset with feature

```

D_SMOTE(N,d,k)
Begin
N - Number of minority class instances
d - Amount of SMOTE-synthetic samples
/* Condition : N+2d is less than the number of majority class samples*/
k - Number of nearest neighbors
/* The amount of oversampling depends on parameter d (SMOTE-synthetic
samples) */
attrnum = Number of attributes
arrnn = To store k nearest neighbors of each minority class instance
Sample: Matrix of minority class instances coordinates
Synlower: Matrix to store synthesized minority class instances coordinates for
lower point
Synhigher: Matrix to store synthesized minority class instances coordinates for
higher point
for  $i = 1$  to  $T$  do
    Calculate k nearest neighbor of  $i^{th}$  minority class instance and store their
    indices in arrnn.
    Populate [N,i,arrnn,k]
end
for  $synindex = 1$  to  $d$  do
    rand = random index between 1 and k
    /* Choosing a random neighbor from arrnn nearest neighbor matrix set */
    for  $j = 1$  to  $attrnum$  do
         $\delta$  = a random number between [0,1]
         $\theta$  = random angle between  $[0^\circ, 360^\circ]$ 
        diff = Sample(arrnn(rand),j) - Sample(i,j)
        Synlower(synindex,j) = Sample(i,j) -  $\delta$ *diff at angle ( $\theta$ )
        Synhigher(synindex,j) = Sample(arrnn(rand),j) +  $\delta$ *diff at angle ( $\theta$ )
        /* Considering Sample(arrnn(rand),j) to be higher coordinated point
        than Sample(i,j) such that 2 new points are generated outside these
        original points */
    end
end
End

```

Figure 4.6: Proposed D_SMOTE

selection. Ranks are calculated using statistical Friedman test on different performance measures, and our proposed technique ranked highest in all techniques.

4.3.1 Data Sets and Classifiers

The 17 multi-class dataset that are chosen for the experiment and analysis is based on the survey. Datasets are chosen from Openml [120], KEEL [185] and UCI [186]. Complete description of dataset is summarized in Table 4.2. The 10-fold cross-validation is used to carry out the experiment on each dataset for the validation of results. We collected diversified dataset varying classes from 3 to 26, features ranging from 8 to 1586 and instances from 122 to 20,000. The Imbalanced Ratio(IR) is calculated as the ratio of total number of majority class instances to the total number of minority class instances. The ratio varies between 1.57 to 6.99. The brief description of our four high dimensional datasets is as follows:

SRBCT- This is a microarray dataset named as small round blue cell tumors. It mainly consist of four classes named as neuroblastoma, Ewing’s family of tumours, Burkitt’s lymphoma, and Rhabdomyosarcoma. Total number of attributes are 1586 with 122 instances.

Burczynski- Gene Expression omnibus is a super dataset of Burczynski. There are 22,823 attributes with 127 number of samples. It mainly consist of three classes normal, crohn’s diseases and ulcerative crisis.

Glioma- This dataset is a textual reports of glioma cancer incidence for mortality analysis. It consist of comprehensive study of PLCO dataset. Nearly 4434 attributes are extracted and 80 instances are considered.

Bullinger- This dataset is a constituent of gene expression of 116 de novo Acute Myeloid Leukemia. It contains high number of features numbered 17,404. We have considered 4 type of classes of leukemia(monocytoid, general leukemia, myeloid and promyelocytic.)

We have shorten the name of dataset `IIF_intensity_all_features_data_set_index`—files as `IIF_intensity` in this work. We have used four different classifiers for our model which are implemented in Python library scikit-learn [187]. The classifiers used for experimentation are: NB [127], RF [129], LR [188] and SVM [127]. All these classifiers are known to produce best results among other classifiers. NB classifier needs less training and converge quickly like LR. RF classifier estimate the importance of features in a reliable manner. SVM can clearly maintains the separation between the classes. SVM works better with high dimensional datasets.

Table 4.2: Description of multi-class dataset

Dataset	class	instances	Attributes	IR
SRBCT	4	122	1586	6.55
Burczynski	3	127	22,823	3.88
Glioma	4	80	4434	6.14
Bullinger	4	170	17,404	11.25
Faults	7	1941	27	14.05
Abalone	18	4139	10	45.93
wine-quality-red	6	1599	11	68.1
Breast tissue	6	1006	9	1.57
Glass	6	214	10	8.44
yeast	10	1484	8	28.1
letter	26	20000	16	1.11
Plates_faults7	7	1941	27	61.11
housing10	10	506	13	152.35
IIF_intensity	3	600	57	4.31
gas_batch1	6	445	128	2.19
Penbased_10an_nn	10	10,992	16	1.95
dataset32pendigits	10	10,992	16	1.16

4.3.2 Assessment measures

For the analysis of performance of imbalanced multi-class dataset, widely used performance measure is AUC [189], G-Mean [190] and F1-score [120]. Precision(P) [101] and Recall(R) [22] are used as a measure for binary class imbalance problem. Precision (P) and Recall (R) is the measure of exactness and completeness respectively [120]. As we have proposed model for multi-class imbalance problem, we have measured performance using F1-score, G-mean and AUC. F1-score computes value using Precision and Recall, to balance the trade-off between the measures [101]. The formula is described in Equation 4.9.

$$F1 - score = \frac{2 \times (Recall \times Precision)}{(Recall + Precision)} \quad (4.9)$$

G-mean indicates how well a classifier can balance the recognition among different classes. High value of G-mean means that all the classes are considered without the ignorance of a single class. G-mean [190] is calculated as geometric mean of the Recall over all classes. Consider a problem for c-class formalized in Equation 5.4:

$$G - mean = \left(\prod_{i=1}^c R_i \right)^{1/c} \quad (4.10)$$

Receiver Operating Characteristics (ROC) curve is created by plotting True Positive Rate (TPR) and False Positive Rate (FPR). Area Under Curve (AUC) is referred to the area lies under the ROC curve [120].

4.4 Results and Analysis

4.4.1 Performance of Distributed SMOTE with Modified BBO

This section analyses and investigates the result of the proposed approach. The results are compared with other approaches as well as the other models proposed by different authors. The comparison and analysis increase the validation of results for the future use by improving the efficiency of text classification. It can be seen through results that proposed D_SMOTE approach over-samples the minority class samples without modifying the originality of minority class. Particularly, in multi-class cases the minority class regions can be wrongly classified into majority class regions. D_SMOTE is performing oversampling along with increasing the density area of minority data instances. In other words, D_SMOTE overcomes the problem of class disjuncts and lack of density by increasing the area to be explored for minority samples. The more instances can be explored and trained, and hence generalization of the results for optimized classification can be achieved. The comparison of D_SMOTE with other six oversampling techniques proved the efficacy of our technique. D_SMOTE performed best in three datasets compared with other techniques and in fourth dataset it performed nearly equal to the best performance by ADASYN. The use of feature selection technique, M_BBO, enhances the classification performance. To analyze the performance of our model, M_BBO+D_SMOTE, with other models, we perform comparison on three performance measures. We have also performed comparison with the state-of-the-art techniques proposed by other authors on seventeen datasets, for AUC performance measure, using SVM classifier only as we find the best results with the SVM classifier for our approach in most of the cases. Our model outperformed other techniques for fourteen datasets. The calculated ranks of different samples using Friedman test also signifies statistically that proposed model produces the best results among the other approaches.

1) Comparison with SMOTE variations--Table 4.3 shows the comparison with other techniques used to handle imbalance problem. The comparison is made using AUC scores for different variants of oversampling techniques exist in literature Random-oversampling [111], SMOTE [110], SMOTE-B [113], ADASYN [112] and SMOTE-SL [115]. No RS signifies the scores for no random sampling and ROS signifies the score for random oversampling. Our proposed technique D_SMOTE performed best in three datasets. ADASYN performed best in Bullinger dataset with 91.8%. The rise with 8% to 25% is seen when compared to best values of different variants. In some cases, SMOTE-B and ADASYN performed nearly good to our technique. Due to space constraint, we have used short form of our datasets as SRBCT (S), Burczynski (B), Glioma (G) and Bullinger (Bu).

The following observations are analyzed:-

1) In SRBCT, Burczynski and Glioma dataset, our proposed approach of D_SMOTE performed best when compared to other techniques with scores 93.4%, 90.3% and 93.9% respectively. For Bullinger dataset ADASYN performed best with 91.8% and D_SMOTE also performed with 91.6%.

2) The SMOTE-SL performed worst in Burczynski dataset with 45.4% score. ROS performed worst in Bullinger dataset with 60% and in Glioma dataset ADASYN technique performed worst with 50.9%. SMOTE also performed poor for Glioma dataset. No RS performed worst for SRBCT dataset with 46.3%.

3) In Glioma dataset the performance of ROS and SMOTE-SL performed good with scores 91.6% and 86.9%. The performance of ROS with LR classifier achieved best score when compared among other techniques.

4) In Bullinger dataset our approach produces equal score with SMOTE combined with SVM classifier. RF classifier performed worst with our technique for this dataset when compared to other techniques.

Our technique D_SMOTE did not work satisfactorily for three datasets with NB classifier and for two datasets with RF classifier. It is due to the presence of noise instances and overlapping of classes, which lead to a decrease in the classification scores. D_SMOTE works best with NB with one dataset which have medium imbalanced ratio. The other two classifiers LR and SVM performed best for all the datasets with D_SMOTE. Hence, D_SMOTE is best for all types of datasets when combined with LR and SVM.

2) AUC Scores- In Table 4.4, comparison of AUC measures is performed for all datasets with other techniques that can be applied to improve the text classification performance. AUC is the correct measure of classes separability and works on probability. The efficiency of a model in terms of classification or separability of classes is measured by AUC. Higher values of AUC signifies that the model classifies the different classes more accurately. From these values, we can analyze that our approach of M_BBO+D_SMOTE has outperformed over all other techniques. SVM performed best for our model for all the datasets. SVM classifier follows kernel trick which increases the feature space size which cannot be performed by other classifiers. SVM generally performed well with small and medium sized feature space. The better density distribution produced by the proposed D_SMOTE is clearly fabricating best results for the datasets in Table 4.4 namely: SRBCT, Bullinger, Burczynski and Glioma. M_BBO+D_SMOTE is best in all datasets, hence it generalizes to use the approach for balancing in multi-class datasets.

The other observations made from Table 4.4 are:

1) In SRBCT dataset, AUC best scores for NB classifier is 85.6%, while with SVM it

Table 4.3: AUC score for various oversampling methods

D	No RS			ROS			SMOTE			SMOTE-B			ADASYN			SMOTE-SL			D_SMOTE		
	NB	RF	LR SVM	NB	RF	LR SVM	NB	RF	LR SVM	NB	RF	LR SVM	NB	RF	LR SVM	NB	RF	LR SVM	NB	RF	LR SVM
S	71.7	73.5	46.3 73.6	79.5	59.9	52.5 78.4	54.0	70.7	59.1 73.0	73.5	67.6	64.4 52.1	72.3	69.2	51.6 78.7	76.5	47.6	65.8 77.4	72.2	75.5	76.3 93.4
B	62.4	69.2	59.7 80.3	72.2	63.3	50.2 78.8	59.1	65.1	53.3 81.5	67.4	72.8	78.9 82.2	62.6	53.3	60.1 75.6	60.1	58.2	45.4 74.5	76.1	83.8	88.6 90.3
G	69.3	84.9	80.6 89.1	90.0	70.6	73.8 91.6	52.6	66.2	68.5 88.7	85.1	75.3	80.8 88.6	82.4	50.9	81.5 79.9	70.1	69.3	77.7 86.9	82.6	70.4	82.8 93.9
Bu	75.1	70.8	69.4 88.7	83.4	60.0	75.1 81.9	79.6	74.1	72.5 90.8	63.2	69.5	65.3 87.4	81.6	61.7	69.1 91.8	79.4	79.2	61.3 83.2	70.8	65.6	75.4 91.6

is 96.2%, with LR is 89.4% and with RF the best scores obtained is 83.7%. The SVM score rise around 17.4% when compared to the worst score. The dataset performed worst with BBO+SMOTE model with three classifiers and with NB, M_BBO+SMOTE is producing lowest results.

2) In Burczynski dataset, the best scores are attained by our technique combined with SVM classifier with score 99.9% which is 17.2% more than the worst score. Our technique M_BBO+D_SMOTE are better than all the other techniques with SVM classifier. In this dataset, the worst score with NB classifier is attained by BBO+D_SMOTE with 76.5%. Rest all three classifiers produced worst score with BBO+SMOTE.

3) In Glioma dataset, the best score of 100 is attained by our proposed technique with SVM classifier, rise with 8.8% when compared to worst score. With LR classifier the best scores attained by our technique with 92.2% and also M_BBO+SMOTE performed well with 92%. In this dataset, BBO+SMOTE performed worst with all classifiers.

4) In Bullinger dataset, the best score is attained by our approach with SVM classifier with score 99.10%, with rise of 7.2% when compared to worst score. Our model outperformed with all the classifiers. The worst score with NB and RF classifiers was produced by BBO+D_SMOTE, and rest two classifiers that is LR and SVM performed worst with BBO+SMOTE.

Our proposed model M_BBO+D_SMOTE performed best in all the datasets with all the classifiers. It is analyzed that BBO+D_SMOTE did not performed well in two datasets with NB and RF in one dataset. Although, we have improved D_SMOTE such that it can understand the dataset more precisely. The classification performance degraded due to the problem of poor exploitation that exists in BBO. The assumption of independent predictor features of NB doesn't combine with BBO+D_SMOTE.

3) G-mean scores- In Table 4.5, G-mean for all techniques is calculated for optimal text classification. G-mean indicates how well a classifier can balance the recognition among different classes. High value of G-mean means that all the classes are considered without the ignorance of a single class. We can clearly conclude from the table that our proposed model outperforms among other techniques with difference of 5% to 30%. The value of G-mean signifies the balance measure, and calculated values signifies that it is correctly predicting the majority and minority class. It can be seen that the best values for most of the datasets are obtained using SVM classifier. The difference between the G-mean values and AUC values lies in the fact that AUC is a performance metric that equals to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. G-mean is a performance metric that combines true negative rate and true positive rate at 'one' specific threshold - where both the errors are considered equal. Hence, G-mean works on absolute values. The various observation made from Table 4.5:

Table 4.4: AUC score

Dataset	BBO+SMOTE				BBO+D_SMOTE				M_BBO+SMOTE				M_BBO+D_SMOTE			
	NB	RF	LR	SVM	NB	RF	LR	SVM	NB	RF	LR	SVM	NB	RF	LR	SVM
SRBCT	79.9	68.4	77.3	78.8	81.6	78.2	81.0	86.7	75.1	79.9	84.5	88.8	85.6	83.7	89.4	96.2
Burczynski	78.7	76.6	79.1	82.7	76.5	82.6	86.4	91	85.4	83.8	87.4	90.9	88.2	97.5	94.3	99.9
Glioma	75.5	74.1	89.4	91.2	76.0	75.8	91.3	93.2	80.6	79.6	92.0	95.5	96.7	86.8	92.2	100
Bullinger	76.2	75.5	71.0	91.9	71.4	73.1	76.5	92.1	74.7	77.3	84.1	93.3	89.9	88.6	92.5	99.10

1) In SRBCT dataset, the best scores are obtained by M_BBO+D_SMOTE technique. For NB classifier score achieved is 86.4, while with SVM is 95.9, with LR it is 91.1 and with RF it is 82.9%. BBO+D_SMOTE with SVM also performed good with 91.5%. The SVM score with our technique rises around 15% when compared to the worst score. The dataset performed worst with BBO+SMOTE model.

2) In Burczynski dataset the best scores are attained by our technique combined with SVM classifier with score 97.7% which is 17.2% more than the worst score. Our technique M_BBO+D_SMOTE are better than all other techniques. In this dataset, the worst scores attained with BBO+SMOTE with two classifiers, NB and SVM with 74.1% and 80.5% respectively. The two classifiers RF and LR performed worst with BBO+D_SMOTE with 62.4% and 72.8% respectively.

3) In Glioma and Bullinger dataset, our proposed technique best with SVM classifier with 98.6% and 97.5% respectively. The worst scores are produced with BBO+SMOTE technique in both the datasets with all classifiers, except in Bullinger dataset with LR classifier. M_BBO+SMOTE performed worst for LR with 62.3%.

Our model performs best with all the classifiers in all the datasets. The best performance is achieved by SVM, although the other classifiers also performed well. M_BBO+SMOTE does not work well for one dataset with LR classifier which is due to the lack of influential outliers. The other reason for degraded performance is the presence of small disjuncts and noise instances. Also, BBO+D_SMOTE does not perform well with RF and LR for Burczynski.

4) F1-score values- The values are shown in Table 4.6. It can be clearly stated that higher value of F1-score shows the balancing of dataset, and our proposed model is generating highest values when compared to other approaches. The value of F1-score gives the information of the balance between the Precision and Recall. So, the value of Precision or Recall affects the value of F1-score, hence any zero value will generate zero. Fairly balanced dataset will generate higher F1-score results and hence our model, which is balancing the multi-class dataset is outperforming other techniques.

The observations made from Table 4.6:

1) In SRBCT dataset, our proposed approach performed best with all classifiers. The best score produced is with SVM classifier with 94.2%, with rise of 14.7% compared to worst score. The worst scores are produced by BBO+SMOTE with all classifiers.

2) In Burczynski dataset, the best score is attained is 96.5% with our approach combined with SVM classifier. The worst score of 42.3% for RF is obtained by BBO+D_SMOTE. The other three classifiers performed worst with BBO+SMOTE.

3) In Glioma dataset, our approach performed best with 99.8% score with SVM classifier, and also performed better with all other classifiers. M_BBO+SMOTE produces good results with LR classifier and worst result with NB classifier. Although NB classi-

Table 4.5: G-mean score

Dataset	BBO+SMOTE				BBO+D_SMOTE				M_BBO+SMOTE				M_BBO+D_SMOTE			
	NB	RF	LR	SVM	NB	RF	LR	SVM	NB	RF	LR	SVM	NB	RF	LR	SVM
SRBCT	63.4	65.6	54.5	80.9	73.3	72.2	64.3	91.5	83.6	70.6	72.5	75.2	86.4	82.9	91.1	95.9
Burczynski	74.1	66.5	74.6	80.5	76.1	62.4	72.8	83.6	78.8	75.5	86.4	89.9	90.2	91.6	95.5	97.7
Glioma	65.4	73.3	79.3	86.2	72.5	80.8	84.7	90.2	68.1	76.4	85.1	92.7	96.1	88.5	93.4	98.6
Bullinger	61.2	55.8	66.3	52.7	69.6	63.0	76.1	65.4	72.5	65.1	62.3	78.6	89.3	88.6	92.3	97.5

fier performed worst with BBO+D_SMOTE. All other three classifiers performed worst with BBO+SMOTE.

4) In Bullinger dataset, the best scores are produced by our approach with all classifiers. The best score with SVM classifier is 97.1%, which is 37.0% value higher than the worst value which is attained by BBO+SMOTE. All classifiers performed worst with BBO+SMOTE.

BBO+D_SMOTE is not recommended with NB and RF classifiers, due to the mixed classification performance values. It is producing good results with LR and SVM classifiers. Our proposed approach can be applied with all the classifiers and to all kind of datasets varying from low dimensional to high dimensional, and low imbalanced to highly imbalanced. The comparison and analysis for AUC measure with other models proposed by authors namely; KNU [122], SMOTE-SF [127], MDO [120], SMOM [121] and DECOC [125], validate our model as it performed better in fourteen datasets. All results in Table 4.7, are generalized by considering the average value of 50 runs. By seeing towards the results, our model outperformed with values in the range from 4% to 25%.

