The Dissertation Committee for Donna Kay Korycinski Certifies that this is the approved version of the following dissertation:

INVESTIGATING THE USE OF TABU SEARCH TO FIND NEAR-OPTIMAL SOLUTIONS IN MULTICLASSIFIER SYSTEMS

Committee:

Melba M. Crawford, Co-Supervisor

J. Wesley Barnes, Co-Supervisor

Joydeep Ghosh

Elmira Popova

John J. Hasenbein

Investigating the Use of Tabu Search to Find Near-Optimal Solutions in Multiclassifier Systems

by

Donna Kay Korycinski, B.S., M.S.E.

Dissertation

Presented to the Faculty of the Graduate School of

the University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

August, 2003

Dedication

To Richard, Sydney and Jordan

Acknowledgements

I would like to thank my advisors, Dr. Melba M. Crawford and Dr. J. Wesley Barnes, whose insightful guidance and unwavering support made this dissertation possible. I would also like to thank all of my friends at the Center for Space Research, especially Amy Neuenschwander, for their help and support during the completion of this work. I am grateful to my parents, Joseph and Phyllis Cecil, for an encouraging word and a helping hand when I needed it most. I wish to extend a sincere heartfelt thank you to my loving husband, Richard, who closed his own window-of-opportunity in life so that mine could be opened; I shall never forget your sacrifice. And to my two daughters, Sydney and Jordan, thank you for understanding when you always heard the words, "Mommy has to study."

Investigating the Use of Tabu Search to Find Near-Optimal Solutions in Multiclassifier Systems

Publication No._____________

Donna Kay Korycinski, Ph.D.

The University of Texas at Austin, 2003

Supervisors: Melba M. Crawford, J. Wesley Barnes

Binary trees provide an ideal framework for many decision problems due to their logical, understandable structures and the computational advantages of the "divide and conquer" paradigm. They can be particularly advantageous for classification applications, which involve categorization of information into groups that are in some sense homogeneous. Algorithms used in construction of decision trees used in classification problems are typically greedy. A new algorithm was developed in this study which incorporates Tabu Search (TS) in the feature selection aspect of hierarchical classification trees. Specifically, it is implemented within the hierarchical classification problem framework of the Binary Hierarchical Classifier (BHC) which has been shown to be advantageous for classification problems with a large number of output classes. The algorithm incorporates feature selection as a means for input space and classifier complexity

reduction for a static tree; the algorithm was also extended and coupled with the BHC to allow TS feature selection to aid in building the class hierarchy. Finally, a new algorithm was developed which uses TS in the rearrangement of the nodes of a binary classification tree. Since the use of highly accurate classification algorithms is vital in fields such as medical diagnoses, character recognition, target detection, and land cover mapping, the primary goal of this research is to attain improved classification accuracies.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Data acquisition is often an expensive undertaking; therefore, many organizations acquire all data possible because it is never known when the data may become useful. Advances in technology have made data storage relatively inexpensive, thereby resulting in enormous increases in the quantity of data being acquired and stored. Unfortunately, the acquisition and storage rates far exceed the current capabilities to process and extract useful information from this data. Thus, large amounts of stored data exist that may never be examined. When data exist in large quantities, it is imperative that computer technologies be used to examine the data and to extract useful information. Even using modern computing capabilities, this task can be extremely difficult. *Classification* involves categorization of information into groups that are in some sense homogeneous. Classification thus achieves both information extraction and compression, and its methods are widely used to perform such diverse actions as labeling and tracking of land cover, making medical diagnoses, target detection, and assessing credit-risks and detecting fraud. The field of statistical classification has been an active area of research for over forty years. Supervised classification is performed in the following manner: out of a set of C known classes, data observations are examined and assigned, or recognized as belonging to one of the known classes. This is accomplished by examining the pattern of the features belonging to each observation and assigning labels to individual observations based on this pattern. Features are also known as attributes or properties, and each observation may possess an associated vector of feature values. In a perfect world, this vector of values would completely determine the correct classification for each observation, but this is a rarity. Typically, observations from classes are random variables with associated probability distributions; often there is substantial overlap between distributions of different classes.

The number of features (attributes) that describe each observation can range from only a few to thousands. This is problematic both because of computational complexity and because the resulting high dimensional input observation space is typically quite sparse. Further, when features are redundant, covariance based classification methods encounter numerical problems. The two general approaches to input space dimensionality reduction involve feature extraction or feature subset selection. The goal of techniques developed under either strategy is to construct a simpler classification algorithm that is more reliable, i.e., possesses greater accuracy and executes faster. Feature extraction is the process of extracting features from the original set to form a lower-dimensional set of potentially different features. This is accomplished through some type of mapping or transformation. For example, principal component analysis is commonly used to project the original feature space onto a lower dimensional feature space. Feature selection reduces the feature space by choosing a subset of the original features to represent the entire set. The goal of feature selection is to find the optimal feature subset such that when the classification algorithm is applied to observations, the resulting labels have the highest accuracy. This selection of the optimal subset out of all possible subsets is an NP-hard problem [1]. Performing an exhaustive search of the solution space of all possible subsets would be required to ensure that the optimal feature subset had been identified. For a very large number of features, exhaustive search is intractable. For a problem with *n* features, the number of all possible feature subsets is 2^n. A few calculations show how the number of possible subsets becomes unmanageable: $2^4 = 16$, $2^{12} = 4{,}096$, $2^{30} = 1{,}073{,}741{,}824$, and $2^{250} = 1.809\text{x}10^{75}$. Current feature selection techniques, which include greedy algorithms and the use of heuristics, do not guarantee optimality but often obtain near-optimal solutions more quickly than an exhaustive search.

The classification algorithm, or classifier, is that function which examines the observations and maps them into the set of *C* known classes. Research has shown that it is very rare when a single classifier can be considered as the best classifier for all of the classes when multiple classes are involved. This realization led to an area of research known as *multiclassifier systems* whereby results from multiple classifiers are combined in such a way as to improve the accuracy of classification relative to that of the single classifier.

This research involves investigation of Tabu Search (TS), a well-known *meta*heuristic that is able to adaptively and reactively guide its own search through the solution space, coupled with the multiclassifier system known as the Binary Hierarchical Classifier (BHC) [2, 3]. The current BHC algorithm utilizes a deterministic annealing-type algorithm which employs Fisher projection based feature extraction to partition the classes and produces a binary hierarchical tree structure that is used to classify all unknown observations. The primary goal in development of this approach was output decomposition for problems with a medium to large number of classes. While the classification accuracies obtained from the BHC are typically good, problems are encountered if the number of inputs is extremely large and the amount of training data is limited. Further, the Fisher weights are not typically stable, and the tree is not necessarily robust to problems where the inputs are perturbed, as would be the case if the classifier were applied to a slightly different problem. A preliminary investigation of a simple greedy based feature selection approach [4] was promising, but inflexible. In this study, new models are developed which incorporate the use of TS in the feature selection aspect of the hierarchical classification trees within the hierarchical problem framework of the BHC. Improved classification accuracies are increasingly more important as the use of classification algorithms becomes more prevalent.

In addition, the combined use of TS for feature selection coupled with tree rearrangement is investigated as a means for input space and classifier complexity reduction. The goals of this research are to extend knowledge and understanding in the areas of classification and to introduce metaheuristics within the hierarchical classification framework. This methodology is applied in the analysis of several datasets, including a standard character set and remotely sensed data acquired by multispectral and hyperspectral sensors, which acquire data simultaneously in hundreds of spectral bands.

Chapter 2

Background and Related Work

This section contains an overview of the characteristics of supervised classification problems and solution approaches, with a focus on the problem of selecting optimal inputs for large data mining problems. It contains a more in-depth discussion of the BHC algorithm and TS as a method for attacking combinatorial problems.

2.1 CLASSIFICATION AND FEATURES

Supervised classification methods derive a set of rules for labeling a (typically) vector-valued observation of *features* as members of one of C known classes. Features can have discrete, continuous or complex values. Discrete features can possess only a finite number of values; ordinal and nominal scale values are of the discrete type. Continuous features possess an infinite number of values within the domain of real numbers. Complex features possess an infinite number of values within the domain of complex numbers, i.e., $x+iy$. Figure 2.1 shows the hierarchy of these feature types.

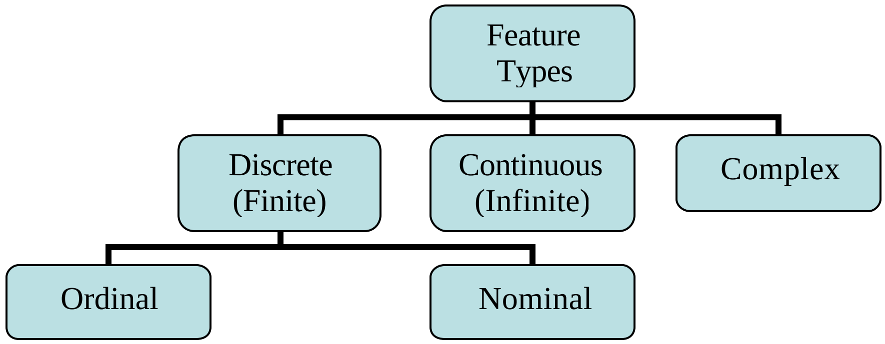

Figure 2.1: Hierarchy of feature types [5].

The vector of feature values that describes each observation forms a pattern that, when examined and compared to known patterns of classes that exist within the dataset, can be used to assign labels to unclassified observations. When there are no clear distinctions between the patterns of different classes, some observations may be misclassified; thus, classification algorithms seek to minimize the expected error rate of classification or maximize some measure of goodness for classification.

2.2 Measures of Goodness

Measures of goodness seek to maximize the classifier's ability to discriminate between the known classes. There are five different types of measures that are commonly used within the area of classification: accuracy, information, distance, dependence and consistency. Accuracy measures directly depend on the classifier used and reflect the predictive accuracy of the classifier by either maximizing the accuracy rate or minimizing the error rate of classification. Accuracy measures are widely used by researchers as the primary measure for evaluation. The other types are measures of *class separability* which are maximized to yield the greatest potential for distinguishing between the classes. Class separability can be further characterized in terms of consistency and the classic measures of information, distance and dependence. Consistency measures reward consistent classification of an observation into the same class as the classifier is iteratively refined. An information measure monitors the likelihood of an observation being classified into its true class by the use of an uncertainty function such as Shannon's entropy, $-\sum_i P(c_i)\log_2 P(c_i)$ [6]. Distance measures attempt to separate the classes as much as possible and label an observation as belonging to its closest class. Typical distance measures include the Mahalanobis distance [7], the Battacharyya distance [7], the Jeffries-Matusita distance [8] and the Patrick-Fisher distance [9]. Finally, dependence measures quantify the association or correlations between features and the classes involved. Figure 2.2 shows the hierarchy of measures typically used in classification.

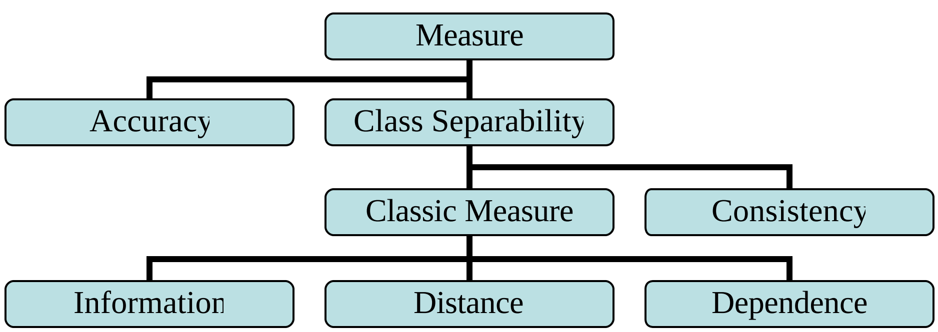

Figure 2.2: Hierarchy of types of measures [5].

2.3 THE CLASSIFIER

A supervised classification procedure takes each unclassified observation and maps it into the set of C known classes, assigning the observation a class label. This process consists of the following steps: (1) determine the set of classes that exists within the dataset, (2) select representative observations that are known to reside in each class (subsequently these will be divided into the set of training data and the set of testing data), (3) use the training data to estimate the parameters of the probability density functions of the individual classes, (4) train the classifier with the training data and evaluate the classifier with the testing data, iterating as necessary, (5) label all unclassified observations using the trained classifier and (6) summarize the results of the classification. This type of classification depends on the ability to model the classes, typically using parametric probability distributions. The classifier can be viewed as a conjecture of the true mapping from a data observation to the correct class. Given new unclassified observations, the classifier predicts the observation's class. Typical classifiers include Bayesian classifiers [10], maximum likelihood classifiers [11] and minimum distance classifiers [12]. During classification, problems can arise when the set of inputs includes features that are irrelevant (do not affect the structure of the data in any way), and/or redundant (do not add any new information to the description of the data structure). These issues are greatly exacerbated when the input space is quite large. This is problematic both because of computational complexity and the resulting high dimensional input observation space is typically quite sparse.

2.4 FEATURE SELECTION

Methods which overcome the problems of irrelevant or redundant features are "input space reduction techniques." The motivation for input space reduction is three-fold: (1) to improve the accuracy of the chosen classifier, (2) to reduce the data dimensionality, while simultaneously reducing the number of observations required to appropriately train the classifier (to estimate the class parameters), and (3) to simplify the classifier by reducing the search space that the classifier must traverse. A welcomed side-effect is the possible reduction of the effort required for the classifier to learn an accurate classification function [13, 14]. Feature extraction and feature subset selection are two general approaches to input space reduction. Feature extraction is the process of extracting a set of new features from the original set of features through a mapping or transformation, for example, the projection of the original feature space onto a lower dimensional feature space (as in principal component analysis). It has been shown that a classifier using irrelevant or redundant features does not perform as well as a classifier that excludes the irrelevant or redundant features [7]. Subset selection is an optimization problem which involves searching the solution space of all possible subsets for an optimal or near-optimal subset of features. Feature selection is usually directed at one of two goals: (1) minimize the number of features selected while satisfying some minimal level of classification capability or (2) maximize classification ability for a subset of prescribed cardinality. Additionally, feature selection potentially provides valuable domain knowledge about the process.

Feature selection can be visualized as a search in a discrete binary space (or Boolean hypercube) where each point depicts a feature subset whose vector of D components identifies the members of the feature subset. For example, a 1 in the vector's j^{th} position indicates the j^{th} feature's inclusion in the subset while a 0 in the j^{th} position indicates its exclusion. This space can be depicted in a lattice structure as depicted in Figure 2.3, where the top node includes all features and the bottom node is the empty set; all other nodes within the lattice are the result of a removal of a feature if the lattice is traversed top-down or the addition of a feature if the lattice is traversed bottom-up. For example, if $D = 4$, the binary vector (1, 0, 1, 0) depicts the feature subset which includes features one and three, i.e. {1, 3} and is highlighted in Figure 2.3.

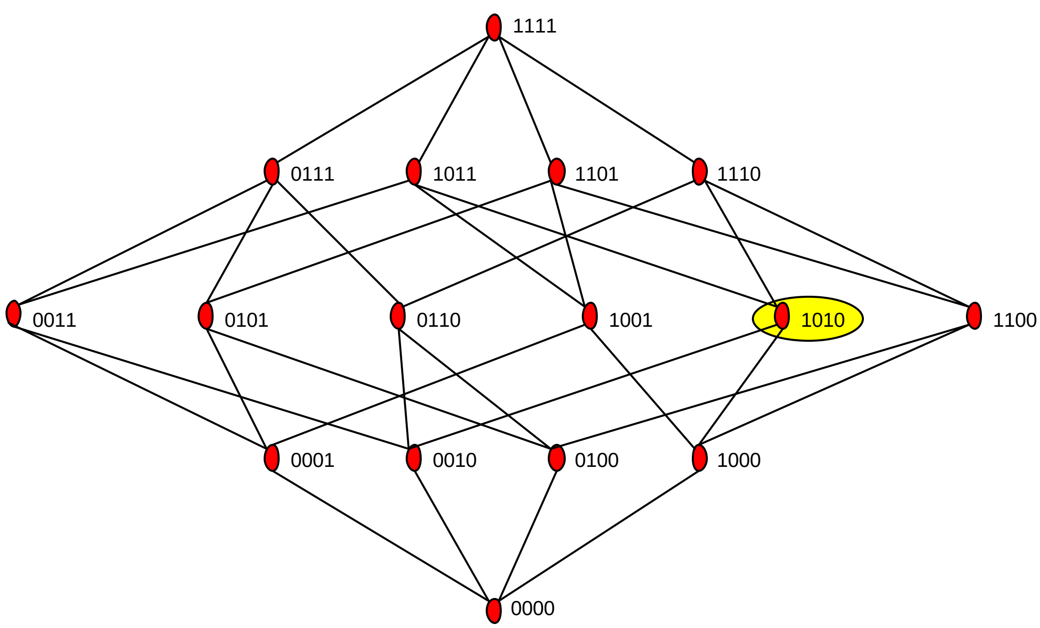

Figure 2.3: Four-dimensional feature selection lattice.

