THE REPUBLIC OF TURKEY
BAHÇEŞEHİR UNIVERSITY

ENSEMBLE PRUNING USING OPTIMIZATION MODELING

Ph.D. Thesis

PINAR KARADAYI ATAŞ

İSTANBUL, 2020

THE REPUBLIC OF TURKEY
BAHÇEŞEHİR UNIVERSITY

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
COMPUTER ENGINEERING

ENSEMBLE PRUNING USING OPTIMIZATION MODELING

Ph.D. Thesis

PINAR KARADAYI ATAŞ

Supervisor: ASSOC. PROF. DR. SÜREYYA ÖZÖĞÜR - AKYÜZ

İSTANBUL, 2020

THE REPUBLIC OF TURKEY
BAHÇEŞEHİR UNIVERSITY
The Graduate School of Natural and Applied Sciences
Computer Engineering

Title of the Ph.D. Thesis : Ensemble Pruning Using Optimization Modeling
Name/Last Name of the Student : Pınar KARADAYI ATAŞ
Date of Thesis Defense : 19 August, 2020

The thesis has been approved by The Graduate School of Natural and Applied Sciences.

Assoc. Prof. Dr. Burak KÜNTAY
Acting Director

I certify that this thesis meets all the requirements as a thesis for the degree of Doctor of Philosophy.

Assist. Prof. Dr. Tarkan AYDIN
Program Coordinator

This is to certify that we have read this thesis and that we find it fully adequate in scope, quality and content, as a thesis for the degree of Doctor of Philosophy in Computer Engineering Department.

Examining Commitee Members: Signature

Assoc. Prof. Dr. Süreyya ÖZÖĞÜR - AKYÜZ (Supervisor) : ....

Prof.Dr. Ayhan DEMİRİZ : ....

Prof. Dr. Mehmet Alper TUNGA : ....

Assoc. Prof. Dr. Birsen EYGİ ERDOĞAN : ....

Assist. Prof. Dr. Tarkan AYDIN : ....

ACKNOWLEDGEMENTS

I am grateful to my father Harun, my mother Zehra, my husband Hakan and his familiy Mustafa, Huriye and Fatih who have provided me opportunities, support and encouragement all the time. Your belief in me keeps me motivated.

To my supervisor Assoc. Prof. Dr. Süreyya Özöğür - Akyüz, thank you for giving me the chance that I can prove myself. Her guidance and support made this thesis possible. I am honoured to work with a scientist who is motivated to introduce her students new ideas and new perspectives.

This study is granted by Scientific and Technological Research Council of Turkey (TÜBİTAK) Project No:116E064

İstanbul, 2020 Pınar KARADAYI ATAŞ

ABSTRACT

ENSEMBLE PRUNING USING OPTIMIZATION MODELING

Pınar KARADAYI ATAŞ

Computer Engineering
Supervisor: Assoc. Prof. Dr. Süreyya ÖZÖĞÜR - AKYÜZ

August 2020, 67 Pages

The performance problem of ensemble clustering in unsupervised learning is a huge concern in data-mining and machine-learning communities. The most crucial concern is diversity and accuracy, both of which determine the outcome of predictive performance. While some great minds have increased the diversity or accuracy of component classifiers to boost performance, some have utilized or manipulated these two metrics to generate excellent ensemble results. This thesis suggests a new clustering ensemble selection model that can overcome some of the ensemble-clustering performance limitations noted in the literature, meaning our goal is to considerably enhance existing ensemble-clustering models. More specifically, we designed our new ensemble model to satisfy the diversity-and-accuracy trade-off and used eleven datasets for each of the three cluster-ensemble methods for comparison. We also ensured that our algorithm did not depend on the data domain. Not only did we realize that diversity or accuracy alone cannot enhance performance, but we also noticed that the cardinality of the ensemble subsets was an important parameter in obtaining better results. After testing and comparing our technique with recent clustering techniques in terms of the number of cardinalities, we found that compared to other ensemble methods, our proposed ensemble-selection method resulted in performance enhancement in terms of providing a better accuracy to a particular problem. Besides that, the proposed methodology was adapted and re-modeled for feature selection problem, which is one of the steps in data pre-processing. In recent years, ensemble based feature selection approaches have been proposed in which, multiple diverse feature selection methods are combined. The proposed algorithm was tested on multiple data sets and learning performances are compared with various feature selection algorithms. The empirical results show that the proposed algorithm performs at higher classification accuracy.

Keywords: Feature Selection, Ensemble Learning, Ensemble Pruning, Clustering, Convex Concave Programming,Dynamic Ensemble Selection(DES)

ÖZET

OPTİMİZASYON MODELLEMESİ KULLANARAK TOPLULUK BUDAMASI

Pınar KARADAYI ATAŞ

Bilgisayar Mühendisliği
Tez Danışmanı: Doç. Dr. Süreyya ÖZÖĞÜR - AKYÜZ

Ağustos 2020, 67 Sayfa

Denetimsiz öğrenmede topluluk kümelenmesinin performans sorunu, veri madenciliği ve makine öğrenimi topluluklarında büyük bir endişe kaynağıdır. En önemli endişe, tahmini performansın sonucunu belirleyen çeşitlilik ve doğruluktur. Bazı büyük fikirler performansı artırmak için bileşen sınıflandırıcılarının çeşitliliğini veya doğruluğunu artırırken, bazıları mükemmel topluluk sonuçları oluşturmak için bu iki metriği kullanmış veya manipüle etmiştir. Bu tez, literatürde belirtilen bazı topluluk kümelenme performans sınırlamalarının üstesinden gelebilecek yeni bir kümelenme topluluğu seçim modelini önermektedir, yani amacımız mevcut topluluk kümeleme modellerini önemli ölçüde artırmaktır. Daha spesifik olarak, yeni topluluk modelimizi çeşitlilik ve doğruluk ödünleşmesini karşılamak için tasarladık ve karşılaştırma için üç küme topluluğu yönteminin her biri için onbir veri seti kullandık. Ayrıca algoritmamızın veri alanına bağlı olmamasını sağladık. Sadece çeşitliliğin veya doğruluğun tek başına performansı artıramayacağını değil, ayrıca topluluk alt kümelerinin kardinalitesinin iyi sonuçlar edinmek için önemli bir parametre olduğunu fark ettik. Tekniklerimizi kardinalite sayısı açısından son kümelenme teknikleriyle test ettikten ve karşılaştırdıktan sonra, diğer topluluk yöntemleriyle karşılaştırıldığında, önerilen topluluk seçme yöntemimizin daha iyi bir doğruluk sağlama açısından performans artışı sağladığını tespit ettik. Bunun yanında, önerilen metodoloji, veri ön işlemedeki adımlardan olan özellik seçim problemi için uyarlanmış, yeniden modellenmiştir. Son yıllarda, çeşitli özellik seçme yöntemlerinin birleştirildiği topluluk temelli özellik seçme yaklaşımları önerilmiştir. Önerilen algoritma birden fazla veri seti üzerinde test edilmiştir ve öğrenme performansları çeşitli özellik seçim algoritmaları ile karşılaştırılmıştır. Ampirik sonuçlar, önerilen algoritmanın yüksek sınıflandırma doğruluğunda performans elde ettiğini göstermektedir.

Anahtar Kelimeler: Öznitelik Seçimi, Topluluk Öğrenimi, Topluluk Budaması, Kümeleme, Dış Bükey İç Bükey Programlama, Dinamik Topluluk Seçimi.

CONTENTS

TABLES

FIGURES

ABBREVIATIONS

ANOVA	:	Multi-Factor Analysis-of-Variance
APMM	:	Alizadeh Parvin Moshki Minaei Criterion
Auto-CES	:	Automatic Clustering Ensemble Selection
CCP	:	Convex- Concave Programming
CFS	:	Correlation Based Feature Selection
CIFE	:	Conditional Infomax Feature Extraction
CLA	:	Complete Linear Aggregation
CMIM	:	Conditional Mutual Information Maximization
CSPA	:	Cluster based Similarity Algorithm
DC	:	Difference Programming
DCCP	:	Disciplined Convex-Concave Programming
DCP	:	Disciplined Convex Programming
DCS	:	Dynamic Classifier Selection Methods
DES	:	Dynamic Ensemble Selection
DISR	:	Double Input Symmetrical Relevance
ECI	:	Ensemble-Based Cluster Validity Index
ENMI	:	Edited Normalized Mutual Information
FCBF	:	Fast Correlation Based Filter
FWLAC	:	Fuzzy Weighted Locally Adaptive Clustering
GKCC	:	Greedy Optimization of K-means-based Consensus Clustering
HGPA	:	Hypergraph Partitioning Algorithm
HMM	:	Hidden Markov Model
JMI	:	Joint Mutual Information
KCC	:	K-means Based Consensus Clustering
LS	:	Laplacian Score
LWGP	:	Locally Weighted Evidence Accumulation
MCFS	:	Multi-Cluster Feature Selection

MCLA	:	Meta-Clustering Algorithm
MIFS	:	Mutual Information Feature Selection
MIM	:	Mutual Information Maximization
MRMR	:	Minimum Redundancy Maximum Relevance
NDFS	:	Nonnegative Discriminative Feature Selection
NMI	:	Normalized Mutual Information
OCT	:	Optimal Classification Trees
PSO	:	Particle Swarm Optimization
RBF	:	Radial Basis Function
Rob-EFS	:	Robust Ensemble Feature Selection
SFR	:	Supervised Ensemble Learning Guided Feature Ranking
SPEC	:	Spectral Feature Selection
SVM	:	Support Vector Machines
SVMS	:	Support Vector Machines
U-SENC	:	Ultra-Scalable Ensemble Clustering
U-SPEC	:	Ultra-Scalable Spectral Clustering
WLAC	:	Weighted Locally Adaptive Clustering

SYMBOLS

A closed convex set	:	X
Quality and diversity matrix for ensemble cluster selection	:	G
Accuracy and diversity matrix for ensemble feature selection	:	T
Bias term	:	b
Composed of the different attribute sets	:	S_i^f
Dot product	:	$<,>$
Ensemble size	:	K
Entropy function of cluster X	:	$H(X)$
Entropy function of cluster Y	:	$H(Y)$
Marginal distribution function of cluster X	:	$p(x)$
Marginal distribution function of cluster Y	:	$p(y)$
Non zero weight	:	l_1
Number of clustering solutions	:	N
Pruning rate of the ensemble	:	k
Regularization (cost) constant	:	C
Regularization constant	:	ρ
Set of Labels	:	Y
The mutual information function between X and Y clusters	:	$I(X,Y)$
The normal vector of the decision surface	:	w
The number of clustering solutions in the ensemble	:	N

1. INTRODUCTION

The advent of information technology (IT) has resulted in the explosion of data, which needs to be transformed and saved into meaningful information. This transformation requires smart data-cleaning techniques that generate minimal errors. The good news is that cloud computing now offers users cheap data storage. Even though today's huge data can be stored efficiently on the cloud, their behaviors need to be fully understood with data mining strategies. Data mining refers to an analytic technique used to extract useful patterns, structures or details from a plethora of raw data. A vast majority of data-mining processes are not performed manually but automatically with software applications such as Sisense, Rapid Miner, Orange, and SSDT (SQL Server Data Tools). Regardless of the data mining tools used, all data mining procedures can be categorized into three groups: clustering, classification, and association rules.

Furthermore, a successful data mining project cannot be achieved without carrying out data clustering analysis, a learning technique used in statistical data analysis to classify data points into desired groups based on what they have in common. By combining different component partitions into one output, clustering algorithms can considerably boost the quality of the final partitions. Clustering analysis has several applications, and it can be employed to find underlying patterns, common meaningful structures, and similar generative features among large amounts of data. In the medical field, for instance, clustering techniques be utilized to find patient segments with similar disease symptoms. A typical clustering algorithm is K-means clustering, which begins with finding similarities based on the location of the data and then classifying them into different groups to produce a consensus solution. Other clustering algorithms used in the literature are the fuzzy C-means (FCM) algorithm, the expectation-maximization (EM) algorithm, and the hierarchical clustering algorithm (Ayad, 2008; Fred &

Jain, 2005; Strehl & Ghosh, 2002; Topchy et al., 2004a). Like all algorithms, each of these clustering algorithms has its benefits and drawbacks.

However, a fundamental problem of clustering is that different cluster ensemble selection algorithms can produce various clustering results, some of which are unreliable (Handl & Knowles, 2007). This means that it is not an easy job in selecting the appropriate ensemble algorithms for the same problem. This problem can be resolved with a cluster ensemble, an efficient method for obtaining predictive performance. Cluster ensemble learning can be referred to as a machine learning technique in which the decisions of multiple base learners are combined to generate better predictions than the ones made by a single base learner. Ensemble learning models are advantageous compared to ordinary machine learning approaches in that they eliminate variance, noise, and biases, all of which distort the final results.

Apart from that, research has confirmed the superiority of ensemble methods to other traditional machine-learning methods, especially in predictive learning. Compared to other learning methods, ensemble learning methods provide an excellent performance in implementing a large number of hypotheses. This is because ensemble models use diverse datasets to predict accurate results. Besides, in contrast to other statistical methods, ensemble models are flexible and cost-efficient. They prune or eliminate redundant hypotheses to produce excellent predictions for problems.

Despite these benefits, existing ensemble models are not immune to problems. Not only do they require a copious amount of computational power, but they also consume lots of memory space. Data over-fitting or under-fitting is another issue of this machine learning paradigm. The effectiveness of ensemble learning algorithms relies on the diversity and quality of ensemble members. Yet only a few studies have investigated how to optimize these two factors. More difficult is even finding the optimal number of classifiers that can solve the same machine problem. While some scientists have found the best performance for a certain ensemble size, no research has focused on the optimal number of component subsets for an ensemble size.

In recent times, researchers have uncovered how to select diverse, accurate solutions from a repository of accurate clustering solutions. However, no one has investigated the optimum number of subset classifiers for an ensemble size. Some statistical tests can determine how many components are required for a solution, yet the law of diminishing returns limits the prediction accuracy as the cardinality of the subsets increases. In other words, there is a trade-off between diversity and accuracy. The greater the diversity of component classifiers, the lesser the quality of the algorithms and vice versa. The ensemble size is another problem. Researchers are yet to find the optimum size of an ensemble subset that has the best effect on prediction accuracy. To achieve this, most of the redundancies in the ensemble solutions must be eliminated. While finding the best model that optimizes both the accuracy of predictions and the diversity of combined models is the objective of this thesis, the final results depend on other important parameters.

The advantages of determining the right number of component subsets in an ensemble construction are numerous yet attractive. There will be less redundancy, variance, and noises, meaning that less computation power will be needed for computers. This also means that predictions will be faster, and there will be an increase in the overall performance of multiple models. In this thesis, our aim is to fill the vacuum in the literature by proposing a new ensemble model that optimizes the number of component classifiers and at the same time produces improved accuracy. Put simply, our priority in this thesis is to find the ideal cardinality number of subsets for an ensemble size. Clustering ensemble methods will be employed to achieve the goal of this research. The final results of this study will be compared with the findings of similar studies done in the past.

Feature Selection is the process of choosing the most relevant and important features which contribute to learning with the highest prediction accuracy. Feature selection methods have various applications (Forman, 2003; Inza et al., 2004), the determination of which constitutes the data pre-processing step of machine learning problems. It is important to eliminate irrelevant features which do not have any dependency on the target value since those features reduce the prediction accuracy of the learning model. There exist many feature selection methods in the literature, including filters based on distinct metrics like probability, entropy, information theory, embedded, and wrapper methods, all using different algorithms (Bolón-Canedo et al., 2016). Most feature selection methods are wrapper methods, which evaluate the features using the learning algorithm. Algorithms based on the filter model examine the intrinsic properties of the data to evaluate the features before the learning tasks. Filter-based approaches almost always rely on class labels, most commonly assessing correlations between features and class label. Some typical filter methods include data variance, Pearson correlation coefficients, Fisher score, and the Kolmogorov-Smirnov test.

Ensemble based feature selection methods are designed to generate an optimum subset of features by combining multiple feature selectors based on the intuition behind ensemble learning. The general idea of ensemble feature selection is to aggregate the decisions of diverse feature selection algorithms to improve representation ability. Recent studies show that the decision of an ensemble of feature selection algorithms gives more accurate predictions than any single feature selection technique (Özöğür-Akyüz et al., 2015; Zhang et al., 2006a).

Ensemble based feature selection methods involve two major steps: generation of diverse feature selectors and aggregation of the decisions. There are three types of generation approaches studied in the literature employed to construct a diverse ensemble library, which can be listed as follows:

a) Data Variation Methods,

b) Function Variation Methods,

c) Hybrid Variation Methods.

The first approach, *Data Variation*, creates subsets of samples by using different methods, such as bagging (Breiman, 1996) or boosting (Freund & Schapire, 1997) or using different feature subspaces and random subspaces (Barandiaran, 1998). In the second method, *Function Variation*, the diversity of an ensemble is provided by the diversity of feature selection functions. Here, the most common functions are filter based rather than wrapper approaches because of their advantages in computational cost. Unlike the first two methods, *Hybrid Variation Methods* aggregate both data variation and function variation steps since it is argued that including data variation or function variation alone is not enough to create a robust ensemble (Guan et al., 2014). In (Dittman et al., 2012), the similarity between the function variation and hybrid variation is higher than the similarity between the data variation. Furthermore, function and hybrid variation methods produce higher classification performance than data variation.

In this study, we propose a novel ensemble based feature selection algorithm that fills the gap in the literature regarding feature selection problems, described above, by using an optimization model to simultaneously optimize the accuracy and diversity trade-off. Since the pruning step of the proposed approach here involves an optimization model, the cardinality of the subset of an ensemble is not a hyper-parameter anymore, as it is obtained directly as a solution of the optimization model.

This remainder of this thesis is divided into six sections. The section 2 will explore the literature review on this research topic. The chapter 3 will explain the methodology used.The mathematical model and experimental results for ensemble clustering selection is presented in the chapter 4 , while the chapter 5 will explain the mathematical model and experimental results for ensemble based feature selection. We conclude with some final remarks and ideas for future work in last chapter.

2. LITERATURE REVIEW

Various mathematical programming methods have employed in the literature, such as statistical methods and stochastic process techniques. Yet each traditional or heuristic optimization algorithm has its cons and pros, meaning that each one is proficient or deficient in one aspect or the other. Traditional optimization algorithms are more flexible and they require the successive interaction of previous or initial solutions to obtain optimal solutions, while heuristic optimization algorithms rely on shortcuts or rules-of-thumb to generate sufficient, but not the best, solutions Ali & Gubran (2002). The main advantage of heuristic optimization algorithms over traditional optimization algorithms is that the former is more practical and faster in producing solutions Ali & Gubran (2002). Nonetheless, most optimization techniques can produce maximum or minimum values for a target function.

Scientists have developed such heuristic optimization algorithms as simulated annealing algorithms, particle swarm (PSO) algorithms and genetic algorithms Liu et al. (2019). PSO algorithms carry out several iterations to produce multiple solutions. That is, the solution for the next iteration is usually better than that for the previous one until the best solution is generated Blondin (2009). Unlike PSO algorithms which start with starts with an initial solution, simulated annealing algorithms pick a random variable rather than the variable to generate optimum global-optima solutions. Totally different from others is the genetic algorithms, which select the fittest variables for simulation. Genetic algorithms are analogous to Darwin's theory of natural selection in that only the best and the fittest parameters are utilized to create high-quality solutions.

The classification of data is a critical step in machine learning. Labeled data are called supervised learning, while the unlabeled ones or data with undefined characteristics are referred to as unsupervised learning Caruana & Niculescu-

Mizil (2006); Hadjitodorov et al. (2006); Margineantu & Dietterich (1997). The major roadblock of ensemble models includes variance, noise, and bias issues, all of which distort the performance of ensemble learning Fern & Lin (2008); Hadjitodorov et al. (2006). Prominent scientists like Fern & Lin (2008) have invented new ways of machine learning with an eye towards uniting the cost function of various objectives. Nonetheless, there is no information in the literature regarding any learning model that can meet the requirements of different objectives at the same time Liu et al. (2019).

Researchers in the artificial-intelligence communities have investigated over the years how clustering for unsupervised data works in ensemble learning Dudoit & Fridlyand (2003); Fern & Brodley (2004); Fred (2001); Strehl & Ghosh (2003); Topchy et al. (2004a). The main objective of clustering is to categorize datasets based on how similar they are in terms of distance or other parameters Hadjitodorov et al. (2006). That is, the quality of results obtained in ensemble learning is a function of not only the methods used but also the parameters taken into consideration. Rather than choosing one clustering results, ensemble models provide solutions to complex problems by combining multiple predictions into a single output with better performance Hubert & Arabie (1985). In ensemble learning, multiple base learners are first trained and then combined with such meta-algorithms as bagging, stacking and boosting, thus generating accurate and improved results Zhou (2012).