Table 4.7: Comparison of performance measure (AUC X 100) with different approaches

Dataset	KNU	SMOTE-SF	MDO	SMOM	DECOC	our approach
SRBCT	70.1	76.1	74.2	82.1	71.3	93.5
Burczynski	62.1	82.1	72.1	71.3	69.2	97.2
Glioma	78.5	89.1	69.3	78.6	74.4	99.4
Bullinger	82.3	92.5	79.10	65.9	72.3	98.3
Faults	81.2	78.3	75.4	82.5	71	90.1
Abalone	79.2	85.5	93.9	86.6	82.1	98.7
wine-quality-red	85.6	91.4	95.3	85.2	90.3	98.5
Breast tissue	86.3	83.7	92.5	80.1	91.5	90.4
Glass	89.2	92.3	93.1	82.3	86.3	96.3
yeast	85.6	81.3	92	90.1	86.3	98.6
letter	93.5	90.3	91.4	83.2	87.2	99
Plates_faults	72.4	62.1	59.9	69.9	78.2	92.3
housing10	77.3	86	63.2	76.2	78.4	96.2
IIF_intensity	86.2	73.17	75.2	82.6	85.4	93.1
gas_batch1	81.3	86.3	73.2	96.3	89.2	99
Penbased_10an_nn	75.2	68.2	76.9	83.3	95	92.2
dataset32pendigits	79.8	87.1	86.2	78.1	97.1	90.8

The comparison with BERT [179] for Accuracy measure [2] is shown in Table 4.8 for four datasets. The implementation of BERT is performed using tensorflow. We have considered small datasets to large datasets. We found that our model outperformed in small datasets but performed equally well in large datasets, when compared with

Table 4.6: F1-score values

Dataset	BBO+SMOTE			BBO+D_SMOTE			M_BBO+SMOTE			M_BBO+D_SMOTE		
	NB	RF	LR	SVM	NB	RF	LR	SVM	NB	RF	LR	SVM
SRBCT	65.2	63.3	69.1	79.5	70.1	72.5	69.6	82.5	82.4	78.4	82.1	86.7
Burczynski	68.4	56.6	59.6	63.4	69.2	42.3	60.5	75.7	80.6	85.7	82.1	85.8
Glioma	64.5	65.1	58.3	65.7	60.1	68.6	77.4	74.8	64.9	81.9	91.3	93.2
Bullinger	69.6	68.5	52	60.1	79.6	74.1	72.9	63.6	85.1	88.3	62.2	89.10

BERT. The training of BERT on small datasets results in over-fitting which affects the classification performance. The pre-training and fine-tuning of BERT is considered similar to [179].

Table 4.8: Comparison of performance measure (ACC X 100) with BERT

Dataset	BERT	Our model
SRBCT	93.3	95.6
Glioma	86.1	97.9
Plates_faults7	92.7	96.2
dataset32pendigits	93.8	93.5

4.4.2 Statistical Analysis

The statistical test helps in determining whether some given hypothesis is unlikely to occur with an actual observed data. It helps in bridging the gap between numbers and methodologies and forming a quantitative picture of the given process. Friedman test is chosen to compare different techniques and check whether there is any significant difference between them or not. Friedman test [124] is a non-parametric test, and it is preferred over the parametric test as for a parametric test we have to take some data assumptions. It assumes k-different experiment-based different techniques and one variable value is determined multiple times. Each cell value is ranked exclusively, columns symbolize treatments and a row indicates the observed values over the dependent variable and later the ranks of each treatment are added up. The large deviation between the sum value indicates a lower p-value indicating the techniques are different. The number of techniques is 6, therefore the authors decided to go with Friedman test.

Friedman test is a highly effective way to prove the technique score by means of a ranking method. Table 4.9 shows the results of Friedman test on AUC for all comparative oversampling approaches. The conclusion was made after analysis that the D_SMOTE over rules all the other techniques. We have computed the results with degree of freedom as 5 which refers to the number of correlated samples minus 1 significant value ($\alpha=0.05$). The obtained p-value is less than 0.001, that is $5.456957e-07$, which signifies the result is significant for the proposed technique. Higher value obtained after applying the test signifies the higher rank of the technique. With the help of rank, we can find out the significant difference between the techniques.

The omnibus p-value is below the respectable critical threshold of 0.05, so post-hoc pairwise multiple comparison tests are conducted to discern which of the pairs have significant differences. These possible post-hoc tests are conducted: the methods of (1) Conover and (2) Nemenyi.

For the (1) Conover method, the p-value is adjusted in two ways, first according to the Family-Wide Error Rate (FWER) procedure of Holm, and next by the false discovery rate (FDR) procedure of Benjamini-Hochberg. The values indicate that the proposed approach is significantly different than the previously discovered techniques.

The obtained Friedman chi-square statistic is 37.201681. The statistical value of chi-square indicates the deviation over mean ranks of the treatments (different techniques). It's zero signifies that the mean of all treatments is the same, and as this numeric value increases, it indicates the deviation between means of different techniques. We have considered the following hypothesis:-

Technique 1:- Rank of model proposed by KNU [122].

Technique 2:- Rank of model proposed by SMOTE-SF [127].

Technique 3:- Rank of model proposed by MDO [120].

Technique 4:- Rank of model proposed by SMOM [121].

Technique 5:- Rank of model proposed by DECOC [125].

Technique 6:- The proposed approach.

Null Hypothesis:- There is no significant difference between the techniques proposed by [122], [127], [120], [121], [125] and the proposed technique.

Alternate hypothesis:- There is a significant difference between the algorithms [122], [127], [120], [121], [125] and the proposed technique.

Table 4.9 shows the average ranks calculated using Friedman test.

Table 4.9: Average rank by Friedman Test results for AUC measure

Techniques	Name	AUC (Ranks)
Technique 1	KNU	1.26
Technique 2	SMOTE-SF	2.38
Technique 3	MDO	2.68
Technique 4	SMOM	3.9
Technique 5	DECOC	4.1
Technique 6	Our approach	5

4.5 Conclusion

The problem of high dimensionality and problems caused due to imbalanced dataset is solved by the proposed model. The technique to handle imbalance problem avoids the problem of small disjuncts and lack of density by creating samples widely. Another limitation of improving classification performance is high number of features, which we handled by modifying mutation operator. The comparison with different variants of oversampling techniques confirmed the model efficiency. The average mean of best values of AUC is 98.8, G-mean is 97.4 and F-score is 96.9, which is satisfac-

tory performance of the proposed model. It was observed that our model performed best with SVM classifier, when compared to other techniques. To validate our model, we compared the proposed approach with other four state-of-the-art algorithms on seventeen datasets. The experiments are carried out on low-dimensional as well as high-dimensional datasets and our model outperformed other techniques for fourteen datasets. Hence, our model can be generalized for low-dimensional as well as high-dimensional datasets. The comparison with the latest classification algorithm BERT on four datasets, where our model outperformed on three datasets and almost performed equally well on fourth dataset. It made our model more promising for improving accuracy with our new techniques. After observation and statistical analysis, it can be said that the classification performance is improved with the proposed feature selection and over-sampling technique. The proposed over-sampling method with a suitable classifier can be used for other applications.

Our new algorithm is efficient to use for imbalanced dataset. The new technique to handle high dimensionality with the help of feature selection is helpful in improving classification performance. In the next chapter we will focus on other technique of tuning parameters of optimization algorithm and classifiers to improve text classification performance.

Publication

The work discussed in this chapter is published in:

Khurana, A, and Verma, O.P. (2020). Optimal Feature Selection for Imbalanced Text Classification. *IEEE Transactions on Artificial Intelligence*.

doi: 10.1109/TAI.2022.3144651.

CHAPTER 5

FINE TUNED GRASSHOPPER OPTIMIZATION ALGORITHM FOR OPTIMAL TEXT CLASSIFICATION

5.1 Introduction

Hyper-parameter optimization is the process of determining the optimal parameter value before beginning the training process. The main objective of hyper-parameter tuning is to generate the optimum model for a given problem. Hyper-parameter can be obtained using several ways. One way is to provide parameters to the objective function to estimate the loss [25]. Another way is to obtain the generalized optimization performance of the model is the use of cross-validation [26]. For the past few years, the focus has been on applying the optimization techniques namely Grid Search (GS) [27], Bayesian Optimization [28], Gradient Based Optimization [29] and Random Search (RS) [26]. Feature selection algorithms emphasize on the selection of important features, hence improve model accuracy [30]. There are many nature inspired algorithms which are used to optimize features, namely: Genetic Algorithm [31], Particle Swarm Optimization [12], Grey Wolf Optimization (GWO) [32], Multiverse Optimizer (MVO) [33] and many more. The rise in classification accuracy relies on the selected number of parameters that need to be tuned and regions of search space of the chosen algorithm. Thus, in hyper-parameter tuning, the main problem lies is to choose a different approach from the existing state-of-the-art approaches. Grasshopper Optimization Algorithm is a new optimizer proposed by [8]. GOA mimics the behavior of grasshoppers. In a study [139], researchers proposed a simultaneous approach for feature selection and parameter optimization using classical GOA. We proposed tuned GOA for feature selection which helps to improve classification accuracy.

For the classification problem, supervised classifiers are used. SVM [136] and K-NN [27] are familiar, high-powered and influential classifiers. For fine tuning, the parameters of classifiers, we chose the Random Search technique. The advantage of Random Search over Grid Search is that fewer combinations are required to evaluate the solution. Hence, high-dimensionality does not cause a problem for the technique. The main motivation of the study lies in the fact that tuning of GOA improved the classification

performance, generating optimal feature subset. We perform classification performance using tuned SVM and K-NN classifiers. All datasets are run for 200 iterations using 10-fold-cross validation. To verify our model we compared the propose model with state-of-the-art models.

5.2 Theoretical Foundations

5.2.1 Grasshopper optimization

The Grasshopper Optimization algorithm (GOA) [8] lies in the domain of nature-inspired algorithm, established on the mimic behavior of grasshoppers. Following three components affects the flying path of grasshoppers: Gravity power (G_j), Air advection (A_j) and Social interaction (S_j) where j is the j -th individual. The mathematical expression of the algorithm is represented in Equation 5.1.

$$X_j = S_j + G_j + A_j \quad (5.1)$$

Grasshopper Optimization Algorithm social interaction is defined as in Equation 5.2.

$$S_j = \sum_{i=1, i \neq j}^N s(d_{ji}) \hat{d}_{ji} \quad (5.2)$$

where, d_{ji} is the distance measure between the j -th and i -th grasshopper, calculated as $d_{ji} = |x_i - x_j|$, s is a function which refers to the power of social forces mentioned in Equation 5.3, and \hat{d}_{ji} is the unit vector between j -th and i -th grasshopper.

$$s(r) = f e^{-r/l'} - e^{-r} \quad (5.3)$$

where f refers to the intensity of attraction and l' represents the length scale. l' is the parameter which we selected for tuning in this research. The value of this function is the cause of attraction and repulsion in grasshopper.

The component G in Equation 5.1 is calculated as in Equation 5.4.

$$G_j = -g \hat{e}_g \quad (5.4)$$

where, g is gravitational constant and \hat{e}_g is a unit vector acting towards center of the earth.

The component A in Equation 5.1 is calculated in Equation 5.5.

$$A_j = u \hat{e}_w \quad (5.5)$$

where u denotes constant drift and e_w denotes unit vector in the wind direction. Substituting values of all components in Equation 5.1 and it constitute into Equation 5.6.

$$X_j = \sum_{j=1, j \neq i}^N s(|x_i - x_j|) \frac{x_i - x_j}{d_{ji}} - g\hat{e}_g + u\hat{e}_w \quad (5.6)$$

where, $s(r)$ is defined in Equation 5.3 and N determines number of grasshoppers.

The swarm model is adjusted and designed, which generates the optimized solution to an algorithm. The mathematical model cannot be directly used for optimized solutions as grasshoppers reach comfort zone quickly. Hence, for optimization problems, to be solved by the algorithm is defined in Equation 5.7.

$$X_i^y = C \sum_{j=1, j \neq i}^N C \frac{ub_y - lb_y}{s} s(|x_i^y - x_j^y|) \frac{x_i - x_j}{d_{ji}} + \hat{T}_y \quad (5.7)$$

where, ub_y refers to the upper bound value in the y -th dimension, lb_y refers to the lower bound value in the y -th dimension, \hat{T}_y is the value of y -th dimension in the best solution of target solution and C is decreasing co-efficient to shrink the repulsion area, attraction area and comfort area. s is the function defined in 5.3. The c parameter is responsible for the convergence of swarm so as to reach to the target.

The Equation 5.7 indicates the updated next position based on the current position of a grasshopper. In GOA algorithm, assume that the best solution obtained so far is the target solution. During the interaction between the grasshoppers and finding the target, the best solution gets updated if a better solution is found. The best solution is selected with respect to the current best position of a grasshopper and other search agents position. In Equation 5.7, the use of two C have different meanings. The most left C is used as inertial weight parameter used in PSO. The second C is responsible for decreasing the different zones namely; repulsion zone, comfort zone and attraction zone. The parameter C is calculated in Equation 5.8.

$$C = C_{max} - l \frac{C_{max} - C_{min}}{L'} \quad (5.8)$$

Where, L' signifies the maximum number of iteration, l is the current iteration, $C_{max}=5$, $C_{min}=0.00005$. So, in this work we have tuned l' from Equation 5.3 and L' , C_{max} , C_{min} from Equation 5.8.

5.2.2 Classifiers

For tuning, the two classifiers are selected namely: K-NN and SVM. We performed hyper-parameter tuning using a Random Search technique to improve the classification accuracy. The K-NN approach is applied to various research areas like pattern recognition [191] and text categorization [192]. The K-NN technique is based on the training of k instances present in the training set. These k instances will classify d documents similar to them (k -nearest neighbor). The majority voting method is used to classify document d , depending on the common category in the neighboring documents.

The improvement in the algorithm depends on the distance of a neighbor, and close neighbor influence the category of classification. The weight of the document is the main criteria for calculating the similarity score in the testing set. If testing document d is similar to more than one k nearest neighbor, then the resulting weight is calculated by summing up the weights of all documents of the similar category. The sorting of similarity score is performed, and the top ranked category is the final category of the test document [193]. For the efficient working we need to know the appropriate value of k [194]. In our tuning method of random search the best value is obtained at $k=7$. The other parameters which yields best value are p (Euclidian distance)=2, Weight=distance.

The other classifier chosen for classification is SVM. In this research, we consider the Gaussian kernel for SVM. The advantage of this kernel is that it can handle more number of kernels when compared to other kernels[195]. The random search technique tuned cost (c) and gamma (γ) parameters. The range for cost hyper-parameter is 2^{-2} to 2^{15} and gamma is 2^{-15} to 2^5 . We have performed 200 runs for each classifier with 10-fold cross-validation. The tuning optimizer is very much dependent on the nature of the problem. According to the literature survey, it can be seen that random search is a good tuning model in most scenarios. Process simplicity and computation time are also advantages of using this model. On comparing different techniques, namely, manual search, random search, grid search, evolutionary algorithms and Artificial Neural Network, random search performed the best and hence was selected by authors as the model to tune the parameters of the classifiers.

5.3 Methodology

As stated before, this research aims to know the effectiveness of the Random Search technique for tuning GOA for feature selection, and for hyper-parameter optimization of the two classifiers. We employed other feature selection techniques to verify the efficiency namely: classical GOA, MVO and GWO. We have used the 10-fold-cross validation method for generalizing the result. Accuracy and ROC measure was used

to evaluate the working of the proposed model. In this research, we have optimized the parameter of the Gaussian kernel [195], (γ), of the SVM classifier. The parameter of SVM, cost (C), is also tuned. The Accuracy and ROC curve values are used as the fitness function to measure the effectiveness. Higher values indicate the propitious hyper-parameter values. Many experiments are employed using meta-heuristic techniques for feature selection and parameter optimization [136], [139]. Tuning parameters helps to improve classification performance [27]. Tuned GOA is formalized to produce a reduced dataset with all essential features. Reducing the dataset helps to improve the classification accuracy. The reduced dataset is streamed to the tuned classifiers to generate the classification results. The model starts working after finalizing the parameter range for tuning of GOA and classifier. The parameters tuned in GOA algorithm using meta-optimization technique are present in Equation 5.3 and 5.8. The process of data normalization is performed on selected features. After pre-processing, the process of tuned GOA for the selected features and tuned parameters of classifiers is prescribed as:

1. Initialization of random population of all candidate solutions which consist of the parameters, corresponds to the selected features of the dataset as discussed in Equation 5.7.
2. The operators used for reproduction for GOA are pertained to all individuals for the formation of new population.
3. The best individuals are determined with the help of fitness function.
4. The termination of the search process after setting the maximum iterations. Then, tuned GOA models produces the best classifier parameters alongside features subset that produces highest fitness function.
5. The subset of features is tested with the developed tuned models.

Parameters that are tuned, range and best values obtained are mentioned in Table 5.1. The flowchart of the methodology is explained in Figure 5.1.

Table 5.1: Tuning Parameters

Technique	Parameters tuned	Range	Best value
GOA	l'	0-4	3.1
	C_{min}	N/A	0.00005
	C_{max}	N/A	5
	L'	N/A	200
K-NN	k	1-10	7
	p	1-3	2
	weight	uniform	distance
SVM	c	2^{-2} to 2^{15}	2^{11}
	γ	2^{-15} to 2^5	2^2

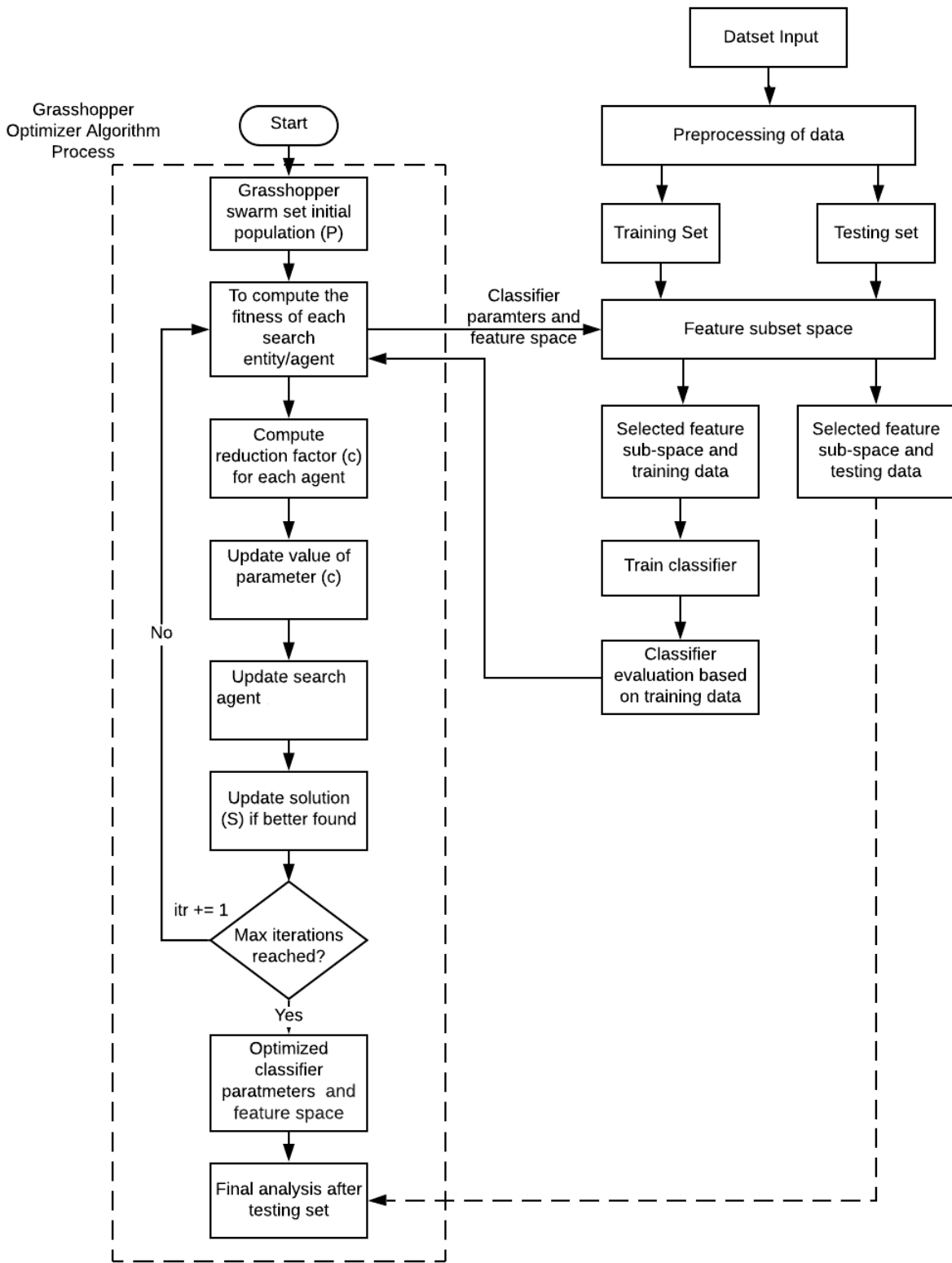


Figure 5.1: Flow process of the model

The Algorithm followed is explained in Figure 5.2.