An example of a greedy heuristic search of the feature selection lattice used in classification is implemented as the Steepest Ascent Algorithm discussed in [15]. This algorithm uses the Jeffries-Matusita (JM) distance as its objective function (maximization) and assumes Gaussian class distributions to simplify computation. In this algorithm, an initial subset is selected and evaluated; all possible one-feature changes are considered; if an improvement can be made, the best improvement is accepted, and the algorithm then considers all one-feature changes from the current subset. These iterations terminate when no improvements can be made, indicating that the process has reached a local optimum, and return the best subset found. This type of algorithm is sensitive to the initial subset. This weakness can be lessened by executing the algorithm several times and comparing the resulting subsets.

Feature selection techniques are characterized either as *filters*, which ignore the classifier to be used, or *wrappers*, which base selection directly on the classifier.

2.4.1 Filters

Computationally more efficient than wrappers, a filter approach performs subset selection based only on the feature qualities within the training data. Since the classifier is ignored, there is no interaction between the biases of the feature selector and the classifier. The quality of the best filter subset is typically not as effective as a subset selected using a wrapper model. Two well-known filter approaches are embodied in the RELIEF and FOCUS algorithms described in [16]. Figure 2.4 depicts a flowchart of a filter model for feature selection.

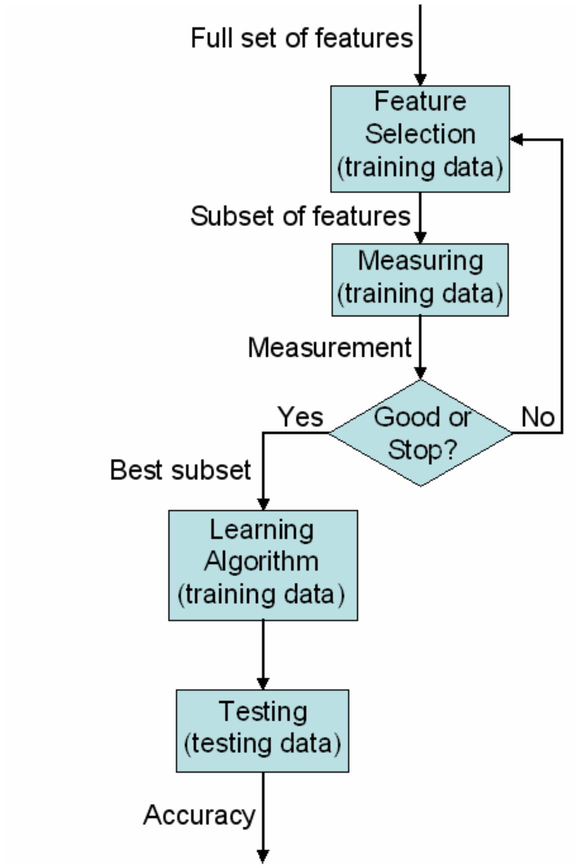

Figure 2.4: A filter model of feature selection [5].

2.4.2 Wrappers

Wrappers select a feature subset based directly on the classifier. The training data are used to train the classifier using different feature subsets; each is then evaluated using the testing data to find the best subset. In this way, the biases inherent in the feature selection algorithm and the classifier strongly interact, and the feature selection is described as being “wrapped around” the classification algorithm. The feature subset with the highest evaluation score is subsequently passed to the classifier to label the remaining unclassified observations. Selecting better subsets can improve the accuracy of a classifier [17], and this is one reason that wrapper models are often preferred over filter models. Unfortunately, depending on the computational intensity of the classifier used and the number of original features, wrapper models can be computationally burdensome and may be intractable for problems having a very large number of features. Another problem associated with wrappers is that they may actually overfit the data [17] by placing undue emphasis on random variations in training data which yields a model that does not generalize well for new data. Figure 2.5 depicts a flowchart of a wrapper model for feature selection.

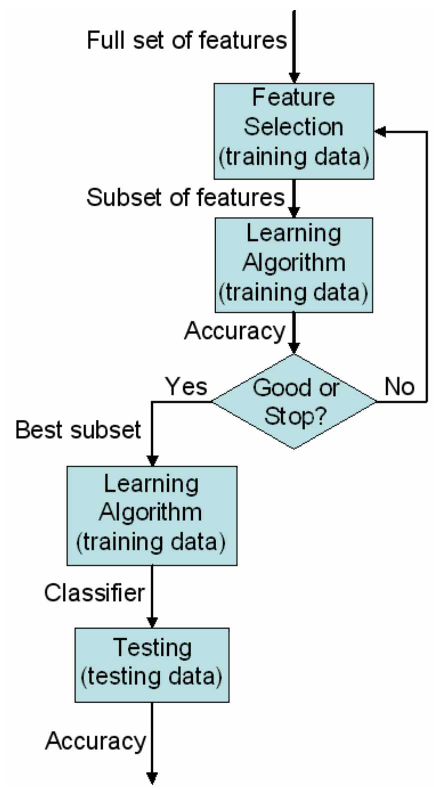

Figure 2.5: A wrapper model of feature selection [5].

2.4.3 Optimal methods

Optimal feature selection methods identify the optimal feature subset which yields the highest possible accuracy for a set of known data. Identification of the optimal subset is guaranteed by an exhaustive search of the solution space of all possible subsets [1]. For a very large number of features, exhaustive search is computationally intractable.

The branch and bound (B&B) method also produces optimal features [18]. A limitation of B&B is that it guarantees the optimal subset only if the performance measure is known to be monotonic, where the addition of features does not deteriorate the performance measure. This condition often cannot be satisfied. In addition, in many situations the effort associated with B&B may still be prohibitive. Other forms of the B&B, automatic B&B and backward automatic B&B [19], have been proposed, but still require the monotonicity property. Approximate B&B [20] is a heuristic B&B which does not require a monotonic performance measure but is computationally more demanding than B&B [5].

2.4.4 Sub-optimal methods

The computational complexities of optimal feature selection methods have resulted in the acceptance of heuristic techniques that find good, near-optimal subsets in relatively short computational times. A comparative study of several of the well-known optimal and sub-optimal feature selection algorithms is contained in [21]. Specifically, the authors contrasted results obtained from the following methods: Sequential Forward/Backward Selection (SFS/SBS), their generalized versions (GSFS(g)/GSBS(g)) and their floating point versions (SFFS/SBFS); Plus *l* take away *r* (PTA(*l*, *r*)) and its generalized version (GPTA(*l*, *r*)); versions of B&B and relaxed B&B; a genetic algorithm; and a parallel algorithm. Genetic algorithms were introduced for the selection of features in [22]. Simulated annealing was used as a feature selector in [20], and the use of the TS *meta*heuristic was shown as a promising approach in [23].

2.5 TABU SEARCH

Tabu Search (TS) is a *meta*heuristic method for solving combinatorial optimization problems. Its first modern formulation is attributed to Glover [24]. TS differs from other search techniques in that modern versions of TS are able to adaptively and reactively guide their search through the solution space while allowing infeasible areas of the solution space to be traversed in its search for the optimal solution. TS uses specialized memory structures to maintain its search history and to avoid becoming trapped in local optima. Its popularity has grown due to its ability to find near-optimal solutions in a short amount of time and its adaptability to many combinatorial optimization problems, including job shop scheduling problems [25], pickup and delivery problems [26], and communication network problems [27, 28]. Group Theoretic TS, a version of TS that makes extensive use of group theory has recently been developed [29, 30] and has been successfully implemented in the aerial fleet refueling problem [31, 32], the theater deployment vehicle routing and scheduling problem [33, 34] and the general crew scheduling problem [35]. Another version known as Extreme Point TS [36] has been proposed to optimize decision trees by representing the tree as a set of disjunctive linear inequalities and optimizing over these inequalities; the results are considered promising.

TS explores from its incumbent solution, looks at neighboring solutions, i.e., those solutions that can be reached by a single move within the specified *move-neighborhood*, and moves to the neighboring solution with the best non-tabu solution. It avoids cycling and escapes from local optima by using a *tabu list*, which incorporates solution attributes of recent solutions that are forbidden for *tabu tenure future* moves. An *aspiration criterion* may be introduced to allow TS to make a tabu move if stipulated conditions are satisfied. TS can include intensification and diversification elements: intensification allows a deeper search into promising areas of the solution space, and diversification encourages movement to yet unexplored or less explored areas of the solution space. Finally, the search will halt and return the best solution found when a *stopping criterion* is satisfied. As with all heuristic methods, the solution returned is not guaranteed to be optimal.

To facilitate the visualization of the TS principles, its application to one of the most researched problems in scientific literature, the combinatorial optimization problem known as the Traveling Salesman Problem (TSP), is discussed here. Simply stated, a single salesman is to travel to several cities, starting from and returning to his home city. Knowing the exact distance between each pair of cities, the salesman desires to plan his route minimizing his entire travel-distance. The simplicity of the problem description is deceiving, as all permutations of the cities must be implicitly examined to identify the tour of minimum length, and thus, the TSP is NP-complete [37]. When the number of cities is large, the TSP becomes intractable. To clarify the general TS approach, a general description of a TS implementation for the TSP is now described.

The TSP, as in all applications of TS, starts with an initial tour (solution). The move-neighborhood is all other solutions in the solution space that are reachable by a single move. While moves can be defined in a number of ways, two common types of moves are known as *swap-moves* and *insert-moves*. In the TSP, a swap-move identifies two cities within the tour and exchanges their positions. For example, suppose for a 6 city TSP that city 1 is the salesman's home. Given incumbent tour 1-2-4-3-6-5, swapping cities 2 and 6 yields 1-6-4-3-2-5. An insert-move removes a single city from its current position and inserts it in a different position. Given incumbent tour 1-2-4-3-6-5 with city 2 selected for insertion, the neighborhood tours are 1-4-2-3-6-5, 1-4-3-2-6-5, 1-4-3-6-2-5 and 1-4-3-6-5-2.

TS is aggressive and will generally choose the best non-tabu move available within its present move neighborhood (characterized by the greatest decrease or smallest increase in the tour length). It differs from simple classical descent methods in that it can escape being trapped in local optima by its ability to learn. An attribute of the new solution is identified and labeled as tabu (not repeatable for a given number of iterations known as tabu tenure). The tabu architecture eliminates cycling, repeatedly returning to and not being able to escape from a local optimum, and allows the search to move away from recently searched areas. Pure TS, a simple but cumbersome and ineffective tabu strategy memorizes all solutions (tours) visited thus far and forbids return to any such solution; an alternative to remembering entire tours is to select an attribute, such as the city just inserted, and then to forbid it from being moved within its tabu tenure. Tabu status can be overridden by a move meeting the aspiration criterion. The simplest such criterion is to allow the move when it leads to a new best solution for the TSP. Diversification may be introduced by counting the number of times the cities have been in particular tour positions and choosing to penalize moves that cause higher counts to be repeated. This drives the search into possibly new unexplored areas of the solution space. Intensification may be implemented by returning to good solutions and searching within the vicinities of such solutions in the hope of finding even better solutions. The stopping criterion is often a specified number of iterations performed or number of iterations performed with no improvement to the tour length. These strategies presented for the TSP are only representative and in no way exhaust the great number of the strategies that may be applied to the TSP and similar problems when using TS.

2.6 MULTIPLE CLASSIFIER SYSTEMS

In classification, it is very rare when a single classifier can be considered the *best* classifier when multiple classes are present [38]. This led to the development of a research area which focuses on developing methods that combine a group of classifiers in such a way as to improve the accuracy of classification relative to that of the single classifier and with greater classification accuracy than any of the individuals within the ensemble [39]. These types of classifiers are said to “divide and conquer” the solution space. Instead of learning one complex classifier, this family of classifiers combines many smaller, easier classifiers into a *multiclassifier* system. Many multiclassifier systems have been developed and continue to be refined; a brief history is presented in [40]. Examples of multiclassifier systems include the Bayesian Pairwise Classifier (BPC) [2], the Bayesian Pairwise Classifier with the Fisher Discriminant (BPC-FD) [2], and the Best-Bases Binary Hierarchical Classifier (B-B BHC) [41, 42]. Much research has been devoted to the exploitation of these multiclassifier-improvements to overall system accuracy in an effort to develop higher quality, more robust classifiers that can contribute to knowledge reuse and transferability of the classifier.

Chapter 3

The Binary Hierarchical Classifier

The focus of this research is multiclassifier problems with large input and output spaces. Large input spaces require input space reduction techniques, while large output spaces are often handled by various output space decomposition techniques. An example of a classifier framework that transforms the feature space and the output space simultaneously is the BHC. The BHC is studied as a means of developing models that attain better overall classification accuracies. In addition, the use of class dependent feature selection within the hierarchical tree schemes is investigated for its impact on retention of domain knowledge. The remainder of this chapter contains a discussion of the current implementation of the BHC and an introduction to the research conducted in this study.

3.1 BINARY HIERARCHICAL CLASSIFIER FOR CLASSIFICATION

The BHC, as developed by Kumar *et al.* [3] for a *C*-class problem, forms a binary tree-type hierarchical classifier (at each node of the tree, only two branches are created). Sets containing more than one class are known as metaclasses and are the *internal nodes* of the tree structure; sets containing individual classes are the *leaf nodes* of the tree which are the final nodes of the branches. The metaclass at the top of the tree structure includes all original classes. The internal nodes of the tree, to include the top node, depict a two-metaclass problem that partitions the classes at each internal node Ω_n into two *child* nodes, Ω_{2n} and Ω_{2n+1}, where

$\Omega_{2n} \cup \Omega_{2n+1} = \Omega_n$; this is accomplished recursively at each internal node until the leaves of the tree structure contain the individual classes and no more partitioning can be executed. Ultimately, this framework yields a hierarchical tree structure of C-1 internal nodes (two-metaclass problems) and C leaf nodes. Figure 3.1 depicts an example of a BHC tree structure with *5* classes.

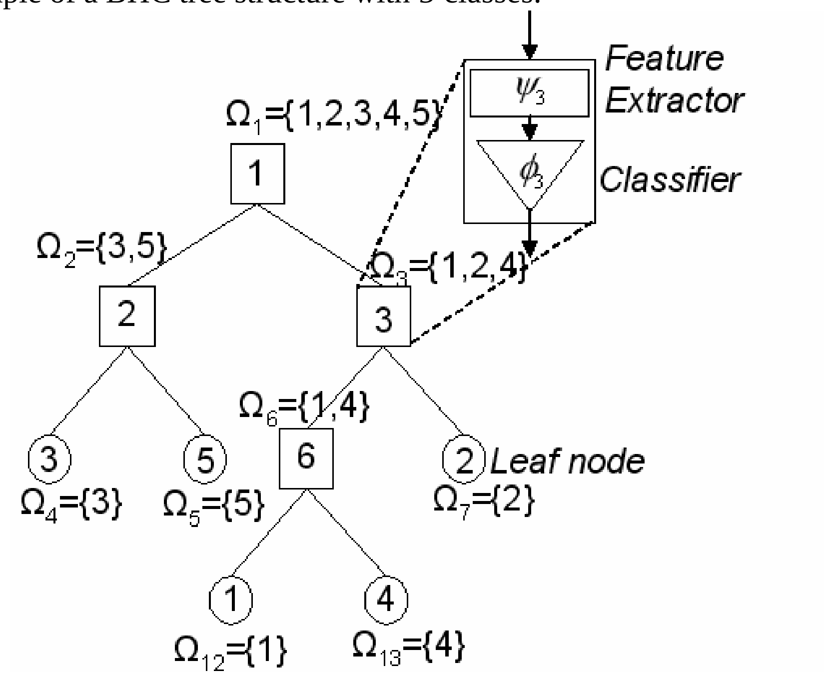

Figure 3.1: Example of a BHC with 5 classes [2].