The function of stacking, bagging and boosting is to minimize and adjust the variance and bias in weak learners to output the best prediction Strehl & Ghosh (2003). Partitioning and hierarchical clustering algorithms are the two forms of traditionally clustering algorithms in the literature Akbari et al. (2015); Naldi et al. (2013). Partitioning clustering divides datasets into multiple groups based on what the datasets have in common Akbari et al. (2015); Treviño et al. (2006). Typically used portioning clustering includes K-means clustering, CLARA (Clustering Large Applications) algorithm, and K-medoids clustering Treviño et al. (2006). Hierarchical clustering, on the other hand, categorizes raw data into hierarchies Fern & Lin (2008). In hierarchical clustering, each similar observation is grouped as clusters, which are turn combined with other clusters with the same characteristics until a dendrogram is obtained Akbari et al. (2015). Commonly used hierarchical clustering methods include average linkage, full linkage, single linkage, and Ward's hierarchical clustering Akbari et al. (2015). These methods are employed to create an optimal number of clusters for raw data (Ye et al., 2010). Nonetheless, partitioning clustering is easy-to-use and efficient especially in creating close clusters Treviño et al. (2006).

The K-means clustering approach has been used in several data-mining project. This is because the K-means clustering method generates tighter clusters and produces faster computation than hierarchical clustering Lourenço et al. (2015). As of today, no clustering method is perfect. As noted by Sarumanthi et al. (2013), all clustering algorithms have flaws. While some clustering methods find the optimal clusters, some just find sufficiently useful clusters. This explains why they produce different solutions for the same datasets Alizadeh et al. (2014); Topchy et al. (2004b). However, with the Cluster Validity Indexes, it is feasible to compare the quality of any clustering technique used Sarumanthi et al. (2013).

The literature is replete with empirical studies of cluster-ensemble methods. Strehl & Ghosh (2003) were among the first few researchers to investigate cluster-ensemble frameworks. Some scientists used random parameters for the clustering algorithm, whereas some employed different clustering models Hong et al. (2008b). Regardless, a vast majority of previous approaches grouped datasets into consensus clusters Topchy et al. (2004b). Apart from that, ample research work has been carried out to verify the efficacy of different clustering methods. In their experiment, Yang & Jiang (2019) utilized the Hidden Markov Model with the meta-clustering approach to work on various datasets. Their method was advantageous in that it improved the efficiency of temporal data clustering. In another research, Nazari et al. (2019) invented a new clustering ensemble selection method that relies on weighing cluster levels. This method requires picking the “fittest” clusters and then assigning a weight to each one to generate accurate results. Parvin & Minaei-Bidgoli (2013) confirmed that clustering ensembles produce better data classification. Developed by Dunn (1973) and improved by Bezdek et al. (1981) fuzzy clustering is another form of clustering that allows one dataset can belong to one or more similar clusters. This clustering approach is widely used in pattern recognition and object detection, and the most popular fuzzy clustering algorithm is the Fuzzy C-means clustering (FCM) algorithm. The FCM clustering procedure is similar to that of K-means clustering. The number of clusters is first selected and coefficients are assigned to each data point in a cluster. Also, like many clustering algorithms, the FCM algorithm cannot find the best clusters for datasets. This difficulty can be resolved either by taking the entropy of outputs or by adjusting the ensemble in the unstable region Fred & Jain (2005); Mimaroglu & Erdil (2013); Strehl & Ghosh (2003). More importantly, the ensemble in the unstable area is needed for the parameters, lest the results will not be accurate Parvin & Minaei-Bidgoli (2015).

Furthermore, Huang et al. (2019) proposed two forms of clustering algorithms, ultra-scalable ensemble clustering (U-SENC) and ultra-scalable spectral clustering (U-SPEC). U-SPEC merges the ability of the K-means algorithm with the random-selection efficiency for the selection process, while U-SENC integrates multiple clusters to produce better performing base clusters Huang et al. (2019). Because machine learning comes with feature-selection problems especially if the datasets contain lots of redundant information, there is a need for a new clustering method that can eliminate redundancies and generate accurate results Liu et al. (2016). Numerous studies have confirmed the performance superiority of ensemble clustering solutions over a single clustering solution Kuncheva & Hadjitodorov (2004); Kuncheva et al. (2006); Zhang et al. (2006b).

It is not unusual for clustering ensemble techniques to be used in obtaining consensus clustering solutions. All clustering ensemble models rely on two metrics. One is the diversity of the base learners, and the other is the accuracy of the base learners Jia et al. (2011); Hadjitodorov et al. (2006); Hong et al. (2008a); Yu et al. (2014). Together, these two factors can determine the outcome of clustering results Hadjitodorov et al. (2006); Yu et al. (2014). Stated another way, the performance of ensemble clustering algorithms is highly dependent on the diversity of component classifiers and the quality of component classifiers Fern & Lin (2008). As Fern & Lin (2008) put it, diversity is a measure of prediction difference exhibited by ensemble components, while accuracy is a measure of the quality of ensemble components. These two factors have an inverse relationship Akbari et al. (2015). If the diversity is high, the quality will be compromised and vice versa. For this reason, a trade-off between these parameters is crucial in obtaining accurate clustering solutions Hadjitodorov et al. (2006); Hong et al. (2008b).

Besides the diversity and quality of component classifiers, another factor that determines the final performance of the clustering algorithm is the ensemble size. While the number of clustering solutions in an ensemble should be optimal, researchers are yet to find an optimization-oriented cluster-pruning technique that can accurately estimate the cardinality of ensemble subsets for a given problem. At best, only a few studies have identified clustering solutions (for a specific ensemble size) that can maximize the diversity and quality trade-off Fern & Lin (2008); Fern & Brodley (2003); Jia et al. (2012, 2011); Naldi et al. (2013). These pieces of research employed different pruning-rate trials to compare the performance of the algorithms, and their methods generated excellent predictions for the pruning rates.

2.1 ENSEMBLE CLUSTERING

There are two classes of clustering algorithms. One is hierarchical clustering algorithms, and the other is partitioning clustering algorithms Akbari et al. (2015); Hadjitodorov et al. (2006). The former ranks datasets in strata by merging them into clusters based on their similarity. Various techniques of hierarchical clustering algorithms include complete-linkage, single- linkage, and minimum-variance algorithms Jain et al. (1999). Unlike hierarchical clustering algorithms that form dendrogram Akbari et al. (2015), partitioning clustering algorithms do not find nested patters Treviño et al. (2006). Rather, they split raw datasets into single-level clusters. Even though both algorithms group clusters based on their similarity, partitioning clustering is more advantageous than hierarchical clustering when it comes to predicting better results Treviño et al. (2006). To be more specific, the best form of partitioning clustering is the K-means clustering algorithm due to its ease-of-use and lower computation requirements Treviño et al. (2006). This current study will employ K-means clustering to create a library of clustering solutions.

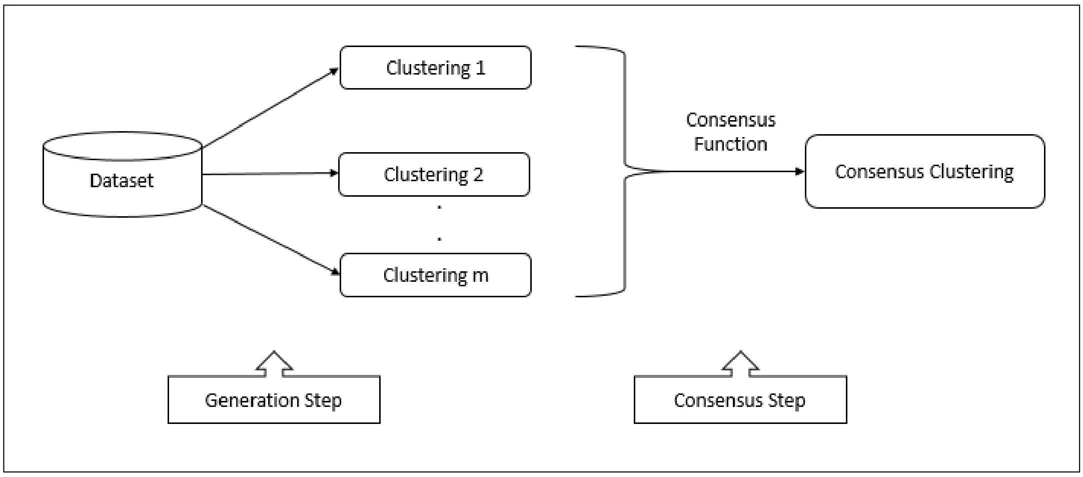
However, in their study, Wang & Liu (2018) developed a selection method that could integrate clustering quality criteria. This method can determine the diversity and quality of clusters selected using data-structure levels and clustering-label levels. With their proposed technique, data characteristics are taken into account, and different evaluation criteria are utilized for clustering partition Wang & Liu (2018). Introduced as a novel clustering algorithm, the Greedy optimization of K-means-based Consensus Clustering (GKCC) can address the challenges of K-means consensus clustering (KCC), such as combining partition steps Li & Liu (2018).

In their research investigation, Li & Liu (2018) discovered that GKCC improves the quality of partitions and accelerates the computation of the algorithm. By changing the clustering parameters, different solutions can be obtained. Another way to boost the performance of the partition is by applying cluster weighting and feature weighting to the problem, a technique that is known as weighted locally adaptive clustering (WLAC) Parvin & Minaei-Bidgoli (2013). More so, more than one cluster can be combined within the cluster uncertainty by employing locally weighted graph partitioning techniques and locally weighted evidence accumulation; however, this method relies heavily on the cluster size Rashidi et al. (2019). Using an object-oriented hierarchical technique for a complex UAV analysis, Yu et al. (2018) were able to create optimal removal and optimal segmentation according to the data of interest. Apart from examining previously computed problems, dynamic programming can also produce the global optimal solution using a structural search Yu et al. (2018). The final consensus clustering solution is the combination of every input clustering. This means the quality of each cluster determines the accuracy of the solution Huang et al. (2018).

Furthermore, scientists have proposed new methods of eliminating clusters that are of a low quality not only by assigning a weight to each cluster but also by classifying them according to their significance Huang et al. (2015); Li & Ding (2008); Yu et al. (2014). The proposed methods, however, are based on the assumption that the quality of clusters in the same base category is similar. Of course, this assumption is flawed in that different clusters have contrasting qualities Huang et al. (2018). More discouraging is that no study has reported any clustering method that can produce the most reliable predictions. No clustering algorithm generates the same result for the same problem. However, Cluster Validity Indexes can be utilized to compare the validity and reliability of different clustering solutions Sarumanthi et al. (2013).

A few years ago, Huang et al. (2018) presented an ensemble clustering approach that considers ensemble cluster uncertainty and clustering weighting. The evaluation of cluster uncertainty is performed by labeling the clusters in the ensem-

Figure 2.1: Diagram of the general process of cluster ensemble .



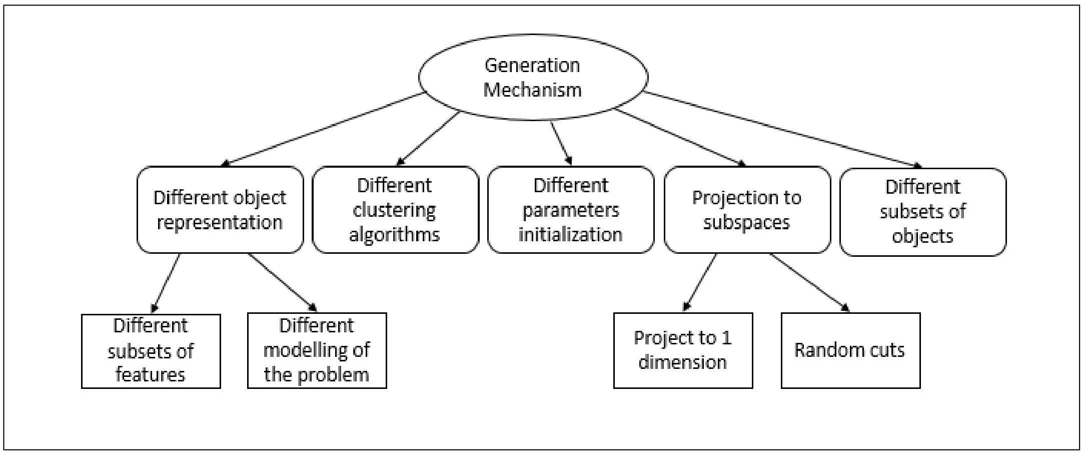
Source: Vega-Pons & Ruiz-Shulcloper (2011)

ble using entropic criteria. Apart from developing a cluster validity index, the researchers proposed two consensus functions for weighting clusters. The consensus functions are Locally Weighted Graph Partitioning (LWGP) and Locally Weighted Evidence Accumulation (LWEA). This method was able to boost the results in terms of efficiency and quality Huang et al. (2018). Another clustering technique, which used a similarity measure between clusters was also suggested to augment ensemble clustering learning. The result of this new method was at best satisfactory.

A clustering ensemble combines more than one ensemble model to produce a result of high quality and efficiency that outperforms one single ensemble algorithm Treviño et al. (2006). The performance of clustering ensembles is dependent on the similarity measures, clustering parameters, clustering techniques, and consensus function Fern & Lin (2008); Azimi & Fern (2009); Joydeep & Ayan (2011); Vega-Pons & Ruiz-Shulcloper (2011). The two main procedures of ensemble clustering comprise the ensemble-library generation and the consensus-function generation. Figure (2.1) illustrates the two steps.

As shown above, the first step does not restrict how different clustering solutions are generated. Nevertheless, it allows 1) different clustering models to be created

Figure 2.2: Diagram of the principal clustering ensemble generation mechanisms.


Generation Mechanism
Different object representation
Different clustering algorithms
Different parameters initialization
Projection to subspaces
Different subsets of objects
Different subsets of features
Different modelling of the problem
Project to 1 dimension
Random cuts

Source: Vega-Pons & Ruiz-Shulcloper (2011)

with different parameters, 2) different data subspaces to be selected, and various objects to be represented. Above is a summary of the step Figure(2.2).

The consensus-function construction is the second step, and it can be divided into two categories based on the median partitions and the object co-occurrence. Unlike the first category which is more heuristic, the second category focus on minimizing the distances between objects. Besides, a similar clustering solution based on the object co-occurrence can be obtained by using the co-association matrix Zhong et al. (2015), votes Charkhabi et al. (2014), finite mixture models Topchy et al. (2004b), graph and hypergraph Strehl & Ghosh (2003), fuzzy techniques and locally adaptive clustering algorithms Fern & Lin (2008).

The median partitioning approach can be categorized into Mirkin distance methods, kernel methods, non-negative matrix factorization methods, and genetic algorithm methods Vega-Pons et al. (2010); Vega-Pons & Ruiz-Shulcloper (2011). The generation step can allow the diversity of clustering solutions to produce reliable ensemble clustering results. To upgrade the consensus step, the combination of the models can be leveraged. Clustering algorithms can also be enhanced with the random projection method, which minimizes the data size by getting rid of redundant partitions Fern & Brodley (2003). Yang et al. (2014)

found in their recent studies that to optimize both quality and diversity, random-projection methods and random sampling methods were useful. An alternative technique for these two methods is the stratified sampling method, in which feature hierarchies are built by grouping data randomly Jing et al. (2015).

A group of scientists also tried another strategy, Auto-CES (an automatic clustering ensemble) method, to generate robust results. This method allowed them to prune the component classifiers based on the selection and clustering steps Amiri Maskouni et al. (2018). Regarding the clustering step, similar categories are grouped with a clustering algorithm, which does not require the cardinality of the final groups. As for the selection step, the cohesiveness metric selects the best trees based on two parameters, quality and diversity Amiri Maskouni et al. (2018). Strehl & Ghosh (2003) also studied how to determine the consensus function. Using different heuristic techniques, they measured and maximized the similarity between groups of clusters. The algorithms were the MCA (Meta-Clustering Algorithm), HPA (Hypergraph Partitioning Algorithm), and CSA (Cluster-based Similarity Algorithm). All of them can change the solutions of clustering into hypergraph notation Strehl & Ghosh (2003). Berikov (2014) suggested another way to aggregate decisions by optimizing the similarity measures of dataset solutions. This method is similar to that proposed by Vega-Pons et al. (2010, 2008), who used a kernel function to determine the similarity measure. Rather than using matrices, Singh et al. (2007) defined the similarity measures by strings to obtain the maximum value, which was later computed with mixed-integer programming.

Combining high-quality component classifiers is part of the goal of ensemble learning even though insignificant clusters can be eliminated in the consensus function. Banerjee et al. (2018) used another approach that optimized clustering without compromising the quality of consensus clusters. However, this method is not immune to scaling problems, especially when the datasets are mammoth.

Sometimes, ensemble algorithms eliminate high-quality clusters. This challenge, according to Alizadeh et al. (2014), can be addressed with the APMM (Alizadeh Parvin Moshki Minaei) criterion rather than with the NMI (Normalized Mutual Information) criterion.

2.2 ENSEMBLE BASED CLUSTERING SELECTION

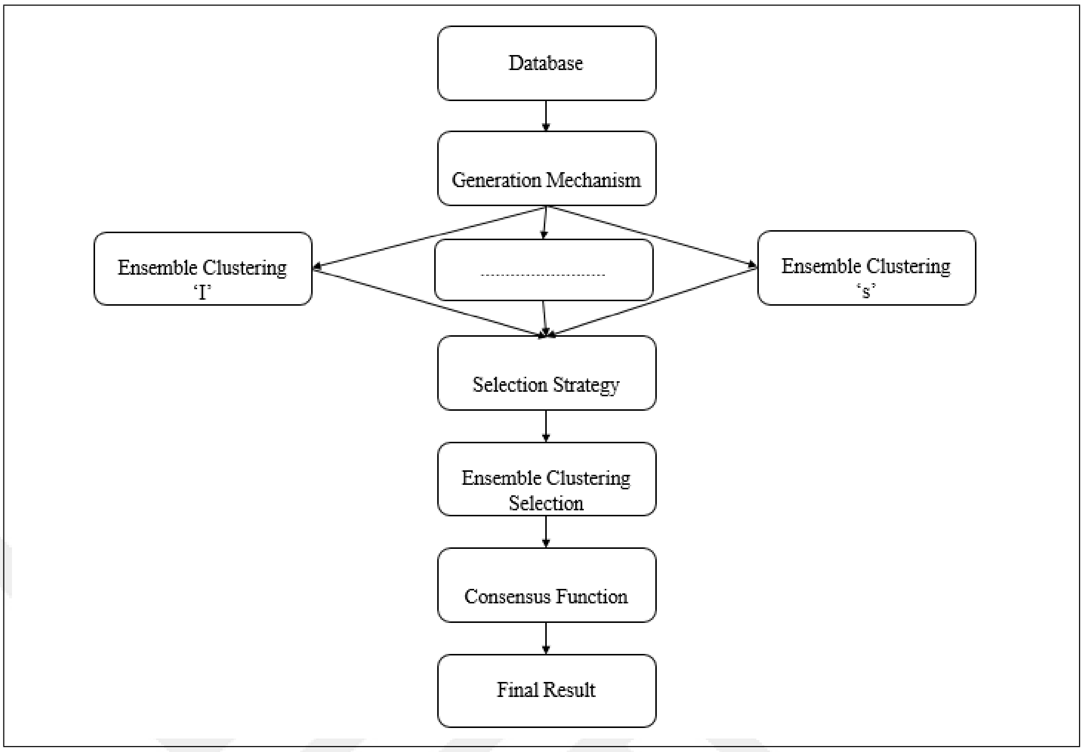
Traditional ensemble learning methods make use of all the component classifiers to produce better predictions. Recent empirical studies, however, have proven that traditional approaches are inefficient in creating ensemble clustering solutions Fern & Lin (2008); Jia et al. (2011); Azimi & Fern (2009); Caruana et al. (2004); Ferrari & De Castro (2015); Yang et al. (2017); Yu et al. (2014). A promising better solution is making sure that all component partitions are different from one another. Many researchers have suggested ways of generating diverse clusters. The procedures are summarized in Figure (2.3).