Parameter(s): P - Initialize control parameters: tuned GoA parameters, tuned classifier parameters, upper band(ub), lower band(lb), maximum number of iterations(T), desired number of features(f)

Output: O - Target solution(S)

//Grasshopper algorithm parameters to be tuned

Meta-optimization technique applied for tuning

C_{min} : 0.00005

C_{max} : 5

l: Range [0-4]

L: 200

$j \leftarrow 0$

Initialize X_i : set of random vectors with initial population, where $i = 1, 2, \dots, n$, each member with the dimension of $1 \times f$ having integer elements in the valid interval of [lb,ub].

while $j < T$ **do**

 Compute c via below equation

while *Member-iterated* < *Population-size* **do**

 Update X_i using Equation 5.7

 Round-up the vales of X_i to the most nearest integer

 Check the outlier values and modify them with random values lies in the provided valid interval.

 Evaluate the duplicate features and modify them.

 Calculate classification error rates of both the classifiers (tuned[K-NN and SVM]) and analyze the fitness values.

//Parameter tuning of classifiers is done using random search technique

end

 Update S if better solution is found in population.

$j = j+1$

end

return S

Figure 5.2: Tuned Grasshopper Algorithm used for feature selection

5.4 Experimental Setup

5.4.1 Dataset

In our research, we have used five multi-class datasets from the UCI repository [196]. The description of the datasets is given in Table 5.2. It can be seen that the datasets we have chosen are varied datasets in terms of number of features, number of classes and number of instances. Our priority focused on high-dimensionality datasets.

Table 5.2: Description of multi-class dataset

Dataset	Features	instances	Classes
Glass	9	214	6
Cleveland	13	303	4
Arrhythmia	297	452	16
Libras	90	360	15
Teaching Assistant evaluation	5	151	3

For training/testing method, we have applied the 10-fold-cross validation. Each dataset is performed with 200 runs to evaluate the results. Teaching Assistant Evaluation is referred as TAE in future Tables.

5.4.2 Objective functions

To evaluate the performance of the proposed model we have computed our results on two objective functions namely: Accuracy and AUC. It was defined in section 3.2.3 and section 4.3.2 respectively.

5.5 Results

In this research, we have implemented all the algorithms on Windows 10 64-bit OS, Intel® Core™ i5-5200U CPU @2.20GHz processor. Results are computed on jupyter notebook using python. All the algorithms are set to run for 200 iterations with 10-fold-cross validation. In Table 5.3, we compared the Accuracy measure of our model with other feature selection techniques namely: classical GOA, MVO, GWO and Grid Search. We have considered two classifiers tuned K-NN and tuned SVMk. It was found that datasets like libras and arrhythmia which are high-dimensional datasets produced accuracy 15-20% higher, datasets like glass, cleveland and TAE shows arise in accuracy for 10 to 15% higher. The result analysis is:

1. In glass dataset, which is low dimensional dataset, tuned SVM is proven to be best, with increase of around 25% higher when compared to lowest score. The

lowest score was obtained with tuned K-NN with GWO and tuned SVM with Grid Search. Tuned K-NN with tuned GOA is also best when compared to other tuned K-NN scores as SVM is good with low dimensional datasets. But, tuned SVM is recommended to use with tuned GOA.

2. In cleveland dataset, our approach, tuned K-NN with Random Search with K-NN classifier is proved best. Tuned K-NN is producing 66.2% Accuracy which is 14% higher than the lowest tuned K-NN with MVO. The performance of tuned K-NN is better than tuned SVM due to random training chosen by the algorithm for this low dimensional dataset.
3. In arrhythmia dataset, tuned SVM with tuned GOA produces highest score when compared to the lowest score produced by tuned SVM with Grid search. Although, the lowest tuned K-NN values are produced with GOA and highest tuned K-NN values are produced with tuned GOA. In this dataset, our approach of tuned SVM with tuned GOA is producing 18% to 23% higher Accuracy.
4. In libras dataset, tuned SVM with tuned GOA is producing 90.2% Accuracy which is 5% to 6% higher than the Accuracy produced by other techniques. This is high-dimensional dataset with large number of instances. Tuned K-NN with tuned GOA is better when compared to other tuned K-NN techniques.
5. In TAE dataset, tuned K-NN with tuned GOA is producing best results with 71.3% Accuracy. The approach is improving results with 8% when compared to lowest values of tuned K-NN with other techniques. Tuned K-NN is producing better than tuned SVM with tuned GOA as classes are clearly separable in the dataset.

The varied datasets is used for verifying the generalization use of the model. Higher Accuracy indicates the better classification performance with optimal parameters. Tuned SVM with tuned GOA is recommended for high-dimensional datasets, while tuned K-NN with tuned GOA for low dimensional datasets.

Table 5.3: Accuracy results for various techniques

Dataset	GOA		MVO		GWO		Grid search		Our approach	
	K-NN	SVM	K-NN	SVM	K-NN	SVM	K-NN	SVM	K-NN	SVM
Glass	61.8	67.1	64.7	64.2	50.9	60.2	56.4	55.1	78.3	83.2
Cleveland	52.1	54.7	50.9	47.0	53.4	56.6	54.1	52.9	66.2	60.6
Arrhythmia	55.0	68.3	64.1	71.4	60.6	68.0	62.4	61.6	76.1	82
Libras	76.4	85.2	75.3	85.8	79.7	84.1	72.1	76	86.6	90.2
TAE	62.9	57.6	68.8	54.1	69.9	54.4	55.6	49.4	71.3	60.3

In Figure 5.3, AUC plot of glass dataset is represented. All the techniques are represented in different colours. Tuned SVM with tuned GOA is represented with red color, Tuned SVM with tuned K-NN is represented in green color, tuned GOA with SVM is shown with dark blue color, tuned SVM with tuned MVO in sky blue color, tuned K-NN with tuned GOA in pink color, tuned K-NN with tuned MVO in yellow color, tuned SVM with tuned GWO in grey color and tuned K-NN with tuned GWO in black color. It can be seen that AUC for glass dataset produces similar results with both tuned classifiers. Tuned GOA and SVM with 85% score shows rise of 3% when compared to tuned SVM with classical GOA technique and 14% rise when compared to tuned SVM with GWO technique.

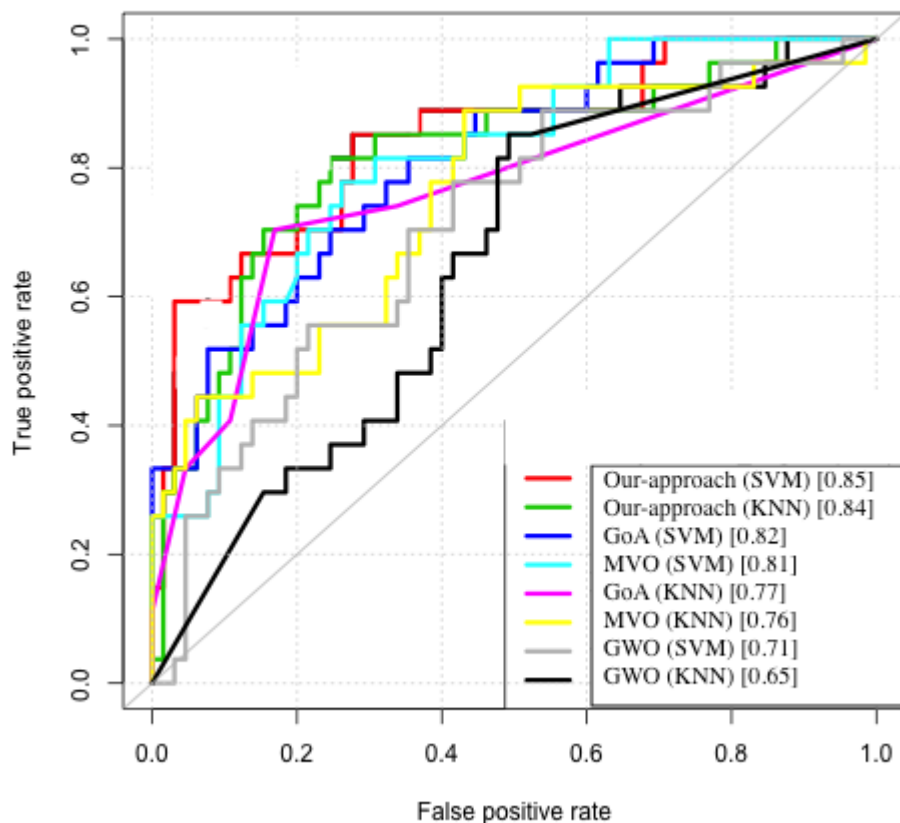


Figure 5.3: AUC plot of Glass dataset

In Figure 5.4, represent the AUC plot for arrhythmia dataset. Tuned SVM with tuned GOA is represented in red color, Tuned SVM with tuned K-NN is represented in green color, tuned GOA with SVM is shown with dark blue color, tuned SVM with tuned MVO in sky blue color, tuned K-NN with tuned GOA in pink color, tuned K-NN with tuned MVO in yellow color, tuned SVM with tuned GWO in grey color and tuned

K-NN with tuned GWO with black color. The dataset is high-dimensional dataset and producing good results with our model. AUC score is 90% with tuned GOA and tuned SVM. The rise of score in our model is from 9% to 25%. Our model perform best with high-dimensional as well as low-dimensional datasets. 10-fold-cross validation method is applied for computation of all the results.

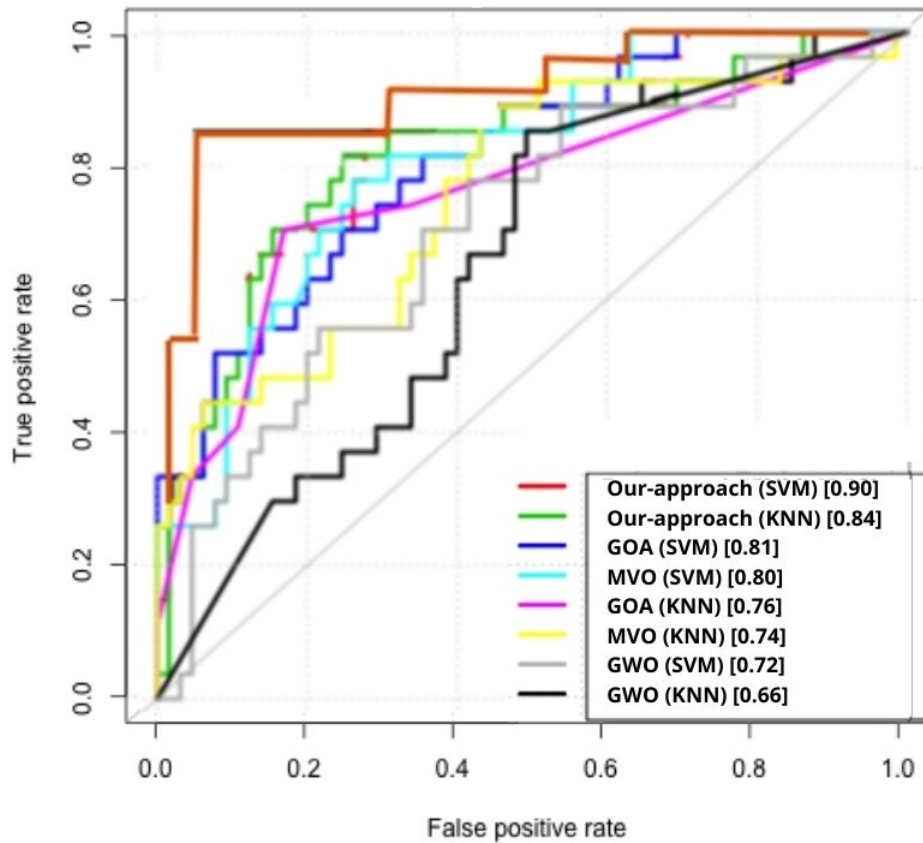


Figure 5.4: AUC plot of arrhythmia dataset

In Table 5.4 comparison of the proposed technique is performed with other proposed researches [139], [136] and [135]. We concluded that our research produced better results than all the techniques with good Accuracy rise.

Table 5.4: Comparison of Accuracy results

Dataset	[139]		[136]		[135]		Our research	
	K-NN	SVM	K-NN	SVM	K-NN	SVM	K-NN	SVM
Glass	68.2	68.5	73.3	70.8	66.2	65.3	78.6	83.5
Cleveland	50.2	53.3	51.6	57.7	40.1	51.0	66.4	60.6
Arrhythmia	70.5	66.5	56.3	50.3	48.1	45.5	76.1	82.3
Libras	89.2	87.5	89.5	90.0	75.6	70.1	86.6	90.2
TAE	50.1	56.1	66.2	70	67.3	60.2	71.3	60.3

5.6 Conclusion

This study propose a novel approach of tuned GOA with tuned classifiers. The tuned GOA is used for feature selection to reduce the problem caused due to high-dimensionality, and increase the classification performance. The Random Search technique is used for the parameter optimization of the two classifiers- K-NN and SVM. The Gaussian kernel of SVM is used for hyper-parameter optimization. We have tested the proposed model on 5-real-world datasets from UCI by computing classification accuracy and AUC measures. In the different performance measure, our model performed better than the other state-of-the-art techniques. The best accuracy was obtained with a high-dimensional dataset, although the proposed model worked well with low-dimensional dataset as well.

The performance of text classification is improved by performing various pre-processing techniques. In the next chapter, improvement in the text classification performance is achieved using a transfer learning framework. The transfer learning framework works on trained model and predicts classes for new data. A new framework for heterogeneous domain adaptation is introduced.

Publication

The work discussed in this chapter is published in:

Khurana, A., and Verma, O. P. (2020, December). A Fine Tuned Model of Grasshopper Optimization Algorithm with Classifiers for Optimal Text Classification. *IEEE 17th India Council International Conference (INDICON)* (pp. 1-7). IEEE.

CHAPTER 6

OPTIMAL HETEROGENEOUS DOMAIN ADAPTATION FOR TEXT CLASSIFICATION

The prime challenge of unsupervised symmetric heterogeneous cross-domain adaptation is to train the source domain and apply the trained knowledge to the target domain. Most of the existing algorithms for unsupervised transfer learning create the subspace of the source domain features and target domain features for training purpose. It is a computational extensive process as most of the techniques require labeled source data. Many techniques also suffer from the loss of originality of features in both domains. This chapter aims to consider the feature vectors of both the source and target domain for training the data based on similarity of exemplar (feature) vectors of different instances, known as Instance Similarity Feature (ISF).

6.1 Introduction

In this digital era, where knowledge content is increasing, managing the data becomes difficult. The abundant amount of data makes the classification of text, training the dataset, data analysis, and many more activities a tedious task [153]. Various models are developed that aim to transfer the knowledge available in source domain to classify the text in target domain [40]. The transfer learning approach is already developed and used in many other areas like object recognition [41] [42] event detection [43], classification of images [39] and text categorization [44]. The basic ideology is to learn from a model that has been already worked upon to transfer the knowledge from source to the target set, but there should be some connection/threshold similarity between the features/domains of the source and target set. In the literature, several transfer learning techniques have been developed for domain adaptation. Techniques like manifold embedded distribution alignment, transfer component analysis, transfer joint matching, have been proposed by researchers in [46]. However, many of the techniques in literature suffered from limitations like optimizing the marginal distribution [197], maximizing the domain separation error [198], preserve the originality of source domain [40] and degenerated transformation of feature space [45]. Sometimes, it is difficult to study

and work on specific datasets and domains due to the lack of available stable data. If the problem is new and there is confined knowledge about the data, then there is no historical data to learn about the domain [45]. Hence, to bridle the limitations of conventional existing techniques, transfer learning techniques are proposed.

When the data pattern and feature learning is a difficult task due to high dimensional dataset [2] sub clusters of data instances are formed to begin learning in the form of small clusters. This technique is also known as divide and conquer. The target object instance is mapped over the clusters, and similarity vectors are formed to decide on the target instance. But many times it is difficult to form cluster categories due to the complex distribution of source object instances. Instead of working on explicit sub-categorization/clustering, the multimodal distribution can also be learned and worked by an exemplary vector-based approach. We propose to use the similarity with a set of exemplars as instances for classification. Contrasted and the low-level highlights removed legitimately from the sensor information, the similarity vectors include a center level component that has semantic implications. A classifier is prepared utilizing the similarity vector, just like the significant level model prepared by the sub-classification scores, accepting that every model is a subclass classifier, and the comparability esteem is the classifier's score. These similarity vectors are used alongside with initial source vector information to train the classifier in a more advanced manner so that the classifier at the end, is able to handle and work on the target space.

The proposed model works with symmetric heterogeneous domain adaptation for homogeneous transfer learning. Choosing the optimal set of exemplary vectors is a complex and rigorous process. Initial studies opted for random selection process, but it did not able to do proper justification on the method. Few other alternatives were proposed by [198] [46] but, not very optimal one. In our proposed work, a learning policy with k-means centroid model is proposed. Further target instance space is spread over the principal components (for better visualization in case of high dimensional data), then K-means clustering is applied to know more about intra and inter relationships for data instances. The centroid are initially selected as the exemplary target vectors. This selection is done within a window space, as the data is large and usually distributed unevenly. Hence, it is difficult to process all of it at once. These vectors are used to develop the similarity mapping. Selecting the optimal value for parameter k and the window size is deduced from the learning policy. Learning policy consists of the old centroid, current centroid and the cost. Cost function is computed using euclidean distance between the old and current centroid and the Accuracy parameter value at the end of each classifier computation iteration. This is to learn about the regions in target space which dominate the output, hence best suited to be chosen as exemplary vectors. We have proposed two models, ISF and KISF, for transfer learning techniques. KISF technique uses K-

Table 6.1: Predictive Results summary of F1-score

Notation	Description
x and y	feature vector
D^s	Source Domain
D^t	Target domain
ω	weight vector
E	Exemplar vector set
$\text{sf}()$	Similarity function
I_s, I_t	Input vector subspace
$I_{s,\perp}$	Orthogonal subspace
M_ϵ	Matrix for exemplar vectors

means clustering to overcome the problem observed in ISF technique. To reduce the problem of dimensionality, nature-based optimization techniques, namely: Ant Lion Optimization (ALO), PSO and BBO are used. For classification purpose, we have used three classifiers LR, RF and SVM. The new hybrid models are proposed with feature selection namely; K-means Instance similarity Feature with ALO (KISFA), K-means Instance Similarity Feature with PSO (KISFP) and K-means Instance Similarity Feature with BBO (KISFB). The comparison of classifiers is performed using F1-score and Accuracy. The comparison with other eleven techniques proposed by the researchers was implemented to explore the performance of our model for symmetric heterogeneous domain adaptation.

6.2 The Proposed framework- KISF

6.2.1 Notations

In this section, we will elaborate the use of some notations. We have used lower case alphabets and upper case alphabets for the source and target domain respectively. We have also used the similarity function (). All notations are listed in Table 6.1.