In its current implementation, a feature extractor at each internal node extracts those features that best discriminate child node pairs in a reduced input space. Kumar *et al.* apply the Fisher discriminant as the feature extractor. Additionally, it is often easier for a classifier to distinguish between two subsets of classes than it is for the classifier to distinguish between all classes simultaneously, thus decomposing (and reducing) the output space. Therefore, the best child node pair which is that pair with the strongest associations based on a posterior probability based criterion, is chosen and the parent node branches on those child nodes.

Based solely on the training data, the construction of the BHC tree can be accomplished in two ways: a top-down approach or a bottom-up approach. The top-down version of the BHC tends to be less greedy than the bottom-up version, and the two versions often yield different results. The top-down approach starts with all of the classes in a single metaclass which is partitioned into two child nodes (subsets) that can be depicted as having three possible combinations of *child* nodes: leaves only, i.e. two single class child nodes if the parent metaclass is comprised of only two classes; smaller metaclasses, i.e. child nodes made up of more than one class but fewer classes than the metaclass that serves as its parent, so that $\Omega_{2n} \cup \Omega_{2n+1} = \Omega_n$ if the parent metaclass is made up of four or more classes; or some combination of a leaf and metaclass if the parent metaclass is comprised of three or more classes. The bottom-up version of tree construction is initiated with all the individual classes as leaf nodes and successively combines those leaves, metaclasses, or combination of leaf and metaclasses determined to be the least distinguishable from each other. This agglomeration is continued until the single metaclass containing all the classes is attained. Once the tree is built, the classifier uses the structure for classification of unlabeled observations. Refer to [2, 3] for more information on the BHC.

3.1.1 Top-down BHC

The top-down BHC framework uses the Generalized Associative Modular Learning System (GAMLS) [43], a deterministic annealing-type algorithm.

GAMLS is used to partition each metaclass into two child nodes until each branch of the tree is reduced to a single class at the leaves. This process decomposes the output space at each branching of the tree by reducing the number of possible allocations of an observation to two choices at each branch of the tree.

Each individual class contained in a metaclass is ultimately assigned to one, and only one, of the two child nodes. This allocation is accomplished by computing the posterior probabilities of each class $v \in \Omega$ belonging to either child node. This requires allocating the classes to nodes and estimating the parameters for the child nodes. Each partition needs to be explored to ensure that the best partition was found. Instead of allocating classes directly, GAMLS "softly associates" classes with child nodes by associating one class with one of the subordinate metaclasses with probability 1 while all other classes are equally associated with each subordinate metaclass with probability .5. The algorithm updates these associations at each step until the associations are clear, i.e., close to 1 for "associated with" and close to 0 for "not associated with". For the metaclass with $C > 2$, GAMLS execution can be summarized as: (1) the Fisher feature extractor reduces the feature space to that which maximally discriminates between the two "soft" metaclasses using the current associations; (2) the mean log-likelihoods of classes in the feature space are computed, (a univariate or multivariate Gaussian distribution is assumed); (3) associations are updated by maximizing the weighted sum of the log-likelihoods subject to an annealing constraint; (4) Steps 1 through 3 are repeated until the incremental increase in the defined gain is insignificant; (5) the stopping threshold is reached then the execution halts returning the current associations, else the temperature is cooled and execution returns to the Fisher feature extractor. (These steps are displayed in Figure 3.2.) As the temperature cools, the associations (posterior probabilities) approach 0 or 1. When the algorithm terminates, the partition is realized and the metaclass is split between those classes that most closely associate with Ω_a and those that most closely associate with Ω_b. This splitting is continued at all internal nodes until only leaf nodes remain. Unclassified observations are ultimately classified using the resulting binary hierarchical classifier.

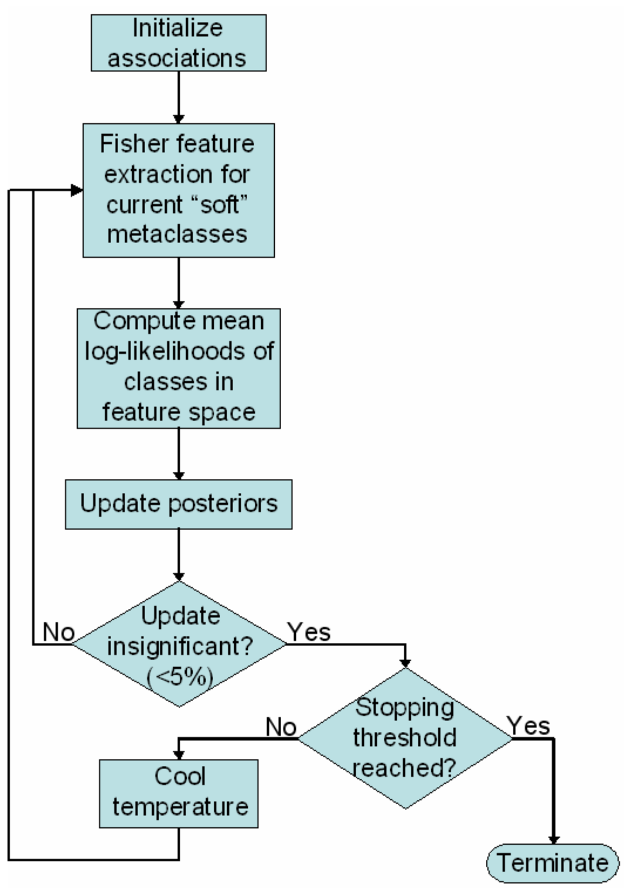

Figure 3.2: Flowchart of GAMLS execution.

3.1.2 Best Bases BHC

An approach referred to as "best bases feature extraction", was developed by Kumar *et al.* [41] for a Bayesian Pairwise Classifier (BPC) to reduce the input candidates in high dimensional remote sensing data. Many of the original features of hyperspectral data, which are comprised of potentially hundreds of narrow, contiguous windows of the electromagnetic spectrum, are highly correlated and provide redundant information. Implemented in both a bottom up band aggregation mode and a top down splitting mode, the method seeks to reduce the number of highly correlated features while maintaining good discrimination between pairs of classes in the BPC. The approach was modified by Morgan *et al.* [42, 44] and incorporated in the BHC. In this bottom up implementation of best bases feature extraction, the features which are contiguous in the spectrum and are highly correlated are combined to form a class dependent feature "group" at every node of the BHC. Spectrally adjacent feature groups are successively combined until some user defined threshold is satisfied. The resulting best bases features then replace the original features, thereby reducing the dimensionality of the input space while exploiting the correlation structure inherent in the data.

3.2 RESEARCH

In this study, Tabu Search (TS) was investigated as a means of improving classification accuracies within the BHC and Best-Bases BHC frameworks. TS was first implemented as a means of generalizing the greedy feature selection within a specified tree structure obtained by the original BHC and Best Bases

BHC. Feature selection extracts the most useful bands/band groups from the feature vector and presents them to the classifier as a vector of lower dimension whose elements retain only the most significant characteristics of the original input space. Feature selection also attempts to remove any redundant and/or irrelevant features. TS uses the greedy feature selection results as an initial solution and searches the solution space for subsets of features (original features for the BHC and combined features for the Best Bases BHC) which yield higher classification accuracies while leaving the hierarchical tree unchanged.

TS is then investigated as the feature selector at each internal node as the hierarchy is being constructed. In this configuration, TS aides in the construction of the binary hierarchical structure and can be applied when using either the original features or the best bases combined features.

While class hierarchies such as those resulting from the BHC generally achieve good classification accuracies, leaf nodes that are statistically close to each other can reside in two unrelated branches. The BHC algorithms do not have the ability to examine the resulting tree structure and to rebuild/rearrange the branches and leaf nodes when the algorithm is unable to effectively partition/merge the metaclasses. The hierarchy, once built, is fixed without any possibility of recourse. This second application of TS provides a method that allows hierarchical classification algorithms to rearrange the resulting class hierarchies through the application of a combinatorial search through the solution space containing all possible class hierarchies. For this component of the study, the tree structure from a hierarchical classifier like the BHC becomes the initial incumbent solution for the TS algorithm and is stored as the "best solution found thus far".

The primary goals of this research were to extend knowledge and understanding in the areas of classification and to implement TS within the hierarchical classification framework in the quest for increased classification accuracies. A secondary goal is to select a meaningful set of features that provide domain knowledge. Finally, robustness of classifiers is important as values of the inputs used to train and test the classifier may not be representative of the population, or the classifier may need to be applied to a similar dataset for which no training data are available. This work should contribute to that longer term goal.

Chapter 4

The BHC with Tabu Search Feature Selection (TS-FS)

The output of the BHC is a binary hierarchical tree that is used to assign a class label to observations whose class is unknown. A typical BHC class hierarchy is displayed in Figure 4.1 for a dataset with five classes where the root node includes all C classes, the leaf nodes are the individual classes, and the internal nodes are metaclasses or subsets of the original set of classes. Starting at the root node, each internal node is partitioned into two child nodes, two mutually exclusive subsets of the classes at that node, where $\Omega_{2n} \cup \Omega_{2n+1} = \Omega_n$. The partitioning continues until the destination node for each branch of the tree results in a leaf node, yielding a binary class hierarchy with $2C$-1 nodes (C leaf nodes and C-1 internal nodes).

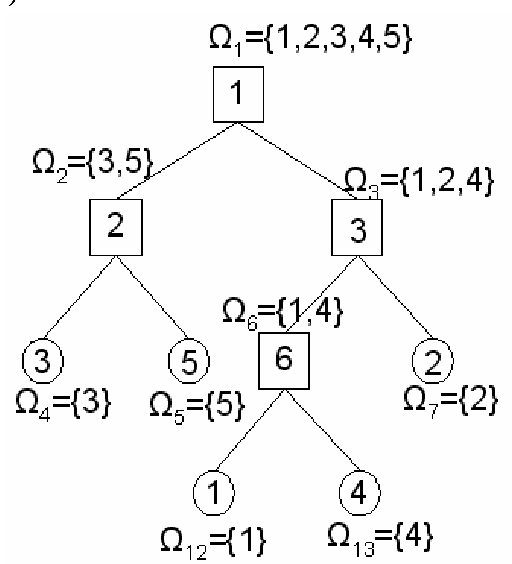

Figure 4.1: Typical BHC hierarchical tree for a dataset with five classes.

Once the BHC class hierarchy is constructed using the entire feature set, an option using feature selection then iteratively examines each internal node, selecting that subset of features which is most useful for discriminating between each internal node's child nodes. This feature selection option is currently accomplished in the operational code [4] by a greedy forward feature selection algorithm. Considering the particular classes present in the current metaclass, the first feature included is the feature that individually yields the highest classification accuracy. The second feature that is considered for inclusion in the feature subset is that feature which, when included, maximizes a log-odds relevance function; the feature is subsequently selected if the classification accuracy at the current node is increased more than an arbitrarily selected threshold (.01). This process continues until the increase in accuracy is less than the defined threshold. Here, features are only added to the subset, never removed. Once the hierarchy is constructed and all metaclass features selected, unclassified observations are labeled as described in Section 3.1.1.

4.1 TABU SEARCH FEATURE SELECTION

A feature subset selection algorithm attempts to find an optimal or near-optimal subset of features. In its simplest implementation, Tabu Search Feature Selection (TS-FS) is a post-processing algorithm that operates on, but does not change the class hierarchy developed by the original BHC. It can add or remove features from the feature subset during the search.

The new TS based feature selection algorithm developed in this study starts with the root node and travels down the hierarchical tree, iteratively considering each internal node for feature subset selection. At the root node, the TS-FS is initiated using the BHC feature subset as an incumbent solution. The objective is to maximize the classification accuracy, i.e., the percentage of correct labels of classes that are members of the metaclass at the current node. This accuracy is computed using the same classification scheme as the original BHC and the same training data. If the classification accuracy for any node is perfect (100%), the node is skipped. The move neighborhood selected for the TS procedure consists of the union of all possible swaps and inserts of features that can be achieved from the current incumbent solution. The swap neighborhood considers all possible single-feature swaps between the sets of used and unused features. This neighborhood does not change the current number of features used. The insert neighborhood considers all single-feature insertions both from the set of selected features into the set of unused features and from the set of unused features into the set of currently selected features. This neighborhood is either incrementing or decrementing a feature from the current set of features selected at the current node. If a feature to be included in the feature subset is highly correlated with any features already present (exceeds a user defined correlation threshold), the move is not allowed. This prohibition, which was included for analysis of remotely sensed hyperspectral data, ensures that features being considered for inclusion are not redundant. The maximum number of features allowed at any node is also user-defined. It can be unrestricted allowing greater search flexibility, or the user may define a maximum number of features based on knowledge of the problem. Other user-defined parameters include the maximum tabu tenure, the minimum tabu tenure, the initial tabu tenure (a number between the maximum and minimum defined tabu tenures), the number of iterations allowed with no improvements before halting the execution, and the maximum allowable number of iterations. The tabu list is initialized as a column vector of zeros with a row for each feature. When a feature is selected for movement either into or out of the subset of features, that feature is marked as tabu and the tabu list records the iteration number of that feature's entry into or exit from the list. That feature cannot be moved again until it has been on the tabu list for the number of iterations specified by the tabu tenure. An exception to this rule is made when moving the feature results in a classification accuracy that is higher than any other accuracy previously achieved. In this case, the tabu status is overruled, and the move is allowed.

The user defined maximum and minimum tabu tenure are employed to determine an adaptive tabu tenure strategy. The tabu tenure is never allowed outside of the boundaries defined by maximum and minimum tabu tenures. An improving classification accuracy decrements (if possible) the tabu tenure to allow an intensified search in the current area of the solution space. If no improving classification accuracies are found, the tabu tenure is incremented (if possible) to encourage the search to leave the current area of the solution space and diversify into other unexplored areas of the solution space.

Given an incumbent solution, the best non-tabu move within the move neighborhood is selected. (Since the best non-tabu move is not necessarily an improving move, TS can escape from local optima.) If the current classification accuracy is the highest value yet achieved, the new subset is recorded as the best yet found. The next iteration is performed. Iterations continue until either the user-defined number of iterations has been completed or no improvements have been found within the specified maximum-number-of-iterations-with-no-improvement. When the TS terminates for the current node, the best subset of features is recorded for that node, and the algorithm progresses to the next node for feature selection until all of the nodes have been processed. A flowchart of the algorithm is displayed in Figure 4.2.

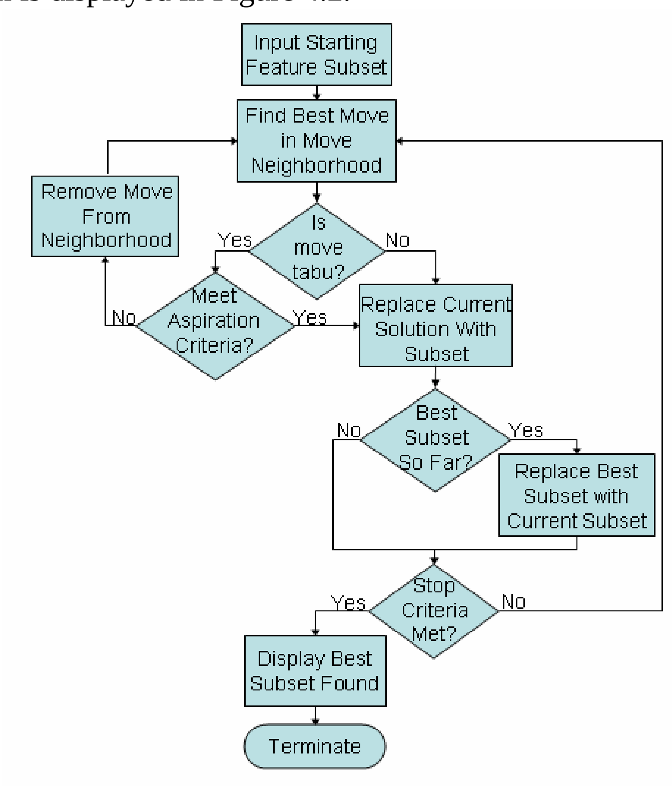

Figure 4.2: Flowchart of TS-FS Algorithm.

When the TS-FS is completed for each node, novel observations are classified using the binary hierarchical tree with feature subsets selected by TS.