With the Joint Criterion method, it is possible to optimize the trade-off between diversity and quality by combining clusters with a similar objective function Fern & Lin (2008). NMI values are ranked to cull the best clusters that can produce the best outcome Azimi & Fern (2009). The expression of $NMI(X,Y)$ is given by equation (2.1) and equation (2.2). $NMI(X,Y)$ represents the information function between two clusters (X and Y), $p(x)$ and $p(y)$ are the distribution functions, and $H(Y)$ and $H(X)$ are the entropy functions of X and Y,

$$I(X;Y) = \sum_{y \epsilon Y} \sum_{x \epsilon X} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}, \tag{2.1}$$

$$NMI(X,Y) = \frac{I(X;Y)}{\sqrt{H(X)H(Y)}}. \tag{2.2}$$

Figure 2.3: **General Flowchart of the Cluster Ensemble Selection.**



In addition, with the spectral clustering method, Jia et al. (2011) updated clusters that were selected randomly with bagging based on selected NMI values. Similar to the objective-function of modeling of Jia et al. (2011) was that of Fern & Lin (2008). However, Fern & Lin (2008) multiplied the NMI values with a natural logarithm. Some researchers such as Jia et al. (2012); Yu et al. (2014) integrated their algorithms with weighted functions, while some scientists like Vega-Pons & Avesani (2015) used lattice information for the optimization. Others developed ensemble models that factored in three parameters such as consistency, cardinality, diversity, and quality Fern & Lin (2008); Yang et al. (2017). In addition, some studies used ensemble subsets with a cardinality of more than 50 Jia et al. (2012, 2011); Naldi et al. (2013); Yu et al. (2014). Azimi & Fern (2009) modified their model, classifying datasets into stable and unstable values (i.e., NMI values less than 0.5 were considered unstable with values greater than 0.5 were regarded as stable). In this study, stable values were ignored, while unstable values were selected Azimi & Fern (2009). Because the cardinality of the ensemble selected affects the final outcome, researchers rate the performance of their models based on the number of elements in the subset. In short, the cardinality of the ensemble subsets is an important parameter in ensemble learning Fern & Lin (2008); Kuncheva & Hadjitodorov (2004); Kuncheva et al. (2006); Ghaemi et al. (2011).

This current study aims to develop an ensemble model that not only optimizes the diversity and accuracy trade-off but also uses a small number of ensemble subsets to produce better predictions. To achieve this goal, our proposed model will be compared with three ensemble cluster ensemble techniques developed by Fern & Lin (2008) and Cruz et al. (2018). The first technique is the Joint Criterion method, which uses a joint objective function that combines quality and diversity. The second technique is the Cluster-And-Select method, which groups similar solutions into clusters and then picks the most competence one among the similar clusters. The third technique is the DES-Cluster method, which produces a scatter plot of clustering solutions with average diversity and quality.

2.3 ENSEMBLE BASED FEATURE SELECTION

In the literature, there are two main approaches regarding the use of ensembles in feature selection. In the first, feature selection steps are used for obtaining the diversity needed for using posterior ensemble classification methods (Cunningham & Carney, 2000). Other authors use ensembles of feature selectors to improve the accuracy, diversity, and stability of the feature selection process (Das et al., 2017; Saeys et al., 2008; Seijo-Pardo et al., 2017a,b). This latter approach is of special interest in knowledge discovery scenarios, and mainly in high dimensional cases.In (Tsymbal et al., 2003), five different pairwise measures of diversity were compared over 21 datasets with fixed ensemble sizes. They aimed to design a fitness function that shows the relation between accuracy and diversity. The results showed that there is a close relationship between the functions employed and the number of ensemble members that produce the highest accuracy. Other works, such as (Bolón-Canedo et al., 2012), used a fixed number of filters in high dimensional scenarios. In (Bolón-Canedo et al., 2014), two different basic methods of heterogeneous type were proposed. The first method applied five filters that fed five classifiers followed by the aggregation step. The same filters were used in the second formulation with the aggregation step, previous to classification. In (Yang & Mao, 2010), a new algorithm, named Multicriterion Fusion-based Recursive Feature Elimination, was developed, whose aim is to increase the robustness of feature selection algorithms by using multiple feature selection evaluation criteria. Another study used Multi-layer perceptrons at the ensemble stage (Windeatt et al., 2011).

Diversity is a factor that deserves specific emphasis. By using several types of feature selectors in an ensemble (Bolón-Canedo et al., 2016) such as rankers, subset methods filters, wrappers, embedded methods, or univariate and multivariate methods as in Bolón-Canedo et al. (2014); Seijo-Pardo et al. (2017b), we can provide diversity.

In Wang et al. (2010), several ensembles of filter rankers were applied to the area of software quality. The combination of individual rankings included simple methods like mean, median, and minimum, and complex methods such as Complete Linear Aggregation Abeel et al. (2009) (CLA), Robust Ensemble Feature Selection (Rob-EFS) Brahim & Limam (2013), SVM-Rank Seijo-Pardo et al. (2017b), and data complexity measures Seijo-Pardo et al. (2019). Meanwhile, there exist many parallel and distributed implementations of feature selection methods Eiras-Franco et al. (2016); Mitchell et al. (2014). Further, various research projects have developed ensembles making use of distributed or parallel schemes. In Seijo-Pardo et al. (2017b), a heterogeneous approach was proposed, with the idea of distributing the dataset in several nodes, applying the same feature selection method in each of them, and then at the end of their work aggregating the results. Hong et al. (2008b) also developed a feature selection algorithm for unsupervised clustering, which put together the clustering ensemble method and the population-based incremental learning algorithm.

The same authors also developed the task of feature ranking for unsupervised clustering Hong et al. (2008a) to guide computation of features relevance. A different approach was followed in the work by Morita et al. (2004), in which they developed an ensemble of classifiers based on unsupervised feature selection. Bellal et al. (2012) developed a new method called semi-supervised ensemble learning guided feature ranking method (SFR), which combined a bagged ensemble of standard semi-supervised approaches with permutation-based out-of-bag feature importance. A new wrapper-type semi-supervised feature selection framework which finds the relevant features using confident unlabeled data was developed by Han et al. (2011). They employed an ensemble classifier that supports the estimation of confidence in unlabeled data. In Ko et al. (2008) developed that the Dynamic classifier selection that the competencies of the individual classifiers are calculated during classification operation. Using the majority voting rule for combining classifiers, perform better than the static selection method.In our study DES was used as classifier method.

It must be noted that in each of the above methods, the number of functions, i.e., the cardinality of an ensemble library, is not determined theoretically. The number of functions in the ensemble acts as a hyper-parameter of these methods, which directly affects the classification performance in the aggregation step. In the generation step of the ensemble, the most accurate and diverse models are desired for better prediction performance at the end. However, there might be models which are weak in the generation phase, causing a decrease in overall accuracy. To eliminate such redundancies in the ensemble, a pruning step is needed to select the optimum subset of the ensemble. To the best of our knowledge, no pruning algorithm has been proposed for ensemble-based feature selection algorithms. There exist pruning methods developed for ensemble classification of multi-class task problems using Error-Correcting Output Codes Özöğür-Akyüz et al. (2015); Zhang et al. (2006a). In these studies, the importance of the accuracy and diversity trade-off is strongly emphasized and optimization-based approaches are proposed for ensemble classification models. This trade-off can be explained as follows: High accuracy in the ensemble leads to a decrease in the diversity of the ensemble, and an increase in diversity sacrifices the accuracy of the overall ensemble.

3. BACKGROUND MATERIAL

In this section, various background methods used in the pruning, classification steps of this study are introduced. In the following subsection, Disciplined Convex-Concave Programming (DCCP), the core of the pruning step, is introduced briefly. In the later subsection, the classifier in this study, Dynamic Ensemble Selection(DES), is summarized.General ideas about Joint Criterion Method, Cluster and Select Method and DES-Clustering is given in this chapter

3.1 DISCIPLINED CONVEX-CONCAVE PROGRAMMING

DCCP is an optimization method which was first introduced in Grant et al. (2006) and which combines two ideas: Disciplined Convex Programming (DCP) and Convex- Concave Programming (CCP) Shen et al. (2016). DCP requires a set of conventions in which problems follow, whereas CCP is an organized heuristic for solving nonconvex problems. Disciplined convex programming can be defined by the following optimization problem:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f_0(x) - g_0(x) \\ \text{subject to} \quad & f_i(x) - g_i(x) \leq 0, \; i = 1, \ldots, m, \end{aligned} \tag{3.1}$$

where $x \in \mathbb{R}^n$ refers to the optimization variable, and the functions $f_i : \mathbb{R}^n \to \mathbb{R}$ and $g_i : \mathbb{R}^n \to \mathbb{R}$ $(i = 1, ..., m)$ are convex functions.

Above problem (3.1) can be rewritten as follows:

$$\begin{aligned} \text{minimize} \quad & f_0(x) - t \\ \text{subject to} \quad & t = g_0(x), \\ & f_i(x) \leq g_i(x), \; i = 1, \ldots, m, \end{aligned} \tag{3.2}$$

where x and t refers to the original optimization variable and a new optimization variable, respectively.

DCCP is an appropriate and simple standard form for Difference Programming (DC), because the linearized problem of CCP is a DCP problem if the original problem is DCCP. The linearized problem can then be transformed into a cone program and solved using generic solvers Shen et al. (2016).

3.2 DYNAMIC ENSEMBLE SELECTION - COMBO

In this study, Dynamic Ensemble Selection(DES) was used as a classifier for the proposed ensemble model. As previously reported, given that selecting only one classifier is very vulnerable to error, some researchers chose to pick a subset of classifiers. Ko et al. (2008) suggested an approach aimed at imitating the Oracle model, which obtained the best ensemble results Kuncheva (2002). The KNORA-E (K Nearest ORAcles - Eliminate) eliminates the classifier from the ensemble if the classifier misclassifies any pattern of the neighbors. There exists moreover a weighted form KNORA-E-W which weights the labels of the chosen classifiers based on the distance between the test sample and the neighbors. This work includes two fusion algorithms: KNORAU (K Nearest ORAcles - Union) and its weighted form KNORA-U-W. Soares et al. (2006) choose the N most correct classifier, according to a defined region of competence, and the J most diverse classifiers in order to produce the ensemble. The values of N and J were settled by the authors. These methods are called dynamic ensemble selection (DES) since they can choose more than one classifier. In this thesis, DecisionTreeClassifier, LogisticRegression, KNeighborsClassifier, RandomForestClassifier,, GradientBoostingClassifier were set as estimators of classifer steps. Here, we used Combo which is a relatively new library specialized in ensemble learning which provides several common methods under a unified Scikit-learn-compatible API so that it maintains compatibility with many estimators from the Scikit-learn ecosystem Zhao et al. (2019). The Combo library delivers algorithms that are capable of combining models for classification, clustering, and anomaly detection tasks, and it has been used widely in the Kaggle predictive modeling community. Combo provides a unified outlook for different ensemble methods whilst remaining compatible with Scikit-learn.

3.3 SUPPORT VECTOR MACHINES

In this study, SVM was used as a base classifier for the proposed ensemble clustering selection model. SVMs were first introduced by Vapnik (1998) and have been used frequently in recent years for classification problems. It is a discriminative classifier formally defined by a separating hyperplane which constructs an optimal hyperplane $y = < \mathbf{w}, \mathbf{x} > +b$ between classes during the training phase. Here, $\mathbf{x}$ is a vector from the input space, $\mathbf{y}$ is the output, $<,>$ refers to dot product, $\mathbf{w}$ refers to the normal vector of the decision surface and b stands for the bias term Cristianini & Shawe-Taylor (2000). The optimal separating hyperplane is obtained by maximizing the margin between two classes with the following optimization problem:

$$\begin{aligned}&\min_{\mathbf{w},\xi} \|w\|^2 + C\sum_{i=1}^{l} \xi_i,\\ &y_i(w^T x_i + b) \geq 1 - \xi_i \quad (i = 1 \ldots l),\end{aligned}$$

where C refers to a regularization (cost) constant corresponding to the error term ξ_i which contributes to the model as a penalty term to avoid over fitting. Determining an optimum value for C is important since large C results in a small margin and, a small C results in a large margin for l training points.

3.4 JOINT CRITERION METHOD

In the Joint Criterion Method, quality and diversity terms are merged into a joint criterion function Fern & Lin (2008). For a given ensemble size K, the following objective function (3.3) is maximized with respect to find the indices of the best candidates among the ensemble

$$\alpha \sum_{i=1,\dots,K} SNMI(C_i, L) + (1-\alpha) \sum_{i \neq j} (1 - NMI(C_i, C_j)), \tag{3.3}$$

where the first term measures the quality, the second term measures the diversity and the parameter α controls the impact assigned to each term Fern & Lin (2008). It starts with a single solution having the highest-quality and the next candidate is added to the ensemble subsequently which maximizes the objective function of the problem (3.3) Fern & Lin (2008).

3.5 CLUSTER AND SELECT METHOD

The Cluster and Select (CAS) method is the invention of Fern & Lin (2008), and the goal of this method is to eliminate all the redundancies among similar clusters and ultimately improve unsupervised learning results. In the library, it is not uncommon to have many similar cluster solutions, which create redundancies and slow down the computation process of the ensemble. As Fern & Lin (2008) put it, if two similar clustering solutions (C_1 and C_2) are in an ensemble, either C_1 or C_2 can be used without compromising the quality of the objective function. Unlike the Joint Criterion method which uses a single AOF (aggregated objective function) to solve problems, the Cluster and Select method continues to regroup the clusters based on their similarity and then select one of the clusters in the group. While clusters can be grouped based on their similarity in many ways, it is crucial to note that the CAS method selects only the highest quality cluster among the similar clusters in the ensemble Fern & Lin (2008). Overall, the CAS method can achieve improvements, which are statistically significant, for clustering ensembles of any size.

3.6 DES-CLUSTERING

In machine learning projects, DESlib makes use of dynamic selection (DS) algorithms to implement several techniques that involve ensemble and dynamic classifiers. This open-source python library assumes that not all classifiers in an ensemble are effective, and the major advantage of using DESlib is that only the most competent classifiers in an ensemble are picked. This library uses three methods: SE (static ensemble) methods, DES (dynamic ensemble) methods, and DCS (dynamic classifier selection) methods. The DES method was employed for comparison in this study. First, component classifiers of an ensemble were selected, and the competence region was defined with the K-means algorithm. Two factors were considered. One, the best cluster in an ensemble of similar clusters was picked to refine performance. Two, the most accurate ensemble classifiers to the given problem were culled from the diverse ensemble classifiers Cruz et al. (2018). Then, the classifiers were categorized into k partitions. After that, they were ranked in decreasing order according to their quality, which is accuracy and diversity. The Euclidean distance was utilized to test the set effectiveness Soares et al. (2006). For instance, the most diverse classifiers (J) were gleaned from N classifiers.

4. ENSEMBLE CLUSTERING SELECTION

4.1 THE MATHEMATICAL MODEL

This section focuses on the cluster ensemble selection model used in this study. Similar to the model used by Zhang et al. (2006b), a constrained optimization technique was incorporated into the partitions, and the ensemble model was developed with an eye towards maximizing diversity and minimizing variances and noises in the objective function. Also considered in the development of the model were pairwise errors of components. With the NMI (normalized mutual information) function, it was possible to obtain the matrix entries by estimating the similarities of clusters in an ensemble. As highlighted in Eq. (4.1) and (4.2), E represents the clustering solutions for n cluster ensembles, $NMI(C,C_i)$ is the function that computes the similarity between solutions of clusters, $SNMI(C,E)$ measures the quality, and the matrix G is the parameter with the diversity and quality information.

$$SNMI(C,E) = \sum_{i=1}^{n} NMI(C,C_i), \tag{4.1}$$

$$G_{ij} = \begin{cases} G_{ii} = SNMI(C,C_i), \ (i = 1,...,n) \\ G_{ij} = 1 - NMI(C_i,C_j), \ (j = 1,...,n). \end{cases} \tag{4.2}$$

As shown above, the quality of the i-th solution and the diversity between two solutions are measurements of each diagonal element and off diagonals, respectively. The matrix G is transformed as shown below to ensure that all the elements are similar.

$$\tilde{G}_{ii} = \frac{G_{ii}}{N},$$

Where N is the clustering-solution N in the ensemble. Then, the new equation is denoted as:

$$\tilde{G}_{ij,i\neq j} = \frac{1}{2}\left(\frac{G_{ij}}{G_{ii}} + \frac{G_{ij}}{G_{jj}}\right). \tag{4.3}$$

As suggested by Zhang et al. (2006b), the diversity and quality trade-off is optimized by the model shown below in (4.4) , With the solution to the Mixed Integer Problem shown in Eq. (4.4) , we can find the clustering-solution indices, which indicate the ensemble subsets that should be included.

$$\begin{aligned} &\max_x x^T \tilde{G} x \\ \text{subject to} &\sum_i x_i = k, \\ &x_i \in \{0,1\}. \end{aligned} \tag{4.4}$$

With Eq. (4.4) depicted above, we can find the cluster's indices for each solution in an ensemble subset. If a solution, for instance, is $x = [1000100011]^T$, the 1st, 4th, 8th, and 10th indices of vector x will be chosen from the ensemble matrix. This equation, which is called an NP-Hard problem, can be solved with SDP (Semi-definite programming). As the ensemble size is a function of the quality of the solution (k) which in turn determines the quality of the ensemble algorithm, it is essential for k to be divorced from insignificant clusters. For this reason, like in Lagrangian Relaxation, the constraint $\sum_i x_i = k$ will be moved to the objective function with the regularization constant *rho*. In short, this constraint can be changed to $||x||_0 = k$. This new expression represents the subset cardinality of the total ensemble. With this new equation, the binary vector x can be transformed into the real vector with the support of sparsity. That way, to become the unconstrained problem, (4.4) can be rewritten as:

$$\max_{x \in R^n} x^T \tilde{G} x - \rho \|x\|_0 . \tag{4.5}$$

Regarding the clustering problem, it is feasible to approximate the second value in the objective function by $\|x\|_1$. This function is approximated as the Student t-distribution negative-log likelihood. In the literature, this approximation is not only more tight than $\|x\|_1$ but also commonly used in many ensemble studies Candès et al. (2008); Fazel et al. (2003); Sriperumbudur et al. (2011); Weston et al. (2003). This objective function can be approximated to $\|x\|_0$, for example, $\|x\|_0 = \sum_{i=1}^{n} (1 - e^{-\alpha |x_i|})$ where $\alpha > 0$ Bradley & Mangasarian (1998). In the objective function, the zero-norm in the function is estimated with the expression below:

$$\|x\|_0 := \sum_{i=1}^{n} 1_{x_i \neq 0} = \lim_{\epsilon \to 0} \sum_{i=1}^{n} \frac{\log(1 + |x_i|/\epsilon)}{\log(1 + 1/\epsilon)} .$$

Based on this, Eq. (4.5) is transformed into the expression below:

$$\max_{x \in R^n} x^T \tilde{G} x - \rho \lim_{\epsilon \to 0} \sum_{i=1}^{n} \frac{\log(1 + |x_i|/\epsilon)}{\log(1 + 1/\epsilon)} . \tag{4.6}$$

As shown, there is a similar between Eq. (4.6) and the expression proposed by Sriperumbudur et al. (2011) , which can compute the difference convex-function (DC). It is feasible to change Eq. (4.6) to a DC problem by defining the positive matrices and making sure $\tilde{G} + \tau I \in S^+$. This means that one can select any $\tau > 0$ so long $\tilde{G} + \tau I \in S^+$. Selecting $\tau > -\lambda_{min}(\tilde{G})$ results in $\tilde{G} + \tau I \in S^+$ when $\tilde{G}$ is indefinite. Note that λ_{min} is the minimum eigenvalue of $\tilde{G}$. Sriperumbudur et al. (2011) used the same approach in their study to solve an eigenvalue problem. If $\tilde{G} \in S^n$ assuming that as $\tau \geq \max(0, -\lambda_{min})$, then it can be said that positive semi-definite matrices exist in the expression. Thus, Eq. (4.6) can be expressed as follows:

$$\min_{x} \tau \|x\|_2^2 - x^T \left(\tilde{G} + \tau I\right) x + \rho \lim_{\epsilon \to 0} \sum_{i=1}^{n} \frac{\log(1 + |x_i|/\epsilon)}{\log(1 + 1/\epsilon)}, \tag{4.7}$$

where $\|.\|_2$ refers to the Euclidean norm. If we neglect the limit of the last term of the problem (4.7) by choosing $\epsilon > 0$, the following convex problem is derived:

$$\min_{x} \left\{ \tau \|x\|_2^2 - \left[x^T \left(\tilde{G} + \tau I\right) x - \rho \sum_{i=1}^{n} \frac{\log(1 + |x_i|/\epsilon)}{\log(1 + 1/\epsilon)} \right] \right\}. \tag{4.8}$$

Under the new expression, Eq. (4.4) is presumed to be independent of the ensemble size k in the model, where ρ is the regularization constant. By taking so much time to select the fittest k, the efficiency of the algorithm performance becomes low. However, our model can overcome this inefficiency by introducing another parameter $\rho = [10^{-1}, 10^{-2}, 10^{-3}, 1, 10^{1}, 10^{2}, 10^{3}, 10^{4}, 10^{5}, 10^{6}]$. Put simply, this method selects the fittest k with the highest NMI values. With this strategy, the search space can be reduced, meaning the computation speed will be high.

$$\min_{x,y} \left\{ \tau \|x\|_2^2 - \left[x^T \left(\tilde{G} + \tau I\right) x - \rho \sum_{i=1}^{n} \frac{\log(1 + y_i/\epsilon)}{\log(1 + 1/\epsilon)} \right] : -y \leq x \leq y \right\}$$
$$\text{such that } \; x^T x = 1. \tag{4.9}$$

The first term $\tau \|x\|_2^2$ is convex in x as $\tau > 0$ and $x^T \left(\tilde{G} + \tau I\right) x - \rho \sum_{i=1}^{n} \frac{\log(1 + y_i/\epsilon)}{\log(1 + 1/\epsilon)}$ is jointly convex in x and y. The Eq. (4.9) represents the minimization of convex difference functions for a data set Sriperumbudur et al. (2011). To solve DC programs, we use such algorithms as cutting-plane and brand-and-bound algorithms. However, these global algorithms are not scalable, especially if the value of n is large. For this reason, we use a majorization-minimization method that uses a local optimization algorithm for the DC problems.