6.2.2 Instance Similarity Features (ISF)

6.2.2.1 Similarity exemplars

Similarity learning is a field in Artificial Intelligence that relates with supervised machine learning. It is associated with classification and regression. However, its main objective is to learn a similarity function that calculates how similar or comparable any two occurrences or instances are. Similarity learning has broad applications in various systems such as ranking, recommendations, visual identity tracking, speaker verification and facial recognition. In the field of machine learning, artificial intelligence, and pattern recognition, we define a feature as an individual quantifiable characteristic or

property of an occurrence being measured. Selecting independent, informative and discriminating features is a significant step for efficient algorithms in recognizing patterns, their classification and regression.

In Machine Learning (ML), representation of feature vectors is in the form of numeric or symbols known as features. It is the mathematical representation of an instance/object and can be analyzed easily. The numerical representation of ML algorithms is necessary to process the features and perform statistical experiments. The features have gained importance in recent times in the era of artificial intelligence especially in the areas of ML and pattern recognition/processing. In classification, feature vectors are same as correspondent vectors used by linear regression for statistical experiments for explanatory variables. Explanatory variable are those variable, which can be independent variable but are not significantly statistically independent.

The introduction feature vectors in ML algorithms helps the practitioners in practical analyses and representation of object in mathematical way more effectively. The analysis with the use of feature vectors becomes easy and significant as many techniques are introduced for the implementation. Euclidean distance is mostly used metric for the comparison of the given feature vectors of two instances or objects. In the field of image processing, pixels in image can be used to represent features, gray-scale intensity, color and edges and many more. Text classification features can be number of occurrences of word in a text. The features in text classification are based on the application like sentiment analysis, spam filtering and other application. Features can vary from application to application. In speech recognition techniques, some generally used features are noise ratio, noise level, sound frequency variation and much more. The use of feature vectors is popular for image processing area, when focusing in the analysis of an attributes of an image. There are many existing algorithms that currently work on feature vectors in classification problem namely: Neural Network techniques, statistical techniques and K-NN algorithms. To support our study, we evaluate a problem on binary classification, in which each of the input is represented in the form of feature vector x and $y \in \{0, 1\}$ is the true label of x . Binary Classification, as the name suggests is the task of classifying elements into one of two classes/groups. Some applications of binary classification are and not limited to: sentiment classification, spam detection, product reviews, and cancer detection.

We consider a set of labelled occurrences from one point which we call as ‘source domain’. This is denoted by Equation 6.1.

$$D^s = (x_i^s, y_i^s)_{i=1}^N \quad (6.1)$$

where, N is the total number of instances in source domain. We also consider a set of unlabeled occurrences from another point, which we call as ‘target domain’. This is

shown in Equation 6.2.

$$D^t = \{X_j^t\}_{j=1}^M \quad (6.2)$$

where X consists of x_1, x_2, \dots, x_n , signifies the features in the target domain and M represents the total number of instances in target domain. In order to train a linear classifier, we assume a general state of a weight vector, that is, ω as such that x is considered likely to be 1, if $\omega^T x \geq 0$. Here, T refers to transpose of the weight vector. The basic mechanism will be to just train a classifier from D^s . In unsupervised heterogeneous domain adaptation, the main aim lies in the operating method by focusing on both D^s and D^t to grasp a righteous ω to test the target domain. To make unsupervised domain adaptation feasible, we have assumed that the source and the target domains belong to the related space and their features have enough similarity.

Our method works as follows. The first step in our method is the random selection of a fraction of target occurrences from D^t and worked on the instances to normalize them. Thus, the resulting vectors are termed as exemplar vectors. These are symbolized and shown in Equation 6.3.

$$E = \{\epsilon^k\}_{k=1}^K \quad (6.3)$$

where ϵ is a single instance of exemplar vector. In the next step, after calculating the similarity with ϵ^k , the transformation of each instance x' has been performed from the source domain and converting it into a contemporary new feature vector. It is represented by Equation 6.4

$$g(x) = [sf(x, \epsilon^1), \dots, sf(x, \epsilon^K)]^T \quad (6.4)$$

where, T signifies the transpose and $sf(x, x')$ can be defined as similarity function between x and x' . Here, x' belongs to the exemplar vector set E . In our method, the dot product is used as sf^2 . We use Equation 6.4 to transform each labeled source domain occurrence into a K -dimensional vector. Once this transformation is completed, the combining of the new feature vector begins with the original source feature vector. The manipulation has been performed by combining the feature vectors of all provided labeled source occurrences for the purpose of training a classifier. The same process is followed by target domain as well. In the target domain, each target occurrence must also conjoin the new K -dimensional instigated feature vector. The exemplar vectors for the method are chosen erratically from the accessible target instances.

6.2.3 Heterogeneous Domain Adaptation

In this section, we describe the training of our model.

6.2.3.1 Learning in Target Space

The research in [197] [199] indicated that the option of unsupervised domain adaptation is possible in a way by combining the new learned common weights for target features with the simultaneously training of related trained features. The establishment of induced feature rendition is performed in the same way. Firstly, we examine the study suggested by the researchers in [199] [197]. It was observed that for NLP tasks, the input vector space ' I ' is usually high-dimensional. However, the existing space where input vectors are present, they may be of lower dimension due to the strong feature dependency that usually belongs to NLP techniques. For instance, when we specify binary features from the same pattern such as the preceding word; they are mutually exclusive. Additionally, for the source and target domains, the actual low-dimensional spaces are generally different, that can be the reason of distributional difference between the domains and domain-specific features. The subspace I_s is defined as the (lowest dimensional) fraction of I traversed by domain input vectors from all sources. We define subspace I_t in a similar way. We designate $I_{s,t} = I_s \cap I_t$, as a shared subspace between the two domains. The $I_{s,\perp}$, is defined to be the subspace that is orthogonal to $I_{s,t}$. The combination of $I_{s,\perp} + I_{s,t}$ is I_s . Similarly, it can be specified $I_{\perp,t}$. Fundamentally $I_{s,t}$, $I_{s,\perp}$ and $I_{\perp,t}$ are mutually orthogonal shared subspaces that are domain-specific.

Three subspaces that were discussed above are projected in terms of the input vector x in the Equation 6.5.

$$x = x_{s,t} + x_{s,\perp} + x_{\perp,t} \quad (6.5)$$

Correspondingly, a linear classifier ' c ' can also be decomposed into $c_{s,t}$, $c_{s,\perp}$ and $c_{\perp,t}$, and or a basic technique where c simply learns from D_s . Assume $c_{\perp,t}$ as 0, the learned component. If the component of any source instance $c_{\perp,t}$ is 0, hence in training error, there will not be any reduction by any non-zero $c_{\perp,t}$. Furthermore, any nonzero $c_{s,\perp}$, that model trained from D_s , would not have impact on target domain because $x_{s,\perp} = 0$ for all target instances. This implies that ω trained from D_s , only $c_{s,t}$ is some use for domain transfer.

The researchers in [197] [157] propose that an untagged target instance can aim to "correlate" the training of $\omega_{\perp,t}$ with $\omega_{s,t}$. It was observed that only use of induced feature representation that does not include original feature vector appended to it, the same above mentioned situation can be achieved. Firstly, the matrix M_ϵ is defined, where exemplar vectors from ϵ are taken as column vectors. Hence, $g(x)$ defined in Equation 6.4, will be formulated as $M_\epsilon^T x$. Let, the linear classifier be ω' that is trained from the reconstructed labeled data. The prediction done by ω' is based on $\omega'^T M_\epsilon^T x$, that is identical as $(M_\epsilon \omega')^T x$. This confirms that the trained classifier ω' for the induced features correspond to a linear classifier $\bar{\omega} = M_\epsilon \omega'$ for the original features. It is easily

visible that $M_c \omega'$ is basically $\sum_k \omega'_k e^k$, which is denoted by E, a linear coalescence of vectors. The 'e' is abstracted from I_t . The Equation that is formulated is shown in Equation 6.6.

$$.e^k = e_{s,t}^k + e_{\perp,t}^k \quad (6.6)$$

Therefore, we have two points that can be noted from the Equation 6.7, 1) The trained classifier $\bar{\omega}$ does not contain any integral component $I_{x,\perp}$ in the subspace. This is satisfactory for target domain as no such a integral was useful. 2) The trained classifier $\bar{\omega}_{\perp,t}$ improbable be zero, as its training is “integrate” with the training of $\bar{\omega}_{s,t}$ along with ω' . Consequently, only those features were picked up that were target-specific or correspond to applicable familiar features.

$$\bar{\omega} = \sum_k \omega'_k e_{s,t}^k + \sum_k \omega'_k e_{\perp,t}^k \quad (6.7)$$

Practically, to attain effective results, the induced features are embedded with original features. It may find implausible as these results in an expand feature space rather than restricted feature space. In this research, the typical L2 regularizer [200] is owned during training. Hence, there is an impulse to transfer the mass to the secondary induced features, and not let the value decreased to zero. It results in non-sparse space solution consist of features. The previous studies have justified the combined use of the new induced features with the original features [201] and marginalized denoising auto-encoders [202], where researchers worked on semi supervised techniques.

6.2.3.2 Reduction in Domain Divergence

The studies in [203] regarding domain adaptation focus on the features in the hypothesis space, to decrease the error occurs in source domain. It was found that their study was not able to separate the source and target instances. In our work, if we continue to use the induced features only, then it was observed that the hypothesis space excludes the $I_{s,\perp}$. It makes the model resistant to distinguish clearly between source and target domain instances. To confirm this finding, the three feature representation error is reported in Table 6.2, these are:

- 1) $\hat{\epsilon}_s$ - signifies the training fault occurs in the source domain.
- 2) When the model is trained to dissociate the source and target instances, then focus is on classification error.
- 3) Once the classifier is trained on source domain instances, the focus is the error

occurs in the target domain and denoted by $\hat{\epsilon}_t$.

Table 6.2: Error reduction Table

Features	$\hat{\epsilon}_s$	Domain separation error	$\hat{\epsilon}_t$
Original	0.000	0.011	0.283
ISF-	0.120	0.129	0.315
ISF	0.006	0.062	0.254

ISF- denotes that only induced ISF are included.

While ISF denotes the combined space of feature vectors of source and target domain. The results depict that ISF have achieved relatively low source domain error, $\hat{\epsilon}_s$, and hence, increases the domain separation error. These two factors that reduction in source error and increase of domain separation error will lead to a reduction in target error $\hat{\epsilon}_t$.

6.2.3.3 Exemplar Vector Selection

Until now, all the exemplar vectors in our method are randomly chosen from the target instances, and one important assumption we make for these exemplar vectors is that they contain some target-specific features. However, this assumption cannot be guaranteed to be satisfied by simply choosing random instances from the target domain. For example, in an extreme case, if all the chosen instances did not contain any target-specific features but only common features, i.e., $I_{\perp,t} = 0$, then our method is not able to learn an appropriate weight vector for target-specific features. Therefore, we examined that random selection of exemplar vectors may lead to two potential limitations:

- 1) It is possible to choose some “poor” exemplar vectors that may only contain common features.
- 2) The result of our method might be relatively unstable, since it is highly dependent on the proportion of “poor” exemplar vectors.

To solve these two limitations, we further propose to apply a clustering approach to cluster all the target instances into K clusters, and then treat the K cluster centroids as our exemplar vectors. This is our second proposed model KISF. Specifically, we first employ the well-known K -Means clustering algorithm to perform clustering on the available target instances, where Euclidean distance is used to measure the instance similarity. Next, the resulting K -cluster centroids are utilized to form our exemplar vectors and Equation 6.4 is reformulated as Equation 6.8:

$$E = \{c_k\}_{k=1}^K \quad (6.8)$$

where $c(k)$ refers to the centroid of the k th cluster. Here, K are the total number of clusters. The chosen K-means clustering technique is an hard partitioning unsupervised technique. The goal was to create K clusters from the target space based on the above objective function 'E'. According to the minimum distance of the vectors, the particular instance is assigned to the cluster. After all the vectors are clustered, the mean of all cluster is calculated, which belongs to a particular cluster. For the next iteration, the calculated mean value is assumed as a new centroid of that cluster. This process is repeated until the present centroid matches with the previous iteration centroid. The prime goal of K-means is to cluster the relatable exemplary vectors with the use of an objective function.

6.2.4 Optimal Feature Selection

From the survey [153], feature selection improves the classification performance. It re-weights the features and consider only relevant features. As we are working on high-dimensional features and mapping them into low-dimensional subspaces, each feature must be different and independent of values. After analyzing the results of both of the algorithms (ISF and KISF), we focused on the problem of high dimensionality. Hence, the use of nature-based optimizer leads to amplify the results of the K-means. We have used nature-based optimization techniques, namely: ALO, PSO [204] and BBO [2] for the feature selection as a pre-processing step. The ALO technique [204] [205] focus on the mechanism of hunting present in the behaviour of ant lions naturally. Five steps that are involved in the technique are trap building, prey-catching, trap re-building, ants entrapment and random walk of ants. In the algorithm initialization of antlion optimizer is done with n random ants or prey. The position of each prey represents the combination of selected features. For hunting a prey, different antlion positions are initialized with k . At every iteration, the antlion is selected for hunting, using roulette wheel mechanism. Algorithm performs random walk around the ants and the antlion. According to the last two random walks, when the fitness of ant becomes better than the fitness of antlion, the ant is eaten by antlion and position of ant is grasped by antlion. The maximum iterations are applied till algorithm converge to produce best solution [206]. This technique is chosen because of high exploration and high convergence results. The technique also avoids the local optima problem. The three proposed hybrid models are K-Means and ALO (KISFA), K-means and PSO (KISFP) and K-means and BBO (KISFB). Finally, we have used Equation 6.4 to derive the K -dimensional vector for each source and target instance. The working of ALO as feature selection is shown with the help of flow chart in Figure 6.1.

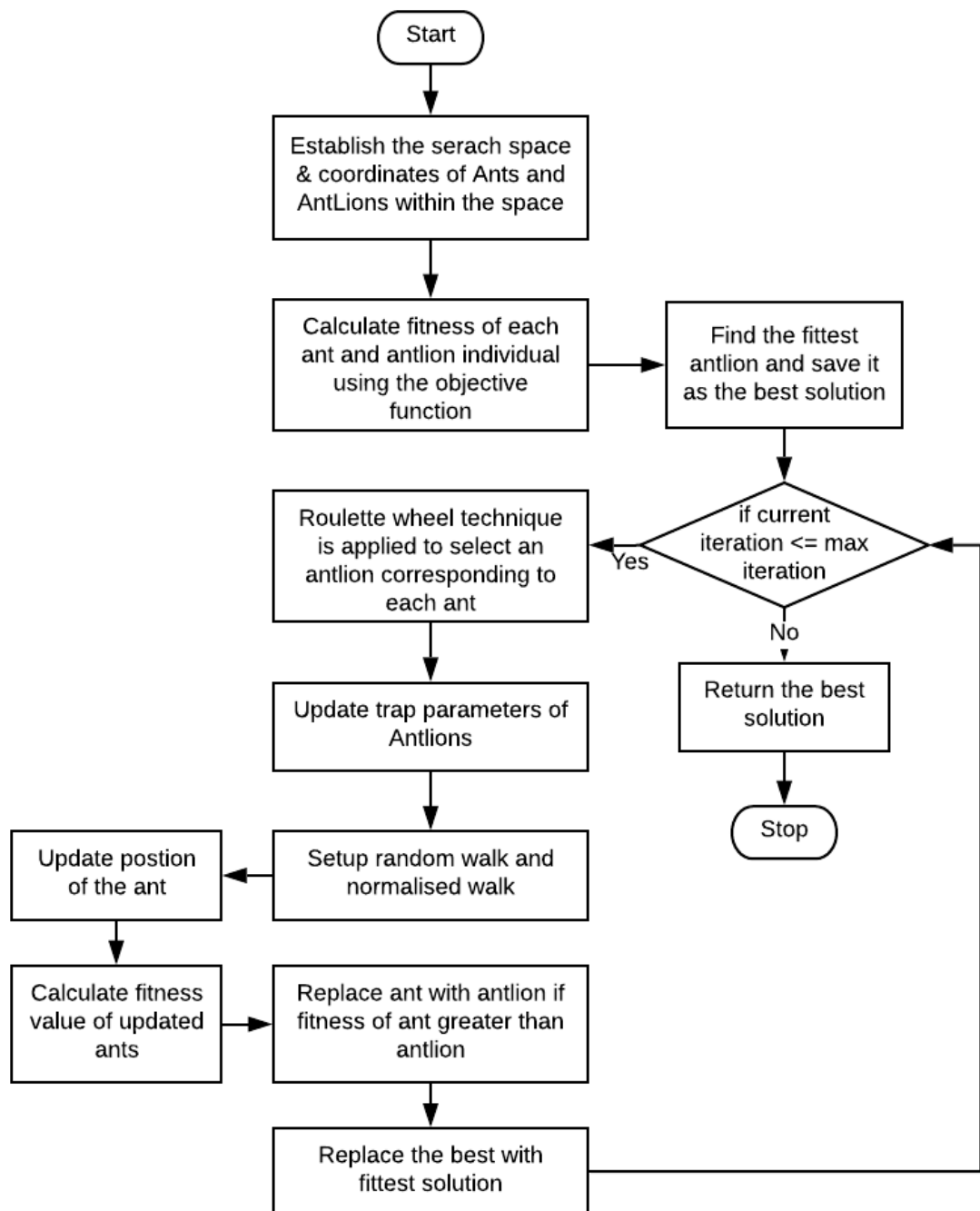


Figure 6.1: Flow chart of ALO as feature selection.

The complete working of the proposed model is shown in Figure 6.2.

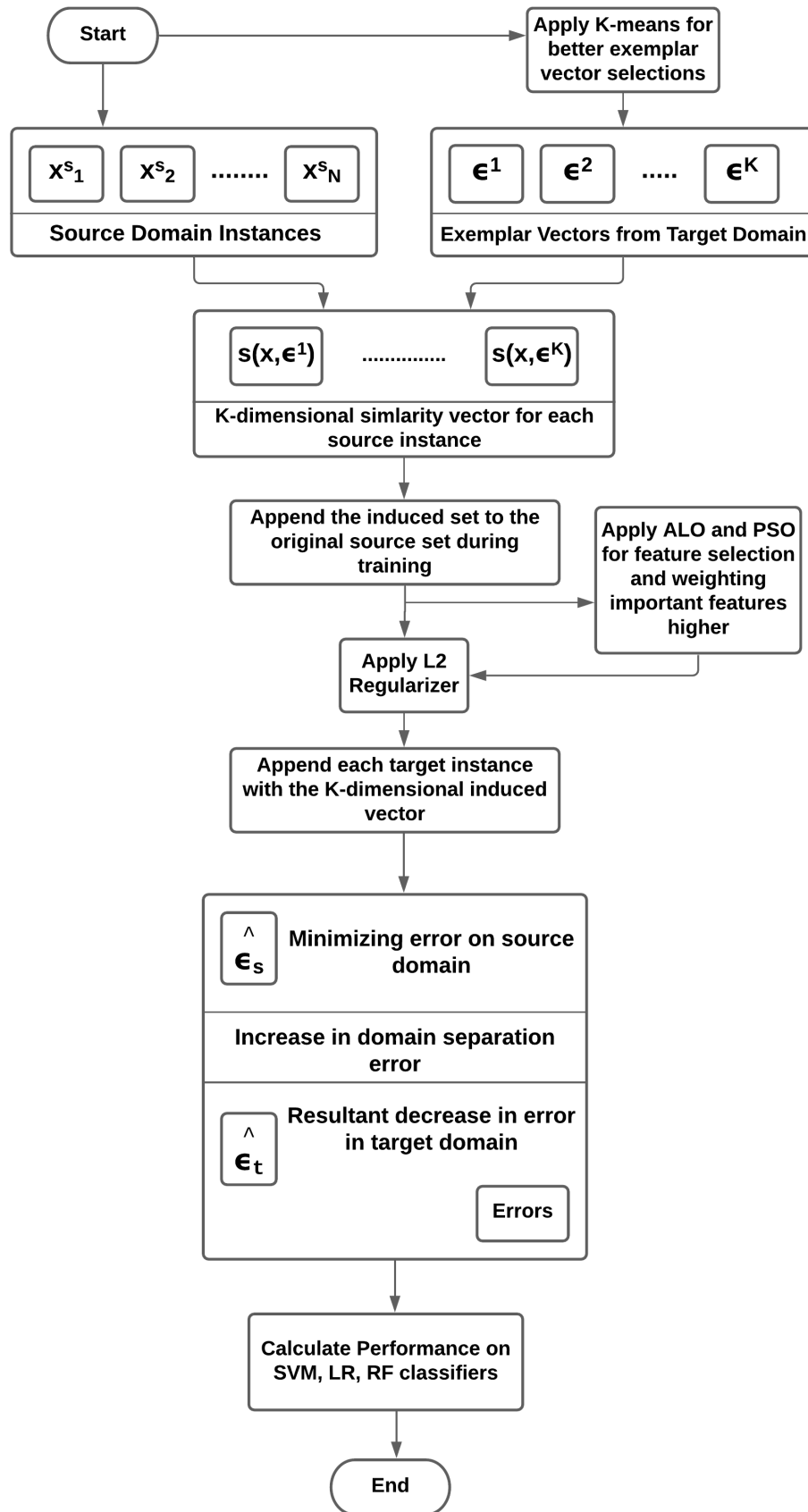


Figure 6.2: Flow process of the model.