All algorithms were executed using MATLAB® student version 6.0.0.42a, release 12 dated 13 November 2000 with the Pentium IV patch applied and implemented on a personal computer with an Intel® Pentium® IV, 2.66GHz and 512 Mb of RAM; all execution times reported are in reference to this system.

4.2 APPLICATION OF TS-FS ALGORITHM TO STATIC TREES

The TS-FS algorithm was applied to the BHC tree obtained from three datasets: multispectral and hyperspectral remotely sensed data acquired over Botswana and a standard character recognition dataset.

4.2.1 Botswana Advanced Land Imager (ALI) Dataset

The Botswana multispectral data were acquired by the Advanced Land Imager (ALI) aboard the Earth Observer 1 (EO1) satellite on 31 May 2001. The mission is being flown to evaluate experimental sensor technology for future space missions. For example, ALI is a prototype sensor for the Landsat Data Continuity Mission (LDCM). The array of data can be displayed as an image where each pixel represents a vector-valued observation. The data cover a subset of the Okavango Delta of Botswana that is undergoing change due to anthropogenic and natural processes such as seasonal flooding. A small subset of the data is displayed in Figure 4.3 to illustrate the difficulty of land cover classification in this particular area.

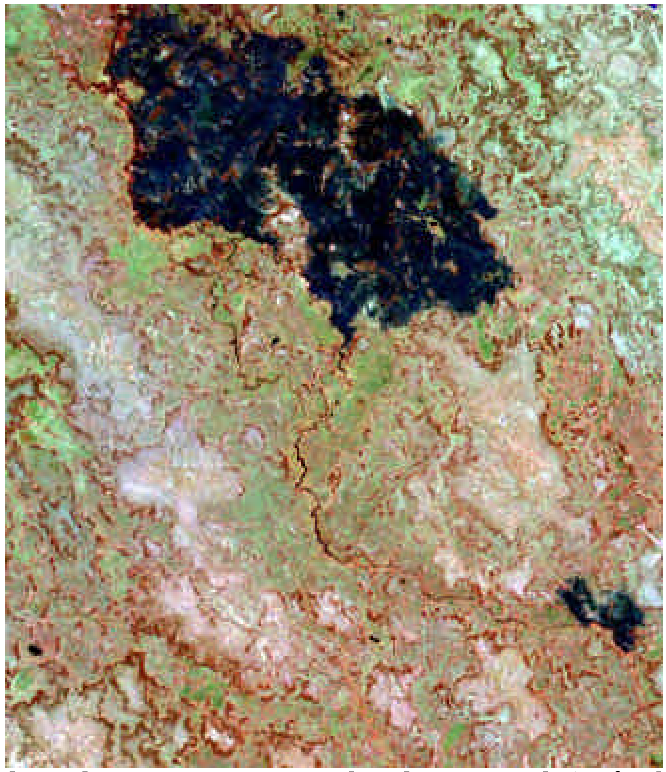
Figure 4.3: False color RGB composite (bands 4p, 5 and 3) of subset of Botswana ALI data.

Data were pre-processed by the UT Center for Space Research (CSR) prior to analysis in this study [45]. The ALI data consist of observations from 23 identified classes representing the land cover types in the area, each with 9 integer features which represent the spectral reflectance of the land cover types within contiguous bands of the visible and near infrared spectrum. The class numbers, names and number of ground truth observations are listed in Table 4.1. In addition to vegetation, soils and water, three types of floodplain are identified: floodplain1 (class 17) is the primary floodplain, floodplain2 (class 18) represents the seasonal floodplain, and floodplain3 (class 19) is considered to be a secondary floodplain. In addition, two fire scar classes are identified: firescar1 (class 22) was recently burned, whereas firescar2 (class 23) was previously burned and exhibits some patches of new vegetation growth. CSR provided ten partitions of the data, where each class was randomly sampled and the data partitioned such that 50% of the data were identified for the training of the classifiers and the remaining 50% identified for the subsequent testing of the classifiers. These ten datasets were maintained, and the same testing/training data utilized for each of the experiments labeled ALI1-ALI10. Appendix A contains selected results (class hierarchies and confusion matrices) for this dataset. Because the training and test data are spatially co-located in regions of known classes, accuracies can be inflated in remote sensing applications. For this reason, an additional independent test set was also provided, and data were classified as novel observations.

Class #	Class Name	Training Sample Size
1	north riparian	157
2	south riparian	193
3	short mopane	303
4	mopane (dense)	249
5	acacia mix	254
6	woodland mix	201
7	acacia woodlands	149
8	acacia shrublands	134
9	acacia grasslands	171
10	mopane/pechuel/grass mix	164
11	grass/pechuel mix	170
12	dry grasses	252
13	island interior	166
14	exposed soil	118
15	reeds1	192
16	backswamp	233
17	floodplain1	202
18	floodplain2	193
19	floodplain3	340
20	water	241
21	aquatic vegetation	151
22	firescar1	248
23	firescar2	156

Table 4.1: Class information for the Botswana ALI dataset.

4.2.2 Botswana Hyperion Dataset

The Hyperion sensor on EO-1 is the first hyperspectral sensor successfully flown in space. It acquires data simultaneously with ALI, but over a smaller area (7.5 km vs. 37 km strip width) that is shifted slightly to the west as the telescopes for the sensors are not co-aligned. The width of the Hyperion strip is smaller because the number of bands is more than 20 times that of ALI, thereby resulting in a dramatic increase in the amount of data recorded. ALI and Hyperion cover the same range of the electromagnetic spectrum [46]. The data were provided to the study after extensive pre-processing was completed by CSR. The Hyperion dataset consists of observations from 14 identified classes representing the land cover types in the area studied, each with 242 candidate features. Uncalibrated and noisy bands that cover water absorption features are removed, and the remaining 145 bands are included as candidate features: [10-55, 82-97, 102-119, 134-164, 187-220]. The class numbers, names and number of ground truth observations are presented in Table 4.2. As with the ALI, CSR provided ten randomly sampled partitions of the data, which were subdivided into 50% for training and 50% for testing the classifiers, and an independent test set. These data splits were maintained throughout the study and are labeled HYP11-HYP20. Selected results (class hierarchies and confusion matrices) for this dataset are contained in Appendices B and C.

Class code	Class	Training sample size
1	water	270
2	hippo grass	101
3	floodplain grasses1	251
4	floodplain grasses2	215
5	reeds1	269
6	riparian	269
7	firescar2	259
8	island interior	203
9	acacia woodlands	314
10	acacia shrublands	248
11	acacia grasslands	305
12	short mopane	181
13	mixed mopane	268
14	exposed soils	95

Table 4.2: Class information for the Botswana Hyperion dataset.

4.2.3 Letter Recognition Dataset

The letter recognition data were obtained from the University of California, Irvine (UCI) [47] Machine Learning Repository with the title, Letter Image Recognition Data. This dataset consists of 20,000 instances, where typically the first 16,000 are used for training and the last 4,000 for testing; this partition was followed for this study. The class labels are contained in Table 4.3.

Class #	Class Name	Training Sample Size	Testing Sample Size
1	A	633	156
2	B	630	136
3	C	594	142
4	D	638	167
5	E	616	152
6	F	622	153
7	G	609	164
8	H	583	151
9	I	590	165
10	J	599	148
11	K	593	146
12	L	604	157
13	M	648	144
14	N	617	166
15	O	614	139
16	P	635	168
17	Q	615	168
18	R	597	161
19	S	587	161
20	T	645	151
21	U	645	168
22	V	628	136
23	W	613	139
24	X	628	159
25	Y	641	145
26	Z	576	158

Table 4.3: Class information for the letter recognition dataset.

Each instance is a black-and-white rectangular pixel display of one of the 26 capital letters of the English alphabet (see Figure 4.4 for example letters which yielded individual data observations) and is described by 16 integer-valued numerical attributes (statistical moments and edge counts), or features (see Table 4.4). The best accuracy obtained for this dataset is reported in the literature as “a little over 80%” [47]. Confusion matrices for this dataset can be found in Appendix D.

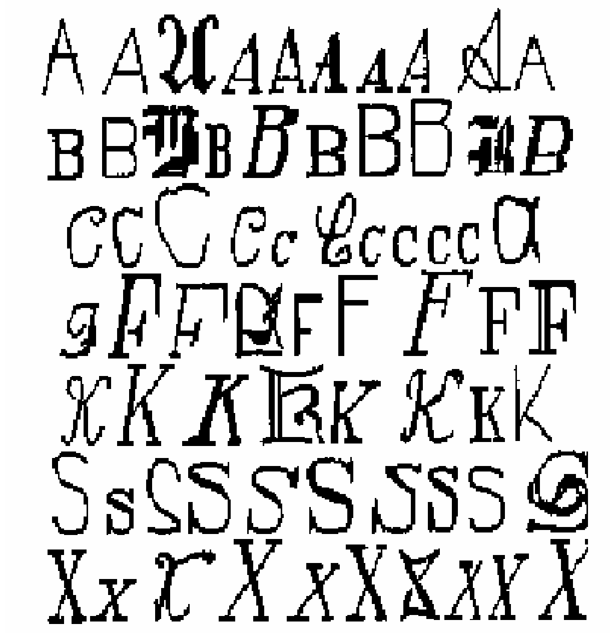

Figure 4.4: Examples of letters which yielded individual data observations for the letter recognition dataset [2].

Feature #	Feature Name	Description
1	x-box	horizontal position of box
2	y-box	vertical position of box
3	width	width of box
4	high	height of box
5	onpix	total # on pixels
6	x-bar	mean x of on pixels in box
7	y-bar	mean y of on pixels in box
8	x2bar	mean x variance
9	y2bar	mean y variance
10	xybar	mean x y correlation
11	x2ybr	mean of x*x*y
12	xy2br	mean of x*y*y
13	x-ege	mean edge count left to right
14	xegvy	correlation of x-ege with y
15	y-ege	mean edge count bottom to top
16	yegvx	correlation of y-ege with x

Table 4.4: Feature information for the letter recognition dataset.

4.3 IMPLEMENTATION OF TS-FS ALGORITHM AND RESULTS

The TS parameters were tuned using the first experiment for the ALI and Hyperion datasets. The parameters were then used for the remainder of the experiments. For the letter recognition dataset, the parameters were tuned with the single data partition.

4.3.1 Feature Selection Results for ALI Remotely Sensed Data

Each of the ten datasets (experiments) was analyzed by the BHC, both with and without the original feature selection (FS) method. The TS-FS was then performed on the static tree structures output by this algorithm using the features selected by the greedy algorithm as the TS starting solutions. The overall classification accuracies for each of the algorithms are displayed in Table 4.5. Tabu tenure was set at 3. Because this dataset has only 9 features, neither the correlation check for inclusion of new features nor the adaptive tabu tenure was utilized. The stopping criterion was set at 30 iterations, and the maximum number of iterations to continue with no improvements was set at 10.

Experiment	BHC	BHC FS	BHC TS-FS
ALI1	88.72 / 72.82	86.38 / 69.18	88.59 / 72.74
ALI 2	87.20 / 71.71	85.30 / 64.92	89.71 / 72.20
ALI 3	86.29 / 69.69	86.64 / 68.16	89.88 / 71.20
ALI 4	86.60 / 70.96	85.99 / 68.62	90.01 / 71.69
ALI 5	88.33 / 73.33	86.34 / 67.73	90.06 / 73.36
ALI 6	87.64 / 73.63	85.82 / 66.97	89.32 / 72.66
ALI 7	86.86 / 70.96	87.68 / 67.48	90.06 / 71.90
ALI 8	85.82 / 75.03	84.48 / 71.82	88.28 / 75.16
ALI 9	87.25 / 69.61	86.73 / 71.39	89.67 / 70.64
ALI 10	88.98 / 72.68	87.42 / 69.37	89.75 / 72.09
Average	**87.37 / 72.04**	**86.28 / 68.56**	**89.53 / 72.36**
Standard Deviation	**1.05 / 1.76**	**0.95 / 2.04**	**0.62 / 1.25**

Table 4.5: BHC, BHC FS and BHC TS-FS overall experiment classification accuracies (%) for Botswana ALI testing/independent test data.

The original BHC utilizes the full set of features (weighted according to the Fisher projection) and consistently yields higher accuracies than the BHC with greedy feature selection. The goal of both the original FS and TS-FS are to reduce the number of features, both to improve interpretability and increase robustness of the classifier. In every experiment (using the test data), the class hierarchy utilizing the TS-FS resulted in higher overall classification accuracies than the BHC with the greedy feature selection by an average of 3.26% per experiment, and in 9 out of the 10 experiments it yielded higher overall classification accuracies than the BHC by an average of 2.16% per experiment (only experiment ALI1 resulted in a lower overall accuracy). Even more significantly, standard deviation of the classification accuracies was also reduced relative to both the BHC and BHC FS. For the testing data, the standard deviation of the accuracies for the TS-FS was only ~60% of that of the BHC and ~65% of that of the BHC-FS. For the independent test data it was ~70% of that obtained by the BHC and ~60% of that for BHC-FS. Thus, TS-FS method yielded a more stable set of features. The tree structures had 22 internal nodes consisting of metaclasses where the feature selection was implemented. On average per class hierarchy, compared to the results of the greedy feature selection: no feature selection was performed at 4 of the metaclass nodes because the classification accuracy at the nodes was 100%; feature selection was performed on 18 of the metaclass nodes, and of these the classification accuracy at 16.3 of the metaclass nodes was improved by an average of 1.65% per metaclass with 3.8 of the nodes improving to 100%; and the classification accuracy at 1.7 of the metaclass nodes could not be improved upon using tabu search feature selection. Of a total 198 possible features per hierarchy (9 features per metaclass node), the greedy feature selection chose an average of 70 per hierarchy while the TS-FS (starting with the features selected by the greedy algorithm) chose an average of 103.7 features per hierarchy and maintained an average of 55.7 of the greedy features per hierarchy. The first feature selected by the greedy algorithm at each metaclass is that feature which is individually the most significant contributor to classification accuracy; these first-chosen features were discarded by the TS feature selector an average of 4.8 times per tree in order to find better feature subsets and to attain better classification accuracies at the metaclasses. Given that these features are considered to be the "most important" in one sense, this clearly illustrates the value of eliminating features subsequent to their initial selection. Using the independent test data, the BHC TS-FS resulted in the highest overall average accuracy of 72.36%. Figure 4.5 is an example of the classification of the data subset from Figure 4.3.

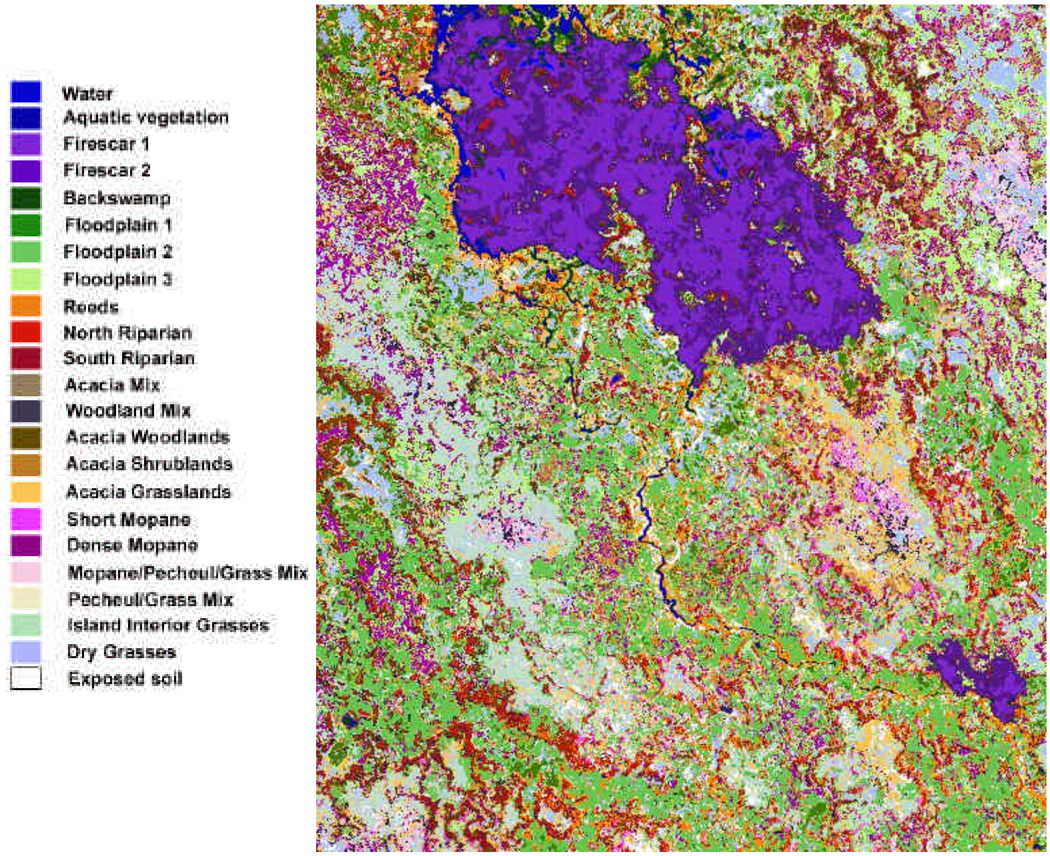

Figure 4.5: Example of a classified subset using the BHC TS-FS classifier (experiment ALI7: test set accuracy 90.06%, independent test set accuracy 71.90%).