The constraint $x^T x = 1$ makes vector x into small values with restricted interval, thereby allowing us to transform the vector solution x into the binary solution. The expression shown below represents the (4.9) Algorithm 1 binary solution.

$$x = \begin{cases} 1 & x \geq 0.1, \\ 0 & x < 0.1. \end{cases}$$

As shown above, the boundary constraints produce excellent results compared to (5.4), which generates a low subset size for an ensemble.

Algorithm 1 Ensemble Cluster Selection with DC Programming

```
Input: Data set,Base learners
Paramters: $\tau$,$\epsilon$,$\rho$
Output: NMI values
Generate Ensemble library $E$ in 3 steps given in Section 5.2
Compute matrix $\tilde{G}$ defined by equation (4.3)
Find a solution $x$ of the below problem by Majorization Minimization OR DCCP by user selection
    $\min_{x,y} \left\{ \tau \|x\|_2^2 - \left[ x^T \left( \tilde{G} + \tau I \right) x - \rho \sum_{i=1}^{n} \frac{\log(1 + y_i/\epsilon)}{\log(1 + 1/\epsilon)} \right] : -y \leq x \leq y \right\}$
    such that $x^T x = 1.$          (4.10)
Find all $x_i^\prime s$ such that $x_i \geq 0.1, \;\; i = 1 \ldots m$ to be the indices of clustering solutions in selected subset
Run consensus method Hypergraph Partitioning Algorithm (HGPA) Hein et al. (2013)
```

The model was evaluated in this thesis with DCCP (convex-concave programming. Algorithm (1) generates (4.9) , and it implements the DCCP package in Python 3.0 library. The numerical optimization approach is introduced below. To obtain robust solutions, the solutions were aggregated in the sub-ensemble. We then employed the HGPA (Hypergraph Partitioning Algorithm) for clustering. Despite the objective-function constrain, mutual information is optimized in the algorithm. The literature provides more information regarding the HGPA approach.

4.2 EXPERIMENTAL RESULTS

In this section, we explain the results of our experiments and make informed conclusions based on our findings. Apart from presenting the tables and figures, we explain our ensemble step-up used in this thesis and then compare the datasets of each ensemble method with others. That way, we can determine whether or not the performance of our proposed ensemble model is better than that of existing ensemble models. As shown in 2.2, the ensemble-library generation step in this study includes three steps. The base learner used was K-means to produce initial partitions of the datasets. By employing the K-means algorithm, we were able to obtain diverse clustering solutions. Added to the ensemble library in the second step were clustering solutions with data attributes, which were randomly selected. Approximately 50 clustering solutions were also created for each generation step with the aim of exploring the data structure, meaning the ensemble library was comprised of 150 diverse clustering solutions. For the clustering solutions, Algorithm 1 was used as the selection strategy, and approaches such as Joint Criterion and Cluster-And-Select Fern & Lin (2008) and DES-ClusteringCruz et al. (2018) methods were utilized to compare the model performance.

In addition to Algorithm 1, the DCCP algorithm was utilized to resolve the proposed model. Table 4.5 presents the best NMI values for the Joint Criterion method, Cluster- Select method, DES-Cluster method, and PrunedOPT method. As recommended by Dheeru & Karra Taniskidou (2017), each method was assessed with 11 benchmark datasets: Frog MFCCs, Glass, Movement, Seeds, Segmentation, Synthetic Control, Wine, Zoo, Yale, Hill-Valley, and USPS. Table 4.1 provides the details of the benchmark dataset, including their Types, Feature Values, Features, Instances, and Class. As shown in the table below, most of the datasets have a continuous feature value, and a class number greater than 3.

Table 4.1: Detailed information of benchmark datasets.

Dataset	Type	Feature Value	# Feature	# Instance	# Class
Frogs MFCCs	Bio	Continuous	22	7195	4
Glass	Physical	Continuous	10	214	6
Movement	Image	Continuous	91	360	15
Seeds	Bio	Continuous	7	210	3
Segmentation	Image	Continuous	19	2310	3
Synthetic Control		Time Series	60	600	6
Wine	Chemical	Continuous	13	178	3
Zoo	Artificial	Categorical	17	101	7
Yale	Image	Continuous	1024	165	15
Hill-Valley	Physical	Continuous	101	606	2
USPS	Image	Continuous	256	9298	10

After listing the datasets and implementing the Cluster-And-Select, Joint Criterion, and DES-Cluster methods, we compared the proposed ensemble clustering algorithm. In the DES-Cluster approach, the parameters' default values (k) was set as 5, the percentage quality of base classifiers was set as 0.5, and the percentage diversity of base classifiers was set as 0.33. For the Joint Criterion method, the value of a in equation(3.3) was set as 0.5. Tables 4.3 and 4.4 indicate the NMI values and their standard deviation, including the confidence interval of Cluster-And-Select and Joint-Criterion methods for different pruning rates. Put simply, each dataset for the cluster-ensemble methods has different NMI values for a pruning rate of 100, 90, 80, 70, 60, and 50.

After comparing each method's performance with the proposed cluster-ensemble method using the NMI mean values assigned to each pruning rate (Table 4.5), we found that the mean value results of our proposed method PrunedOPT outperformed the Cluster-And-Select, DES-Cluster and Joint-Criterion methods. The values in bold in Tables 4.3, 4.4 and 4.5 indicate the best NMA results. In Figure 4.1, we further compared the 11 benchmark datasets by combining the NMA values for various pruning rates in Tables 4.3, 4.4 and 4.5. It can be concluded from the graph that the proposed model offers the best NMA values for the vast majority of the pruning rates.

In addition to the data comparison, student-test was carried out to estimate the likely errors in the confidence interval. The error rates found in the confidence interval are displayed in Tables 4.3, 4.4 and 4.5. For each dataset, the standard deviation was performed, and findings are illustrated in each table. The statistical significance and confidence level are $p > 0.99$ and 93 percent , respectively. This means the null hypothesis, in which error rates are random for 5 different tests, should be rejected. Table 4.2 indicates the time cost of each cluster-ensemble method. For all the datasets, the time cost of the PrunedOPT method was higher than that of other ensemble methods. This longer time is a drawback of our proposed method; nonetheless, this deficiency is not a fair comparison. After all, our proposed cluster-ensemble method produces better quality. Unlike the Cluster-And-Select, Joint-Criterion, and DES-Cluster methods, PrunedOPT can automatically find the optimal pruning rate without compromising the diversity and quality of the base learners. It is, therefore, pertinent to consider the trial costs of different pruning-rate selections.

Table 4.2: Time cost in second of each method.

	Time Cost (Second)			
Data	**Cluster-Select**	**Joint Criterion**	**DES-Cluster**	**PrunedOPT**
Frogs MFCCs	57.131	25.549	**23.67**	776.241
Glass	57.045	**24.564**	31.457	862.456
Movement	78.976	28.890	**2.21**	890.125
Seeds	94.589	25.146	**21.567**	764.125
Segmentation	92.067	**28.648**	27.891	779.960
Synthetic Control	58.017	**27.578**	32.654	859.167
Wine	76.045	26.639	**16.456**	884.843
Zoo	75.030	37.645	**33.876**	789.100
Yale	97.321	89.546	**78.34**	894.678
Hill-Valley	67.45	72.567	**48.453**	796.435
USPS	202.56	198.45	**98.565**	1400.11
Mean	76.247	**38.677**	39.439	826.713

Table 4.3: NMI values, standart deviation and confidence interval for various pruning rates of Joint Criterion Method.

Joint Criterion			**Pruning Rate**			
Data	**50**	**60**	**70**	**80**	**90**	**100**
Frogs MFCCs	0.239 (±0.012) Cl=[0.217, 0.271]	**0.272** (±0.013) **Cl=[0.283, 0.296]**	0.209 (±0.011) Cl=[0.205, 0.238]	0.217 (±0.021) Cl=[0.207, 0.256]	0.252 (±0.012) Cl=[0.229, 0.293]	0.251 (±0.021) Cl=[0.221, 0.285]
Glass	**0.789** (±0.014) **Cl=[0.738, 0.810]**	0.708 (±0.019) Cl=[0.701, 0.728]	0.723 (±0.013) Cl=[0.714, 0.756]	0.726 (±0.012) Cl=[0.713, 0.801]	0.745 (±0.012) Cl=[0.741, 0.749]	0.708 (±0.019) Cl=[0.701, 0.723]
Movement	0.526 (±0.019) Cl=[0.500, 0.601]	**0.541** (±0.019) **Cl=[0.531, 0.582]**	0.53 (±0.019) Cl=[0.401, 0.701]	0.532 (±0.019) Cl=[0.502, 0.604]	0.514 (±0.019) Cl=[0.499, 0.612]	0.531 (±0.019) Cl=[0.498, 0.621]
Seeds	**0.654** (±0.001) **Cl=[0.640, 0.681]**	0.654 (±0.012) Cl=[0.630, 0.660]	0.530 (±0.014) Cl=[0.512, 0.620]	0.620 (±0.011) Cl=[0.588, 0.720]	0.530(±0.013) Cl=[0.512, 0.569]	0.620(±0.019) Cl=[0.599, 0.701]
Segmentation	0.260 (±0.014) Cl=[0.189, 0.301]	0.184 (±0.012) Cl=[0.174, 0.192]	0.262 (±0.023) Cl=[0.219, 0.329]	0.345 (±0.021) Cl=[0.321, 0.419]	0.354(±0.012) Cl=[0.310, 0.420]	**0.371** (±0.014) **Cl=[0.320, 0.419]**
Synthetic Control	0.743 (±0.013) Cl=[0.701, 0.765]	0.753 (±0.014) Cl=[0.712, 0.787]	**0.775** (±0.022) **Cl=[0.728, 0.790]**	0.693 (±0.018) Cl=[0.601, 0.721]	0.688 (±0.037) Cl=[0.655, 0.721]	0.654 (±0.013) Cl=[0.630, 0.700]
Wine	0.751 (±0.014) Cl=[0.710, 0.789]	0.718 (±0.014) Cl=[0.689, 0.730]	0.781 (±0.014) Cl=[0.772, 0.791]	0.781 (±0.014) Cl=[0.770, 0.801]	**0.781** (±0.001) **Cl=[0.780, 0.781]**	0.781 (±0.001) Cl=[0.780, 0.781]
Zoo	0.645 (±0.016) Cl=[0.629, 0.667]	0.706 (±0.019) Cl=[0.701, 0.712]	0.704 (±0.004) Cl=[0.701, 0.707]	**0.753** (±0.013) **Cl=[0.742, 0.780]**	0.642 (±0.015) Cl=[0.635, 0.666]	0.646 (±0.021) Cl=[0.631, 0.678]
Yale	0.543 (±0.011) Cl=[0.520, 0.578]	0.545 (±0.015) Cl=[0.520, 0.580]	0.620 (±0.016) Cl=[0.612, 0.667]	0.587 (±0.015) Cl=[0.543, 0.612]	**0.632** (±0.018) **Cl=[0.612, 0.654]**	0.400 (±0.012) Cl=[0.368, 0.434]
Hill-Valley	0.644 (±0.027) Cl=[0.601, 0.712]	0.547 (±0.021) Cl=[0.539, 0.578]	0.538 (±0.023) Cl=[0.511, 0.589]	0.542 (±0.018) Cl=[0.511, 0.580]	0.543 (±0.032) Cl=[0.501, 0.602]	**0.653** (±0.027) **Cl=[0.603, 0.689]**
USPS	0.656 (±0.028) Cl=[0.612, 0.718]	0.523 (±0.018) Cl=[0.511, 0.598]	0.545 (±0.013) Cl=[0.519, 0.561]	0.512 (±0.011) Cl=[0.511, 0.513]	0.571 (±0.014) Cl=[0.567, 0.587]	**0.671** (±0.045) **Cl=[0.622, 0.692]**
Mean	0.586 (±0.015)	**0.577** (±0.013)	0.565 (±0.017)	0.573 (±0.015)	0.568 (±0.015)	0.570 (±0.014)

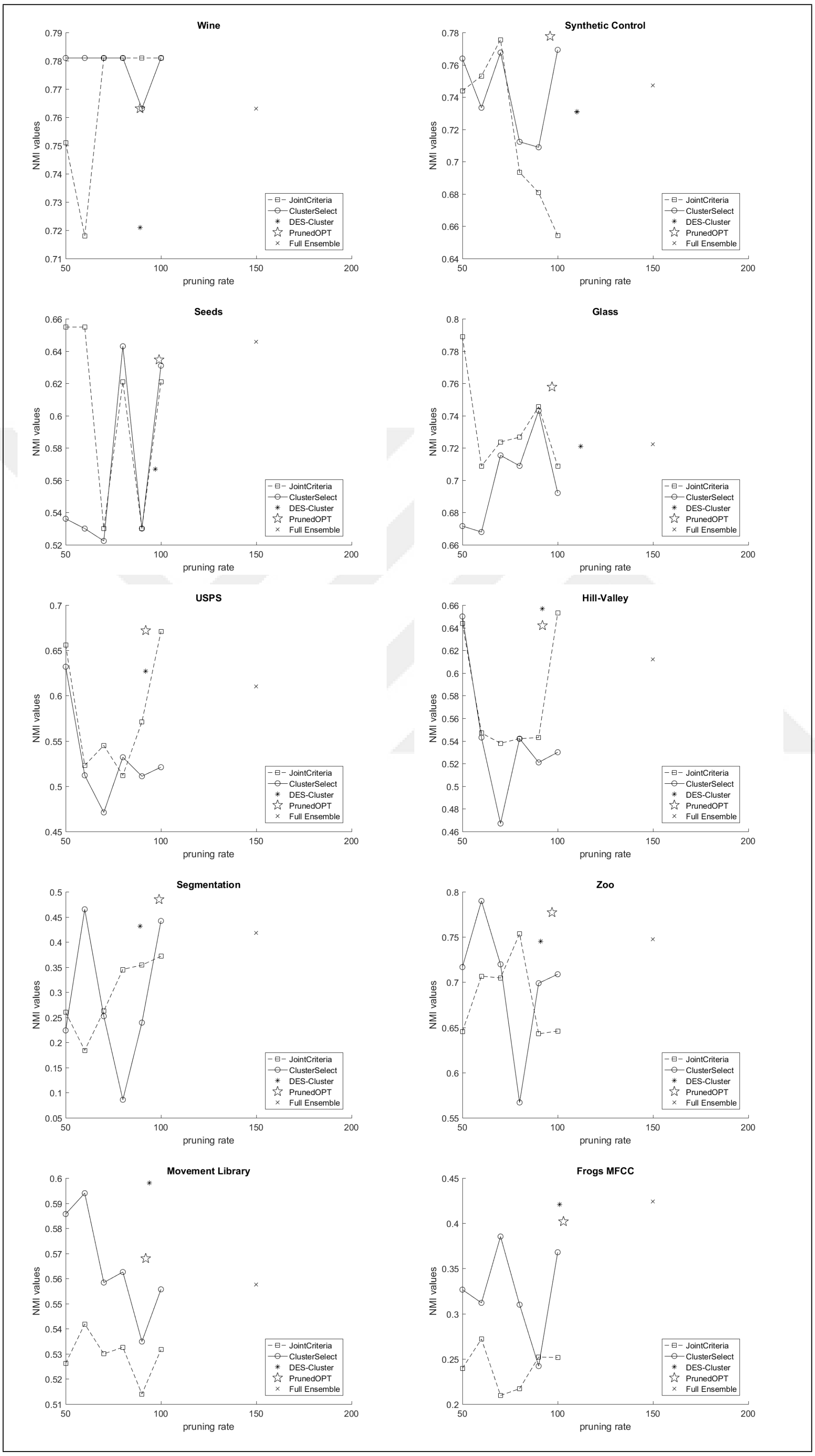
Table 4.4: NMI values, standart deviation and confidence interval for various Pruning Rates of Cluster-Select Method.

Cluster-Select	Pruning Rate					
Data	**50**	**60**	**70**	**80**	**90**	**100**
Frogs MFCCs	0.326 (±0.015) Cl=[0.311, 0.345]	0.312(±0.058) Cl=[0.311, 0.315]	0.385(±0.098) Cl=[0.321, 0.378]	0.310(±0.013) Cl=[0.298, 0.354]	0.242(±0.043) Cl=[0.231, 0.278]	**0.368**(±0.089) **Cl=[0.301, 0.389]**
Glass	0.671 (±0.043) Cl=[0.656, 0.690]	0.667 (±0.034) Cl=[0.651, 0.689]	0.715 (±0.026) Cl=[0.654, 0.809]	0.708 (±0.014) Cl=[0.702, 0.723]	**0.743** (±0.024) **Cl=[0.734, 0.756]**	0.692 (±0.032) Cl=[0.689, 0.702]
Movement	0.585 (±0.0032) Cl=[0.534, 0.651]	**0.594** (±0.011) **Cl=[0.570, 0.602]**	0.558 (±0.034) Cl=[0.548, 0.579]	0.562 (±0.056) Cl=[0.552, 0.598]	0.534 (±0.034) Cl=[0.512, 0.578]	0.555 (±0.098) Cl=[0.543, 0.609]
Seeds	0.536 (±0.021) Cl=[0.521, 0.556]	0.530 (±0.001) Cl=[0.401, 0.680]	0.522 (±0.045) Cl=[0.512, 0.578]	**0.643** (±0.054) **Cl=[0.634, 0.666]**	0.530 (±0.025) Cl=[0.511, 0.589]	0.631 (±0.032) Cl=[0.602, 0.678]
Segmentation	0.224 (±0.021) Cl=[0.211, 0.234]	**0.465** (±0.023) **Cl=[0.432, 0.489]**	0.252 (±0.01) Cl=[0.211, 0.301]	0.085 (±0.021) Cl=[0, 0.120]	0.239 (±0.056) Cl=[0.201, 0.267]	0.442 (±0.046) Cl=[0.441, 0.444]
Synthetic Control	0.763 (±0.056) Cl=[0.734, 0.779]	0.733 (±0.058) Cl=[0.721, 0.745]	0.767 (±0.057) Cl=[0.721, 0.802]	0.712 (±0.044) Cl=[0.701, 0.732]	0.708 (±0.055) Cl=[0.689, 0.754]	**0.769** (±0.068) **Cl=[0.732, 0.798]**
Wine	0.781 (±0.035) Cl=[0.705, 0.810]	0.781 (±0.014) Cl=[0.777, 0.790]	**0.781** (±0.001) **Cl=[0.763, 0.799]**	0.781 (±0.054) Cl=[0.712, 0.810]	0.763 (±0.006) Cl=[0.701, 0.810]	0.781 (±0.011) Cl=[0.710, 0.802]
Zoo	0.716 (±0.034) Cl=[0.701, 0.734]	**0.789** (±0.065) **Cl=[0.754, 0.790]**	0.719 (±0.036) Cl=[0.701, 0.749]	0.566 (±0.064) Cl=[0.534, 0.58]	0.698 (±0.065) Cl=[0.601, 0.843]	0.708 (±0.089) Cl=[0.658, 0.723]
Yale	0.578 (±0.036) Cl=[0.502, 0.604]	0.643 (±0.01) Cl=[0.612, 0.678]	**0.720** (±0.036) **Cl=[0.699, 0.739]**	0.533 (±0.043) Cl=[0.512, 0.45]	0.523 (±0.040) Cl=[0.503, 0.579]	0.627 (±0.046) Cl=[0.603, 0.660]
Hill-Valley	**0.650** (±0.043) **Cl=[0.623, 0.707]**	0.543 (±0.043) Cl=[0.504, 0.589]	0.467 (±0.054) Cl=[0.430, 0.498]	0.542 (±0.064) Cl=[0.518, 0.578]	0.521 (±0.065) Cl=[0.510, 0.591]	0.530 (±0.076) Cl=[0.521, 0.567]
USPS	**0.632** (±0.040) **Cl=[0.621, 0.643]**	0.512 (±0.094) Cl=[0.510, 0.514]	0.471 (±0.009) Cl=[0.465, 0.486]	0.532 (±0.006) Cl=[0.513, 0.543]	0.511 (±0.01) Cl=[0.501, 0.522]	0.521 (±0.013) Cl=[0.513, 0.531]
Mean	0.587 (±0.034)	0.598 (±0.037)	0.578 (±0.037)	0.543 (±0.040)	0.547 (±0.043)	**0.602** (±0.055)

Table 4.5: The best NMI values, standart deviation and confidence interval for various pruning rates of each method.