6.3 Experimental Setup

6.3.1 Dataset

To prove the model effectiveness, we performed experiments extensively on four datasets. We have also compared our model with state-of-the-art techniques for six datasets. The four datasets on which we performed experiments are namely; Amazon product review [152] referred as AP, ACE 2005 multilingual training corpus [207] referred as ACE, Gene Named Recognition [208] referred as NER and Spambase dataset [209] referred as spam. ACE dataset is retrieved from Linguistic Data Consortium, spam from UCI and other datasets from different website. These dataset are comprised of different domain and every dataset is randomly divided into training and testing dataset irrespective of domain. All datasets are validated using 10-fold cross validation.

AP dataset have 24 categories consist of review of particular product in form of text, rating, help votes, meta data of product and links of the product viewed by user. We sampled 800,000 reviews and analyzed the sentiment using positive and negative polarity. The dataset has around 83 million reviews and with 5 classes.

ACE 2005 multilingual training corpus contains English, Arabic and Chinese dataset. The site which stores data perform on the areas like recognition of entities, relations, temporal expressions, values and events. This corpus focuses on event extraction mechanism.

NER also known as entity identification, entity extraction and entity chunking. This dataset contains chemical protein interaction track. It consist of information about development set abstracts, entity mention annotations and chemical protein detailed relation annotations. We have extracted dataset from the link. ¹

Spambase dataset [210]-This dataset consist of emails from various advertisement sources of web sites , products, fraud money schemes and chain bulk mails. This dataset has 58 attributes and 4601 instances. 20Newsgroup (20NG) [151] [40] contains 20,000 documents that are evenly distributed into 20 different domains. We have extracted this dataset from the link ² .

AFPBB dataset [152] is "AFPBBroadBand News" that includes Japanese articles consisting of different categories namely: Sports, Politics, IT Science, environment, Lifestyle and 10 more categories. It consist of 20,000 articles. We have extracted this dataset from the link ³. We have compared our results with nine different state-of-the art approaches on 12 combinations of datasets. The result is displayed in the Table 6.7. We have performed corpora preprocessing for outlier detection using DBSCAN to improve

¹http://www.biocreative.org/media/store/files/2017/chemprot_sample.zip.

²<https://snap.stanford.edu/data/web-Amazon.html>

³<http://www.afpbb.com/>

performance. It gathers the points which are closer to each other according to euclidean distance and considers those points which lies in low-density region and known as outliers.

6.3.2 Baselines

1) SVM, LR and RF- We proposed our model with these supervised classifiers. The linear Kernel of SVM [153] is used. Default parameters of LR and RF is set for the experiments. SVM is a classification algorithm that can create the boundary between group of vectors that belongs to a particular class to those that do not belong to a particular class. Linear kernel produces high performance when there are high dimensional features. Logistic Regression is widely used for binary classification problem like cancer detection and more binary problems [211]. LR also produced good performance in classification in the literature. Another classifier for high-dimensional data used is Random Forest [212].

2) Structural Correspondence Learning (SCL)- This is cross-domain adaptation semi supervised technique [198]. It basically selects the pivot features to establish the correlation between the features in source and target domain. In final supervised technique the matrix for mapping is obtain to train source domain dataset.

3) ALO - This metaheuristic technique [204] focus on the mechanism of hunting present in the behaviour of antlions naturally. Five steps that are involved in the technique are trap building, prey-catching, trap re-building, ants entrapment and random walk of ants. This technique is chosen because of high exploration and high convergence results. The technique also avoids the local optima problem.

4) PSO - It is nature population-based technique introduced by [213]. The technique is dependent on the group of flocking birds trying to find the optimal path in search of food. Each bird known as 'particle' owns a velocity and position vector to find the optimal solution. It works on the two principles: communication between particles and knowledge of optimal path. The particles share the optimal path to other particle and to store the best position and velocity vector, PSO uses pbest and gbest [2].

5) BBO - Another benchmark optimization technique for feature selection is BBO. BBO is based on migration of species between the islands. The Habitat suitability index, immigration rate and emigration rate represents mathematical model of BBO. The immigration rate and the emigration rate depends on SIV (Suitability Index Variables). BBO as feature selection is implemented in [2].

6) Transfer Independently Together (TIT)- The TIT [153] is generalized method for transferring information for different domains. This method used multiple transformation with re-weighting of samples and landmark samples. Graphs are used for landmark selection.

- 7) Softly Associative Transfer Learning (sa-TL)- sa-TL [151] uses non-negative matrix transformations (NMTF). The two NMTF's are joined with sa-TL. It reduces the difference between word cluster matrix and cluster association matrix.
- 8) Temporal Convolutional Neural Network (TDCN)- The TDCN [152] approach is used to transfer information based on Deep Convolutional Network. The input is given in the form of characters. Input number of character is fixed to 1014. The deep model is constructed with 6- Convolutional layers and 3-fully connected layer. The size of large frame is 1024 and small frame is 256 with two window size of 7 and 3.
- 9) Kullback–Leibler (KL)- The KL [214] is a feature based transfer learning technique which depends on the distribution similarity of features. The features are re-weighted according to the similar distribution in source and target distribution respectively.
- 10) Triplex Transfer Learning (Tri-TL)- The Tri-TL [44] technique uses distinct and also shared theory for cross-domain text classification. It divides the whole theory into three fields: 1) distinct theory, 2) alike theory and 3) identical theory. The total number of word clusters is set as 50. The number of distinct, alike and identical theory is set to 10, 30 and 30.
- 11) HIDC- It is a general probabilistic framework [215] for cross-domain learning in text classification. The number of distinct concepts are set as 10, identical concepts as 20 and homogeneous concepts as 20 and number of clusters of word k as 60.
- 12) FSUTL-PSO- In the study [46], PSO is used for feature selection based on the fitness function. To reduce the fear of selection of degenerated features the common features are selected according to the score of fitness function.
- 13) PA- The researchers in [154] proposed novel method of progressive alignment for heterogeneous domain adaptation. The model is trained to learn new feature space which is transferred to dictionary coding scheme.
- 14) 10-fold-cross-validation - To verify the all experiments performed we have used 10-fold-cross validation technique.
- 15) Statistical significance- To prove statistical significance we have used ANOVA test. Our proposed method KISFA proved to be significantly better than other method.

6.3.3 Objective Functions

To measure the performance of the classification technique, Accuracy [153] [216] and F1-score [194] was chosen. Accuracy (A) is a prosaic way to calculate the performance of classification techniques. It is defined as the number of correct identification of classes by the classification model. F1-score is also known as F-measure, and it calculates the classes which are incorrectly classified. Accuracy and F1-score are calculated using Precision (P), Recall (R) with the help of CM (Confusion Matrix) [45] [2].

6.4 Result

We have implemented and analyzed our model on Windows 10 professional on 2.2 GHZ, 16 GB RAM and i7 processor. The platform we have used for running our algorithms is python3 with libraries tensorflow, keras, numpy and openCV. This section shows the results of the experiments carried out on the four datasets. Each dataset is randomly divided into two different sub-datasets, referred as source and target dataset. Every dataset has cross-domain task to be trained and tested.

A) F1-Score without feature selection- Table 6.3 displays the results of F1-score for the four datasets without feature selection. It can be depicted that K-means Instance Similarity Feature technique performed better than Instance Similarity Feature technique for mostly all of the datasets, except ACE dataset. The other three datasets performed well with SVM classifier. The KISF technique in AP dataset shown improvement with all classifiers with 6% rise in F1-score. The ISF technique performs best with SVM classifier for ACE dataset. The spam dataset performs very well with KISF technique with a rise of 8% when combined with SVM. The NER dataset also showed improved result with all the classifiers with KISF, rise with 4.4%. It was analyzed that in the three datasets, KISF with SVM proved to be the best.

Table 6.3: F1-score

Dataset	ISF			KISF		
	LR	RF	SVM	LR	RF	SVM
ACE	50.1	44.3	58.5	43.3	48.2	55.5
AP	40.3	46.5	52.1	42.6	51.3	58.4
spam	43.4	45.1	40.8	48.2	48.6	52.5
NER	46.2	54.4	45.3	44.9	56.5	59.8

B) Accuracy without feature selection- Table 6.4 indicates the Accuracy of all the datasets. It can be easily seen that KISF performed best with SVM classifier in all the datasets. In NER dataset RF performed same as SVM with KISF. The performance of ACE model is raised from 3% to 8% with rise of SVM classifier to 8%. The performance of Amazon and spam increases from 3% to 25%. The best value in amazon and spam dataset is exhibit with SVM classifier with Accuracy of 76.2% and 74.4% respectively. In NER dataset, RF and SVM classifier performed same with the rise in performance from 6% to 11%. The following observations can be made from Table 6.3 and Table 6.4:-

a) The F1 score values for KISF model ranges from 43.3% to 59.8% and Accuracy values for the same model ranges from 45.8% to 69.5%, which is very satisfactory performance of the model.

Table 6.4: Accuracy

Dataset	ISF			KISF		
	LR	RF	SVM	LR	RF	SVM
ACE	42.5	48.8	52.6	53.7	50.3	60.4
Amazon	43.2	48.5	53.7	48.5	56.7	76.2
spam	30.5	48.0	50.5	45.8	62.1	74.6
NER	56.4	58.3	58.1	62.67	70.4	69.5

b) It can be analyzed that KISF with SVM for all the datasets for both fitness functions produced higher results when compared to other classifiers, except in case of NER which is giving better results with RF classifier. Thus, using SVM with KISF is highly recommended.

c) The KISF model is producing better results with rise of 3% to 13% for F1 score and 5% to 25% in Accuracy score, with respect to ISF values. The k-means clustering approach has been proven effective for all the datasets. The advantage of clustering algorithm in our approach made clear difference in the results, which can be due to easily adaptation to the new patterns.

C) F1-score with feature selection- Table 6.5 displays the results of F1-score of the four datasets with feature selection approach. We have applied feature selection approach to KISF techniques, as it was depicted from Table 6.3 and Table 6.4, that KISF was better technique to be used for transfer learning domain adaptation than our ISF technique. The proposed optimal models namely; KISFA, KISFP and KISFB are being analyzed for F1-score and Accuracy. The F1-score values is displayed in Table 6.5 and Accuracy scores are displayed in Table 6.6. We have compared the results of our proposed techniques. It is analyzed that best values is produced by KISFA in most of the datasets. In Table 6.5, for ACE dataset, the rise of 18.3% is depicted for its best value, with SVM classifier. In AP dataset, KISFB performs best with 4.5% than KISFA, it can be due to different variance of instances. In spam and NER dataset, KISFA performed outstanding with rise of 13.3% and 23.3% in the best values respectively. Both the values performed best with SVM classifier. The observations that can be made from Table

Table 6.5: F1-score with feature selection

Dataset	KISFA			KISFP			KISFB		
	LR	RF	SVM	LR	RF	SVM	LR	RF	SVM
ACE	51.7	50.8	78.2	52.5	49.3	60.5	48.2	52.6	63.4
AP	43.5	58.6	63.1	22.1	46.8	59.9	50.4	57.3	67.6
spam	50.5	47.4	75.4	49.3	58.6	48.8	46.9	55.7	62.3
NER	40.2	66.1	88.5	50.0	60.9	70.3	53.8	61.1	65.8

6.5 are:-

a) The KISFA technique is producing better rise from 2% to 28% from KISFB technique

and 1.5% to 26.6% than KISFP technique. Their results produced exception result in case of AP dataset which is producing slightly better results for KISFB technique with SVM.

b) The KISFA technique is not performing well with RF classifier for ACE, AP and spam dataset but performed well with NER dataset. LR classifier with KISFA technique doesn't work well with AP and NER dataset.

c) The comparison of values of Table 6.3 and Table 6.5 depicts that optimal feature selection gives us better performance than using techniques with classical conventional classifier. This can be due to removal of unnecessary features and high convergence rate of optimization algorithms.

D) Accuracy with feature selection- Table 6.6 shows the Accuracy score of all the four datasets. It can be clearly seen that our proposed model KISFA perform better than the other proposed model KISFP and KISFB. Among all the three traditional classifiers SVM performed best in our model. LR performance is the least preferred classifier for our model. The proposed model with SVM classifier boost up the F1-score by 10% to 20%.

In Table 6.6, Accuracy of all the datasets are calculated. The Accuracy is calculated

Table 6.6: Accuracy with feature selection

Dataset	KISFA			KISFP			KISFB		
	LR	RF	SVM	LR	RF	SVM	LR	RF	SVM
ACE	68.1	70.3	85.5	56.0	42.5	50.2	60.6	57.4	74.9
AP	70.9	78.6	97.7	55.3	70.4	81.5	60.7	82.8	84.2
spam	61.2	82.4	93.7	58.2	57.1	72.6	64.8	66.2	84.5
NER	70.5	82.9	93.8	72.6	70.8	82.3	78.1	78.3	92.6

using feature selection algorithm with KISF technique, as it performed better than the other proposed ISF technique. It can be observed that Accuracy achieved by KISFA with SVM classifier is the highest when compared to other techniques. In all the four datasets the model KISFA with SVM performed best, although other two classifiers also performed well in some of the cases. LR performance was the worst among all classifiers for all the datasets and RF performed nearly good in two datasets. It can be clearly seen that the performance of ACE rises nearly to 7.4%, AP performance rises to 17.3%, spam to 21.1% and NER to 11.5%, when compared to best values among KISFA, KISFP and KISFB technique respectively. Hence, our model, KISFA performed best among the proposed optimal three models. We performed 50 iterations to check the significance of the techniques.

The following other observations can be made:-

a) The KISFA technique performed best with all the classifiers in comparison to KISFB except in NER dataset with LR classifier. The SVM classifier with KISFA technique is

recommended to use for transfer learning approach for heterogeneous domain adaptation.

b) In comparison to KISFP technique KISFA produced better results with all the classifiers except for NER dataset with LR classifier. The KISFA technique is producing better values than both KISFP and KISFB techniques respectively. This can be due to the high convergence of Ant Lion Optimization algorithm.

c) From Table 6.4 and 6.6, the use of K-means clustering with feature selection approach is justified. The nature of guaranteed convergence and adaptation to new examples of K-means algorithms have proved significance addition to our proposed model ISF. The hybrid of KISF technique with ALO has been proven as more better models than KISF with PSO and KISF and BBO. We perform the parameter sensitivity test on population size. After analyzing the best Accuracy produced on the population size of a technique, the experiments were performed.

E) Comparison with State-of-the-art techniques- In Table 6.7, we compared our model with nine different models on six datasets, proposed by researchers. We performed the comparison with other models, mentioned in 6.3.2 and proposed in ([153], [151], [152], [214], [198], [44], [215], [46] and [154]) . It can be observed that our model achieved the best average Accuracy among all the models. The proposed model enhanced the Accuracy for four datasets ranging from 6% to 25%. Our proposed model, KISFA outperformed the transfer learning task in four out of six datasets. We have calculated the average Accuracy to generalize the best performance, and it can be clearly seen that our model performed better in terms of average Accuracy. TIT, sa-TL, TDCN and KL techniques perform better than other techniques but our model is better. TriTL and HIDC, produces second worst results because of not separating associate cluster matrix for heterogeneous domain adaptation. It was observed from Table 6.7 that although technique TIT uses an auxiliary data to train the dataset, but, our model performs better than TIT with 6% higher average Accuracy. Our model also performed better with higher average Accuracy of 21% than semi supervised SCL which have labelled source data. We have conducted 50 iterations and assume best Accuracy among all the iterations. TIT perform better than our model in ACE and AFPBB dataset that can be because of multiple transformations and re-weighting of samples. The techniques sa-TL and TDCN produced good performance, but inferior scores to our model. In 20NG dataset PA and HIDC produces third and second best Accuracy results respectively, but our model still produces best result with 2.5% higher than HIDC. This can be due to the use of cluster centroids as exemplar vectors. The proposed model is better than the FP technique for all the datasets with rise from 20% to 30%. Our model outperformed the PA technique in five datasets and performed almost equivalent in one dataset.

Table 6.7: Predictive Accuracy

Dataset	TIT	sa-TL	TDCN	KL	SCL	TriTL	HIDC	FP	PA	Our model
ACE	91.6	75.0	85.8	83.7	70.3	79.3	80.1	63.4	85.6	88.5
AP	89.3	74.8	80.5	86.4	72.4	71.2	73.4	67.2	90.1	97.7
spam	76.2	90.6	91.7	75.2	73.7	67.5	71.8	72.4	81.5	92.5
NER	86.2	84.8	82.3	91.6	66.6	80.9	86.2	64.5	85.6	93.8
20NG	88.8	94.1	80.6	78.0	85.4	82.3	96.1	68.4	96.0	98.6
AFPBB	93.5	91.3	90.6	82.5	68.9	76.8	68.6	42.0	89.9	90.2
Average	87.6	86.6	85.25	82.91	72.88	76.33	79.3	62.8	88.11	93.38

6.4.1 Error reduction

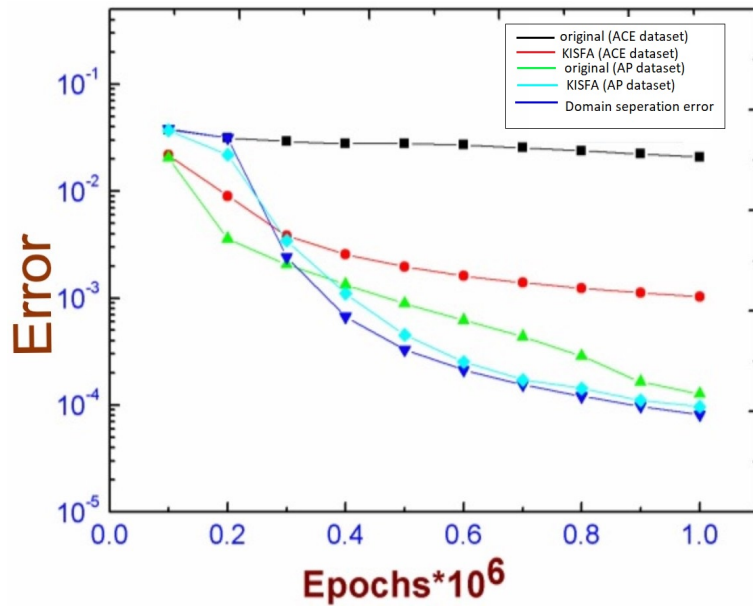


Figure 6.3: Error graph of ACE and AP

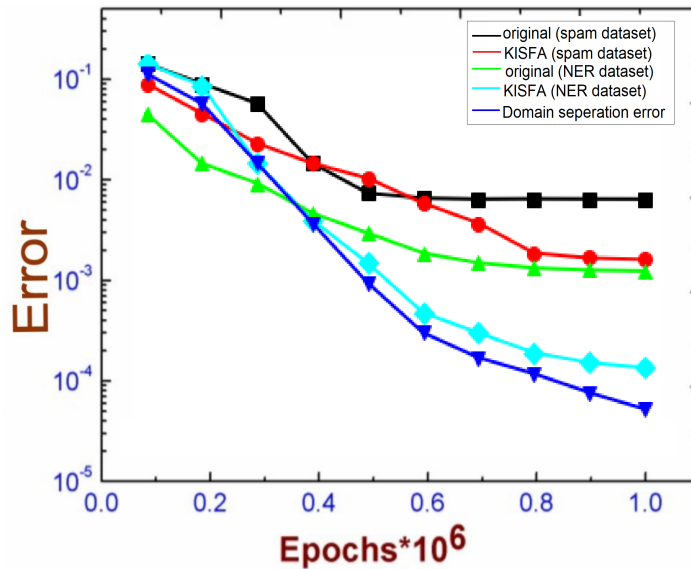


Figure 6.4: Error graph of Spam and NER

In Figure 6.3 and Figure 6.4, x-axis denotes the epochs which means iterations for which the graph is constructed and y-axis denotes the error calculated by reduction formulae given in the section 6.2.3.2. The Figure 6.3, error reduction graph is for ACE and AP dataset. The black colored line denotes the error point for the original model on the given dataset ACE without feature selection. A red line denotes KISFA technique on same dataset, green line denotes original feature set of AP dataset, blue line denotes the KISFA on AP dataset. In Figure 6.4, the graph shows the error reduction results for

spam and NER dataset. The black and green colored lines denote the original feature set with ISF technique on spam and NER dataset respectively. The red line and blue line depicts the error points for KISFA technique for spam and NER dataset respectively. The dark blue line is of the domain separation error in terms of error reductions in both the graph. It can be seen that domain separation increases then domain separation error increases, hence decreasing the target error. The reduction in target error depends on both the training error on the source domain and domain separation error, when we train the features on different domains of the source and target instances.