The average classification accuracies for each class for each algorithm are displayed in Table 4.6 highlighting the highest average accuracy per class. Each of the algorithms performs well in classifying selected classes, but the TS-FS is able to classify a majority of the classes more consistently for this dataset using both the test and independent test data. While the standard deviations of the class classification accuracies are comparable for the BHC and the BHC TS-FS, there was a reduction relative to the BHC FS. For the test data, the standard deviation of the class accuracies for the TS-FS was ~70% of that of the BHC FS, and for the independent test data, it was ~65% of that for the BHC FS, again indicating that the TS-FS yielded a more stable set of features than the BHC FS. Class 8, which is consistently classified with low accuracy by all algorithms, is a mixed class.

Class #	Class	BHC	BHC FS	BHC TS-FS
1	north riparian	76.93 / 54.05	72.81 / 56.49	75.38 / 60.54
2	south riparian	88.12 / 85.08	88.23 / 69.59	89.38 / 72.18
3	short mopane	95.76 / 88.48	91.52 / 87.63	95.25 / 88.53
4	mopane (dense)	82.59 / 77.88	82.41 / 75.96	87.49 / 75.05
5	acacia mix	87.96 / 92.82	87.24 / 87.39	88.67 / 90.48
6	woodland mix	96.00 / 87.50	96.10 / 98.08	97.10 / 98.42
7	acacia woodlands	84.98 / 38.54	84.19 / 51.58	87.15 / 46.32
8	acacia shrublands	66.26 / 40.17	65.23 / 36.39	69.69 / 41.37
9	acacia grasslands	84.95 / 16.84	72.71 / 18.51	78.00 / 17.59
10	mopane/pechuel/grass mix	94.02 / 93.06	90.36 / 92.50	92.79 / 93.06
11	grass/pechuel mix	88.71 / 85.71	90.95 / 94.70	91.41 / 93.34
12	dry grasses	81.74 / 88.43	76.68 / 77.36	82.13 / 83.72
13	island interior	87.47 / 76.15	87.48 / 76.80	87.48 / 75.60
14	exposed soil	79.67 / 63.63	94.56 / 79.84	92.86 / 75.97
15	reeds1	93.34 / 95.03	89.38 / 87.89	93.87 / 91.64
16	backswamp	84.83 / 70.00	77.66 / 55.00	84.84 / 75.04
17	floodplain1	81.98 / 34.70	87.71 / 27.07	94.43 / 37.38
18	floodplain2	85.20 / 69.92	77.82 / 59.84	92.20 / 77.34
19	floodplain3	80.14 / 59.43	83.00 / 52.20	86.81 / 55.69
20	water	96.93 / 90.51	97.84 / 86.87	96.92 / 88.82
21	aquatic vegetation	82.67 / 82.22	96.94 / 90.15	89.48 / 86.26
22	firescar1	99.28 / 65.13	98.80 / 61.91	98.96 / 54.47
23	firescar2	98.31 / 99.20	89.35 / 64.63	97.44 / 90.37
	Average	**86.86 / 71.93**	**86.04 / 69.06**	**89.12 / 72.57**
	Standard Deviation	**4.27 / 5.27**	**5.33 / 8.34**	**3.86 / 5.36**

Table 4.6: BHC, BHC FS and BHC TS-FS average testing/independent test classification accuracies (%) by class for Botswana ALI data.

For the 10 experiments, the BHC constructed 7 different class hierarchies, and no hierarchy was duplicated more than twice. A representative class hierarchy is displayed in Figure 4.6. The partition of the root node is identical for all of the experiments; subtle differences in the structure become apparent at and below the third level of the trees. Closer inspection of the trees reveals that the acacia shrublands (class 8) was paired with four different classes; this result is not unexpected, as accuracies listed in Table 4.6 reflect that this class is the most difficult for each of the algorithms to classify using the test data. Exposed soil (class 14), which is not closely related phenologically to any other class, was assigned to two different major branches of the class hierarchies in different experiments. While the BHC class hierarchies differ with respect to the exposed soil class, the feature selection is able to isolate those features that are useful for labeling the class and to improve the accuracies for this class. Interestingly, when class signatures are quite similar (e.g. acacia grasslands (class 9) and the dry grasses (class 12)), feature selection may tend to exacerbate the problem of misclassification. This problem is illustrated in Figure 4.7, which contains plots of the training data for experiment ALI8. For illustration of the overall within-class variation, all class 9 training observations are plotted with the class means for classes 9 and 12. Classes 9 and 12 are paired on 9 of the 10 hierarchies, with the acacia grasslands most often misclassified as dry grasses for both test and independent test data due to their similar patterns and variations in the observations sampled for training and test data. When distinguishing between classes 9 and 12, the greedy feature selection generally tended to choose the features 2, 4, 8 and 9 while the TS-FS most often chose features 1, 4, 5, 6 and 9. This difference is significant for such a small number of total features.

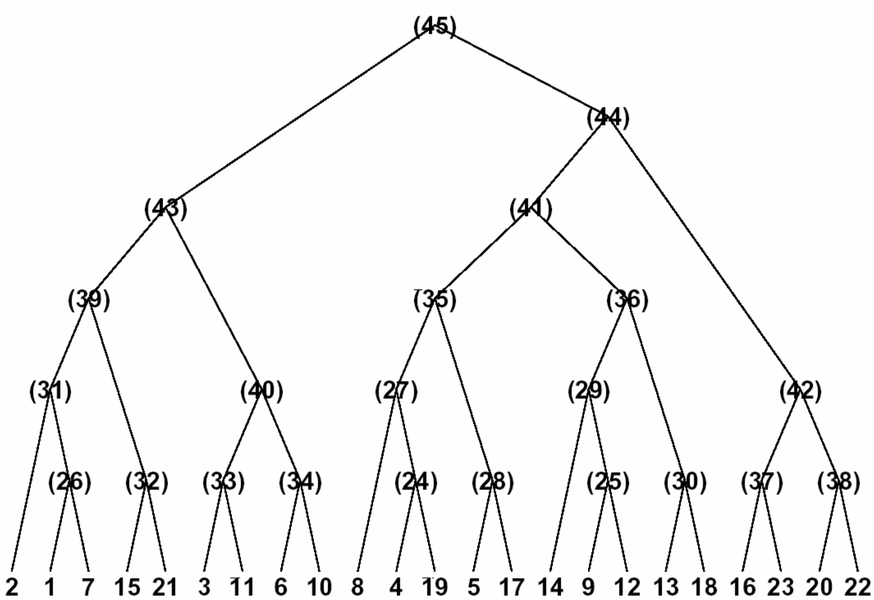

Figure 4.6: Representative BHC tree structure for the Botswana ALI dataset.

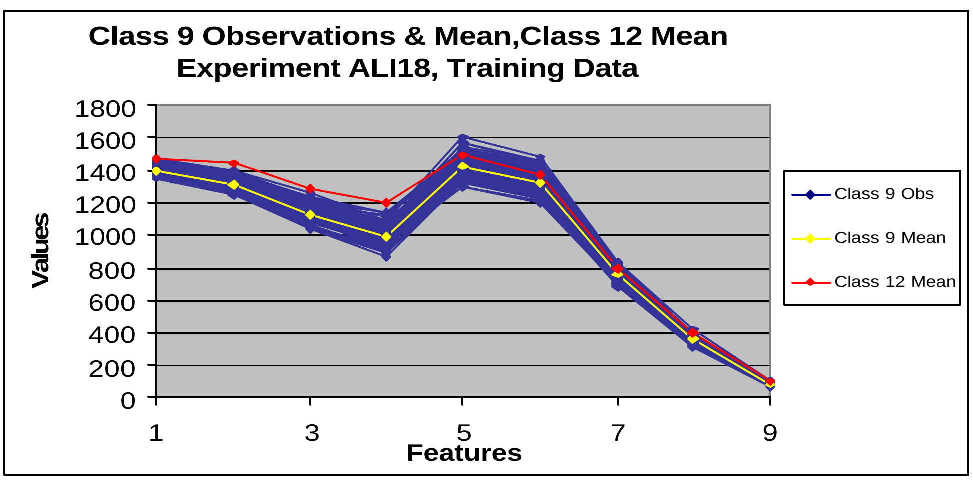

Figure 4.7: Plot of experiment ALI8 training data: class 9 Observations and mean, class 12 mean.

4.3.2 Feature Selection Results for Hyperion Remotely Sensed Data

The Hyperion experiments were also analyzed with the BHC, with and without feature selection. As with the ALI data, the TS-FS was then performed on the static tree structure utilizing the features output by the greedy feature selection algorithm as its starting solution. The overall classification accuracies for each of the experiments are contained in Table 4.7. Dynamic tabu tenure was initialized at 5 and allowed to range from 3 to 10. Because this dataset has 145 total candidate features, the correlation check for inclusion of new features was utilized. The TS stopping criterion was set at 30 iterations and the maximum number of iterations to continue with no improvements was set at 10.

Experiment	BHC	BHC FS	BHC TS-FS
HYP11	92.71 / 61.23	89.13 / 66.32	93.51 / 60.87
HYP12	88.76 / 56.87	86.53 / 63.44	89.75 / 67.52
HYP13	88.08 / 69.36	90.30 / 69.36	92.77 / 64.08
HYP14	91.91 / 60.07	86.72 / 62.88	92.59 / 58.35
HYP15	89.99 / 58.55	85.98 / 59.47	90.67 / 62.07
HYP16	91.85 / 60.63	87.34 / 64.04	92.16 / 66.92
HYP17	91.60 / 59.43	86.29 / 62.15	92.90 / 61.35
HYP18	91.91 / 60.59	90.80 / 68.88	92.34 / 63.88
HYP19	89.19 / 63.52	85.05 / 68.92	89.13 / 63.40
HYP20	90.67 / 62.07	85.55 / 63.04	91.54 / 62.15
Average	**90.67 / 61.23**	**87.37 / 64.85**	**91.74 / 63.06**
Standard Deviation	**1.58 / 3.39**	**2.01 / 3.36**	**1.44 / 2.75**

Table 4.7: BHC, BHC FS and BHC TS-FS overall experiment classification accuracies (%) for Botswana Hyperion testing/independent test data.

In every experiment using the test data, the tree structure utilizing the TS-FS resulted in higher overall classification accuracies than the BHC with the greedy feature selection by an average of 4.33% per experiment, and in 9 out of the 10 experiments it resulted in higher overall classification accuracies than the BHC by an average of 1.07% per experiment (only experiment HYP19 resulted in a lower overall accuracy). In addition, the standard deviation of the classification accuracies was reduced relative to the other algorithms: for the test data, the standard deviation of the accuracies for the TS-FS was ~90% of that of the BHC and ~70% of that of the BHC FS, and for the independent test data, the standard deviation was ~80% of both the BHC and BHC FS. The class hierarchies had 13 metaclasses where feature selection was implemented. On average per hierarchy, when compared with the results of the greedy feature selection: no feature selection was performed at 3.6 of the metaclass nodes because the classification accuracy was 100%; feature selection was performed on 9.4 of the metaclass nodes and of these the classification accuracy at all of the metaclass nodes was improved by an average of 2.03% per metaclass (with 2.7 of the nodes improving to 100% with the test data). With a maximum of 1885 features per tree (145 features per metaclass node), the greedy feature selection chose an average of 40.6 per tree while the TS-FS chose an average of 62.5 features per tree, maintaining an average of 22.7 of the greedy features per class hierarchy. The greedy first-chosen features at each metaclass were discarded by the TS-FS an average of 6.6 metaclasses per tree.

The average classification accuracies for each class for each algorithm are listed in Table 4.8 highlighting the highest average accuracy per class. The BHC and the BHC with TS-FS both outperform the BHC with greedy feature selection. The BHC is able to classify a majority of the classes more consistently than the BHC with TS-FS for this dataset; however, when the BHC using TS classifies an individual class with higher average accuracy, it is able to do so with greater improvements in the accuracies (for example, class 14 BHC accuracy: 76.16% and BHC TS-FS accuracy: 98.73%). When the BHC results in higher class accuracy than the BHC TS-FS, it averages 1.56% improvement, while the BHC TS-FS averages 5.10% better than the BHC for individual average class accuracies. The standard deviations of the class classification accuracies are somewhat elevated due to the large differences in the capabilities of individual experiments to classify some individual classes. For example, for the BHC, experiment HYP13 classifies exposed soil (class 14) with an accuracy of 25.5% while experiment HYP18 is able to classify it with an accuracy of 89.4%. Using the same class hierarchy and the same training/testing data, the TS-FS is able to increase the exposed soil classification accuracies for these experiments to 97.9% and 100% respectively, while greatly reducing the standard deviation for this particular class from 20.41 (BHC) to 2.67 (TS-FS). Particularly significant, were the reductions in standard deviations for the testing data, where the average standard deviation of the accuracies for TS-FS was ~78% of that of the BHC and ~50% of that of the BHC FS.

Class #	Class	BHC	BHC FS	BHC TS-FS
1	water	100.00 / 99.92	99.41 / 98.81	99.41 / 99.53
2	hippo grass	87.60 / 15.68	96.80 / 51.29	97.60 / 40.12
3	floodplain grasses1	95.12 / 81.39	88.16 / 51.58	96.08 / 53.93
4	floodplain grasses2	96.92 / 72.00	96.34 / 81.88	96.37 / 66.61
5	reeds1	86.03 / 48.93	72.25 / 43.39	84.71 / 58.69
6	riparian	80.09 / 60.76	67.69 / 56.87	83.43 / 63.56
7	firescar2	98.96 / 82.27	93.55 / 88.58	97.20 / 88.01
8	island interior	95.05 / 84.90	93.75 / 83.06	94.35 / 78.98
9	acacia woodlands	88.07 / 69.27	87.01 / 69.67	86.56 / 64.50
10	acacia shrublands	90.86 / 86.74	80.98 / 83.32	87.42 / 85.74
11	acacia grasslands	93.02 / 18.49	90.31 / 30.61	90.45 / 26.68
12	short mopane	87.66 / 66.67	91.34 / 72.75	92.68 / 76.80
13	mixed mopane	84.40 / 57.86	84.34 / 61.20	90.58 / 49.53
14	exposed soils	76.16 / 77.98	98.30 / 99.89	98.73 / 99.78
	Average	**90.00 / 65.92**	**88.59 / 69.49**	**92.54 / 68.03**
	Standard Deviation	**4.82 / 8.48**	**7.39 / 9.51**	**3.77 / 8.54**

Table 4.8: BHC, BHC FS and BHC TS-FS average testing/independent test classification accuracies (%) by class for Botswana Hyperion data.

For the 10 experiments, the BHC constructed 8 different class hierarchies, and no hierarchy was duplicated more than twice. A representative class hierarchy is displayed in Figure 4.10. All of the hierarchies do not share the same partition of the root node. Experiments HYP12 and HYP19 place the acacia woodlands (class 9) with the left branch while all other experiments place it with the right branch. Experiment HYP12 and HYP19 yield the two lowest overall BHC TS-FS classification accuracies while yielding two of the three lowest accuracies for the BHC. Discrimination of acacia shrublands and woodlands is greatly improved in the Hyperion data, presumably due to the increased number of bands. Labeling of acacia grasslands (class 11) is still problematic for the independent test set, as is hippo grass (class 2). This may be due to incorrect labeling of the independent test data, which have not been field validated, changes in signature, or overtraining. The most difficult class for the Hyperion data to discriminate in the test data is the riparian (class 6), which also proved a challenge when classifying the ALI data (north riparian was the second most difficult to classify). The firescar and water classes were most consistently classified with a high degree of accuracy for both the Hyperion and ALI data. These results are not unexpected as the plots of the class means in Figures 4.8 and 4.9 show that the spectra of the most difficult classes to label are clustered toward the centers of plots.