Data	Methods				
	Joint Criterion	Cluster-Select	DES-Cluster	PrunedOPT	
	NMI Value (SD)	NMI Value (SD)	NMI Value (SD)	Pruning Rate	NMI Value(SD)
Frogs MFCCs	0.240 (±0.0.015)	0.323 (±0.052)	0.421 (±0.033)	103	0.401 (±0.034)
Glass	0.733 (±0.015)	0.701 (±0.028)	0.721 (±0.023)	97	**0.757** (±0.001)
Movement	0.529 (±0.019)	0.564 (±0.044)	**0.598** (±0.065)	92	0.568 (±001.)
Seeds	**0.601** (±0.007)	0.565 (±0.029)	0.567 (±0.065)	99	0.634 (±0.009)
Segmentation	0.296 (±0.014)	0.284 (±0.029)	0.432 (±0.078),	99	**0.484** (±0.011)
Synthetic Control	0.717 (±0.023)	0.741 (±0.056)	0.731 (±0.065)	96	**0.777** (±0.055)
Wine	0.765 (±0.011)	**0.778** (±0.029)	0.721 (±0.065)	89	0.763 (±0.065)
Zoo	0.682 (±0.009)	0.699 (±0.058)	0.745 (±0.013)	97	**0.777** (±0.01)
Yale	0.587 (±0.015)	0.604 (±0.035)	0.713 (±0.011)	89	**0.723** (±0.030)
Hill-Valley	0.574 (±0.025)	0.542 (±0.057)	**0.657** (±0.056)	92	0.642 (±0.076)
USPS	0.579 (±0.016)	0.529 (±0.029)	0.627 (±0.088)	92	**0.672** (±0.022)
Mean	0.573 (±0.015)	0.575 (±0.04)	0.627 (±0.088)	96	0.630 (±0.051)

Figure 4.1: Graphical illustration of NMI values versus various pruning rates for all data sets.



5. ENSEMBLE BASED FEATURE SELECTION METHODS

5.1 GENERATION OF ENSEMBLE LIBRARY

In this study, 28 different traditional feature selection algorithms are used to create the ensemble of feature selection methods. These are grouped into four categories similarity based, information theoretical based, sparse learning based and statistical based methods.

5.1.1 Similarity Based Methods

In general, feature selection algorithms utilize a variety of criteria distance, separability, information, correlation, dependency, and reconfiguration error to define attribute appropriateness. Similarity-based feature selection methods assess the importance of preserving data similarity and the importance of features. They are divided into five sub-categories as follows:

a) **Laplacian Score**
The Laplacian Score (LS) is an uncontrolled and three-phase attribute selection algorithm that can best protect the data manifold structure He et al. (2006). It is generated by Laplacian Eigenmaps Belkin & Niyogi (2002) and Locality Preserving Projection He & Niyogi (2003) which evaluates the features according to their locality preserving power.

b) **Spectral Feature Selection (SPEC)**
SPEC is a graph based feature selection method which is an extension of LS. It can be used for both supervised and unsupervised scenarios. For example, in the unsupervised case, Radial Basis Function (RBF) kernel function is used to measure data similarity. In the supervised case, a diagonal matrix

is constructed using affinity matrix information Zhao & Liu (2007).

c) **Fisher Score**
Fisher score is supervised feature selection methods that selects each feature independently according to their scores under the Fisher criterion Duda et al. (2012).

d) **Trace Ratio Criterion**
In the Trace Ratio Criterion method, a feature subset is selected based on the corresponding subset-level score, which is calculated in a trace ratio form Nie et al. (2008).

e) **ReliefF**
ReliefF algorithm is one of the most successful filtering feature selection methods. It selects features to separate instances from different classes Robnik-Šikonja & Kononenko (2003). It assesses the quality of features based on how well their values discriminate between samples that are near each other.

5.1.2 Information Theoretical Based Methods

Information theoretical based methods use different heuristic filter criteria to measure the importance of the attributes which maximize the relevance of the attributes and minimize their redundancy Duda et al. (2012). These types of methods can be divided into nine sub-categories as follows:

a) **Mutual Information Maximization (MIM) (or Information Gain)**
MIM evaluates the significance of a feature by its correlation with the class label Lewis (1992).

b) **Mutual Information Feature Selection (MIFS)**

The MIFS criterion considers both feature relevance and feature redundancy in the feature selection phase Battiti (1994).

c) **Minimum Redundancy Maximum Relevance (MRMR)**

The MRMR criterion considers both feature with maximum relevance and feature with minimum redundancy in the feature selection phase Peng et al. (2005).

d) **Conditional Infomax Feature Extraction (CIFE)**

As long as the feature redundancy of a given class label is stronger than the intra feature redundancy, the feature selection is affected negatively Tang et al. (2006). CIFES takes this into account by including a third term which maximizes the conditional redundancy between unselected features and already selected features for a given class label.

e) **Joint Mutual Information (JMI)**

MIFS and MRMR reduce feature redundancy in the feature selection process. It is recommended that JMI, an alternative criterion, increase the shared information between the new selected attribute, and the selected attributes given the class labels Yang & Moody (2000). The basic idea of JMI consists of adding new features that are complementary to existing features for a given class label.

f) **Conditional Mutual Information Maximization (CMIM**)

CMIM selects features iteratively by maximizing the mutual information with the class labels given the selected features Vidal-Naquet & Ullman (2003); Fleuret (2004).

g) **Double Input Symmetrical Relevance (DISR)**

DISR performs normalization techniques to normalize mutual information Meyer & Bontempi (2006).

h) **Fast Correlation Based Filter (FCBF)**

FCBF is an algorithm that has the capability of being employed as the approximation method for relevance and redundancy analysis Yu & Liu (2003).

i) **Interaction Capping**

The Interaction Capping feature selection criterion is similar to CMIM except that Interaction Capping restricts the term $I(X_j;X_k) - I(X_j;X_k|Y)$ to be nonnegative where $I(.,.)$ refers to the information gain function Jakulin (2005).

5.1.3 Sparse Learning Based Methods

Filter based feature selection methods select attributes that are independent of any learning algorithm. The bias of the learning algorithm is not taken into account in filter type approaches, so that the selected attributes may not be optimal for a specific problem. In order to overcome this issue, embedded type approaches are developed which embed the feature selection step into the learning model construction so that each step feeds into one other. There are three types of embedded feature selection methods: The first is based on pruning redundant features by assigning binary weights to features while maintaining prediction accuracy. The second type consists of a built-in feature selection mechanism such as ID3 Quinlan (1986) and C4.5 Ross Quinlan (1993). The last type refers to sparse learning based methods, which minimize empirical error by inducing a regularization term to the objective function so that some feature coefficients are small or exactly zero. There are different types of sparse based approaches, but we will introduce only those that are used in this study.

a) **Multi-Cluster Feature Selection (MCFS)**

Most of the existing sparse feature selection methods use label information of the data where the feature selection step is modeled after determining the sparse feature coefficients. Since labeled data is costly and time consuming to collect, unsupervised sparse learning based feature selection has gained increasing attention in recent years Du & Shen (2015); Liu et al. (2014). MCFS is one of the first unsupervised feature selection algorithms developed and performs spectral clustering and sparse coefficient learning before the feature selection step Cai et al. (2010).

b) **Feature Selection with L_1 norm Regularization**

This method performs feature selection by assigning insignificant input features with zero weight and useful features with a non zero weight by incorporating l_1 norm penalty functions to the objective function while minimizing the empirical error on the training set Tibshirani (1996).

c) $l_{2,1}$ **norm Regularized Discriminative Feature Selection**

A widely accepted criterion for choosing an unsupervised feature is to select attributes that best protect the manifold structure of the data He et al. (2006). One crucial property of 2,1-norm regularization is that it allows multiple predictors to share similar sparsity patterns. However, the resulting optimization problem is difficult to solve because of the nonsmoothness of 2,1-norm regularization.

d) **Nonnegative Discriminative Feature Selection (NDFS)**

NDFS is an algorithm that performs spectral clustering and attribute selection at the same time to select a subset of distinctive attributes. Li et al. (2012). Unlike other spectral clustering methods, NDFS executes nonnegative and orthogonal constraints in the spectral clustering phase which causes the learned pseudo class labels to be closer to real cluster results.

5.1.4 Statistical based Methods

Another feature selection algorithm category is based on various statistical measurements. Because they rely on statistical criteria instead of learning the algorithm to assess the appropriateness of attributes, most of these methods are filter-based methods. We can divide statistical methods into three categories:

a) **F-score**
In statistical analysis of binary classification, F-score is a measure of a test's accuracy. It considers both the precision and the recall of the test to obtain the scores Wright (1965).

b) **Gini Index**
Gini index is a statistical measure to quantify if the feature is able to separate instances from different classes Gini (1912).

c) **Correlation Based Feature Selection (CFS)**
The basic idea of CFS is a heuristic approach based on a correlation to evaluate the value of the attribute subset Hall & Smith (1999).

5.1.5 Feature Selection with Structure Features

Most of the feature selection algorithms are based on the assumption that the features are independent from each other though the essential structures among them are disregarded. Yet, in many real problems, features reveal various types of structures such as spatial or temporal smoothness, disjointed groups, trees and graphs Tibshirani et al. (2005). Next, we briefly give the idea of graph based and group based approaches.

a) **Feature Selection with Graph Feature Structures**
In many cases, strong dependencies may occur between the attributes so

that an unverified graph can be used to encode these dependencies such that nodes represent features and edges between two nodes showing the pairwise dependencies between features Cai et al. (2010). Those dependencies on the graph can be transformed to a more mathematical representation by adjacency matrices consisting of binary entries.

b) **Feature Selection with Group Feature Structures**

In many real-world applications, features represent group structures. One of the most common examples is seen in multi-factor analysis-of-variance (ANOVA), where each factor is associated with several groups. When selecting attributes, this method obtains accurate predictions when the group structure between attributes is considered Cai et al. (2010).

5.1.6 Wraper Methods

Wrapper methods consider the selection of a set of features as a search problem, where different combinations are prepared, evaluated, and compared to other combinations Whitney (1971); Marill & Green (1963).

5.2 MATHEMATICAL MODEL

In this study, we developed a model which determines the optimum subset of the different solutions among a library of 28 feature selection methods (summarized in the previous sections). After the generation step of the ensemble of feature selectors on the training set,the accuracy and diversity trade off within the ensemble are introduced with a matrix T which is defined by equation 5.1 below:

$$[T] = \begin{cases} Acc_i, & i = j \\ \sum\{Y_i^{\mathrm{DES}} \neq Y_j^{\mathrm{DES}}\}, & i \neq j \end{cases} \tag{5.1}$$

In the matrix T, the total number of correct predictions of the i-th feature selector by base classifier DES, represented by Acc_i are defined on the diagonal entries and the total number of uncommon predictions of these feature selectors are assigned to the off-diagonal entries as a measure of diversity in T. In this way, diagonal elements of the matrix T represent the accuracy criterion whereas the off-diagonal elements represent diversity.

Here, we adapted our previous study for *ensemble clustering selection* approach in Otar & Akyüz (2017) and Üçüncü et al. (2018) to the *ensemble based feature selection model*. The previous model differs by involving different accuracy and diversity metrics in the matrix T. In Otar & Akyüz (2017) and Üçüncü et al. (2018), since the problem was a clustering problem which did not have label information, normalized mutual information values were previously used to define accuracy and diversity metrics.

In this thesis, quality (accuracy) and diversity metrics are defined to be the total number of accurate predictions and the uncommon pairwise predictions of each binary couple of feature selectors, respectively. Basically, the main intuition behind maximizing the accuracy and diversity trade off within the ensemble learning library studied previously in Özöğür-Akyüz et al. (2015); Zhang et al. (2006a); Otar & Akyüz (2017); Üçüncü et al. (2018) is to adapt the ensemble of feature selectors with the same optimization model (5.2) by adapting the accuracy diversity notion using metrics based on feature selectors as follows:

$$\begin{aligned}
&\text{maximize } x^T T x \\
&\text{subject to } \textstyle\sum_{i=1}^{n} x_i = k, \\
&x_i \in \{0,1\}\ (i = 1,2,\ldots,n),
\end{aligned} \tag{5.2}$$

where k stands for the pruning rate of the ensemble, i.e., the cardinality of the subset of the ensemble. Since above 0- 1 binary integer problem (5.2) is NP-hard in general. However, there exist studies that approximates the solution of integer programming optimally in the literature for many years. One of the recent study in the field of machine learning, presents a new formulation of the classical univariate decision tree problem as an Mixed Integer Programming (MIP) problem that motivates their new classification method which is called Optimal Classification Trees (OCT) Bertsimas & Dunn (2017). In one of the applications of MIP involves with the tensor complementarity problem where a global solver LINGO was used to obtain optimal solution Du & Zhang (2019). The proposed ensemble pruning model for problem in this thesis was inspired from the MIP in Zhang et al. (2006a) which includes the parameter k in its constraint. The constraint in problem (5.2) determines the size of the subset of the ensemble, in other words, the parameter k is given by the user beforehand as a pruning rate. Furthermore, its variables are integer because it is defined with a sum that counts the number of elements to be selected in the new subset of ensemble which can be defined as cardinality constraint by a zero norm. Thus, the solution of the MIP and the accuracy of the machine learning model highly depend on the optimal value of parameter k. The objective of our study is to get rid of the parameter k to automate finding the optimal value of k while selecting the best candidates considering both accuracy and diversity within the optimization problem. In order to do this, we moved that cardinality constraint to the objective function with a regularization constant so that the whole MIP turned into continuous optimization problem. This procedure can also be regarded as regularization in statistical learning to overcome the complexity as in Lasso regularization.

This relaxation by moving the cardinality constraint to the objective function with a regularization constant ρ is further improved by adding bound constraint to variable x to obtain sparse solution as shown below:

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & -x^T T x + \rho \|x\|_0 \\ \text{subject to} \quad & x^T x = 1, \end{aligned} \tag{5.3}$$

Since the model (5.2) is relaxed to a continuous programming, in order to keep the sparsity, an additional constraint that bounds x is added to the problem (5.3).

The proposed ensemble based feature selection model introduced by equation (5.3) is non-convex because of the second term and the matrix T might be negative definite. Approximating the zero norm with the student log likelihood distribution and adding/subtracting the term τI to the first term leads to a difference of convex functions where τ is defined to be $\tau \geq \max\{0, -\lambda_{min}(T)\}$. Hence the optimization problem (5.3) can be rewritten as:

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & \left\{ \tau \|x\|_2^2 - \left[x^T \left(\tilde{T} + \tau I \right) x - \rho \sum_{i=1}^{n} \frac{\log(1 + |x_i|/\epsilon)}{\log(1 + 1/\epsilon)} \right] \right\} \\ \text{subject to} \quad & x^T x = 1, \end{aligned} \tag{5.4}$$

where $\hat{T}$ is the normalized form of the matrix T and ρ refers to the regularization parameter corresponding to the cardinality of a subset of the ensemble which is introduced by a zero norm.

If the absolute value in equation (5.4) is replaced with an additional variable y_i by adding an extra constraint $-y \leq x \leq y$, the model (5.4) takes the following final form:

$$\begin{aligned} \underset{x,y \in \mathbb{R}^n}{\text{minimize}} \quad & \left\{ \tau \|x\|_2^2 - \left[x^T \left(\tilde{T} + \tau I \right) x - \rho \sum_{i=1}^{n} \frac{\log(1 + y_i/\epsilon)}{\log(1 + 1/\epsilon)} \right] \right\} \\ \text{subject to} \quad & -y \leq x \leq y, \\ & x^T x = 1. \end{aligned} \tag{5.5}$$

5.3 EXPERIMENTAL RESULTS

In this work, the eight most popular feature selection data sets, selected from different domains are used Li et al. (2018). The features that exist in these datasets are either numerical or categorical values. The number of features, number of instances and number of classes are presented in Table (5.1).

Table 5.1: Detailed information of benchmark datasets.

Dataset	Type	Feature Value	# Feature	# Instance	# Class
Lung small	Bio	Discrete	325	73	7
Madelon	Artificial	Continuous	500	2600	2
Yale	Image	Continuous	1024	165	15
WarpAR10P	Image	Continuous	2400	130	10
Colon	Bio	Discrete	2000	62	2
Urban Land Cover	Physical	Continuous	148	168	9
Libras Image	Bio	Continuous	91	360	15
Hill-Valley	Physical	Continuous	101	606	2

The dataset is divided into three parts:testing, validation, and training. 20 percent of the entire dataset is used for testing, and remaining 80 percent is divided into (20 percent) validation and (80 percent) training folds. 28 different feature selection algorithms were applied on training data. SVMs were employed with 5-fold cross-validation with the selected features by the 28 different feature selection techniques.

Optimization problem (5.5) is solved by the DCCP algorithm Grant et al. (2006) of cvxpy library of Python 3.7. The solution of the optimization model defined by equation (5.5) provides the indices of the selection of the results of the twenty eight attribute selection methods. The hyper-parameter ρ was determined by five fold cross-validation on the validation set among the different numbers which is $\rho = \left[10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^{2}, 10^{3}\right]$ and the threshold value of 0.01 was determined experimentally for x values to binarize it.The coordinates of vector x represent the feature selection methods and the indices of vector x having 0 values corresponds to redundant feature selectors being eliminated whereas indices of values of 1 correspond to the methods which are to be chosen.

The solution of the DCCP model corresponding to the optimum ρ parameter refers to the best subset among 28 feature selection methods. The elements of this subset are composed of the different attribute sets S_i^f introduced in Algorithm 2 that are generated by the those feature selection methods outputted by the DCCP method. The voting algorithm, was applied to aggregate the results of subsets S_i^f of the feature selectors. In this voting step, the solutions of the algorithms that were voted more than 50 percent were included in the selected attributes. We used voting method to select features as an aggregation of feature selector in the final subset determined by DCCP. In full ensemble there may be methods which decrease the performance of the ensemble. Our main goal is to eliminate such methods by pruning. One can increase the number of possible method candidates in the ensemble so that diversity increases. All pruning methods compared including the proposed DCCP model and also full ensemble results find the final set of features using voting. In this manner, voting should be considered as an aggregation function for both pruned and unpruned cases.

For example, if the solutions of 15 methods from 28 methods were selected to the subset by using the DCCP model (5.5), the attributes of those 15 techniques which passed the 50 percent of threshold would be considered to be the final attributes of the test data. All the steps described here are given in the flowchart shown by Algorithm 2 and Figure 5.1 and the performance of the models was compared with the methods in the literature called Joint Criterion Fern & Lin (2008).