6.4.2 Parameter Sensitivity Test

To measure the parameter sensitivity of optimization techniques, we have considered population size as important variable. Although there are many other control variables in optimization algorithms, but the variable population size can highly affect the performance of the models. The population size is highly dependent on the application. Hence, to study the affect of population size on our application we used different variants of population size. The population of model means the number of individuals taking part in the evaluation as habitats in BBO, number of swarms in PSO and in ALO it is the number of Ant Lions. Each technique is tested with five different population sizes 20, 50, 100, 200 and 400. The Accuracy is considered as the fitness function to measure the analysis. Different colours are used for different population size, red for 20 population size, 50 with blue colour, 100 with green colour, 200 with yellow colour and 400 with pink colour. Iterations are plotted on x-axis, while Accuracy measures are on Y-axis. Every model runs for 1000 iterations for the setting of optimal parameters. The result is a convergence curve showing the change in Accuracy measure. Figure 6.5 represents the ALO, Figure 6.6 shows PSO and Figure 6.7 shows BBO. In Figure 6.5, it can be easily analyzed that population size of 400 is giving best results and highest value of 97.7 at 798th iteration. Figure 6.6, produces the best value at population size of 100 at 800th iteration. At population size 200, the Accuracy started decreasing. Figure 6.7, depicts the curve and it shows that at population size of 50 with 995th iteration, it obtains best value. The best value is obtained is 92.6. BBO again gained the good results at population size of 200 at 1000th iteration.

Figure 6.8, compares the Accuracy measure with respect to population size. On X-axis, the population size is considered and on y-axis, Accuracy measures are plotted. A red colour denotes line of PSO technique, blue colour line for BBO technique and green colour line for ALO technique. As we can depict that all the three techniques increases the Accuracy with the increasing number of iterations. After analysis, ALO, PSO and BBO is performing best at 400, 100 and 200 than the other population size. It can also be depicted that Accuracy score is always upward in ALO, while fluctua-

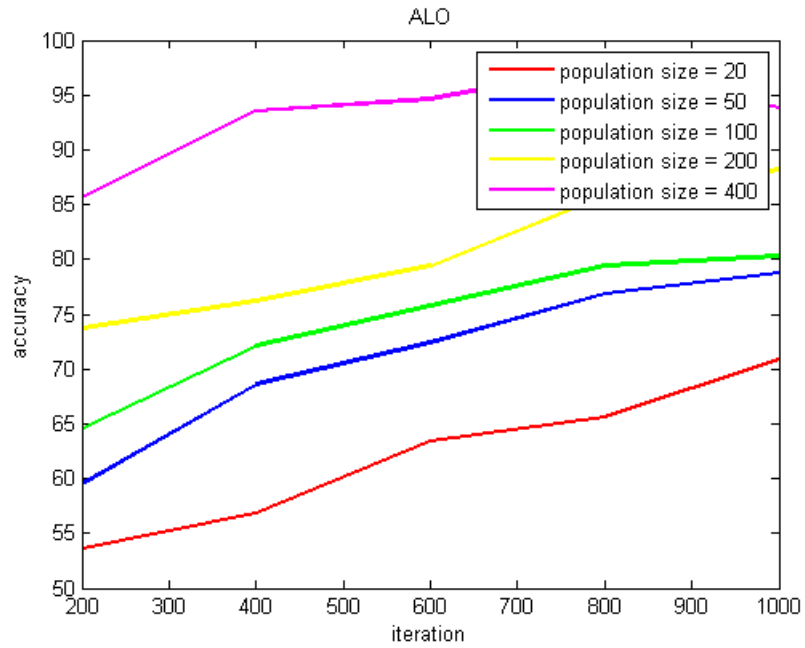


Figure 6.5: sensitivity analysis for ALO

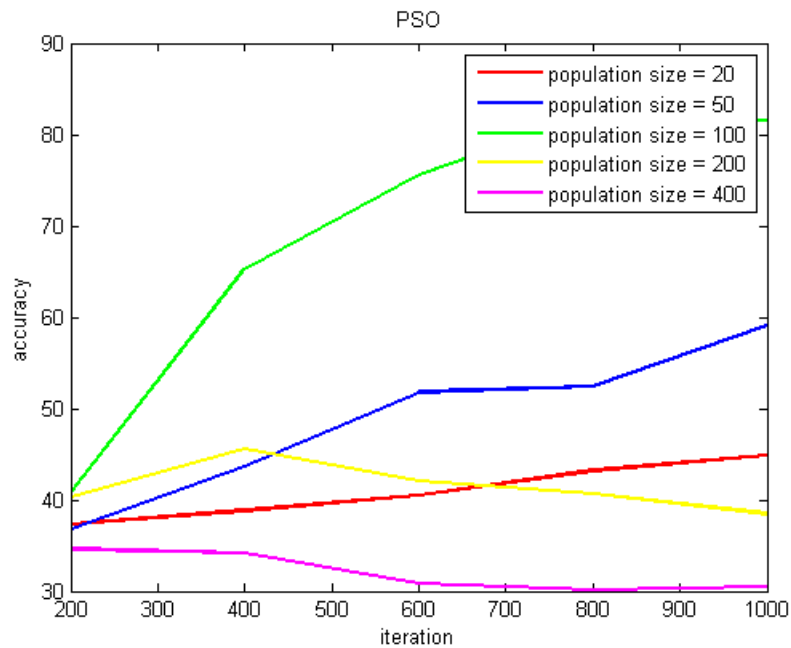


Figure 6.6: Sensitivity analysis for PSO

tion can be observed in other two techniques. The proposed model KISFA achieve the highest Accuracy.

6.4.3 Statistical Significance

To validate the working of our model, the statistical test ANOVA is executed. ANOVA test is considered when the results are variable and is dependent on the experimental

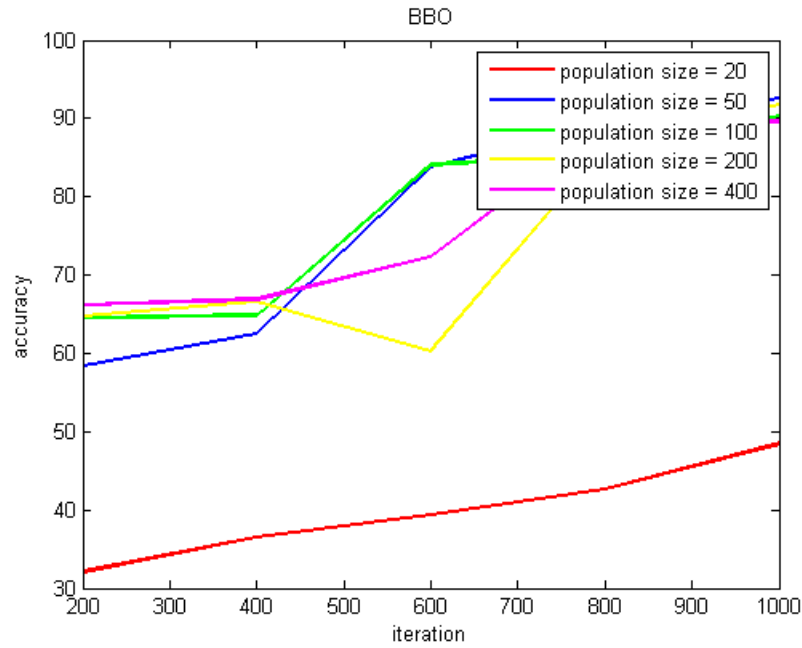


Figure 6.7: Sensitivity analysis for BBO

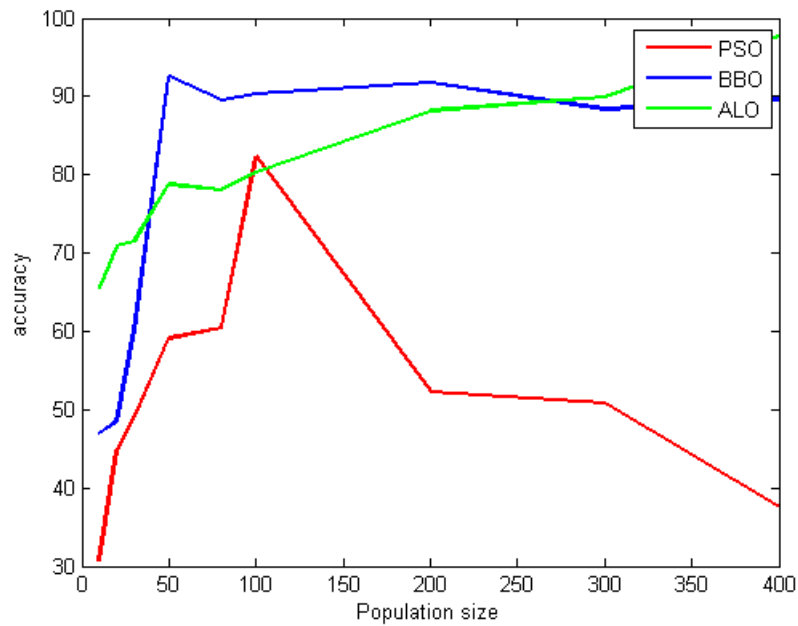


Figure 6.8: Comparison of Sensitivity analysis

values of factors. The selection of ANOVA test is done on the basis that we have more than two groups to be compared. To execute test, the experimental factors that are significant to change the scores are determined. In our study, we have mainly considered one-way factor which causes significant change in results, that is techniques. The result of ANOVA model is shown in Table 6.8, where leftmost column describes the techniques, with subsequent columns as Degree of freedom (DoF), Sum of Squares (SS),

Mean square (MS), F-statistics and P-values. The row represent the results Between Groups (BG), within Groups (WG) and Total results. The ANOVA model signifies that individual techniques as well as grouped techniques are significantly different, with 95% confidence level ($p\text{-value} \leq 0.05$). In Table 6.9, the ANOVA test summary is represented to analyze the significance. In results, it can be depicted that highest mean score and low standard deviation and standard error is achieved by our model, which signifies that our model is significantly better than other techniques. As p-value is less

Table 6.8: Descriptive statistics

Technique	DOF	SS	MS	F-stat	p-value
BG	8	3945.5139	493.1892	9.1797	0.002
WG	45	2417.6701	53.726		
Total	53	6363.184			

than 0.05, so null hypothesis is rejected, stating that there is significant difference between the groups.

Table 6.9: Data summary of ANOVA

Groups	N	Mean	Std.Dev	Std.Error
TIT	6	87.6	6.1165	2.4971
Sa-TL	6	85.1	8.4603	3.4539
TDCN	6	85.25	4.9682	2.0283
KL	6	82.9	5.8658	2.3947
SCL	6	72.883	6.627	2.7055
Tri-TL	6	76.33	5.8284	2.3794
HIDC	6	79.3057	10.3539	4.227
FP	6	62.8667	10.6884	4.6366
our model	6	93.55	4.0153	1.6393

6.5 Conclusion

In this research, we have proposed a generalized optimal model for heterogeneous domain adaptation, KISFA. We have proposed and tested five models for heterogeneous domain adaptation in transfer learning namely: ISF, KISF, KISFA, KISFP and KISFB. The KISFA, KISFP and KISFB are optimal models. The proposed approach is mainly based on feature vectorization, exemplar vector selection using similarity between the features and use of K-means clustering technique in choosing the best exemplar vectors.

Moreover, we can further observe that by using the cluster centroids as exemplar vectors, it can bring significant improvements over our ISF method in all the four datasets. It can be analyzed that best model depicted after extensive experiments on four datasets with 10-fold cross validation is KISFA. We have chosen the varied datasets to verify the working on different domains. The use of feature selection techniques proven to be beneficiary for the better classification results. The mean Accuracy varies from 60.1 to 92.6 and F1 score 43.37 to 76.3 for optimal models. The parametric sensitivity test shows the convergence of ALO is best, hence producing the best model when combined with KISF. The model validation is completed with statistical ANOVA test which signifies the KISFA better technique than other techniques. The comparison with other nine state-of-the-art techniques prove the working of our model as best and can be used for transfer learning process.

For the future scope we will develop heterogeneous transfer learning. We will also propose model to work on parameter sensitivity values and experiment for feature selection with more upcoming nature based techniques.

Publication

The work discussed in this chapter is under review in:

Khurana, A., and Verma, O.P. Optimal Heterogeneous Domain Adaptation for Text Classification in Transfer Learning, Expert Systems and Applications.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

In this thesis, we formalize the algorithms for the improvement of text classification. The models can be used to classify text for review datasets, healthcare datasets, sentiment classification etc. The developed models have been compared with the existing state-of-the-art models and shown significant results for different performance measures like AUC curve, Precision, Recall, Accuracy, G-mean and F-measure. Proposed models perform well quantitatively and qualitatively.

7.1 Summary of the work done in the thesis

To address the problems that occur during text classification, different models were proposed.

- The novel approach of nature-based optimization as feature selection with an ensemble classifier for optimal text classification is proposed. To overcome the problem of high-dimensionality, technique using an ensemble classifier with a feature selection approach using BBO is explored. The proposed technique of BBO and ensemble classifier is compared with other individual models with state-of-the-art algorithms available in the literature on all datasets. The model produces better results as compared to individual classifier. Results are compared using different performance measures.
- The technique is developed to handle imbalance problem which avoids the problem of small disjuncts and lack of density by creating samples widely. Another limitation of improving classification performance is high number of features, which we handled by modifying mutation operator of the optimization technique. The comparison with different variants of oversampling techniques and state-of-the-art techniques confirmed the model efficiency.
- Another way to improve text classification is with the help of hyper parameter tuning of optimization techniques. We proposed a novel approach of tuned GOA with tuned classifiers. The tuned GOA is used for feature selection to decrease

the problem caused due to high-dimensionality, and increase the classification performance. We have used random search technique for the parameter optimization of the two classifiers- k-NN and SVM. We have tested the proposed model on 5-real-world datasets from UCI by computing classification accuracy and AUC measure. In the different performance measure, our model performed better than other state-of-the-art techniques. The best accuracy was obtained with high-dimensional dataset, although proposed model worked good with low-dimensional dataset as well.

- Another model is proposed using transfer learning approach. We have proposed a generalized optimal model for heterogeneous domain adaptation, KISFA. We have proposed and tested five models for heterogeneous domain adaptation in transfer learning namely: ISF, KISF, KISFA, KISFP and KISFB. The KISFA, KISFP and KISFB are optimal models. The proposed approach is mainly based on feature vectorization, exemplar vector selection using similarity between the features and use of K-means clustering technique in choosing the best exemplar vectors. Moreover, we can further observe that by using the cluster centroids as exemplar vectors, it can bring significant improvements over our ISF method in all the four datasets. It can be analyzed that best model depicted after extensive experiments on four datasets with 10-fold cross validation is KISFA. We have chosen the varied datasets to verify the working on different domains. The use of feature selection techniques proven to be beneficiary for the better classification results.

7.2 Future work

- The future work will include more ensemble classifiers with other upcoming optimization algorithms.
- Focus on training deep neural networks using tuned meta-heuristic algorithms.
- Transfer learning on heterogeneous data to be explored in future. Heterogeneous data have different features on both the datasets that is used for training and testing. The information extracted from image will also be explored to training textual dataset.
- Model for the improvement of text classification with multi-objective optimization algorithm.

REFERENCES

- [1] C. C. Aggarwal and C. Zhai, “A survey of text classification algorithms,” in *Mining text data*, pp. 163–222, Springer, 2012.
- [2] A. Khurana and O. P. Verma, “Novel approach with nature-inspired and ensemble techniques for optimal text classification,” *Multimedia Tools and Applications*, vol. 79, no. 33, pp. 23821–23848, 2020.
- [3] A. K. Uysal and S. Gunal, “Text classification using genetic algorithm oriented latent semantic features,” *Expert Systems with Applications*, vol. 41, no. 13, pp. 5938–5947, 2014.
- [4] J. Kaur and J. R. Saini, “A study of text classification natural language processing algorithms for indian languages,” *The VNSGU Journal of Science Technology*, vol. 4, no. 1, pp. 162–167, 2015.
- [5] A. Khurana and O. P. Verma, “Optimal feature selection for imbalanced text classification,” *IEEE Transactions on Artificial Intelligence*, 2022.
- [6] D. R. O. Reddy, R. N. Reddy, M. Radha, and S. Vani, “A review of machine learning approaches in data sensitive real-world applications ‘;’,” *Journal of advanced research in dynamical and control systems*, vol. 9, no. 3, pp. 165–171, 2017.
- [7] H. Chen, W. Jiang, C. Li, and R. Li, “A heuristic feature selection approach for text categorization by using chaos optimization and genetic algorithm,” *Mathematical problems in Engineering*, vol. 2013, 2013.
- [8] S. Saremi, S. Mirjalili, and A. Lewis, “Grasshopper optimisation algorithm: theory and application,” *Advances in Engineering Software*, vol. 105, pp. 30–47, 2017.
- [9] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, “An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics,” *Information sciences*, vol. 250, pp. 113–141, 2013.