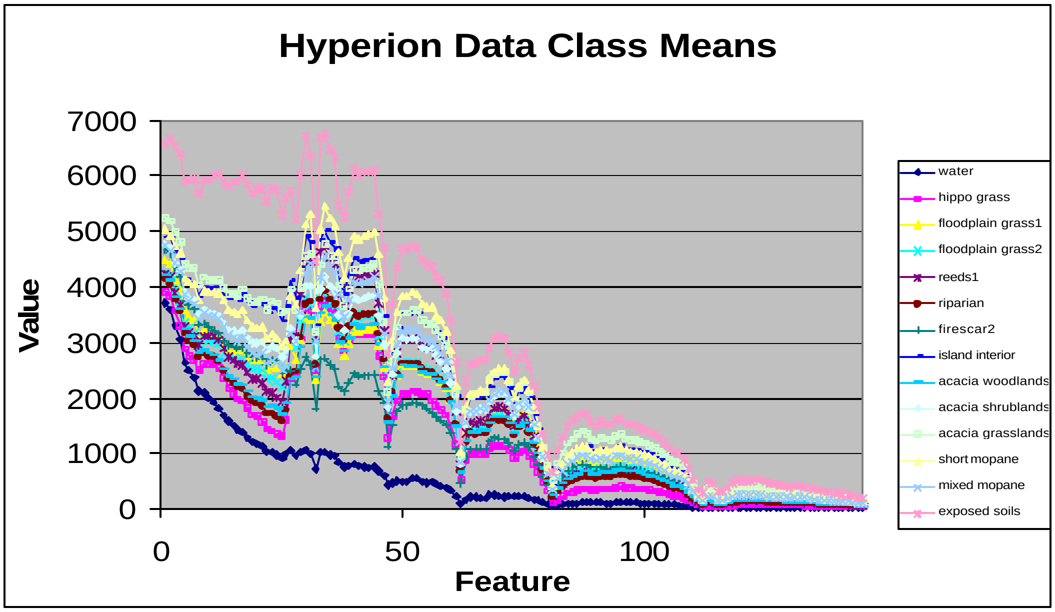

Figure 4.8: Plot of Hyperion data class means.

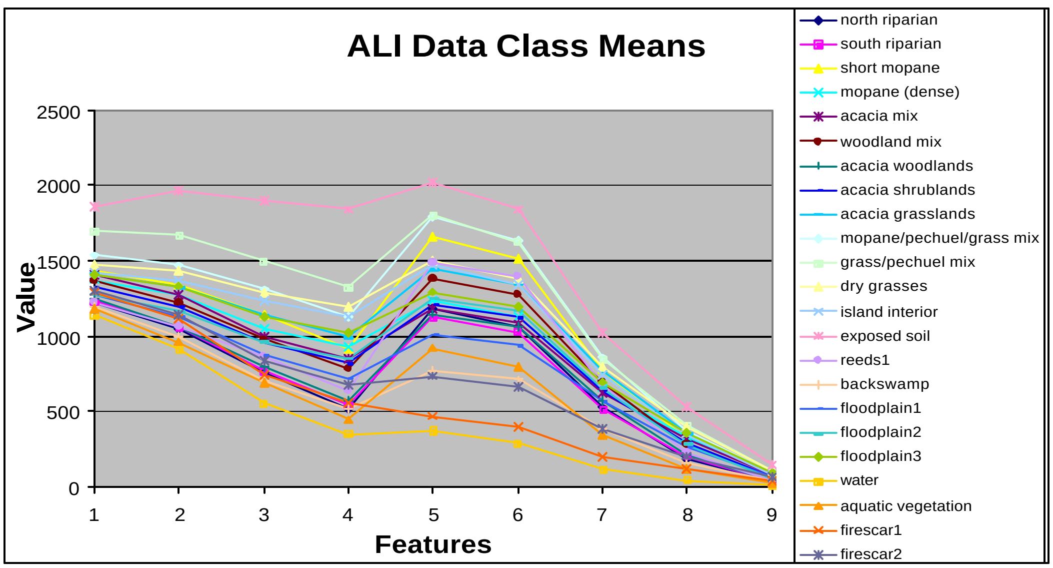

Figure 4.9: Plot of ALI data class means.

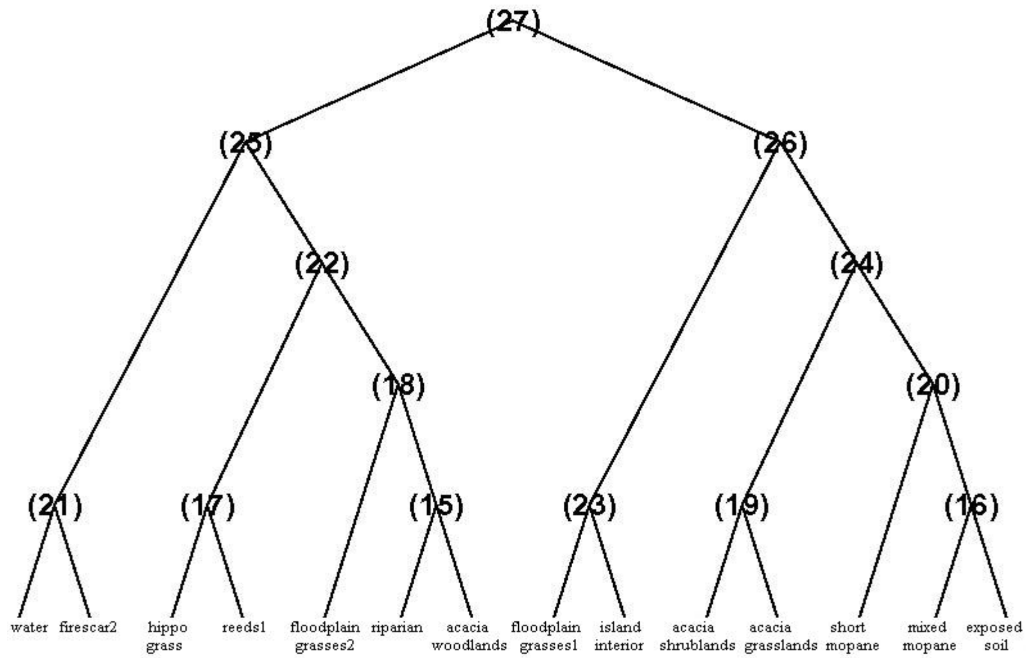

Figure 4.10: Representative BHC tree structure for the Botswana Hyperion dataset.

4.3.3 Feature Selection Results for Hyperion Data using Best Bases

The best bases method for band aggregation described in Section 3.2.2 was applied to the Hyperion data. Since the features are combined differently and thereby best bases features subsequently are selected differently for each metaclass on each tree, only general results are presented here. The overall classification accuracies for the BHC BB, BHC BB with greedy FS and the BHC BB with TS-FS are displayed in Table 4.9. A dynamic tabu tenure initialized at 5 was allowed to vary from 3 to 8. Because the combining of the features is implemented to reduce redundant correlated features, the correlation check for inclusion of new features was not utilized. The stopping criterion was set at 30 iterations and the maximum number of iterations to continue with no improvements was set at 10.

Experiment	BHC BB	BHC BB FS	BHC BB TS-FS
HYP11	89.38 / 56.59	88.94 / 63.84	91.60 / 65.76
HYP12	91.54 / 61.43	86.66 / 68.00	90.92 / 66.80
HYP13	91.17 / 58.55	85.36 / 64.84	93.27 / 69.92
HYP14	92.16 / 61.43	87.77 / 61.51	93.02 / 60.75
HYP15	92.28 / 59.99	86.35 / 61.71	93.70 / 64.80
HYP16	91.54 / 60.15	88.14 / 65.48	93.39 / 62.80
HYP17	91.72 / 61.19	89.31 / 66.84	91.41 / 66.88
HYP18	92.46 / 61.83	86.41 / 64.60	93.21 / 65.84
HYP19	90.30 / 66.08	89.19 / 64.80	90.80 / 70.56
HYP20	92.22 / 62.07	85.55 / 61.67	91.97 / 62.60
Average	**91.48 / 60.93**	**87.37 / 64.33**	**92.33 / 65.67**
Standard Deviation	**0.98 / 2.48**	**1.49 / 2.20**	**1.10 / 3.12**

Table 4.9: BHC BB, BHC BB FS and BHC BB TS-FS overall experiment classification accuracies (%) for Botswana Hyperion testing/independent test data.

The BHC BB reduced the 1885 original features per tree to an average of 850.7 BB features per tree (averaging 65.44 BB features per metaclass). The greedy feature selection chose an average of 40.10 BB features per tree while the TS-FS chose an average of 91.70 BB features. The TS-FS kept an average of 25.30 BB greedy features while maintaining an average of 7.70 of the first-chosen BB features per tree. In every experiment using the test data, the tree structure utilizing the TS-FS with BB resulted in higher overall classification accuracies than the BHC BB with the greedy feature selection by an average of 4.96% per experiment. In 7 of the 10 experiments it achieved higher overall classification accuracies than the BHC BB, and it resulted in a higher overall average accuracy. The independent test data results were similar with the BHC BB TS-FS having the highest average accuracy per experiment. The standard deviations of the experiments, however, are increased for both the test and independent test data when using BB TS-FS over that of the BHC BB. For the test data, the standard deviation of the accuracy for the BHC BB is ~90% of that of the BB TS-FS and ~80% of that of the BB TS-FS for the independent test data. The results are mixed when comparing the standard deviations of the BHC BB FS and the BHC BB TS-FS.

The average classification accuracies for each class for each BB algorithm are displayed in Table 4.10 highlighting the highest average accuracy per class for both the test and independent test data. The BHC BB TS-FS clearly outperforms the BHC BB with greedy feature selection, and it exhibits the ability to classify a majority of the classes more consistently for this dataset. While not improving the classification accuracies for all individual classes when compared to the prior application without BB, the average overall classification accuracies were improved.

Class #	BHC BB	BHC BB FS	BHC BB TS-FS
1	100.00 / 100.00	99.40 / 98.49	99.78 / 98.49
2	94.00 / 15.06	95.60 / 52.90	96.00 / 49.44
3	95.36 / 86.65	89.12 / 52.98	94.40 / 54.94
4	96.92 / 74.30	94.77 / 82.12	95.44 / 72.37
5	89.19 / 50.06	75.15 / 47.26	88.97 / 55.83
6	80.16 / 60.05	60.81 / 51.09	82.99 / 66.63
7	98.80 / 80.57	91.77 / 88.01	95.81 / 88.98
8	96.52 / 87.64	96.41 / 79.30	96.21 / 78.60
9	86.18 / 70.53	89.09 / 80.66	84.83 / 64.64
10	90.06 / 87.00	90.72 / 92.11	90.00 / 88.84
11	92.45 / 17.91	89.08 / 23.13	93.70 / 26.87
12	89.09 / 66.08	91.90 / 72.16	92.46 / 75.43
13	88.36 / 49.57	78.27 / 56.91	92.38 / 63.91
14	81.06 / 77.42	98.44 / 98.99	99.16 / 98.99
Average	**91.30 / 65.92**	**88.61 / 69.72**	**93.01 / 70.28**
Standard Deviation	**4.37 / 7.03**	**6.65 / 9.67**	**3.63 / 7.82**

Table 4.10: BHC BB, BHC BB FS and BHC BB TS-FS average testing/independent test classification accuracies (%) by class for Botswana Hyperion data.

For the 10 experiments, the BHC BB constructed 7 different tree structures, and no tree was duplicated more than twice. Comparing the BHC BB tree structures with the BHC tree structures, again, no tree structures were identical; there were 15 different resulting tree structures for 10 partitions of a single dataset when the BHC and BHC BB algorithms were implemented. As was noticed with the BHC, acacia woodlands (class 9) branches left 6 times and right 4. When the acacia woodlands were grouped with the first right branch, it resulted in the 4 lowest overall average accuracies for the BHC BB TS-FS; the results utilizing TS-FS are very sensitive to the tree structures selected by the original BHC, indicating the importance of possibly incorporating TS into the building of the tree.

4.3.4 Feature Selection Results for Letter Recognition Data

Overall classification accuracies for the different BHC algorithms when implemented on the letter recognition dataset are: BHC, 68.82%; BHC with greedy feature selection, 62.31%; and BHC with TS-FS, 76.27%. Tabu tenure was initially set at 3 and allowed to range from 3 to 5. Because this dataset only has 16 features, the correlation check for inclusion of new features was disabled. The stopping criterion was set at 30 iterations and the maximum number of iterations to continue with no improvements was set at 10. With a total of 400 possible features for the entire tree (25 internal nodes each with 16 features), the greedy feature selection chose 143, and the TS-FS chose 336. TS-FS was implemented at every node because no metaclass was able to classify with 100% accuracy with the greedy features, and was able to increase the classification accuracy at all but one metaclass with an average accuracy increase of 4.68% per metaclass for this single tree when compared with the greedy feature selection tree. The standard deviations for the algorithms are dramatically different with the TS-FS more consistently classifying the individual letters resulting in the standard deviation being ~75% of the BHC and ~60% of the BHC FS.

The single data partition classification accuracies for each class are displayed in Table 4.11 highlighting the highest accuracy achieved per class. The BHC with TS-FS outperforms the BHC and BHC with greedy feature selection as it is able to classify a majority of the classes more consistently and often with markedly improved accuracies. The BHC resulting tree structure is displayed in Figure 4.11. BHC with feature selection yields consistently poorer results than the other two methods.

Class	BHC	BHC FS	BHC TS-FS
A	85.26	86.54	87.82
B	43.38	26.47	61.03
C	71.83	74.65	80.99
D	80.24	34.13	73.65
E	52.63	53.29	58.55
F	71.90	71.90	78.43
G	39.63	68.29	70.73
H	47.68	33.77	63.58
I	73.94	70.30	83.03
J	77.03	77.70	81.08
K	60.96	18.49	50.00
L	73.25	79.62	77.71
M	85.42	86.81	94.44
N	87.35	74.10	88.55
O	43.17	49.64	69.06
P	70.83	76.79	79.76
Q	50.60	71.43	67.86
R	54.66	57.76	56.52
S	58.39	49.69	73.91
T	80.79	65.56	78.81
U	74.40	46.43	81.55
V	88.97	77.21	86.76
W	85.61	86.33	89.93
X	73.58	67.30	76.10
Y	80.69	55.86	80.69
Z	77.22	60.13	92.41
Average	**68.82**	**62.31**	**76.27**
Standard Deviation	**15.26**	**18.83**	**11.47**

Table 4.11: BHC, BHC FS and BHC TS-FS classification accuracies (%) by letter for letter recognition data.