The percentage of accuracy results on the test set for the proposed model PrunedOPT is given in the first column of Table 5.2 and accuracy values of the un-

Algorithm 2 Improved Ensemble Feature Selection with DCCP

```
Input:$X^{tr}$, $X^{val}$, $X^{test}$, $S_n$ (Feature Selection Algorithms)
Parameter:$\rho$ (DCCP parameter), k (Number of Features)
Output: percentage of accuracy
for $i \leftarrow 1$ to $n$ do
    $S_i^f = S_n(X)$ /* the feature selection subset that each feature selection algorithm chooses /*
    $Acc_i = acc(\text{DES}(X^{tr}_{S_i^f}))$ /* DES percentage of accuracies using the training set /*
end for
$T_{ii} = Acc_i$ /* The percentage of accuracies for feature selection methods/*
$T_{ij} = \sum_{i \neq j} Y_i^{\text{DES}} \neq Y_j^{\text{DES}}$ /* non-common estimation results for i-th and j-th feature selection methods /*
$\check{\text{S}}_\rho = \text{DCCP}(T, \rho)$ /* Obtaining the optimum subset of feature selection algorithms with DCCP method on $X^{val}$ /*
$F = Votting(\check{\text{S}}_\rho)$ on $X^{test}$
$Percentage\ of\ Accuracy = \text{DES}(X^{test})$
```

pruned case corresponding to Full Ensemble, and the Joint Criterion with its best pruning rates are illustrated by second and the third columns respectively. Here, the best pruning rate of Joint Criterion is selected among the values $[5, 10, 15, 20]$ based on corresponding accuracies. It is clear from the experimental results that the proposed ensemble based feature selection approach PrunedOPT achieves better prediction accuracies than both unpruned case and Joint Criterion. In Table 5.2, the bold numbers correspond to the best accuracy values measured by the ratio of correct predictions to the total number of examples in the test set.

In order to perform a thorough comparison of the results, we presented these results in Tables 5.2 and visually by Figure 5.2 for each of the eight data sets. In Figure 5.2, each subfigure stands for different data sets in which our proposed method called PrunedOPT and the method Joint Criterion are compared against their accuracy values versus pruning rate k. It is clear from each subfigure in Figure 5.2 that the proposed optimization model PrunedOPT approximates the optimal accuracy value with respect to its optimal pruning rate when compared with accuracy values corresponding to various pruning rates of Joint Criterion method.

Figure 5.1: Flow chart of the proposed method.

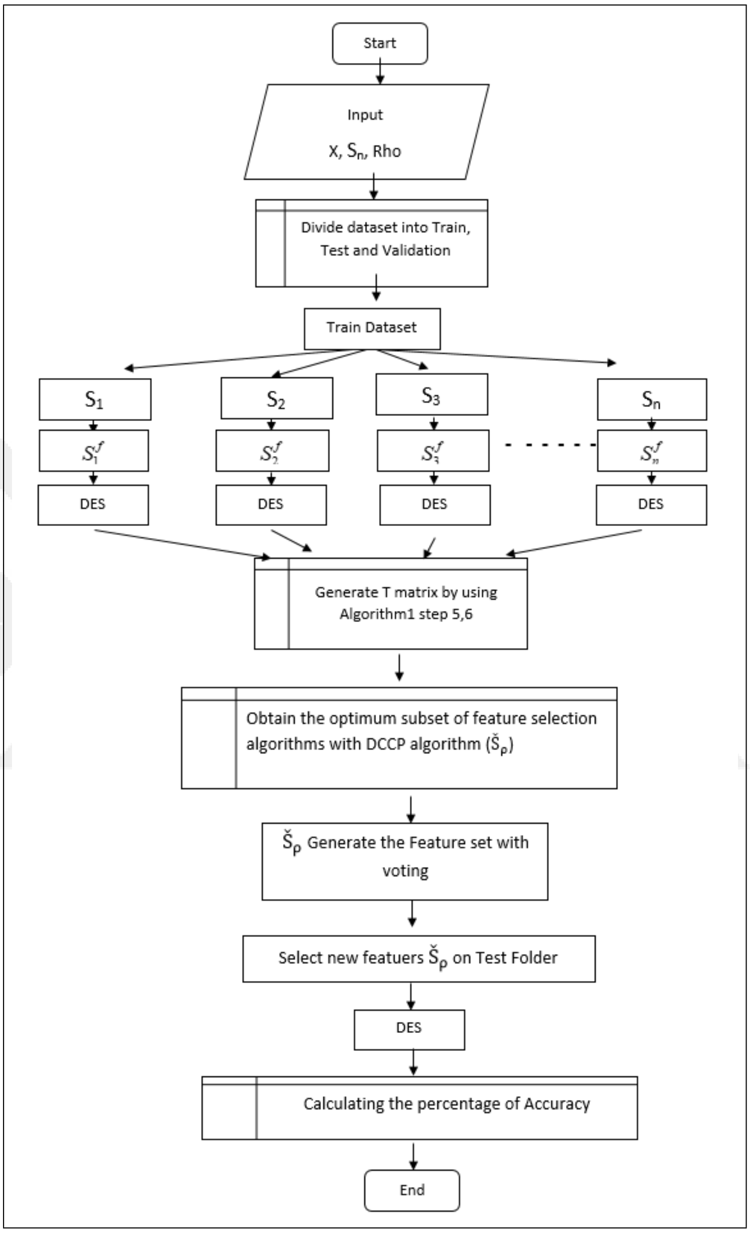
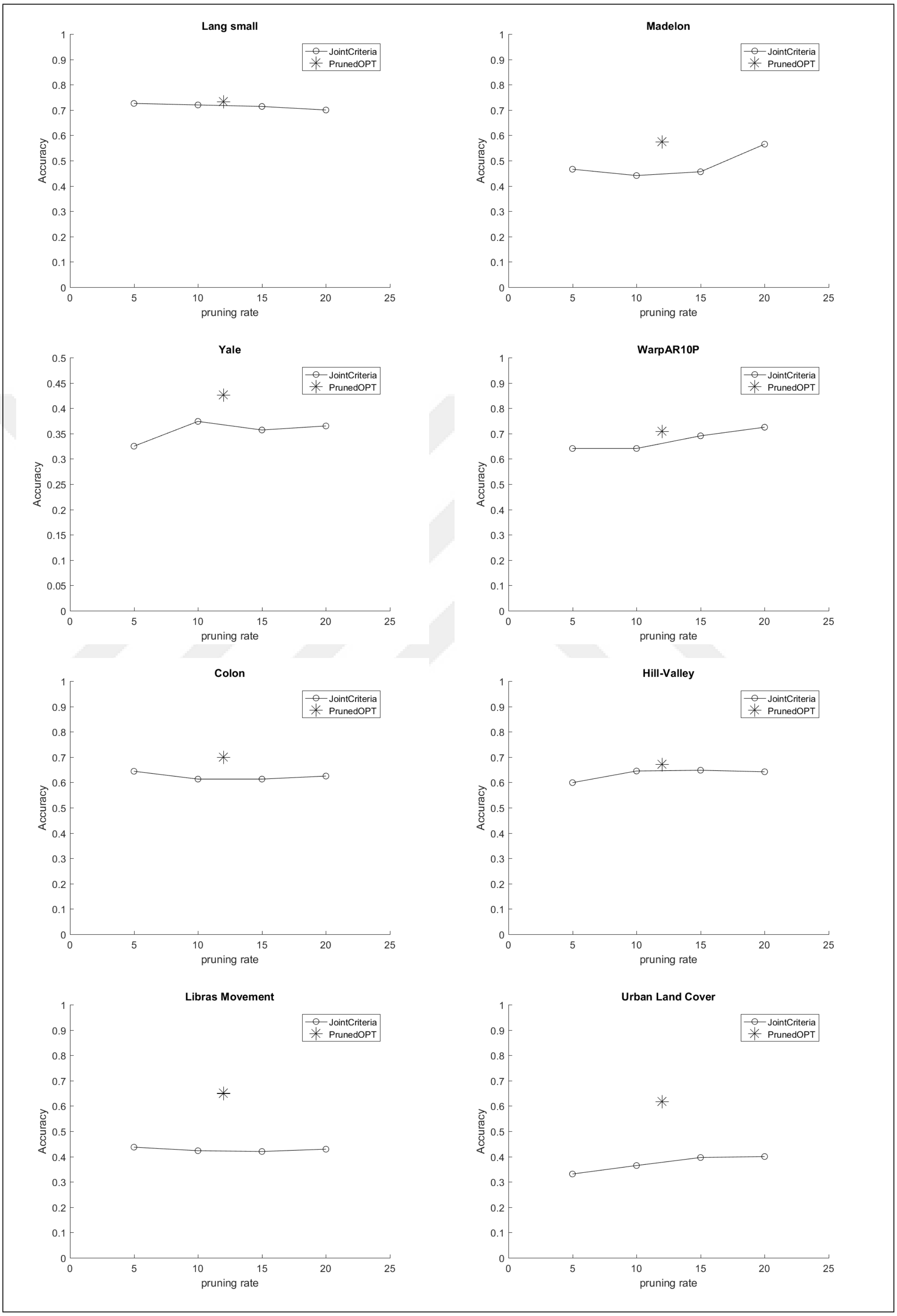

Start
Input
X, S_n, Rho
Divide dataset into Train, Test and Validation
Train Dataset
S1
S2
S3
Sn
DES
DES
DES
DES
Generate T matrix by using Algorithm1 step 5,6
Obtain the optimum subset of feature selection algorithms with DCCP algorithm (Šρ)
Šρ Generate the Feature set with voting
Select new featuers Šρ on Test Folder
DES
Calculating the percentage of Accuracy
End

Table 5.2: **Proposed feature selection algorithm percentage of accuracy.**

Dataset	Ensemble Feature Selection PrunedOPT by DCCP	Ensemble Feature Selection Unpruned Case	Joint Criterion (Pruning Rate)
Lung small	**0.740**	0.712	0.711 (5)
Madelon	**0.653**	0.580	0.578 (20)
Yale	**0.655**	0.467	0.422(20)
WarpAR10P	**0.741**	0.642	0.683 (20)
Colon	**0.759**	0.690	0.664 (20)
Urban Land Cover	**0.680**	0.591	0.40 (5)
Libras Movement	**0.701**	0.658	0.50 (20)
Hill-Valley	**0.690**	0.667	0.632 (20)

Figure 5.2: Graphical illustration of accuracy values versus various pruning rates for all data sets.



As the Joint Criterion method requires pruning rate to be an input parameter for each different input parameters, we get different accuracy values. In order to compare the best accuracy results of the Joint Criterion method with the proposed ensemble pruning, we illustrated the best of each method for each data referring to their corresponding pruning rates in Table 5.2. It should be noted that our proposed approach achieves better accuracy values than both unpruned case and Joint Criterion.

For each of the 8 datasets, the performance evaluation was measured against each of those 28 constituent feature selection methods. Those methods were run 5 times randomly. The performance (accuracy) results of these methods are shown in Table 5.4 -5.11 where the first column represents the method names, remaning second, third, fourth and fifth columns show the accuracy values calculated by using equation (5.6) below:

$$Accuracy = \frac{\#\ of\ correct\ predictions}{\#\ of\ samples}. \tag{5.6}$$

The last two columns named by AVG and STD show the average accuracy and average standard deviation values of 5 random iterations for each data set.

The average running time required for DCCP were calculated as two minutes while time required for the Joint Criterion method was three seconds. From these results, we observe that our proposed ensemble pruning method takes longer time than Joint Criterion. However, this drawback is not a fair comparison between two methods since the proposed pruning method gives not only higher accuracy but it also automates finding the optimal pruning rate with a unified framework of optimizing accuracy and diversity simultaneously; whereas the joint criterion finds the pruning rate combinatorially. Thus, when one compares the running times, time spend for trials of different selections of pruning rates should be considered. Further, one cannot know the optimal pruning rate without trials in Joint Criterion which differs also in each data set.

Table 5.3: Proposed feature selection algorithm percentage of best accuracy for each classifaction method.

Methods	PrunedOPT by DCCP	Unpruned Case	Joint Criterion
Linear SVM	**0.634**	0.615	0.557
Non-linear SVM	**0.625**	0.612	0.608
Decision Tree	0.608	**0.610**	0.578
DES	**0.702**	0.625	0.557

In our experimental analysis, the selected features by PrunedOPT, unpruned case and Joint Criterion for each data set were tested with widely used ML methods such as decision tree algorithm, nonlinear SVM and DES in Table 5.2. It is clear from the Table 5.2 that the proposed algorithm PrunedOPT has always better accuracy values in average for all data sets in Table 5.1 when compared with other methods for all classification algorithms except decision tree with a slight difference. The reason behind this can come from the well-known fact that there is no unique and best algorithm that works for all type of data sets. This slight differene can be seen as negligibile.

Table 5.4: The accuracy measurements for each feature selection algorithms for Small Lung dataset.

Data	Accuracy						
Small Lung Dataset	1st	2nd	3rd	4th	5th	AVG	STD
LS	0,53	0,21	0,33	0,00	0,32	0,28	0,19
(SPEC)	0,63	0,47	0,58	0,57	0,32	0,52	0,13
Fisher Score	0,42	0,37	0,67	0,29	0,47	0,44	0,14
Trace Ratio Criterion	0,32	0,32	0,00	0,86	0,21	0,34	0,32
ReliefF	0,16	0,53	0,58	0,43	0,68	0,48	0,20
MIM	0,58	0,42	0,42	0,29	0,47	0,44	0,11
MIFS	0,53	0,11	0,25	0,14	0,37	0,28	0,17
MRMR	0,47	0,32	0,50	0,57	0,68	0,51	0,14
CIFE	0,32	0,26	0,33	0,57	0,68	0,43	0,18
JMI	0,53	0,32	0,42	0,57	0,21	0,41	0,15
CMIM	0,21	0,42	0,17	0,43	0,32	0,31	0,12
DISR	0,53	0,26	0,17	0,29	0,47	0,34	0,15
FCBF	0,32	0,21	0,50	0,71	0,53	0,45	0,20
Interaction Capping	0,63	0,37	0,33	0,57	0,11	0,40	0,21
MCFS	0,26	0,32	0,33	0,86	0,37	0,43	0,24
L1 norm Regularization	0,47	0,26	0,58	0,43	0,47	0,44	0,12
l2;1 norm Regularized	0,37	0,11	0,42	0,57	0,47	0,39	0,17
NDFS	0,26	0,63	0,67	0,71	0,37	**0,53**	0,20
F-score	0,32	0,26	0,17	0,29	0,32	0,27	0,06
Gini Index	0,37	0,21	0,42	0,57	0,42	0,40	0,13
CFS	0,42	0,21	0,58	0,57	0,42	0,44	0,15
Wraper	0,53	0,21	0,25	0,14	0,26	0,28	0,15
Group Feature Structures	0,47	0,26	0,50	0,43	0,21	0,38	0,13
UDFS	0,26	0,32	0,67	0,29	0,21	0,35	0,18
Tree-fs	0,42	0,21	0,50	0,57	0,42	0,42	0,14
RFS	0,16	0,63	0,33	0,57	0,26	0,39	0,20
SVMBackward	0,63	0,26	0,50	0,71	0,26	0,47	0,21
SVMForward	0,32	0,42	0,50	0,57	0,32	0,42	0,11
AVG	0,40	0,35	0,41	0,48	0,37	0,40	0,16

Table 5.5: The accuracy measurements for each feature selection algorithms for Madelon dataset.

Data Set	Accuracy						
Madelon Data	1st	2nd	3rd	4th	5th	AVG	STD
LS	0,38	0,60	0,50	0,56	0,30	0,47	0,13
(SPEC)	0,38	0,50	0,17	0,56	0,50	0,42	0,16
Fisher Score	0,38	0,30	0,33	0,75	0,10	0,37	0,24
Trace Ratio Criterion	0,38	0,50	0,67	0,44	0,20	0,44	0,17
ReliefF	0,31	0,50	0,50	0,69	0,40	0,48	0,14
MIM	0,75	0,30	0,33	0,63	0,70	0,54	0,21
MIFS	0,63	0,60	0,17	0,56	0,60	0,51	0,19
MRMR	0,63	0,50	0,33	0,50	0,40	0,47	0,11
CIFE	0,38	0,40	0,50	0,50	0,40	0,44	0,06
JMI	0,56	0,60	0,33	0,56	0,70	0,55	0,13
CMIM	0,44	0,50	0,33	0,63	0,50	0,48	0,11
DISR	0,56	0,40	0,67	0,81	0,60	0,61	0,15
FCBF	0,56	0,40	0,50	0,69	0,40	0,51	0,12
Interaction Capping	0,44	0,60	0,33	0,56	0,70	0,53	0,14
MCFS	0,69	0,50	0,33	0,63	0,50	0,53	0,14
L1 norm Regularization	0,44	0,70	0,33	0,69	0,40	0,51	0,17
l2;1 norm Regularized	0,50	0,40	0,50	0,56	0,50	0,49	0,06
NDFS	0,44	0,60	0,83	0,88	0,60	**0,67**	0,18
F-score	0,56	0,20	0,33	0,63	0,40	0,42	0,17
Gini Index	0,63	0,40	0,33	0,69	0,60	0,53	0,15
CFS	0,50	0,20	0,33	0,69	0,20	0,38	0,21
Wraper	0,31	0,50	0,17	0,56	0,30	0,37	0,16
Group Feature Structures	0,44	0,30	0,50	0,81	0,30	0,47	0,21
UDFS	0,44	0,50	0,33	0,69	0,40	0,47	0,13
Tree-fs	0,44	0,10	0,83	0,50	0,60	0,49	0,27
RFS	0,44	0,30	0,17	0,56	0,40	0,37	0,15
SVMBackward	0,56	0,50	0,33	0,63	0,50	0,50	0,11
SVMForward	0,50	0,50	0,33	0,50	0,10	0,39	0,18
AVG	0,48	0,47	0,40	0,62	0,43	0,47	0,15

Table 5.6: **The accuracy measurements for each feature selection algorithms for Yale dataset.**

Data	**Accuracy**						
Yale Dataset	1st	2nd	3rd	4th	5th	AVG	STD
LS	0,31	0,30	0,12	0,55	0,86	0,43	0,29
(SPEC)	0,17	0,37	0,41	0,64	0,57	0,43	0,18
Fisher Score	0,36	0,44	0,53	0,36	0,57	0,45	0,10
Trace Ratio Criterion	0,12	0,26	0,41	0,45	0,29	0,31	0,13
ReliefF	0,24	0,33	0,24	0,36	0,29	0,29	0,06
MIM	0,19	0,44	0,47	0,55	0,71	0,47	0,19
MIFS	0,24	0,37	0,29	0,36	0,43	0,34	0,07
MRMR	0,36	0,22	0,65	0,64	0,57	0,49	0,19
CIFE	0,33	0,30	0,47	0,82	0,43	0,47	0,21
JMI	0,40	0,48	0,59	0,82	0,57	**0,57**	0,16
CMIM	0,24	0,48	0,47	0,36	0,43	0,40	0,10
DISR	0,29	0,44	0,59	0,64	0,71	0,53	0,17
FCBF	0,33	0,41	0,47	0,64	0,57	0,48	0,12
Interaction Capping	0,29	0,48	0,59	0,36	1,00	0,54	0,28
MCFS	0,19	0,33	0,12	0,27	0,57	0,30	0,17
L1 norm Regularization	0,26	0,41	0,65	0,55	0,57	0,49	0,15
l2;1 norm Regularized	0,38	0,37	0,41	0,82	0,71	0,54	0,21
NDFS	0,33	0,33	0,24	0,55	0,57	0,40	0,15
F-score	0,24	0,22	0,24	0,27	0,29	0,25	0,03
Gini Index	0,29	0,37	0,41	0,36	0,43	0,37	0,06
CFS	0,17	0,19	0,35	0,18	0,57	0,29	0,17
Wraper	0,31	0,52	0,53	0,45	0,29	0,42	0,12
Group Feature Structures	0,14	0,48	0,53	0,27	0,86	0,46	0,27
UDFS	0,17	0,56	0,59	0,45	0,57	0,47	0,18
Tree-fs	0,26	0,41	0,53	0,45	0,14	0,36	0,16
RFS	0,33	0,48	0,24	0,27	0,71	0,41	0,20
SVMBackward	0,24	0,30	0,47	0,64	0,86	0,50	0,25
SVMForward	0,31	0,41	0,35	0,18	0,57	0,36	0,14
AVG	0,26	0,42	0,42	0,47	0,56	0,42	0,16

Table 5.7: **The accuracy measurements for each feature selection algorithms for Warp dataset.**