- [10] A. K. Uysal, “An improved global feature selection scheme for text classification,” *Expert systems with Applications*, vol. 43, pp. 82–92, 2016.
- [11] X. Tang and L. Chen, “Artificial bee colony optimization-based weighted extreme learning machine for imbalanced data learning,” *Cluster Computing*, pp. 1–16, 2018.
- [12] C.-S. Yang, L.-Y. Chuang, J.-C. Li, and C.-H. Yang, “Chaotic maps in binary particle swarm optimization for feature selection,” in *2008 IEEE Conference on Soft Computing in Industrial Applications*, pp. 107–112, IEEE, 2008.
- [13] S. Fong, R. Wong, and A. V. Vasilakos, “Accelerated pso swarm search feature selection for data stream mining big data,” *IEEE transactions on services computing*, vol. 9, no. 1, pp. 33–45, 2016.
- [14] D. Y. Eroglu and K. Kilic, “A novel hybrid genetic local search algorithm for feature selection and weighting with an application in strategic decision making in innovation management,” *Information Sciences*, vol. 405, pp. 18–32, 2017.
- [15] E. Zorarpacı and S. A. Özel, “A hybrid approach of differential evolution and artificial bee colony for feature selection,” *Expert Systems with Applications*, vol. 62, pp. 91–103, 2016.
- [16] J. Grande, M. del Rosario Suárez, and J. R. Villar, “A feature selection method using a fuzzy mutual information measure,” in *Innovations in Hybrid Intelligent Systems*, pp. 56–63, Springer, 2007.
- [17] B. Tang, H. He, P. M. Baggenstoss, and S. Kay, “A bayesian classification approach using class-specific features for text categorization,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 6, pp. 1602–1606, 2016.
- [18] P. Moradi and M. Gholampour, “A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy,” *Applied Soft Computing*, vol. 43, pp. 117–130, 2016.
- [19] H. Wang, X. Zhou, H. Sun, X. Yu, J. Zhao, H. Zhang, and L. Cui, “Firefly algorithm with adaptive control parameters,” *Soft computing*, vol. 21, no. 17, pp. 5091–5102, 2017.
- [20] Y. Zhang, D.-w. Gong, and J. Cheng, “Multi-objective particle swarm optimization approach for cost-based feature selection in classification,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 14, no. 1, pp. 64–75, 2017.

- [21] D. Simon, “Biogeography-based optimization,” *IEEE transactions on evolutionary computation*, vol. 12, no. 6, pp. 702–713, 2008.
- [22] H. He and E. A. Garcia, “Learning from imbalanced data,” *IEEE Transactions on Knowledge and Data Engineering*, no. 9, pp. 1263–1284, 2008.
- [23] A. Amin, S. Anwar, A. Adnan, M. Nawaz, N. Howard, J. Qadir, A. Hawalah, and A. Hussain, “Comparing oversampling techniques to handle the class imbalance problem: A customer churn prediction case study,” *IEEE Access*, vol. 4, pp. 7940–7957, 2016.
- [24] A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla, “Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary,” *Journal of artificial intelligence research*, vol. 61, pp. 863–905, 2018.
- [25] M. Claesen and B. De Moor, “Hyperparameter search in machine learning,” *arXiv preprint arXiv:1502.02127*, 2015.
- [26] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *Journal of machine learning research*, vol. 13, no. Feb, pp. 281–305, 2012.
- [27] R. Ghawi and J. Pfeffer, “Efficient hyperparameter tuning with grid search for text categorization using knn approach with bm25 similarity,” *Open Computer Science*, vol. 9, no. 1, pp. 160–180, 2019.
- [28] Z. Wang, F. Hutter, M. Zoghi, D. Matheson, and N. de Freitas, “Bayesian optimization in a billion dimensions via random embeddings,” *Journal of Artificial Intelligence Research*, vol. 55, pp. 361–387, 2016.
- [29] S. Diab, “Optimizing stochastic gradient descent in text classification based on fine-tuning hyper-parameters approach. a case study on automatic classification of global terrorist attacks,” *arXiv preprint arXiv:1902.06542*, 2019.
- [30] M. A. Tahir, A. Bouridane, and F. Kurugollu, “Simultaneous feature selection and feature weighting using hybrid tabu search/k-nearest neighbor classifier,” *Pattern Recognition Letters*, vol. 28, no. 4, pp. 438–446, 2007.
- [31] R. Leardi and A. L. Gonzalez, “Genetic algorithms applied to feature selection in pls regression: how and when to use them,” *Chemometrics and intelligent laboratory systems*, vol. 41, no. 2, pp. 195–207, 1998.
- [32] E. Emary, H. M. Zawbaa, C. Grosan, and A. E. Hassenian, “Feature subset selection approach by gray-wolf optimization,” in *Afro-European conference for industrial advancement*, pp. 1–13, Springer, 2015.

- [33] H. Faris, M. A. Hassonah, A.-Z. Ala'M, S. Mirjalili, and I. Aljarah, "A multi-verse optimizer approach for feature selection and optimizing svm parameters based on a robust system architecture," *Neural Computing and Applications*, vol. 30, no. 8, pp. 2355–2369, 2018.
- [34] B. Ionescu, M. Lupu, M. Rohm, A. L. Gînsca, and H. Müller, "Datasets column: diversity and credibility for social images and image retrieval," *ACM SIGMulti-media Records*, vol. 9, no. 3, pp. 7–7, 2018.
- [35] M. Long, J. Wang, J. Sun, and S. Y. Philip, "Domain invariant transfer kernel learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 6, pp. 1519–1532, 2014.
- [36] J. Li, J. Zhao, and K. Lu, "Joint feature selection and structure preservation for domain adaptation.," in *IjCAI*, pp. 1697–1703, 2016.
- [37] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2066–2073, IEEE, 2012.
- [38] Z. Cui, W. Li, D. Xu, S. Shan, X. Chen, and X. Li, "Flowing on riemannian manifold: Domain adaptation by shifting covariance," *IEEE transactions on cybernetics*, vol. 44, no. 12, pp. 2264–2273, 2014.
- [39] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer joint matching for unsupervised domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1410–1417, 2014.
- [40] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.
- [41] L. Niu, J. Cai, and D. Xu, "Domain adaptive fisher vector for visual recognition," in *European Conference on Computer Vision*, pp. 550–566, Springer, 2016.
- [42] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *2011 international conference on computer vision*, pp. 999–1006, IEEE, 2011.
- [43] L. Duan, I. W. Tsang, and D. Xu, "Domain transfer multiple kernel learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 465–479, 2012.
- [44] F. Zhuang, P. Luo, C. Du, Q. He, Z. Shi, and H. Xiong, "Triplex transfer learning: Exploiting both shared and distinct concepts for text classification," *IEEE transactions on cybernetics*, vol. 44, no. 7, pp. 1191–1203, 2013.

- [45] H. Wang, W. Wang, C. Zhang, and F. Xu, “Cross-domain metric learning based on information theory,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 28, 2014.
- [46] R. K. Sanodiya, M. Tiwari, J. Mathew, S. Saha, and S. Saha, “A particle swarm optimization-based feature selection for unsupervised transfer learning,” *Soft Computing*, vol. 24, no. 24, pp. 18713–18731, 2020.
- [47] L. Jiang, C. Li, S. Wang, and L. Zhang, “Deep feature weighting for naive bayes and its application to text classification,” *Engineering Applications of Artificial Intelligence*, vol. 52, pp. 26–39, 2016.
- [48] K. Nag and N. R. Pal, “A multiobjective genetic programming-based ensemble for simultaneous feature selection and classification,” *IEEE transactions on cybernetics*, vol. 46, no. 2, pp. 499–510, 2016.
- [49] V. Balasubramanian, S.-S. Ho, and V. Vovk, *Conformal prediction for reliable machine learning: theory, adaptations and applications*. Newnes, 2014.
- [50] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003.
- [51] X. Li, Y. Rao, H. Xie, R. Y. K. Lau, J. Yin, and F. L. Wang, “Bootstrapping social emotion classification with semantically rich hybrid neural networks,” *IEEE Transactions on Affective Computing*, vol. 8, no. 4, pp. 428–442, 2017.
- [52] Y. Liu, J.-W. Bi, and Z.-P. Fan, “Multi-class sentiment classification: The experimental comparisons of feature selection and machine learning algorithms,” *Expert Systems with Applications*, vol. 80, pp. 323–339, 2017.
- [53] R. H. Pinheiro, G. D. Cavalcanti, R. F. Correa, and T. I. Ren, “A global-ranking local feature selection method for text categorization,” *Expert Systems with Applications*, vol. 39, no. 17, pp. 12851–12857, 2012.
- [54] B. Yang, Y. Zhang, and X. Li, “Classifying text streams by keywords using classifier ensemble,” *Data and Knowledge Engineering*, vol. 70, no. 9, pp. 775–793, 2011.
- [55] C.-T. Su and H.-C. Lin, “Applying electromagnetism-like mechanism for feature selection,” *Information Sciences*, vol. 181, no. 5, pp. 972–986, 2011.
- [56] M. A. Esseghir, G. Goncalves, and Y. Slimani, “Adaptive particle swarm optimizer for feature selection,” in *International Conference on Intelligent Data Engineering and Automated Learning*, pp. 226–233, Springer, 2010.

- [57] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of machine learning*, pp. 760–766, Springer, 2011.
- [58] D. M. Diab and K. M. El Hindi, "Using differential evolution for fine tuning naïve bayesian classifiers and its application for text classification," *Applied Soft Computing*, vol. 54, pp. 183–199, 2017.
- [59] A. S. Ghareb, A. A. Bakar, and A. R. Hamdan, "Hybrid feature selection based on enhanced genetic algorithm for text categorization," *Expert Systems with Applications*, vol. 49, pp. 31–47, 2016.
- [60] G. I. Sayed, A. E. Hassanien, and A. T. Azar, "Feature selection via a novel chaotic crow search algorithm," *Neural Computing and Applications*, vol. 31, no. 1, pp. 171–188, 2019.
- [61] A. I. Hafez, H. M. Zawbaa, E. Emary, H. A. Mahmoud, and A. E. Hassanien, "An innovative approach for feature selection based on chicken swarm optimization," in *2015 7th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, pp. 19–24, IEEE, 2015.
- [62] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, 2016.
- [63] A. I. Hafez, H. M. Zawbaa, E. Emary, and A. E. Hassanien, "Sine cosine optimization algorithm for feature selection," in *2016 International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*, pp. 1–5, IEEE, 2016.
- [64] S.-W. Lin, K.-C. Ying, S.-C. Chen, and Z.-J. Lee, "Particle swarm optimization for parameter determination and feature selection of support vector machines," *Expert systems with applications*, vol. 35, no. 4, pp. 1817–1824, 2008.
- [65] G. I. Sayed, A. Darwish, A. E. Hassanien, and J.-S. Pan, "Breast cancer diagnosis approach based on meta-heuristic optimization algorithm inspired by the bubble-net hunting strategy of whales," in *International Conference on Genetic and Evolutionary Computing*, pp. 306–313, Springer, 2016.
- [66] M. Schiezero and H. Pedrini, "Data feature selection based on artificial bee colony algorithm," *EURASIP Journal on Image and Video Processing*, vol. 2013, no. 1, p. 47, 2013.
- [67] H. M. Zawbaa, E. Emary, B. Parv, and M. Sharawi, "Feature selection approach based on moth-flame optimization algorithm," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 4612–4617, IEEE, 2016.

- [68] S. A.-F. Sayed, E. Nabil, and A. Badr, “A binary clonal flower pollination algorithm for feature selection,” *Pattern Recognition Letters*, vol. 77, pp. 21–27, 2016.
- [69] R. Storn and K. Price, “Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [70] S. J. Russell and P. Norvig, “A modern, agent-oriented approach to introductory artificial intelligence,” *SIGART Bulletin*, vol. 6, no. 2, pp. 24–26, 1995.
- [71] K. El Hindi, “Fine tuning the naïve bayesian learning algorithm,” *AI Communications*, vol. 27, no. 2, pp. 133–141, 2014.
- [72] K. El Hindi, “A noise tolerant fine tuning algorithm for the naïve bayesian learning algorithm,” *Journal of King Saud University-Computer and Information Sciences*, vol. 26, no. 2, pp. 237–246, 2014.
- [73] L. Ying, “Analysis on text classification using naive bayes,” *Computer Knowledge and Technology (Academic Exchange)*, vol. 11, 2007.
- [74] H.-s. LIANG, J.-m. XU, and Y.-p. CHENG, “An improving text categorization method of na ve bayes,” *Journal of Hebei University (Natural Science Edition)*, no. 3, p. 24, 2007.
- [75] A. McCallum, K. Nigam, *et al.*, “A comparison of event models for naive bayes text classification,” in *AAAI-98 workshop on learning for text categorization*, vol. 752, pp. 41–48, Citeseer, 1998.
- [76] J. D. Rennie, L. Shih, J. Teevan, and D. R. Karger, “Tackling the poor assumptions of naive bayes text classifiers,” in *Proceedings of the 20th international conference on machine learning (ICML-03)*, pp. 616–623, 2003.
- [77] L. Jiang, Z. Cai, D. Wang, and H. Zhang, “Improving tree augmented naive bayes for class probability estimation,” *Knowledge-Based Systems*, vol. 26, pp. 239–245, 2012.
- [78] A. Melo and H. Paulheim, “Local and global feature selection for multilabel classification with binary relevance,” *Artificial intelligence review*, vol. 51, no. 1, pp. 33–60, 2019.
- [79] C. N. Silla and A. A. Freitas, “A survey of hierarchical classification across different application domains,” *Data Mining and Knowledge Discovery*, vol. 22, no. 1-2, pp. 31–72, 2011.

- [80] M. Jiang, Y. Liang, X. Feng, X. Fan, Z. Pei, Y. Xue, and R. Guan, "Text classification based on deep belief network and softmax regression," *Neural Computing and Applications*, vol. 29, no. 1, pp. 61–70, 2018.
- [81] B. Z. Dadaneh, H. Y. Markid, and A. Zakerolhosseini, "Unsupervised probabilistic feature selection using ant colony optimization," *Expert Systems with Applications*, vol. 53, pp. 27–42, 2016.
- [82] Q. Qin, W. Hu, and B. Liu, "Feature projection for improved text classification," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8161–8171, 2020.
- [83] G. Kou, P. Yang, Y. Peng, F. Xiao, Y. Chen, and F. E. Alsaadi, "Evaluation of feature selection methods for text classification with small datasets using multiple criteria decision-making methods," *Applied Soft Computing*, vol. 86, p. 105836, 2020.
- [84] S. L. Marie-Sainte and N. Alalyani, "Firefly algorithm based feature selection for arabic text classification," *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 3, pp. 320–328, 2020.
- [85] X. Luo, "Efficient english text classification using selected machine learning techniques," *Alexandria Engineering Journal*, vol. 60, no. 3, pp. 3401–3409, 2021.
- [86] K. N. Singh, S. D. Devi, H. M. Devi, and A. K. Mahanta, "A novel approach for dimension reduction using word embedding: An enhanced text classification approach," *International Journal of Information Management Data Insights*, vol. 2, no. 1, p. 100061, 2022.
- [87] C. Catal and S. Guldan, "Product review management software based on multiple classifiers," *Iet Software*, vol. 11, no. 3, pp. 89–92, 2017.
- [88] A. Onan, S. Korukoğlu, and H. Bulut, "Ensemble of keyword extraction methods and classifiers in text classification," *Expert Systems with Applications*, vol. 57, pp. 232–247, 2016.
- [89] R. Prabowo and M. Thelwall, "Sentiment analysis: A combined approach," *Journal of Informetrics*, vol. 3, no. 2, pp. 143–157, 2009.
- [90] R. Xia, C. Zong, and S. Li, "Ensemble of feature sets and classification algorithms for sentiment classification," *Information Sciences*, vol. 181, no. 6, pp. 1138–1152, 2011.

- [91] N. F. Da Silva, E. R. Hruschka, and E. R. Hruschka Jr, “Tweet sentiment analysis with classifier ensembles,” *Decision Support Systems*, vol. 66, pp. 170–179, 2014.
- [92] G. Wang, J. Sun, J. Ma, K. Xu, and J. Gu, “Sentiment classification: The contribution of ensemble learning,” *Decision support systems*, vol. 57, pp. 77–93, 2014.
- [93] E. Fersini, E. Messina, and F. A. Pozzi, “Sentiment analysis: Bayesian ensemble learning,” *Decision support systems*, vol. 68, pp. 26–38, 2014.
- [94] H. Kwon and S. Lee, “Ensemble transfer attack targeting text classification systems,” *Computers and Security*, p. 102695, 2022.
- [95] R. C. Prati, G. E. Batista, and D. F. Silva, “Class imbalance revisited: a new experimental setup to assess the performance of treatment methods,” *Knowledge and Information Systems*, vol. 45, no. 1, pp. 247–270, 2015.
- [96] 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)*, vol. 42, no. 4, pp. 463–484, 2011.
- [97] Y. Liu, Y. Wang, X. Ren, H. Zhou, and X. Diao, “A classification method based on feature selection for imbalanced data,” *IEEE Access*, 2019.
- [98] V. García, J. S. Sánchez, and R. A. Mollineda, “On the effectiveness of pre-processing methods when dealing with different levels of class imbalance,” *Knowledge-Based Systems*, vol. 25, no. 1, pp. 13–21, 2012.
- [99] V. López, A. Fernández, M. J. Del Jesus, and F. Herrera, “A hierarchical genetic fuzzy system based on genetic programming for addressing classification with highly imbalanced and borderline data-sets,” *Knowledge-Based Systems*, vol. 38, pp. 85–104, 2013.
- [100] B. W. Yap, K. A. Rani, H. A. A. Rahman, S. Fong, Z. Khairudin, and N. N. Abdullah, “An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets,” in *Proceedings of the first international conference on advanced data and information engineering (DaEng-2013)*, pp. 13–22, Springer, 2014.
- [101] S. García and F. Herrera, “Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy,” *Evolutionary computation*, vol. 17, no. 3, pp. 275–306, 2009.

- [102] H. A. Fayed and A. F. Atiya, “A novel template reduction approach for the k -nearest neighbor method,” *IEEE Transactions on Neural Networks*, vol. 20, no. 5, pp. 890–896, 2009.
- [103] M. Kubat, S. Matwin, *et al.*, “Addressing the curse of imbalanced training sets: one-sided selection,” in *Icml*, vol. 97, pp. 179–186, Nashville, USA, 1997.
- [104] H. Niemann and R. Goppert, “An efficient branch-and-bound nearest neighbour classifier,” *Pattern Recognition Letters*, vol. 7, no. 2, pp. 67–72, 1988.
- [105] K. Gowda and G. Krishna, “The condensed nearest neighbor rule using the concept of mutual nearest neighborhood (corresp.),” *IEEE Transactions on Information Theory*, vol. 25, no. 4, pp. 488–490, 1979.
- [106] I. Tomek, “Two modifications of cnn,” *IEEE Trans. Systems, Man and Cybernetics*, vol. 6, pp. 769–772, 1976.
- [107] F. Cheng, J. Zhang, and C. Wen, “Cost-sensitive large margin distribution machine for classification of imbalanced data,” *Pattern Recognition Letters*, vol. 80, pp. 107–112, 2016.
- [108] Z. Sun, Q. Song, X. Zhu, H. Sun, B. Xu, and Y. Zhou, “A novel ensemble method for classifying imbalanced data,” *Pattern Recognition*, vol. 48, no. 5, pp. 1623–1637, 2015.
- [109] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, “Smoteboost: Improving prediction of the minority class in boosting,” in *European conference on principles of data mining and knowledge discovery*, pp. 107–119, Springer, 2003.
- [110] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: synthetic minority over-sampling technique,” *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.
- [111] B. X. Wang and N. Japkowicz, “Imbalanced data set learning with synthetic samples,” in *Proc. IRIS Machine Learning Workshop*, vol. 19, 2004.
- [112] H. He, Y. Bai, E. A. Garcia, and S. Li, “Adasyn: Adaptive synthetic sampling approach for imbalanced learning,” in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pp. 1322–1328, IEEE, 2008.
- [113] H. Han, W.-Y. Wang, and B.-H. Mao, “Borderline-smote: a new over-sampling method in imbalanced data sets learning,” in *International conference on intelligent computing*, pp. 878–887, Springer, 2005.