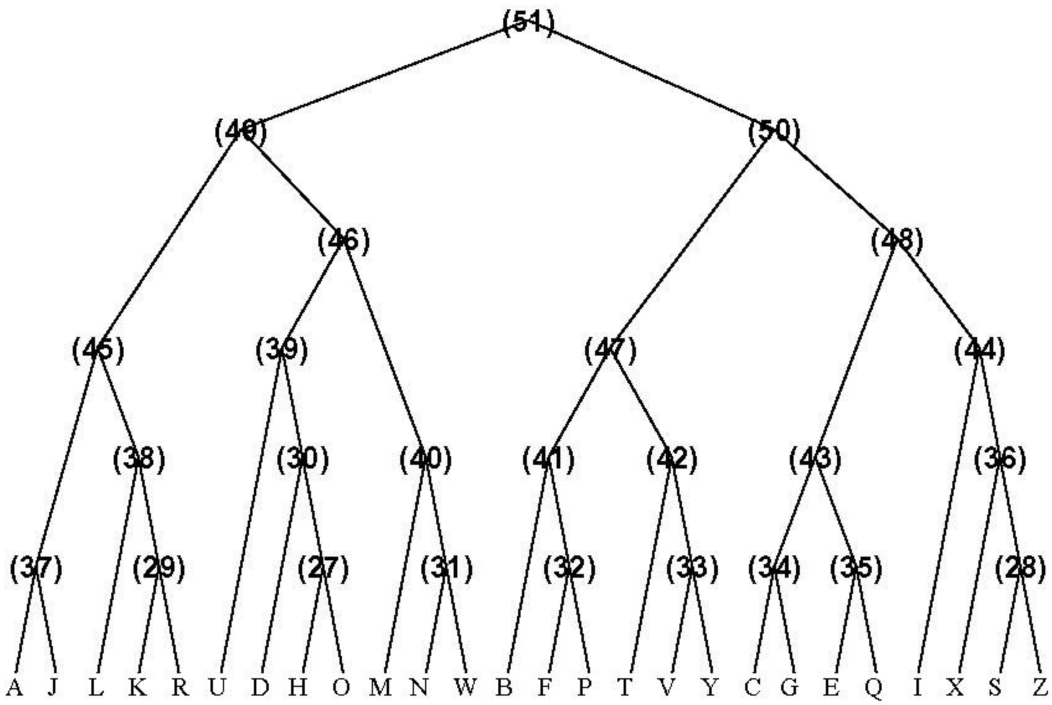

Figure 4.11: BHC class hierarchy for the single partition of the letter recognition data.

4.4 CONCLUSIONS

The algorithms average execution times are displayed in Table 4.12. The BHC and BHC FS algorithms are very fast, averaging a fraction of a minute to execute analysis of the ALI data with its small number of 9 features and slightly greater execution times for the Hyperion data with its larger set of 145 features. The TS-FS average execution times for the ALI and Hyperion data are not substantially increased in comparison. In contrast, the Hyperion BB TS-FS average execution time is noticeably increased compared to the other execution times; this is due to matrix multiplication required for the large BB matrices and the large feature vectors. Further investigation may reveal more efficient coding methods to execute the TS-FS using the BB.

Algorithm	ALI	Hyperion	Hyperion BB
BHC	0.07707	0.65949	0.36346
BHC FS	0.12904	1.06725	0.76155
BHC TS-FS	1.23707	6.04634	15.38026

Table 4.12: Average algorithm execution times in minutes for BHC, BHC FS and BHC TS-FS.

The impact of TS-FS upon the BHC classification accuracies was demonstrated to be positive. When feature selection was conducted, TS's ability to find improved feature subsets significantly improved the overall classification accuracies. TS-FS is aided by searching from a good starting solution, the set of greedy selected features, which on average, more than half are found in the TS-FS subset of features. The TS-FS algorithm also significantly increased the total number of features used by approximately one-third in most instances, but approximately doubling the number of features used in the case of the BB. These improved feature subsets are more beneficial for domain knowledge, overall classifier interpretability and possible transportability of the classifiers. The TS implementations are sensitive to the resulting class hierarchy structures; therefore, if better hierarchical trees can be constructed, the TS implementations will be enhanced and ultimately more useful for increasing classification accuracies. Using the TS-FS in the construction of the class hierarchy is one method to accomplish this goal.

Chapter 5

Building the Binary Hierarchical Classifier Tree with the Aid of Tabu Search Feature Selection

The top-down Binary Hierarchical Classifier (BHC) builds its class hierarchy iteratively starting with all of the classes in a single metaclass at the root node. Subsequently, nodes at each level of the tree are partitioned into two child nodes (subsets) until the leaves of the tree, consisting of a single class, are reached. The top-down BHC framework uses the Generalized Associative Modular Learning System (GAMLS) [43], described in Section 3.1.1 and Figure 3.2. Whereas TS Feature Selection (TS-FS) was implemented initially as a post-processor after the BHC was built, here it is incorporated into the development of the BHC hierarchical tree.

5.1 TABU SEARCH FEATURE SELECTION

The TS-FS method in this application is utilized exactly as described in Section 4.1. Now, it reduces the GAMLS input space and is instrumental in building the binary classification hierarchy. The algorithm, TS Build, is initiated with all classes in the root node at the top of the class hierarchy. The first split is accomplished using GAMLS (with all of the original features) resulting in two child nodes. As a result of this first partitioning, those features with the greatest Fisher weights are identified, and GAMLS is used to make a second binary split of the classes at the current node using only the identified highly-weighted features. This new partition becomes the current partition. Using the set of highly-weighted features as its incumbent solution, TS-FS is then performed at the root node to obtain the best subset of the total set of original features to discriminate between the two current child nodes. This resulting subset of features is passed to GAMLS which makes a third, and final, partitioning of the classes at the current node using only those features selected by TS-FS. This final partitioning becomes the binary split for the current node. Subsequent to this final partitioning, TS-FS is performed one final time using the current set of features as its incumbent solution, and the resulting feature subset becomes the feature subset used at the current node for classification. This partitioning process is then repeated at each of the current node's child nodes that contain more than a single class, moving down the tree to perform the partitioning at all multiclass nodes until only leaf nodes remain. The resulting class hierarchy is then used for classification exactly as with the BHC. The flowchart for this algorithm is presented in Figure 5.1.

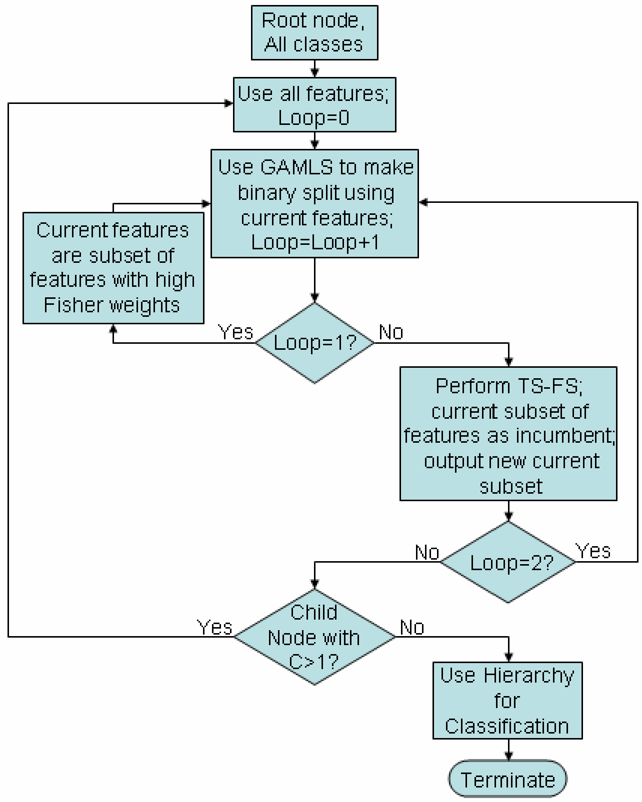

Figure 5.1: Flowchart for building the BHC tree using GAMLS and TS-FS.

5.2 Results Building the Tree using TS For ALI Data

Tabu tenure was set at 3, and the correlation check was not implemented. The maximum number of iterations was defined as 30 with an early termination criterion of 10 iterations with no improvement. Overall accuracies are shown in Table 5.1. Results from Section 4.3.1 are duplicated here for comparison.

Experiment	BHC	BHC FS	BHC TS-FS	TS Build
ALI1	88.72 / 72.82	86.38 / 69.18	88.59 / 72.74	89.15 / 73.54
ALI 2	87.20 / 71.71	85.30 / 64.92	89.71 / 72.20	89.23 / 71.38
ALI 3	86.29 / 69.69	86.64 / 68.16	89.88 / 71.20	89.93 / 72.84
ALI 4	86.60 / 70.96	85.99 / 68.62	90.01 / 71.69	90.49 / 72.84
ALI 5	88.33 / 73.33	86.34 / 67.73	90.06 / 73.36	88.41 / 67.31
ALI 6	87.64 / 73.63	85.82 / 66.97	89.32 / 72.66	90.36 / 71.14
ALI 7	86.86 / 70.96	87.68 / 67.48	90.06 / 71.90	90.32 / 70.76
ALI 8	85.82 / 75.03	84.48 / 71.82	88.28 / 75.16	87.76 / 69.65
ALI 9	87.25 / 69.61	86.73 / 71.39	89.67 / 70.64	88.37 / 71.32
ALI 10	88.98 / 72.68	87.42 / 69.37	89.75 / 72.09	90.40 / 72.57
Average	**87.37 / 72.04**	**86.28 / 68.56**	**89.53 / 72.36**	**89.44 / 71.33**
Standard Deviation	**1.05 / 1.76**	**0.95 / 2.04**	**0.62 / 1.25**	**1.00 / 1.84**

Table 5.1: BHC, BHC FS, BHC TS-FS and TS Build overall experiment classification accuracies (%) for Botswana ALI testing/independent test data.

Ten different binary tree structures were constructed with TS Build; none were identical to the BHC class hierarchies constructed for the same experiments, nor were they identical to any drawn by BHC indicating that the TS Build is having an effect on the tree-building process. The most notable differences were the TS Build placement of exposed soil (class 14) and floodplain1 (class 17). Experiments ALI3 and ALI6 resulted in different root node partitions than the BHC, affecting the subset placement of the exposed soil class. When classifying the test data, the TS Build class hierarchy outperformed the BHC and the BHC with feature selection in all experiments, and it bested the overall classification accuracies of the BHC with TS-FS in 6 of the 10 experiments. When classifying the test data, TS Build resulted in the second highest average overall classification accuracy behind the BHC with TS-FS, although by only .09%, and resulted in a slightly lower average overall classification accuracy than the BHC and the BHC with TS-FS when classifying the independent test set. In the two experiments where the TS Build class hierarchy was least effective in classifying the independent test data (experiments ALI5 and ALI8), the resulting class hierarchies had more difficulty than those developed in the other experiments classifying the acacia shrublands (class 8) and the acacia grasslands (class 9). Both experiments exhibited similar trends by repeatedly classifying acacia shrublands as acacia grasslands and acacia grasslands as dry grasses (class 12). The similarities of the class signatures for the acacia shrublands, acacia grasslands and the dry grasses are illustrated in Figures 5.2 and 5.3.

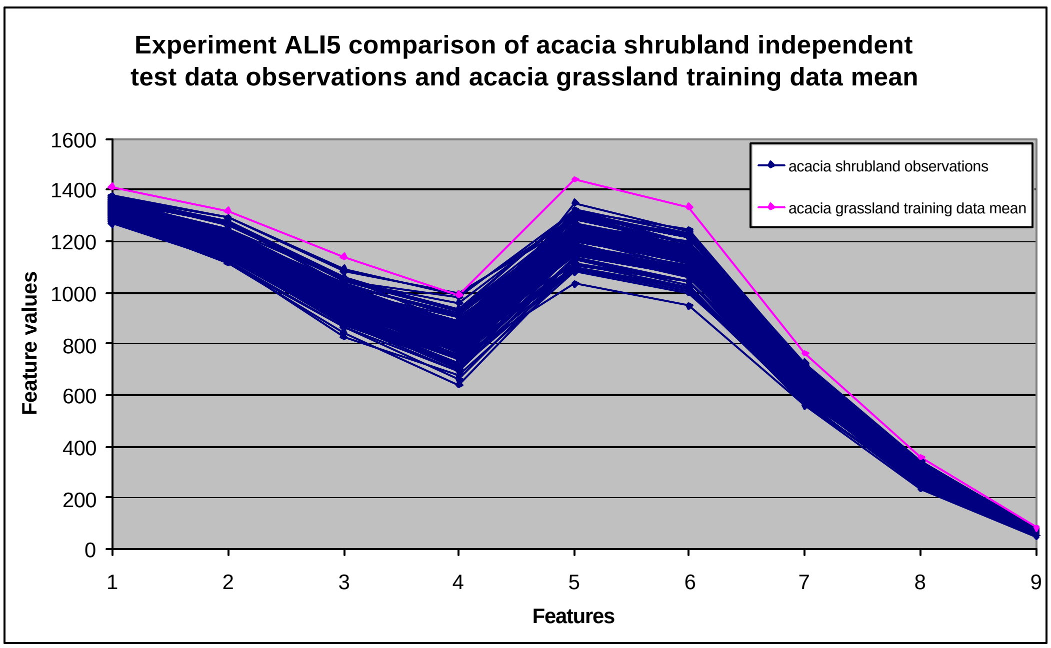

Figure 5.2: Experiment ALI5 comparison of acacia shrubland independent test data observations and acacia grassland training data mean.

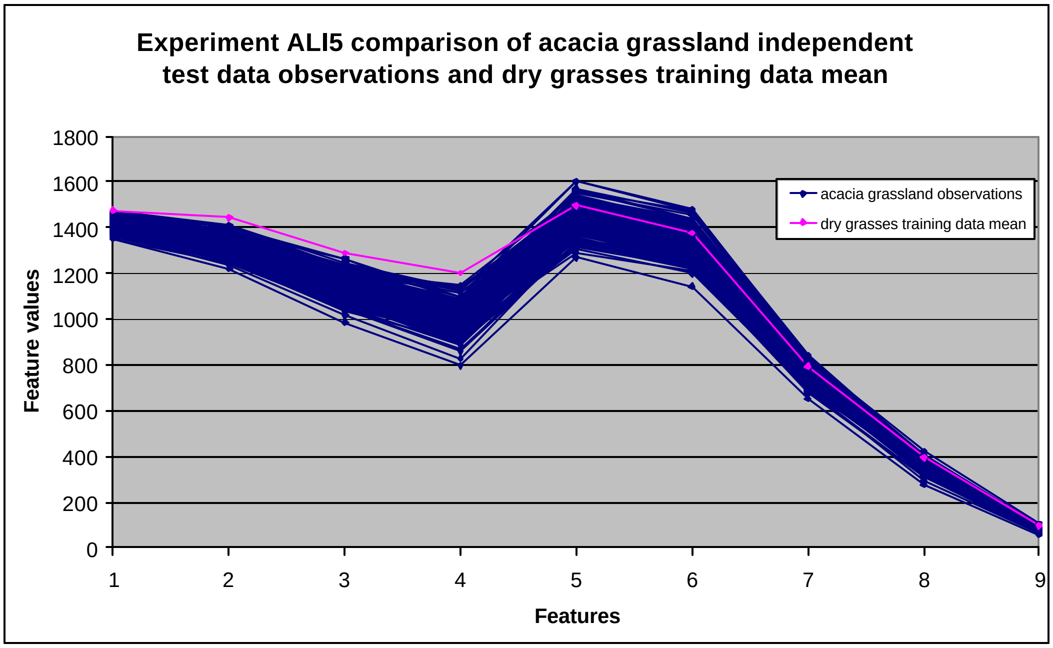

Figure 5.3: Experiment ALI5 comparison of acacia grassland independent test data observations and dry grasses training data mean.

Average overall classification accuracies are listed by class in Table 5.2, where the highest class accuracies are highlighted. Each of the algorithms exhibits strengths in the classification of individual classes with the TS algorithms resulting in the two highest average class accuracies for the test data. Although the maximum number of features can be specified, this implementation of TS-FS allowed the algorithm to seek the best cardinality of the feature subset; the TS Build trees averaged 141.8 features per tree compared to 103.7 for the BHC TS-FS. Figure 5.4 is an example of the classification of the data subset from Figure 4.3.