Data	**Accuracy**						
Warp Dataset	1st	2nd	3rd	4th	5th	AVG	STD
LS	0,38	0,31	0,30	0,13	0,13	0,25	0,12
(SPEC)	0,44	0,25	0,70	0,50	0,31	0,44	0,18
Fisher Score	0,50	0,44	0,90	0,56	0,63	0,61	0,18
Trace Ratio Criterion	0,44	0,56	0,90	0,19	0,69	0,56	0,27
ReliefF	0,50	0,44	0,60	0,25	0,19	0,40	0,17
MIM	0,69	0,69	0,80	0,50	0,63	0,66	0,11
MIFS	0,13	0,31	0,60	0,50	0,31	0,37	0,18
MRMR	0,56	0,38	0,80	0,56	0,44	0,55	0,16
CIFE	0,56	0,25	0,60	0,56	0,19	0,43	0,20
JMI	0,75	0,50	0,70	0,56	0,50	0,60	0,12
CMIM	0,56	0,56	0,50	0,38	0,50	0,50	0,08
DISR	0,69	0,75	0,60	0,31	0,44	0,56	0,18
FCBF	0,19	0,63	1,00	0,75	0,75	0,66	0,30
Interaction Capping	0,56	0,56	0,70	0,38	0,50	0,54	0,12
MCFS	0,69	0,56	0,90	0,69	0,56	0,68	0,14
L1 norm Regularization	0,50	0,19	0,70	0,56	0,19	0,43	0,23
l2;1 norm Regularized	0,88	0,50	0,50	0,69	0,56	**0,63**	0,16
NDFS	0,88	0,38	0,80	0,50	0,56	0,62	0,21
F-score	0,44	0,25	0,60	0,50	0,50	0,46	0,13
Gini Index	0,06	0,44	0,60	0,44	0,25	0,36	0,21
CFS	0,50	0,88	0,90	0,19	0,31	0,56	0,32
Wraper	0,38	0,50	0,50	0,50	0,31	0,44	0,09
Group Feature Structures	0,56	0,38	0,90	0,56	0,19	0,52	0,26
UDFS	0,63	0,06	0,70	0,50	0,44	0,47	0,25
Tree-fs	0,38	0,88	0,70	0,38	0,19	0,50	0,28
RFS	0,44	0,69	0,70	0,38	0,38	0,52	0,17
SVMBackward	0,63	0,31	0,90	0,63	0,31	0,56	0,25
SVMForward	0,38	0,63	0,50	0,88	0,19	0,51	0,26
AVG	0,51	0,51	0,7	0,48	0,39	0,51	0,19

Table 5.8: **The accuracy measurements for each feature selection algorithms for Colon dataset.**

Data	**Accuracy**						
Colon Dataset	1st	2nd	3rd	4th	5th	AVG	STD
LS	0,31	0,88	0,80	0,50	0,44	0,59	0,24
(SPEC)	0,56	0,75	0,80	0,44	0,88	0,69	0,18
Fisher Score	0,81	1,00	0,90	0,63	0,94	0,86	0,15
Trace Ratio Criterion	0,88	0,94	1,00	0,75	0,69	0,85	0,13
ReliefF	0,88	1,00	0,50	0,69	0,81	0,78	0,19
MIM	0,88	1,00	0,90	0,75	0,88	**0,88**	0,09
MIFS	0,63	0,94	0,50	0,44	0,63	0,63	0,19
MRMR	0,56	0,94	0,80	0,63	0,88	0,76	0,16
CIFE	1,00	0,88	0,40	0,88	0,88	0,81	0,23
JMI	0,94	1,00	0,80	0,63	0,88	0,85	0,14
CMIM	0,69	0,88	1,00	0,69	0,81	0,81	0,13
DISR	0,94	0,63	1,00	0,88	0,88	0,86	0,14
FCBF	0,88	0,94	1,00	0,88	0,88	0,91	0,06
Interaction Capping	0,56	0,69	0,40	0,31	0,38	0,47	0,15
MCFS	0,88	0,75	0,50	0,44	0,88	0,69	0,21
L1 norm Regularization	0,50	0,50	0,50	0,75	0,56	0,56	0,11
l2;1 norm Regularized	0,94	0,75	0,90	0,56	0,88	0,81	0,15
NDFS	0,88	0,81	0,60	0,56	0,75	0,72	0,13
F-score	0,63	0,44	0,50	0,56	0,75	0,58	0,12
Gini Index	0,81	0,69	0,60	0,56	0,56	0,65	0,11
CFS	0,50	0,88	0,50	0,88	0,81	0,71	0,20
Wraper	0,56	0,75	0,90	0,81	0,38	0,68	0,21
Group Feature Structures	0,75	0,88	0,80	0,56	0,69	0,74	0,12
UDFS	0,56	0,81	0,70	0,63	0,81	0,70	0,11
Tree-fs	0,94	0,69	0,80	0,56	0,69	0,74	0,14
RFS	0,81	0,50	0,80	0,50	0,81	0,69	0,17
SVMBackward	0,69	0,63	0,50	0,81	0,69	0,66	0,11
SVMForward	0,88	0,69	0,60	0,63	0,81	0,72	0,12
AVG	0,74	0,83	0,71	0,63	0,74	0,72	0,14

Table 5.9: The accuracy measurements for each feature selection algorithms for Urban Land Cover dataset.

Data	**Accuracy**						
Urban Land Cover Dataset	1st	2nd	3rd	4th	5th	AVG	STD
LS	0,56	0,64	0,61	0,49	0,70	0,60	0,08
(SPEC)	0,47	0,70	0,70	0,61	0,35	0,57	0,15
Fisher Score	0,88	0,64	0,65	0,44	0,85	0,69	0,18
Trace Ratio Criterion	0,70	0,58	0,61	0,61	0,55	0,61	0,06
ReliefF	0,65	0,51	0,65	0,38	0,45	0,53	0,12
MIM	0,65	0,45	0,38	0,44	0,60	0,50	0,12
MIFS	0,52	0,58	0,70	0,32	0,45	0,51	0,14
MRMR	0,79	0,58	0,43	0,55	0,50	0,57	0,14
CIFE	0,65	0,58	0,79	0,44	0,60	0,61	0,13
JMI	0,70	0,58	0,56	0,61	0,35	0,56	0,13
CMIM	0,61	0,51	0,70	0,61	0,50	0,59	0,08
DISR	0,65	0,95	0,75	0,49	0,35	0,64	0,23
FCBF	0,65	0,64	0,84	0,38	0,75	0,65	0,17
Interaction Capping	0,79	0,70	0,70	0,73	0,65	**0,71**	0,05
MCFS	0,56	0,58	0,52	0,26	0,70	0,52	0,16
L1 norm Regularization	0,56	0,58	0,75	0,55	0,65	0,62	0,08
l2;1 norm Regularized	0,56	0,64	0,75	0,49	0,55	0,60	0,10
NDFS	0,79	0,58	0,65	0,49	0,45	0,59	0,14
F-score	0,65	0,58	0,56	0,49	0,55	0,57	0,06
Gini Index	0,52	0,51	0,70	0,44	0,50	0,53	0,10
CFS	0,56	0,58	0,61	0,38	0,50	0,52	0,09
Wraper	0,47	0,51	0,65	0,61	0,55	0,56	0,07
Group Feature Structures	0,52	0,45	0,56	0,32	0,50	0,47	0,09
UDFS	0,61	0,70	0,70	0,61	0,40	0,60	0,12
Tree-fs	0,70	0,64	0,70	0,67	0,45	0,63	0,10
RFS	0,79	0,51	0,52	0,44	0,60	0,57	0,14
SVMBackward	0,70	0,76	0,43	0,61	0,70	0,64	0,13
SVMForward	0,61	0,45	0,56	0,55	0,50	0,54	0,06
AVG	0.63	0,59	0,63	0,50	0,54	0,58	0,11

Table 5.10: The accuracy measurements for each feature selection algorithms for Libras Movement dataset.

Data	Accuracy						
Libras Movement Dataset	1st	2nd	3rd	4th	5th	AVG	STD
LS	0,49	0,30	0,72	0,67	0,56	0,55	0,17
(SPEC)	0,36	0,36	0,88	0,76	0,56	0,58	0,23
Fisher Score	0,43	0,59	0,62	0,62	0,69	0,59	0,10
Trace Ratio Criterion	0,36	0,36	0,62	0,76	0,69	0,56	0,19
ReliefF	0,61	0,30	0,35	0,29	0,50	0,41	0,14
MIM	0,30	0,54	0,56	0,71	0,56	0,54	0,15
MIFS	0,43	0,42	0,56	0,67	0,69	0,55	0,13
MRMR	0,43	0,36	0,72	0,90	0,75	0,63	0,23
CIFE	0,49	0,36	0,46	0,67	0,75	0,54	0,16
JMI	0,43	0,48	0,41	0,67	0,69	0,53	0,14
CMIM	0,43	0,36	0,51	0,52	0,63	0,49	0,10
DISR	0,61	0,48	0,83	0,52	0,75	**0,64**	0,15
FCBF	0,43	0,30	0,77	0,43	0,50	0,49	0,18
Interaction Capping	0,49	0,42	0,51	0,52	0,50	0,49	0,04
MCFS	0,49	0,42	0,46	0,62	0,69	0,53	0,11
L1 norm Regularization	0,43	0,36	0,72	0,57	0,56	0,53	0,14
l2;1 norm Regularized	0,49	0,48	0,72	0,62	0,69	0,60	0,11
NDFS	0,74	0,36	0,51	0,76	0,75	0,62	0,18
F-score	0,61	0,30	0,62	0,67	0,63	0,56	0,15
Gini Index	0,30	0,48	0,67	0,62	0,50	0,51	0,14
CFS	0,55	0,42	0,51	0,57	0,50	0,51	0,06
Wraper	0,43	0,48	0,67	0,71	0,69	0,59	0,13
Group Feature Structures	0,49	0,36	0,62	0,62	0,56	0,53	0,11
UDFS	0,49	0,48	0,51	0,71	0,56	0,55	0,10
Tree-fs	0,49	0,30	0,62	0,57	0,63	0,52	0,13
RFS	0,61	0,48	0,67	0,48	0,63	0,57	0,09
SVMBackward	0,49	0,42	0,77	0,71	0,81	0,64	0,18
SVMForward	0,43	0,42	0,67	0,43	0,56	0,50	0,11
AVG	0,47	0,42	0,61	0,62	0,62	0,54	0.13

Table 5.11: The accuracy measurements for each feature selection algorithms for Hill-Valley dataset.

Data	**Accuracy**						
Hill-Valley Dataset	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	AVG	STD
LS	0,63	0,33	0,50	0,31	0,50	0,45	0,13
(SPEC)	0,56	0,27	0,44	0,44	0,63	0,47	0,14
Fisher Score	0,50	0,53	0,44	0,44	0,50	0,48	0,04
Trace Ratio Criterion	0,63	0,40	0,56	0,31	0,44	0,47	0,13
ReliefF	0,63	0,27	0,56	0,56	0,50	0,50	0,14
MIM	0,50	0,60	0,63	0,50	0,69	**0,58**	0,08
MIFS	0,44	0,53	0,38	0,56	0,38	0,46	0,09
MRMR	0,50	0,40	0,44	0,50	0,56	0,48	0,06
CIFE	0,69	0,53	0,44	0,50	0,63	0,56	0,10
JMI	0,50	0,40	0,63	0,63	0,44	0,52	0,10
CMIM	0,44	0,40	0,38	0,50	0,44	0,43	0,05
DISR	0,63	0,60	0,38	0,50	0,69	0,56	0,12
FCBF	0,44	0,53	0,56	0,25	0,50	0,46	0,12
Interaction Capping	0,88	0,67	0,38	0,44	0,44	0,56	0,21
MCFS	0,81	0,60	0,38	0,38	0,63	0,56	0,19
L1 norm Regularization	0,69	0,60	0,44	0,31	0,56	0,52	0,15
l2;1 norm Regularized	0,63	0,40	0,63	0,63	0,56	0,57	0,10
NDFS	0,63	0,53	0,44	0,31	0,50	0,48	0,12
F-score	0,63	0,40	0,38	0,63	0,31	0,47	0,15
Gini Index	0,50	0,53	0,38	0,44	0,69	0,51	0,12
CFS	0,56	0,40	0,44	0,56	0,56	0,51	0,08
Wraper	0,50	0,33	0,56	0,56	0,31	0,45	0,12
Group Feature Structures	0,75	0,40	0,31	0,50	0,56	0,51	0,17
UDFS	0,56	0,40	0,63	0,38	0,63	0,52	0,12
Tree-fs	0,75	0,60	0,44	0,50	0,19	0,50	0,21
RFS	0,56	0,40	0,44	0,38	0,38	0,43	0,08
SVMBackward	0,63	0,33	0,56	0,50	0,69	0,54	0,14
SVMForward	0,63	0,27	0,56	0,56	0,50	0,50	0,14
AVG	0,6	0.50	0,47	0,46	0,51	0.50	0.12

6. RESULT AND DISCUSSION

This study evaluates the performance of the novel ensemble-pruning model by comparing it with that of other similar models such as Cluster-And-Select, Joint-Criterion, and DES-Cluster methods. The same datasets and settings are utilized throughout the experiments, and a total of 11 benchmark datasets were applied, such as Frog MFCCs, Glass, Movement, Seeds, Segmentation, Synthetic Control, Wine, Zoo, Yale, Hill-Valley, and USPS. Experimental results revealed that the proposed model contains no cardinality of the subset chosen. More importantly, our experimental evaluation demonstrates that our ensemble approach produces better performance than other cluster ensemble selection models. Our method is also advantageous in that it minimizes pruning-rate search space. As well as maximizing the diversity-accuracy trade-off, we observe that our proposed pruning algorithm is independent of the data domain. Even after comparing our prediction results with existing cluster ensemble-selection methods, our suggested ensemble model is superior in terms of finding better optimal solutions.

Same model was adapted to ensemble learning based feature selection.The proposed approach is validated on the most well known data sets and the performance results are compared with an un-pruned case of ensemble learning and Joint criterion method. DES is used as a classifier of these classification tasks. In addition to this, we implemented our model with other classification algorithms such as Linear SVM, nonlinear SVM and decision tree method where our proposed approach gave better accuracy performance with those classification techniques as well. As a future study, ensemble library can be enhanced by considering data variation techniques such as bagging.The performance evaluation was carried out against each of those 28 constituent feature selection methods. When analyzing feature selection algorithms individually versus Ensemble Feature selection method performance, ensemble methods show great promise for large feature domains. It turns out that the best trade-off between accuracy and diversity performance depends on the ensemble feature selection model, giving rise to a new model selection strategy.

REFERENCES

Books

Cristianini, N. & Shawe-Taylor, J., 2000. An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods. New York, NY, USA: Cambridge University Press.

Duda, R. O., Hart, P. E., & Stork, D. G., 2012. Pattern classification. John Wiley & Sons.

Vapnik, V. N., 1998. Statistical Learning Theory. Wiley-Interscience.

Zhou, Z.-H., 2012. Ensemble Methods: Foundations and Algorithms. Chapman & Hall/CRC, 1st ed.

Periodicals

Abeel, T., Helleputte, T., Van de Peer, Y., Dupont, P., & Saeys, Y., 2009. Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. Bioinformatics **26(3)**, pp. 392–398.

Akbari, O. S., Bellen, H. J., Bier, E., Bullock, S. L., Burt, A., Church, G. M., Cook, K. R., Duchek, P., Edwards, O. R., Esvelt, K. M. et al., 2015. Safeguarding gene drive experiments in the laboratory. Science **349(6251)**, pp. 927–929.

Alizadeh, H., Minaei-Bidgoli, B., & Parvin, H., 2014. Cluster ensemble selection based on a new cluster stability measure. Intell. Data Anal. **18(3)**, pp. 389–408.

Ayad, H., 2008. Voting-based consensus of data partitions .

Barandiaran, I., 1998. The random subspace method for constructing decision forests. IEEE transactions on pattern analysis and machine intelligence **20(8)**.

Battiti, R., 1994. Using mutual information for selecting features in supervised neural net learning. IEEE Transactions on neural networks **5(4)**, pp. 537–550.

Bellal, F., Elghazel, H., & Aussem, A., 2012. A semi-supervised feature ranking method with ensemble learning. Pattern Recognition Letters **33(10)**, pp. 1426–1433.

Berikov, V., 2014. Weighted ensemble of algorithms for complex data clustering. Pattern Recognition Letters **38**, pp. 99 – 106.

Bertsimas, D. & Dunn, J., 2017. Optimal classification trees. Machine Learning **106(7)**, pp. 1039–1082.

Bezdek, J. C., Coray, C., Gunderson, R., & Watson, J., 1981. Detection and characterization of cluster substructure i. linear structure: Fuzzy c-lines. SIAM Journal on Applied Mathematics **40(2)**, pp. 339–357.

Bolón-Canedo, V., Sánchez-Maroño, N., & Alonso-Betanzos, A., 2012. An ensemble of filters and classifiers for microarray data classification. Pattern Recognition **45(1)**, pp. 531–539.

Bolón-Canedo, V., Sánchez-Marono, N., & Alonso-Betanzos, A., 2014. Data classification using an ensemble of filters. Neurocomputing **135**, pp. 13–20.

Bolón-Canedo, V., Sánchez-Maroño, N., & Alonso-Betanzos, A., 2016. Feature selection for high-dimensional data. Progress in Artificial Intelligence **5(2)**, pp. 65–75.

Breiman, L., 1996. Bagging predictors. Machine learning **24(2)**, pp. 123–140.

Candès, E. J., Wakin, M. B., & Boyd, S. P., 2008. Enhancing sparsity by reweighted l1 minimization. Journal of Fourier Analysis and Applications **14(5)**, pp. 877–905.

Charkhabi, M., Dhot, T., & A. Mojarad, S., 2014. Cluster ensembles, majority vote, voter eligibility and privileged voters **4**, pp. 275–278.

Cruz, R. M., Hafemann, L. G., Sabourin, R., & Cavalcanti, G. D., 2018. Deslib: A dynamic ensemble selection library in python. arXiv preprint arXiv:1802.04967 .

Das, A. K., Das, S., & Ghosh, A., 2017. Ensemble feature selection using bi-objective genetic algorithm. Knowledge-Based Systems **123**, pp. 116–127.

Du, S. & Zhang, L., 2019. A mixed integer programming approach to the tensor complementarity problem. Journal of Global Optimization **73(4)**, pp. 789–800.

Dudoit, S. & Fridlyand, J., 2003. Bagging to improve the accuracy of a clustering procedure. Bioinformatics **19(9)**, pp. 1090–1099.

Dunn, J. C., 1973. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters .

Eiras-Franco, C., Bolón-Canedo, V., Ramos, S., González-Domínguez, J., Alonso-Betanzos, A., & Tourino, J., 2016. Multithreaded and spark parallelization of feature selection filters. Journal of Computational Science **17**, pp. 609–619.

Fern, X. Z. & Lin, W., 2008. Cluster ensemble selection. Stat. Anal. Data Min. **1(3)**, pp. 128–141.

Ferrari, D. & De Castro, L., 2015. Clustering algorithm selection by meta-learning systems: A new distance-based problem characterization and ranking combination methods **301**, pp. 181–194.

Fleuret, F., 2004. Fast binary feature selection with conditional mutual information. Journal of Machine Learning Research **5(Nov)**, pp. 1531–1555.

Forman, G., 2003. An extensive empirical study of feature selection metrics for text classification. Journal of machine learning research **3(Mar)**, pp. 1289–1305.

Fred, A. L. & Jain, A. K., 2005. Combining multiple clusterings using evidence accumulation. IEEE transactions on pattern analysis and machine intelligence **27(6)**, pp. 835–850.

Freund, Y. & Schapire, R. E., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences **55(1)**, pp. 119–139.

Ghaemi, R., Sulaiman, N. b., Ibrahim, H., & Mustapha, N., 2011. A review: accuracy optimization in clustering ensembles using genetic algorithms. Artificial Intelligence Review **35(4)**, pp. 287–318.

Gini, C., 1912. Variability and mutability, contribution to the study of statistical distribution and relaitons. Studi Economico-Giuricici della R .

Guan, D., Yuan, W., Lee, Y.-K., Najeebullah, K., & Rasel, M. K., 2014. A review of ensemble learning based feature selection. IETE Technical Review **31(3)**, pp. 190–198.

Hadjitodorov, S. T., Kuncheva, L. I., & Todorova, L. P., 2006. Moderate diversity for better cluster ensembles. Information Fusion **7(3)**, pp. 264–275.

Handl, J. & Knowles, J., 2007. An evolutionary approach to multiobjective clustering. IEEE transactions on Evolutionary Computation **11(1)**, pp. 56–76.