- [114] S. Chen, G.-D. Guo, and L.-F. Chen, “Clustering ensembles based classification method for imbalanced data sets,” *Pattern Recognition and Artificial Intelligence*, vol. 6, 2010.
- [115] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, “Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem,” in *Pacific-Asia conference on knowledge discovery and data mining*, pp. 475–482, Springer, 2009.
- [116] A. Fernández, V. López, M. Galar, M. J. Del Jesus, and F. Herrera, “Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches,” *Knowledge-based systems*, vol. 42, pp. 97–110, 2013.
- [117] R. Rifkin and A. Klautau, “In defense of one-vs-all classification,” *Journal of machine learning research*, vol. 5, no. Jan, pp. 101–141, 2004.
- [118] T. Hastie and R. Tibshirani, “Classification by pairwise coupling,” in *Advances in neural information processing systems*, pp. 507–513, 1998.
- [119] K. Jiang, J. Lu, and K. Xia, “A novel algorithm for imbalance data classification based on genetic algorithm improved smote,” *Arabian journal for science and engineering*, vol. 41, no. 8, pp. 3255–3266, 2016.
- [120] L. Abdi and S. Hashemi, “To combat multi-class imbalanced problems by means of over-sampling techniques,” *IEEE transactions on Knowledge and Data Engineering*, vol. 28, no. 1, pp. 238–251, 2015.
- [121] T. Zhu, Y. Lin, and Y. Liu, “Synthetic minority oversampling technique for multiclass imbalance problems,” *Pattern Recognition*, vol. 72, pp. 327–340, 2017.
- [122] A. Roy, R. M. Cruz, R. Sabourin, and G. D. Cavalcanti, “A study on combining dynamic selection and data preprocessing for imbalance learning,” *Neurocomputing*, vol. 286, pp. 179–192, 2018.
- [123] J. F. Díez-Pastor, J. J. Rodríguez, C. García-Osorio, and L. I. Kuncheva, “Random balance: ensembles of variable priors classifiers for imbalanced data,” *Knowledge-Based Systems*, vol. 85, pp. 96–111, 2015.
- [124] S. Chen, H. He, and E. A. Garcia, “Ramoboost: ranked minority oversampling in boosting,” *IEEE Transactions on Neural Networks*, vol. 21, no. 10, pp. 1624–1642, 2010.

- [125] J. Bi and C. Zhang, “An empirical comparison on state-of-the-art multi-class imbalance learning algorithms and a new diversified ensemble learning scheme,” *Knowledge-Based Systems*, vol. 158, pp. 81–93, 2018.
- [126] Y. Zhang, B. Liu, J. Cai, and S. Zhang, “Ensemble weighted extreme learning machine for imbalanced data classification based on differential evolution,” *Neural Computing and Applications*, vol. 28, no. 1, pp. 259–267, 2017.
- [127] S. Maldonado, J. López, and C. Vairetti, “An alternative smote oversampling strategy for high-dimensional datasets,” *Applied Soft Computing*, vol. 76, pp. 380–389, 2019.
- [128] P. John, “Van de geer, “some aspects of minkowski distance”, department of data theory, leiden university,” tech. rep., RR-95-03.
- [129] L. Ma and S. Fan, “Cure-smote algorithm and hybrid algorithm for feature selection and parameter optimization based on random forests,” *BMC bioinformatics*, vol. 18, no. 1, p. 169, 2017.
- [130] A. Rojas-Domínguez, L. C. Padierna, J. M. C. Valadez, H. J. Puga-Soberanes, and H. J. Fraire, “Optimal hyper-parameter tuning of svm classifiers with application to medical diagnosis,” *IEEE Access*, vol. 6, pp. 7164–7176, 2017.
- [131] M. Pavlinek and V. Podgorelec, “Text classification method based on self-training and lda topic models,” *Expert Systems with Applications*, vol. 80, pp. 83–93, 2017.
- [132] X. Zhang, X. Chen, and Z. He, “An aco-based algorithm for parameter optimization of support vector machines,” *Expert Systems with Applications*, vol. 37, no. 9, pp. 6618–6628, 2010.
- [133] I. Aljarah, H. Faris, S. Mirjalili, and N. Al-Madi, “Training radial basis function networks using biogeography-based optimizer,” *Neural Computing and Applications*, vol. 29, no. 7, pp. 529–553, 2018.
- [134] I. Aljarah, H. Faris, and S. Mirjalili, “Optimizing connection weights in neural networks using the whale optimization algorithm,” *Soft Computing*, vol. 22, no. 1, pp. 1–15, 2018.
- [135] A. V. Phan, M. Le Nguyen, and L. T. Bui, “Feature weighting and svm parameters optimization based on genetic algorithms for classification problems,” *Applied Intelligence*, vol. 46, no. 2, pp. 455–469, 2017.

- [136] H. T. Ibrahim, W. J. Mazher, O. N. Ucan, and O. Bayat, "A grasshopper optimizer approach for feature selection and optimizing svm parameters utilizing real biomedical data sets," *Neural Computing and Applications*, vol. 31, no. 10, pp. 5965–5974, 2019.
- [137] R. Hans and H. Kaur, "Binary multi-verse optimization (bmvo) approaches for feature selection.," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 1, 2020.
- [138] Y. Wan, M. Wang, Z. Ye, and X. Lai, "A feature selection method based on modified binary coded ant colony optimization algorithm," *Applied Soft Computing*, vol. 49, pp. 248–258, 2016.
- [139] I. Aljarah, A.-Z. Ala'M, H. Faris, M. A. Hassonah, S. Mirjalili, and H. Saadeh, "Simultaneous feature selection and support vector machine optimization using the grasshopper optimization algorithm," *Cognitive Computation*, vol. 10, no. 3, pp. 478–495, 2018.
- [140] X. Shi, Q. Liu, W. Fan, S. Y. Philip, and R. Zhu, "Transfer learning on heterogeneous feature spaces via spectral transformation," in *2010 IEEE international conference on data mining*, pp. 1049–1054, IEEE, 2010.
- [141] P. Prettenhofer and B. Stein, "Cross-language text classification using structural correspondence learning," in *Proceedings of the 48th annual meeting of the association for computational linguistics*, pp. 1118–1127, 2010.
- [142] C. Wang and S. Mahadevan, "Heterogeneous domain adaptation using manifold alignment," in *Twenty-second international joint conference on artificial intelligence*, 2011.
- [143] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, p. 9, 2016.
- [144] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.
- [145] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proceedings of the 24th international conference on Machine learning*, pp. 193–200, 2007.
- [146] D. Pardoe and P. Stone, "Boosting for regression transfer," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pp. 863–870, 2010.

- [147] C. Wan, R. Pan, and J. Li, “Bi-weighting domain adaptation for cross-language text classification,” in *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.
- [148] S. Si, D. Tao, and B. Geng, “Bregman divergence-based regularization for transfer subspace learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 7, pp. 929–942, 2009.
- [149] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, “Domain adaptation via transfer component analysis,” *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2010.
- [150] Y. Xu, S. J. Pan, H. Xiong, Q. Wu, R. Luo, H. Min, and H. Song, “A unified framework for metric transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 6, pp. 1158–1171, 2017.
- [151] D. Wang, C. Lu, J. Wu, H. Liu, W. Zhang, F. Zhuang, and H. Zhang, “Softly associative transfer learning for cross-domain classification,” *IEEE transactions on cybernetics*, 2019.
- [152] M. Sato, R. Orihara, Y. Sei, Y. Tahara, and A. Ohsuga, “Text classification and transfer learning based on character-level deep convolutional neural networks,” in *International Conference on Agents and Artificial Intelligence*, pp. 62–81, Springer, 2017.
- [153] J. Li, K. Lu, Z. Huang, L. Zhu, and H. T. Shen, “Transfer independently together: A generalized framework for domain adaptation,” *IEEE transactions on cybernetics*, vol. 49, no. 6, pp. 2144–2155, 2018.
- [154] J. Li, K. Lu, Z. Huang, L. Zhu, and H. T. Shen, “Heterogeneous domain adaptation through progressive alignment,” *IEEE transactions on neural networks and learning systems*, vol. 30, no. 5, pp. 1381–1391, 2018.
- [155] J. Li, E. Chen, Z. Ding, L. Zhu, K. Lu, and H. T. Shen, “Maximum density divergence for domain adaptation,” *IEEE transactions on pattern analysis and machine intelligence*, 2020.
- [156] J. Li, M. Jing, K. Lu, L. Zhu, and H. T. Shen, “Locality preserving joint transfer for domain adaptation,” *IEEE Transactions on Image Processing*, vol. 28, no. 12, pp. 6103–6115, 2019.
- [157] C. Sun and K.-M. Lam, “Multiple-kernel, multiple-instance similarity features for efficient visual object detection,” *IEEE transactions on image processing*, vol. 22, no. 8, pp. 3050–3061, 2013.

- [158] L. Demidova, E. Nikulchev, and Y. Sokolova, “The svm classifier based on the modified particle swarm optimization,” *arXiv preprint arXiv:1603.08296*, 2016.
- [159] K.-C. Lin, K.-Y. Zhang, Y.-H. Huang, J. C. Hung, and N. Yen, “Feature selection based on an improved cat swarm optimization algorithm for big data classification,” *The Journal of Supercomputing*, vol. 72, no. 8, pp. 3210–3221, 2016.
- [160] V. Panchal, P. Singh, N. Kaur, and H. Kundra, “Biogeography based satellite image classification,” *arXiv preprint arXiv:0912.1009*, 2009.
- [161] V. Panchal, H. Kundra, and A. Kaur, “An integrated approach to biogeography based optimization with case based reasoning for retrieving groundwater possibility,” *International Journal of Computer Applications*, vol. 1, no. 8, pp. 975–8887, 2010.
- [162] A. Bhattacharya and P. K. Chattopadhyay, “Solving complex economic load dispatch problems using biogeography-based optimization,” *Expert Systems with Applications*, vol. 37, no. 5, pp. 3605–3615, 2010.
- [163] D. Simon, “A probabilistic analysis of a simplified biogeography-based optimization algorithm,” *Evolutionary computation*, vol. 19, no. 2, pp. 167–188, 2011.
- [164] M. M. Mirończuk and J. Protasiewicz, “A recent overview of the state-of-the-art elements of text classification,” *Expert Systems with Applications*, 2018.
- [165] A. K. Das, S. Sengupta, and S. Bhattacharyya, “A group incremental feature selection for classification using rough set theory based genetic algorithm,” *Applied Soft Computing*, vol. 65, pp. 400–411, 2018.
- [166] I. Fister Jr, X.-S. Yang, I. Fister, J. Brest, and D. Fister, “A brief review of nature-inspired algorithms for optimization,” *arXiv preprint arXiv:1307.4186*, 2013.
- [167] P.-F. Pai, C.-T. Chen, Y.-M. Hung, W.-Z. Hung, and Y.-C. Chang, “A group decision classifier with particle swarm optimization and decision tree for analyzing achievements in mathematics and science,” *Neural Computing and Applications*, vol. 25, no. 7-8, pp. 2011–2023, 2014.
- [168] W. Gong, Z. Cai, and C. X. Ling, “De/bbo: a hybrid differential evolution with biogeography-based optimization for global numerical optimization,” *Soft Computing*, vol. 15, no. 4, pp. 645–665, 2010.
- [169] Z. Boynukalin, “Emotion analysis of turkish texts by using machine learning methods,” *Middle East Technical University*, 2012.

- [170] A. Sboev, T. Litvinova, D. Gudovskikh, R. Rybka, and I. Moloshnikov, “Machine learning models of text categorization by author gender using topic-independent features,” *Procedia Computer Science*, vol. 101, pp. 135–142, 2016.
- [171] A. A. Akinyelu and A. O. Adewumi, “Classification of phishing email using random forest machine learning technique,” *Journal of Applied Mathematics*, vol. 2014, 2014.
- [172] S. Mehta *et al.*, “Concept drift in streaming data classification: Algorithms, platforms and issues,” *Procedia computer science*, vol. 122, pp. 804–811, 2017.
- [173] T. Sabbah, A. Selamat, M. H. Selamat, F. S. Al-Anzi, E. H. Viedma, O. Krejcar, and H. Fujita, “Modified frequency-based term weighting schemes for text classification,” *Applied Soft Computing*, vol. 58, pp. 193–206, 2017.
- [174] L. Liu, Y. Cai, W. Lu, K. Feng, C. Peng, and B. Niu, “Prediction of protein–protein interactions based on psea composition and hybrid feature selection,” *Biochemical and biophysical research communications*, vol. 380, no. 2, pp. 318–322, 2009.
- [175] M. D. Del Castillo and J. I. Serrano, “A multistrategy approach for digital text categorization from imbalanced documents,” *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 70–79, 2004.
- [176] C. Phua, D. Alahakoon, and V. Lee, “Minority report in fraud detection: classification of skewed data,” *Acm sigkdd explorations newsletter*, vol. 6, no. 1, pp. 50–59, 2004.
- [177] T. R. Hoens, Q. Qian, N. V. Chawla, and Z.-H. Zhou, “Building decision trees for the multi-class imbalance problem,” in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 122–134, Springer, 2012.
- [178] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, “Learning from class-imbalanced data: Review of methods and applications,” *Expert Systems with Applications*, vol. 73, pp. 220–239, 2017.
- [179] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [180] S. González-Carvajal and E. C. Garrido-Merchán, “Comparing bert against traditional machine learning text classification,” *arXiv preprint arXiv:2005.13012*, 2020.

- [181] B. S. Khehra and A. P. S. Pharwaha, “Comparison of genetic algorithm, particle swarm optimization and biogeography-based optimization for feature selection to classify clusters of microcalcifications,” *Journal of The Institution of Engineers (India): Series B*, vol. 98, no. 2, pp. 189–202, 2017.
- [182] M. Karnan and K. Thangavel, “Automatic detection of the breast border and nipple position on digital mammograms using genetic algorithm for asymmetry approach to detection of microcalcifications,” *Computer methods and programs in biomedicine*, vol. 87, no. 1, pp. 12–20, 2007.
- [183] K. Yan and D. Zhang, “Feature selection and analysis on correlated gas sensor data with recursive feature elimination,” *Sensors and Actuators B: Chemical*, vol. 212, pp. 353–363, 2015.
- [184] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, “Dbsmote: density-based synthetic minority over-sampling technique,” *Applied Intelligence*, vol. 36, no. 3, pp. 664–684, 2012.
- [185] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, “Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework.,” *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17, 2011.
- [186] A. Frank, A. Asuncion, *et al.*, “Uci machine learning repository, 2010,” *URL <http://archive.ics.uci.edu/ml>*, vol. 15, p. 22, 2011.
- [187] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, “Scikit-learn: Machine learning in python,” *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.
- [188] G. Douzas, F. Bacao, and F. Last, “Improving imbalanced learning through a heuristic oversampling method based on k-means and smote,” *Information Sciences*, vol. 465, pp. 1–20, 2018.
- [189] D. J. Hand and R. J. Till, “A simple generalisation of the area under the roc curve for multiple class classification problems,” *Machine learning*, vol. 45, no. 2, pp. 171–186, 2001.
- [190] Y. Sun, M. S. Kamel, and Y. Wang, “Boosting for learning multiple classes with imbalanced class distribution,” in *Sixth International Conference on Data Mining (ICDM’06)*, pp. 592–602, IEEE, 2006.

- [191] M. Kumar, M. Jindal, and R. Sharma, “k-nearest neighbor based offline handwritten gurmukhi character recognition,” in *2011 International Conference on Image Information Processing*, pp. 1–4, IEEE, 2011.
- [192] X.-F. Zhong, S.-Z. Guo, L. Gao, H. Shan, and J.-H. Zheng, “An improved k-nn classification with dynamic k,” in *Proceedings of the 9th International Conference on Machine Learning and Computing*, pp. 211–216, 2017.
- [193] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown, “Auto-weka: Combined selection and hyperparameter optimization of classification algorithms,” in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 847–855, 2013.
- [194] A. Khurana and O. P. Verma, “Pso based optimal text classification using tuned k-nn and feature weighting,” *International Journal of Information Systems and Management Science*, vol. 1, no. 1, 2018.
- [195] T. A. Gomes, R. B. Prudêncio, C. Soares, A. L. Rossi, and A. Carvalho, “Combining meta-learning and search techniques to select parameters for support vector machines,” *Neurocomputing*, vol. 75, no. 1, pp. 3–13, 2012.
- [196] M. Lichman *et al.*, “Uci machine learning repository,” 2013.
- [197] J. Blitzer, S. Kakade, and D. Foster, “Domain adaptation with coupled subspaces,” in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 173–181, JMLR Workshop and Conference Proceedings, 2011.
- [198] L. Britto, R. Lima, and L. Pacífico, “Structural correspondence learning for cross-domain sentiment analysis in brazilian portuguese,” in *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*, pp. 812–817, IEEE, 2019.
- [199] X. Fang, N. Han, G. Zhou, S. Teng, Y. Xu, and S. Xie, “Dynamic double classifiers approximation for cross-domain recognition,” *IEEE Transactions on Cybernetics*, 2020.
- [200] H. Fan, S. Gao, X. Zhang, X. Cao, H. Ma, and Q. Liu, “Intelligent recognition of ferrographic images combining optimal cnn with transfer learning introducing virtual images,” *IEEE Access*, vol. 8, pp. 137074–137093, 2020.
- [201] L. A. Pereira and R. da Silva Torres, “Semi-supervised transfer subspace for domain adaptation,” *Pattern Recognition*, vol. 75, pp. 235–249, 2018.

- [202] M. Long, Z. Cao, J. Wang, and M. I. Jordan, “Conditional adversarial domain adaptation,” in *Advances in Neural Information Processing Systems*, pp. 1640–1650, 2018.
- [203] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, “A theory of learning from different domains,” *Machine learning*, vol. 79, no. 1-2, pp. 151–175, 2010.
- [204] S. Mirjalili, “The ant lion optimizer,” *Advances in engineering software*, vol. 83, pp. 80–98, 2015.
- [205] S. K. Majhi and S. Biswal, “Optimal cluster analysis using hybrid k-means and ant lion optimizer,” *Karbala International Journal of Modern Science*, vol. 4, no. 4, pp. 347–360, 2018.
- [206] E. Emary and H. M. Zawbaa, “Feature selection via lèvy antlion optimization,” *Pattern Analysis and Applications*, vol. 22, no. 3, pp. 857–876, 2019.
- [207] S. Strassel and A. W. Cole, “Corpus development and publication,” *Proceedings of LREC, Genoa, Italy*. <http://papers.ldc.upenn.edu/LREC2006/CorpusDevelopmentAndPublication.pdf>, 2006.
- [208] M. Krallinger, O. Rabal, A. Lourenco, J. Oyarzabal, and A. Valencia, “Information retrieval and text mining technologies for chemistry,” *Chemical reviews*, vol. 117, no. 12, pp. 7673–7761, 2017.
- [209] S. Singh and S. Kaur, “Improved spambase dataset prediction using svm rbf kernel with adaptive boost,” *Int J Res Eng Technol*, vol. 4, no. 6, pp. 383–386, 2015.
- [210] S. M. Abdulhamid, M. Shuaib, O. Osho, I. Ismaila, and J. K. Alhassan, “Comparative analysis of classification algorithms for email spam detection,” *International Journal of Computer Network and Information Security*, vol. 10, no. 1, 2018.
- [211] H. M. Alshamlan, G. H. Badr, and Y. A. Alohal, “Abc-svm: artificial bee colony and svm method for microarray gene selection and multi class cancer classification,” *Int. J. Mach. Learn. Comput*, vol. 6, no. 3, p. 184, 2016.
- [212] N. Segev, M. Harel, S. Mannor, K. Crammer, and R. El-Yaniv, “Learn on source, refine on target: A model transfer learning framework with random forests,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 9, pp. 1811–1824, 2016.

- [213] R. Eberhart and J. Kennedy, “Particle swarm optimization,” in *Proceedings of the IEEE international conference on neural networks*, vol. 4, pp. 1942–1948, Citeseer, 1995.
- [214] X. Zhong, S. Guo, H. Shan, L. Gao, D. Xue, and N. Zhao, “Feature-based transfer learning based on distribution similarity,” *IEEE Access*, vol. 6, pp. 35551–35557, 2018.
- [215] F. Zhuang, P. Luo, P. Yin, Q. He, and Z. Shi, “Concept learning for cross-domain text classification: A general probabilistic framework,” in *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.
- [216] A. Khurana and O. P. Verma, “A fine tuned model of grasshopper optimization algorithm with classifiers for optimal text classification,” in *2020 IEEE 17th India Council International Conference (INDICON)*, pp. 1–7, IEEE, 2020.