He, X. & Niyogi, P., 2003. Locality preserving projection, neural information processing symposium, vancouver. British Columbia, Canada .

Hong, Y., Kwong, S., Chang, Y., & Ren, Q., 2008a. Consensus unsupervised feature ranking from multiple views. Pattern Recognition Letters **29(5)**, pp. 595–602.

Hong, Y., Kwong, S., Chang, Y., & Ren, Q., 2008b. Unsupervised feature selection using clustering ensembles and population based incremental learning algorithm. Pattern Recognition **41(9)**, pp. 2742–2756.

Huang, D., Lai, J.-H., & Wang, C.-D., 2015. Combining multiple clusterings via crowd agreement estimation and multi-granularity link analysis. Neurocomput. **170(C)**, pp. 240–250.

Huang, D., Wang, C., & Lai, J., 2018. Locally weighted ensemble clustering. IEEE Transactions on Cybernetics **48(5)**, pp. 1460–1473.

Huang, D., Wang, C., Wu, J., Lai, J., & Kwoh, C., 2019. Ultra-scalable spectral clustering and ensemble clustering. CoRR **abs/1903.01057**.

Hubert, L. & Arabie, P., 1985. Comparing partitions. Journal of classification **2(1)**, pp. 193–218.

Inza, I., Larrañaga, P., Blanco, R., & Cerrolaza, A. J., 2004. Filter versus wrapper gene selection approaches in dna microarray domains. Artificial intelligence in medicine **31(2)**, pp. 91–103.

Jain, A. K., Murty, M. N., & Flynn, P. J., 1999. Data clustering: A review. ACM Comput. Surv. **31(3)**, pp. 264–323.

Jia, J., Xiao, X., Liu, B., & Jiao, L., 2011. Bagging-based spectral clustering ensemble selection. Pattern Recogn. Lett. **32(10)**, pp. 1456–1467.

Jing, L., Tian, K., & Huang, J. Z., 2015. Stratified feature sampling method for ensemble clustering of high dimensional data. Pattern Recognition **48(11)**, pp. 3688–3702.

Joydeep, G. & Ayan, A., 2011. Cluster ensembles. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **1(4)**, pp. 305–315.

Ko, A. H., Sabourin, R., & Britto Jr, A. S., 2008. From dynamic classifier selection to dynamic ensemble selection. Pattern recognition **41(5)**, pp. 1718–1731.

Kuncheva, L. I., 2002. A theoretical study on six classifier fusion strategies. IEEE Transactions on pattern analysis and machine intelligence **24(2)**, pp. 281–286.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H., 2018. Feature selection: A data perspective. ACM Computing Surveys (CSUR) **50(6)**, p. 94.

Li, X. & Liu, H., 2018. Greedy optimization for k-means-based consensus clustering. Tsinghua Science and Technology **23(2)**, pp. 184–194.

Liu, J., Chi, Y., He, S., & Liu, Z., 2019. An ensemble multi-objective evolutionary algorithm for gene regulatory network reconstruction based on fuzzy cognitive maps. CAAI Transactions on Intelligence Technology **4**.

Liu, X., Wang, L., Zhang, J., Yin, J., & Liu, H., 2014. Global and local structure preservation for feature selection. IEEE Transactions on Neural Networks and Learning Systems **25(6)**, pp. 1083–1095.

Lourenço, A., Bulò, S. R., Rebagliati, N., Fred, A. L., Figueiredo, M. A., & Pelillo, M., 2015. Probabilistic consensus clustering using evidence accumulation. Machine Learning **98(1-2)**, pp. 331–357.

Marill, T. & Green, D., 1963. On the effectiveness of receptors in recognition systems. IEEE transactions on Information Theory **9(1)**, pp. 11–17.

Mimaroglu, S. & Erdil, E., 2013. An efficient and scalable family of algorithms for combining clusterings. Engineering Applications of Artificial Intelligence **26(10)**, pp. 2525–2539.

Mitchell, L., Sloan, T. M., Mewissen, M., Ghazal, P., Forster, T., Piotrowski, M., & Trew, A., 2014. Parallel classification and feature selection in microarray data using sprint. Concurrency and computation: practice and experience **26(4)**, pp. 854–865.

Naldi, M. C., Carvalho, A. C. P. L. F., & Campello, R. J. G. B., 2013. Cluster ensemble selection based on relative validity indexes. Data Mining and Knowledge Discovery **27(2)**, pp. 259–289.

Nazari, A., Dehghan, A., Nejatian, S., Rezaie, V., & Parvin, H., 2019. A comprehensive study of clustering ensemble weighting based on cluster quality and diversity. Pattern Anal. Appl. **22(1)**, pp. 133–145.

Özöğür-Akyüz, S., Windeatt, T., & Smith, R., 2015. Pruning of error correcting output codes by optimization of accuracy–diversity trade off. Machine Learning **101(1-3)**, pp. 253–269.

Parvin, H. & Minaei-Bidgoli, B., 2013. A clustering ensemble framework based on elite selection of weighted clusters. Adv. Data Anal. Classif. **7(2)**, pp. 181–208.

Parvin, H. & Minaei-Bidgoli, B., 2015. A clustering ensemble framework based on selection of fuzzy weighted clusters in a locally adaptive clustering algorithm. Pattern Anal. Appl. **18(1)**, pp. 87–112.

Peng, H., Long, F., & Ding, C., 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on pattern analysis and machine intelligence **27(8)**, pp. 1226–1238.

Quinlan, J. R., 1986. Induction of decision trees. Machine learning **1(1)**, pp. 81–106.

Rashidi, F., Nejatian, S., Parvin, H., & Rezaie, V., 2019. Diversity based cluster weighting in cluster ensemble: An information theory approach. Artif. Intell. Rev. **52(2)**, pp. 1341–1368.

Robnik-Šikonja, M. & Kononenko, I., 2003. Theoretical and empirical analysis of relieff and rrelieff. Machine learning **53(1-2)**, pp. 23–69.

Ross Quinlan, J., 1993. C4. 5: programs for machine learning. Mach. Learn **16(3)**, pp. 235–240.

Sarumanthi, S., Shanthi, N., & Sharmila, M., 2013. A review: Comparative analysis of different categorical data clustering ensemble methods. International Journal of Computer and Information Engineering **7**, pp. 1622–1632.

Seijo-Pardo, B., Bolón-Canedo, V., & Alonso-Betanzos, A., 2017a. Testing different ensemble configurations for feature selection. Neural Processing Letters **46(3)**, pp. 857–880.

Seijo-Pardo, B., Bolón-Canedo, V., & Alonso-Betanzos, A., 2019. On developing an automatic threshold applied to feature selection ensembles. Information Fusion **45**, pp. 227–245.

Seijo-Pardo, B., Porto-Díaz, I., Bolón-Canedo, V., & Alonso-Betanzos, A., 2017b. Ensemble feature selection: homogeneous and heterogeneous approaches. Knowledge-Based Systems **118**, pp. 124–139.

Sriperumbudur, B. K., Torres, D. A., & Lanckriet, G. R., 2011. A majorization-minimization approach to the sparse generalized eigenvalue problem. Machine Learning **85(1-2)**, pp. 3–39.

Strehl, A. & Ghosh, J., 2002. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. Journal of machine learning research **3(Dec)**, pp. 583–617.

Strehl, A. & Ghosh, J., 2003. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. **3**, pp. 583–617.

Tang, L., Lin, Z., & Li, Y.-m., 2006. Effects of different magnitudes of mechanical strain on osteoblasts in vitro. Biochemical and biophysical research communications **344(1)**, pp. 122–128.

Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological) , pp. 267–288.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., & Knight, K., 2005. Sparsity and smoothness via the fused lasso. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **67(1)**, pp. 91–108.

Treviño, L. K., Weaver, G. R., & Reynolds, S. J., 2006. Behavioral ethics in organizations: A review. Journal of management **32(6)**, pp. 951–990.

Tsymbal, A., Puuronen, S., & Patterson, D. W., 2003. Ensemble feature selection with the simple bayesian classification. Information fusion **4(2)**, pp. 87–100.

Vega-Pons, S. & Avesani, P., 2015. On pruning the search space for clustering ensemble problems. Neurocomputing **150(Part B)**, pp. 481–489.

Vega-Pons, S., Correa-Morris, J., & Ruiz-Shulcloper, J., 2010. Weighted partition consensus via kernels. Pattern Recogn. **43(8)**, pp. 2712–2724.

Vega-Pons, S. & Ruiz-Shulcloper, J., 2011. A survey of clustering ensemble algorithms. International Journal of Pattern Recognition and Artificial Intelligence **25**, pp. 337–372.

Wang, H. & Liu, G., 2018. Two-level-oriented selective clustering ensemble based on hybrid multi-modal metrics. IEEE Access **PP**, pp. 1–1.

Weston, J., Elisseeff, A., Schölkopf, B., & Tipping, M., 2003. Use of the zero norm with linear models and kernel methods. J. Mach. Learn. Res. **3**, pp. 1439–1461.

Whitney, A. W., 1971. A direct method of nonparametric measurement selection. IEEE Transactions on Computers **100(9)**, pp. 1100–1103.

Windeatt, T., Duangsoithong, R., & Smith, R., 2011. Embedded feature ranking for ensemble mlp classifiers. IEEE transactions on neural networks **22(6)**, pp. 988–994.

Wright, S., 1965. The interpretation of population structure by f-statistics with special regard to systems of mating. Evolution **19(3)**, pp. 395–420.

Yang, F., Li, T., Zhou, Q., & Xiao, H., 2017. Cluster ensemble selection with constraints. Neurocomputing **235**, pp. 59 – 70.

Yang, F., Li, X., Li, Q., & Li, T., 2014. Exploring the diversity in cluster ensemble generation: Random sampling and random projection. Expert Systems with Applications **41(10)**, pp. 4844 – 4866.

Yang, F. & Mao, K., 2010. Robust feature selection for microarray data based on multicriterion fusion. IEEE/ACM Transactions on Computational Biology and Bioinformatics **8(4)**, pp. 1080–1092.

Yang, Y. & Jiang, J., 2019. Adaptive bi-weighting toward automatic initialization and model selection for hmm-based hybrid meta-clustering ensembles. IEEE Transactions on Cybernetics **49(5)**, pp. 1657–1668.

Yu, H., Wang, J., Bai, Y., Yang, W., & Xia, G.-S., 2018. Analysis of large-scale uav images using a multi- scale hierarchical representation. Geo-spatial Information Science **21(1)**, pp. 33–44.

Yu, Z., Li, L., Gao, Y., You, J., Liu, J., Wong, H.-S., & Han, G., 2014. Hybrid clustering solution selection strategy. Pattern Recognition **47(10)**, pp. 3362 – 3375.

Zhang, Y., Burer, S., & Street, W. N., 2006a. Ensemble pruning via semi-definite programming. Journal of Machine Learning Research **7(Jul)**, pp. 1315–1338.

Zhang, Y., Burer, S., & Street, W. N., 2006b. Ensemble pruning via semi-definite programming. J. Mach. Learn. Res. **7**, pp. 1315–1338.

Zhao, Y., Wang, X., Cheng, C., & Ding, X., 2019. Combining machine learning models using combo library. arXiv preprint arXiv:1910.07988 .

Zhong, C., Yue, X., Zhang, Z., & Lei, J., 2015. A clustering ensemble. Pattern Recogn. **48(8)**, pp. 2699–2709.

Other References

Ali, M. & Gubran, H., 2002. Traditional optimization methods in engineering applications.

Amiri Maskouni, M., Hosseini, S., Mohammadzadeh Abachi, H., Kangavari, M., & Zhou, X., 2018. Auto-CES: An Automatic Pruning Method Through Clustering Ensemble Selection. pp. 275–287.

Azimi, J. & Fern, X., 2009. Adaptive cluster ensemble selection. In Proceedings of the 21st International Jont Conference on Artifical Intelligence, IJCAI'09. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 992–997.

Banerjee, A., Pati, B., & Rani Panigrahi, C., 2018. SC^2 : A Selection-Based Consensus Clustering Approach. pp. 177–183.

Belkin, M. & Niyogi, P., 2002. Laplacian eigenmaps and spectral techniques for embedding and clustering. In Advances in neural information processing systems. pp. 585–591.

Blondin, J., 2009. Particle swarm optimization: A tutorial.

Bradley, P. S. & Mangasarian, O. L., 1998. Feature selection via concave minimization and support vector machines. In Proceedings of the Fifteenth International Conference on Machine Learning, ICML '98. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 82–90.

Brahim, A. B. & Limam, M., 2013. Robust ensemble feature selection for high dimensional data sets. In 2013 International Conference on High Performance Computing & Simulation (HPCS). IEEE, pp. 151–157.

Cai, D., Zhang, C., & He, X., 2010. Unsupervised feature selection for multi-cluster data. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 333–342.

Caruana, R. & Niculescu-Mizil, A., 2006. An empirical comparison of supervised learning algorithms. In Proceedings of the 23rd international conference on Machine learning. pp. 161–168.

Caruana, R., Niculescu-Mizil, A., Crew, G., & Ksikes, A., 2004. Ensemble selection from libraries of models. In Proceedings of the Twenty-first International Conference on Machine Learning, ICML '04. New York, NY, USA: ACM, pp. 18–.

Cunningham, P. & Carney, J., 2000. Diversity versus quality in classification ensembles based on feature selection. In European Conference on Machine Learning. Springer, pp. 109–116.

Dheeru, D. & Karra Taniskidou, E., 2017. Uci machine learning repository.

Dittman, D. J., Khoshgoftaar, T. M., Wald, R., & Napolitano, A., 2012. Comparing two new gene selection ensemble approaches with the commonly-used approach. In Machine Learning and Applications (ICMLA), 2012 11th International Conference on, vol. 2. IEEE, pp. 184–191.

Du, L. & Shen, Y.-D., 2015. Unsupervised feature selection with adaptive structure learning. In Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp. 209–218.

Fazel, M., Hindi, H., & Boyd, S., 2003. Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices. In Proceedings of the American Control Conference, vol. 3. pp. 2156 – 2162 vol.3.

Fern, X. Z. & Brodley, C. E., 2003. Random projection for high dimensional data clustering: A cluster ensemble approach. In Proceedings of the Twentieth International Conference on International Conference on Machine Learning, ICML'03. AAAI Press, pp. 186–193.

Fern, X. Z. & Brodley, C. E., 2004. Solving cluster ensemble problems by bipartite graph partitioning. In Proceedings of the twenty-first international conference on Machine learning. p. 36.

Fred, A., 2001. Finding consistent clusters in data partitions. In International Workshop on Multiple Classifier Systems. Springer, pp. 309–318.

Grant, M., Boyd, S., & Ye, Y., 2006. Disciplined convex programming. In Global optimization. Springer, pp. 155–210.

Hall, M. A. & Smith, L. A., 1999. Feature selection for machine learning: comparing a correlation-based filter approach to the wrapper. In FLAIRS conference, vol. 1999. pp. 235–239.

Han, Y., Park, K., & Lee, Y.-K., 2011. Confident wrapper-type semi-supervised feature selection using an ensemble classifier. In 2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC). IEEE, pp. 4581–4586.

He, X., Cai, D., & Niyogi, P., 2006. Laplacian score for feature selection. In Advances in neural information processing systems. pp. 507–514.

Hein, M., Setzer, S., Jost, L., & Rangapuram, S. S., 2013. The total variation on hypergraphs - learning on hypergraphs revisited. In Burges, C. J. C., Bottou, L., Ghahramani, Z., & Weinberger, K. Q., eds., NIPS. pp. 2427–2435.

Jakulin, A., 2005. Machine learning based on attribute interactions: phd dissertation. Ph.D. thesis, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko.

Jia, J., Xiao, X., & Liu, B., 2012. Similarity-based spectral clustering ensemble selection. In 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery. pp. 1071–1074.

Kuncheva, L. I. & Hadjitodorov, S. T., 2004. Using diversity in cluster ensembles. In 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583), vol. 2. pp. 1214–1219 vol.2.

Kuncheva, L. I., Hadjitodorov, S. T., & Todorova, L. P., 2006. Experimental comparison of cluster ensemble methods. In 2006 9th International Conference on Information Fusion. pp. 1–7.

Lewis, D. D., 1992. Feature selection and feature extraction for text categorization. In Proceedings of the workshop on Speech and Natural Language. Association for Computational Linguistics, pp. 212–217.

Li, T. & Ding, C., 2008. Weighted Consensus Clustering. pp. 798–809.

Li, Z., Yang, Y., Liu, J., Zhou, X., Lu, H. et al., 2012. Unsupervised feature selection using nonnegative spectral analysis. In AAAI, vol. 2. pp. 1026–1032.

Liu, H., Shao, M., & Fu, Y., 2016. Consensus guided unsupervised feature selection. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI'16. AAAI Press, pp. 1874–1880.

Margineantu, D. D. & Dietterich, T. G., 1997. Pruning adaptive boosting. In ICML, vol. 97. Citeseer, pp. 211–218.

Meyer, P. E. & Bontempi, G., 2006. On the use of variable complementarity for feature selection in cancer classification. In Workshops on applications of evolutionary computation. Springer, pp. 91–102.

Morita, M., Oliveira, L. S., & Sabourin, R., 2004. Unsupervised feature selection for ensemble of classifiers. In Ninth International Workshop on Frontiers in Handwriting Recognition. IEEE, pp. 81–86.

Nie, F., Xiang, S., Jia, Y., Zhang, C., & Yan, S., 2008. Trace ratio criterion for feature selection. In AAAI, vol. 2. pp. 671–676.

Otar, B. Ç. & Akyüz, S., 2017. Ensemble clustering selection by optimization of accuracy-diversity trade off. In Signal Processing and Communications Applications Conference (SIU), 2017 25th. IEEE, pp. 1–4.

Saeys, Y., Abeel, T., & Van de Peer, Y., 2008. Robust feature selection using ensemble feature selection techniques. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, pp. 313–325.

Shen, X., Diamond, S., Gu, Y., & Boyd, S., 2016. Disciplined convex-concave programming. In Decision and Control (CDC), 2016 IEEE 55th Conference on. IEEE, pp. 1009–1014.

Singh, V., Mukherjee, L., Peng, J., & Xu, J., 2007. Ensemble clustering using semidefinite programming. In Proceedings of the 20th International Conference on Neural Information Processing Systems, NIPS'07. USA: Curran Associates Inc., pp. 1353–1360.

Soares, R. G., Santana, A., Canuto, A. M., & de Souto, M. C. P., 2006. Using accuracy and diversity to select classifiers to build ensembles. In The 2006 IEEE International Joint Conference on Neural Network Proceedings. IEEE, pp. 1310–1316.

Topchy, A., Jain, A. K., & Punch, W., 2004a. A mixture model for clustering ensembles. In Proceedings of the 2004 SIAM international conference on data mining. SIAM, pp. 379–390.

Topchy, A., Minaei-Bidgoli, B., Jain, A. K., & Punch, W. F., 2004b. Adaptive clustering ensembles. In Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., vol. 1. pp. 272–275 Vol.1.

Üçüncü, D., Akyüz, S., Gül, E., & Wilhelm-Weber, G., 2018. Optimality conditions for sparse quadratic optimization problem. In International Conference on Engineering Optimization. Springer, pp. 766–777.

Vega-Pons, S., Correa-Morris, J., & Ruiz-Shulcloper, J., 2008. Weighted cluster ensemble using a kernel consensus function. In Ruiz-Shulcloper, J. & Kropatsch, W. G., eds., Progress in Pattern Recognition, Image Analysis and Applications. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 195–202.

Vidal-Naquet, M. & Ullman, S., 2003. Object recognition with informative features and linear classification. In ICCV, vol. 3. p. 281.

Wang, H., Khoshgoftaar, T. M., & Napolitano, A., 2010. A comparative study of ensemble feature selection techniques for software defect prediction. In 2010 Ninth International Conference on Machine Learning and Applications. IEEE, pp. 135–140.

Yang, H. H. & Moody, J., 2000. Data visualization and feature selection: New algorithms for nongaussian data. In Advances in Neural Information Processing Systems. pp. 687–693.

Yu, L. & Liu, H., 2003. Feature selection for high-dimensional data: A fast correlation-based filter solution. In Proceedings of the 20th international conference on machine learning (ICML-03). pp. 856–863.

Zhao, Z. & Liu, H., 2007. Spectral feature selection for supervised and unsupervised learning. In Proceedings of the 24th international conference on Machine learning. ACM, pp. 1151–1157.