Class #	BHC	BHC FS	BHC TS-FS	TS Build
1	76.93 / 54.05	72.81 / 56.49	75.38 / 60.54	76.03 / 62.43
2	88.12 / 85.08	88.23 / 69.59	89.38 / 72.18	89.16 / 73.39
3	95.76 / 88.48	91.52 / 87.63	95.25 / 88.53	96.23 / 86.90
4	82.59 / 77.88	82.41 / 75.96	87.49 / 75.05	83.87 / 71.82
5	87.96 / 92.82	87.24 / 87.39	88.67 / 90.48	89.60 / 89.10
6	96.00 / 87.50	96.10 / 98.08	97.10 / 98.42	95.60 / 98.42
7	84.98 / 38.54	84.19 / 51.58	87.15 / 46.32	85.94 / 44.39
8	66.26 / 40.17	65.23 / 36.39	69.69 / 41.37	70.14 / 34.92
9	84.95 / 16.84	72.71 / 18.51	78.00 / 17.59	85.42 / 16.44
10	94.02 / 93.06	90.36 / 92.50	92.79 / 93.06	92.92 / 92.45
11	88.71 / 85.71	90.95 / 94.70	91.41 / 93.34	93.05 / 95.17
12	81.74 / 88.43	76.68 / 77.36	82.13 / 83.72	83.57 / 83.29
13	87.47 / 76.15	87.48 / 76.80	87.48 / 75.60	83.87 / 71.30
14	79.67 / 63.63	94.56 / 79.84	92.86 / 75.97	93.71 / 72.50
15	93.34 / 95.03	89.38 / 87.89	93.87 / 91.64	93.34 / 90.64
16	84.83 / 70.00	77.66 / 55.00	84.84 / 75.04	81.46 / 75.00
17	81.98 / 34.70	87.71 / 27.07	94.43 / 37.38	91.77 / 37.81
18	85.20 / 69.92	77.82 / 59.84	92.20 / 77.34	96.77 / 79.19
19	80.14 / 59.43	83.00 / 52.20	86.81 / 55.69	86.16 / 49.51
20	96.93 / 90.51	97.84 / 86.87	96.92 / 88.82	98.41 / 88.67
21	82.67 / 82.22	96.94 / 90.15	89.48 / 86.26	86.81 / 85.27
22	99.28 / 65.13	98.80 / 61.91	98.96 / 54.47	98.80 / 61.98
23	98.31 / 99.20	89.35 / 64.63	97.44 / 90.37	95.63 / 84.88
Average	**86.86 / 71.93**	**86.04 / 69.06**	**89.12 / 72.57**	**89.05 / 71.54**
Standard Deviation	**4.27 / 5.27**	**5.33 / 8.34**	**3.86 / 5.36**	**4.37 / 7.44**

Table 5.2: BHC, BHC FS, BHC TS-FS and TS Build average testing/independent test classification accuracies (%) by class for Botswana ALI data.

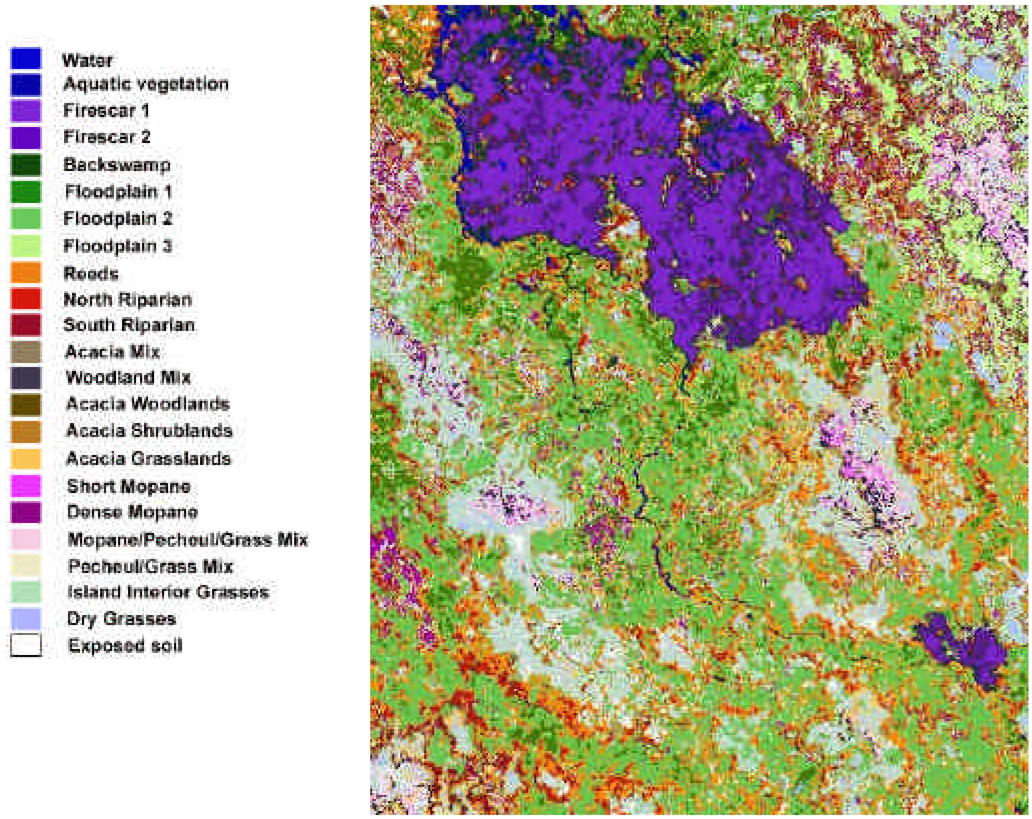

Figure 5.4: Example of a classified subset using the TS Build classifier (experiment ALI3: test set accuracy 89.93%, independent test set accuracy 72.84%).

5.3 RESULTS BUILDING THE TREE USING TS FOR HYPERION DATA USING ORIGINAL FEATURES

Tabu tenure was initialized at 5 within the allowable range of 3 to 10. Due to the large number of candidate features, the correlation check for feature inclusion was used. The number of iterations was set at 30, but execution was halted if 10 iterations were performed without improvement. Results from Section 4.3.2 are included for comparison (see Table 5.3).

Experiment	BHC	BHC with FS	BHC TS-FS	TS Build
HYP11	92.71 / 61.23	89.13 / 66.32	93.51 / 60.87	92.22 / 68.20
HYP12	88.76 / 56.87	86.53 / 63.44	89.75 / 67.52	92.16 / 65.08
HYP13	88.08 / 69.36	90.30 / 69.36	92.77 / 64.08	91.48 / 68.92
HYP14	91.91 / 60.07	86.72 / 62.88	92.59 / 58.35	93.02 / 62.68
HYP15	89.99 / 58.55	85.98 / 59.47	90.67 / 62.07	91.79 / 73.93
HYP16	91.85 / 60.63	87.34 / 64.04	92.16 / 66.92	92.40 / 71.89
HYP17	91.60 / 59.43	86.29 / 62.15	92.90 / 61.35	91.85 / 63.88
HYP18	91.91 / 60.59	90.80 / 68.88	92.34 / 63.88	92.84 / 64.60
HYP19	89.19 / 63.52	85.05 / 68.92	89.13 / 63.40	91.48 / 69.16
HYP20	90.67 / 62.07	85.55 / 63.04	91.54 / 62.15	91.17 / 60.95
Average	**90.67 / 61.23**	**87.37 / 64.85**	**91.74 / 63.06**	**92.04 / 66.93**
Standard Deviation	**1.58 / 3.39**	**2.01 / 3.36**	**1.44 / 2.75**	**0.60 / 4.17**

Table 5.3: BHC, BHC FS, BHC TS-FS and TS Build overall experiment classification accuracies (%) for Botswana Hyperion testing/independent test data.

For this particular dataset, the initial partitioning of the root node proves to be a very important factor; in 7 of the 10 experiments, TS Build partitioned the root node differently than the BHC. As was noted in Section 4.3.2, in 8/10 experiments, the BHC grouped riparian (class 6) and acacia woodlands (class 9) together at the bottom of the hierarchy, but in experiments HYP12 and HYP19, these classes were in different subsets at the root node partition. In contrast, TS Build grouped riparian and acacia woodlands together at the bottom of the class hierarchy in all of the experiments. Using the test data, TS Build outperformed: the BHC in 9/10 experiments, the BHC with feature selection in all of the experiments, and the BHC with TS-FS in 6/10 of the experiments; in addition, TS Build yielded significantly reduced standard deviations of the accuracies for the test set relative to the other algorithms (~40% of that of the BHC, ~30% of that of the BHC FS and ~40% of that of the BHC TS-FS). When classifying the independent test data, TS Build resulted in higher accuracies in 5/10 experiments and the highest overall average accuracy for all of the algorithms, but resulted in the highest standard deviation of the accuracies of the algorithms. This appears to imply that the TS Build may be overtraining, but further investigation is required. Average class accuracies are listed in Table 5.4 (where the results from Table 4.8 are duplicated for comparison and the greatest are highlighted); the TS Build resulted in the highest overall average accuracy for both the test and independent test data. It is noteworthy that significant improvement was achieved in classification of both hippo grass (class 2) and acacia grasslands (class 11) in the independent test set. There was also substantial improvement in the classification accuracy of mixed mopane (class 13) using TS Build. The TS Build tree structures averaged 145.1 features per tree which compares to 62.5 for TS-FS. With the present settings, the TS Build does not reduce the input space as dramatically as the other algorithms, and the class standard deviations are somewhat comparable.

Class #	BHC	BHC FS	BHC TS-FS	TS Build
1	100.00 / 99.92	99.41 / 98.81	99.41 / 99.53	99.12 / 97.70
2	87.60 / 15.68	96.80 / 51.29	97.60 / 40.12	94.60 / 52.41
3	95.12 / 81.39	88.16 / 51.58	96.08 / 53.93	93.60 / 48.67
4	96.92 / 72.00	96.34 / 81.88	96.37 / 66.61	94.03 / 75.88
5	86.03 / 48.93	72.25 / 43.39	84.71 / 58.69	80.98 / 55.95
6	80.09 / 60.76	67.69 / 56.87	83.43 / 63.56	83.74 / 66.30
7	98.96 / 82.27	93.55 / 88.58	97.20 / 88.01	98.97 / 88.01
8	95.05 / 84.90	93.75 / 83.06	94.35 / 78.98	91.38 / 69.21
9	88.07 / 69.27	87.01 / 69.67	86.56 / 64.50	88.61 / 89.37
10	90.86 / 86.74	80.98 / 83.32	87.42 / 85.74	90.55 / 36.87
11	93.02 / 18.49	90.31 / 30.61	90.45 / 26.68	93.62 / 78.11
12	87.66 / 66.67	91.34 / 72.75	92.68 / 76.80	94.57 / 56.61
13	84.40 / 57.86	84.34 / 61.20	90.58 / 49.53	93.72 / 95.96
14	76.16 / 77.98	98.30 / 99.89	98.73 / 99.78	98.51 / 82.29
Average	**90.00 / 65.92**	**88.59 / 69.49**	**92.54 / 68.03**	**92.57 / 70.95**
Standard Deviation	**4.82 / 8.48**	**7.39 / 9.51**	**3.77 / 8.54**	**3.40 / 8.74**

Table 5.4: BHC, BHC FS, BHC TS-FS and TS Build average testing/independent test classification accuracies (%) by class for Botswana Hyperion data.

5.4 RESULTS BUILDING THE TREE USING TS AND BEST BASES FOR HYPERION DATA

In this implementation, new BB features are computed for the current node, and TS-FS is performed on these new BB features. Otherwise, the algorithm progresses as previously described. Parameters were defined as: tabu tenure, 3; maximum tabu tenure, 10; minimum tabu tenure, 3; stopping criterion, 30 iterations; and terminate after 10 iterations with no improvement. In 8 of the 10 experiments, TS Build partitioned the root node differently than the BHC. The two classes most affected were the acacia woodlands (class 9) and the exposed soil (class 14). While the classification accuracy of the acacia woodlands is not significantly impacted by TS Build, the classification accuracy of the exposed soil class is noticeably impacted with an increased average accuracy of 98.30% over the 81.06% average accuracy of the BHC (see Table 5.6). Using the test data, the TS Build classifier resulted in higher accuracies in 9/10 experiments than the BHC BB, in 10/10 experiments over the BHC BB with feature selection (indicating that the TS-FS is outperforming the greedy FS when using BB), and in 7/10 experiments over the BHC BB with TS-FS (see Table 5.5, results of Table 4.9 are duplicated for comparison). The average class accuracies are recorded in Table 5.6 (results of Table 4.10 are duplicated for comparison) where the highest average accuracy per class is highlighted. Consistent with earlier results, the BB algorithms yielded lower standard deviations of accuracies than when the original feature set was used. Further, the TS Build BB significantly reduced the standard deviations for both the test and independent test data relative to the other BB algorithms. For example, the TS Build BB standard deviation is ~70% of that of the BHC for the test data and ~80% of that of the BHC for the independent test data. Thus, the TS Build BB method yielded the most stable classifier. This result is duplicated in the class standard deviations in Table 5.6. The TS Build class hierarchies averaged 147.7 features per hierarchy compared to 91.70 chosen by BB TS-FS.

Experiment	BHC BB	BHC BB FS	BHC BB TS-FS	TS Build BB
HYP11	89.38 / 56.59	88.94 / 63.84	91.60 / 65.76	92.90 / 64.64
HYP12	91.54 / 61.43	86.66 / 68.00	90.92 / 66.80	92.22 / 66.28
HYP13	91.17 / 58.55	85.36 / 64.84	93.27 / 69.92	93.27 / 60.39
HYP14	92.16 / 61.43	87.77 / 61.51	93.02 / 60.75	92.09 / 67.20
HYP15	92.28 / 59.99	86.35 / 61.71	93.70 / 64.80	94.01 / 64.08
HYP16	91.54 / 60.15	88.14 / 65.48	93.39 / 62.80	93.33 / 64.12
HYP17	91.72 / 61.19	89.31 / 66.84	91.41 / 66.88	92.46 / 65.04
HYP18	92.46 / 61.83	86.41 / 64.60	93.21 / 65.84	93.39 / 63.96
HYP19	90.30 / 66.08	89.19 / 64.80	90.80 / 70.56	91.85 / 66.60
HYP20	92.22 / 62.07	85.55 / 61.67	91.97 / 62.60	92.84 / 66.48
Average	**91.48 / 60.93**	**87.37 / 64.33**	**92.33 / 65.67**	**92.84 / 64.88**
Standard Deviation	**0.98 / 2.48**	**1.49 / 2.20**	**1.10 / 3.12**	**0.68 / 1.98**

Table 5.5: BHC BB, BHC BB FS, BHC BB TS-FS and TS Build BB overall experiment classification accuracies (%) for Botswana Hyperion testing/independent test data.

Class #	BHC BB	BHC BB FS	BHC BB TS-FS	TS Build BB
1	100.00 / 100.00	99.40 / 98.49	99.78 / 98.49	99.56 / 52.22
2	94.00 / 15.06	95.60 / 52.90	96.00 / 49.44	96.60 / 62.67
3	95.36 / 86.65	89.12 / 52.98	94.40 / 54.94	95.92 / 99.21
4	96.92 / 74.30	94.77 / 82.12	95.44 / 72.37	94.67 / 50.25
5	89.19 / 50.06	75.15 / 47.26	88.97 / 55.83	88.00 / 59.70
6	80.16 / 60.05	60.81 / 51.09	82.99 / 66.63	83.27 / 64.74
7	98.80 / 80.57	91.77 / 88.01	95.81 / 88.98	98.58 / 87.44
8	96.52 / 87.64	96.41 / 79.30	96.21 / 78.60	96.23 / 77.01
9	86.18 / 70.53	89.09 / 80.66	84.83 / 64.64	85.99 / 64.50
10	90.06 / 87.00	90.72 / 92.11	90.00 / 88.84	93.00 / 88.11
11	92.45 / 17.91	89.08 / 23.13	93.70 / 26.87	94.28 / 33.27
12	89.09 / 66.08	91.90 / 72.16	92.46 / 75.43	93.34 / 73.20
13	88.36 / 49.57	78.27 / 56.91	92.38 / 63.91	90.67 / 57.30
14	81.06/ 77.42	98.44 / 98.99	99.16 / 98.99	98.30 / 97.30
Average	**91.30 / 65.92**	**88.61 / 69.72**	**93.01 / 70.28**	**93.46 / 69.06**
Standard Deviation	**4.37 / 7.03**	**6.65 / 9.67**	**3.63 / 7.82**	**2.93 / 7.25**

Table 5.6: BHC BB, BHC BB FS, BHC BB TS-FS and TS Build BB average testing/independent test classification accuracies (%) by class for Botswana Hyperion data.

5.5 RESULTS BUILDING THE TREE USING TS FOR LETTER RECOGNITION DATA

TS Build overall classification accuracy for the letter recognition dataset is 76.49% which is .22% greater than the highest accuracy reported in Section 4.3.4. Tabu tenure was set at 5 and allowed to range from 3 to 8. The correlation check for inclusion of new features was disabled as it is not appropriate for this data set, the stopping criterion was 30 iterations, and the maximum number of iterations to continue with no improvements was 10. The class hierarchy was constructed using a total of 323 features. When compared to the BHC class hierarchy, the root node partition is identical to the BHC, but the overall class hierarchy differs. For example, TS Build brings the letter U closer to the letters M, N and W and also B closer to S and Z (see Figure 5.5).

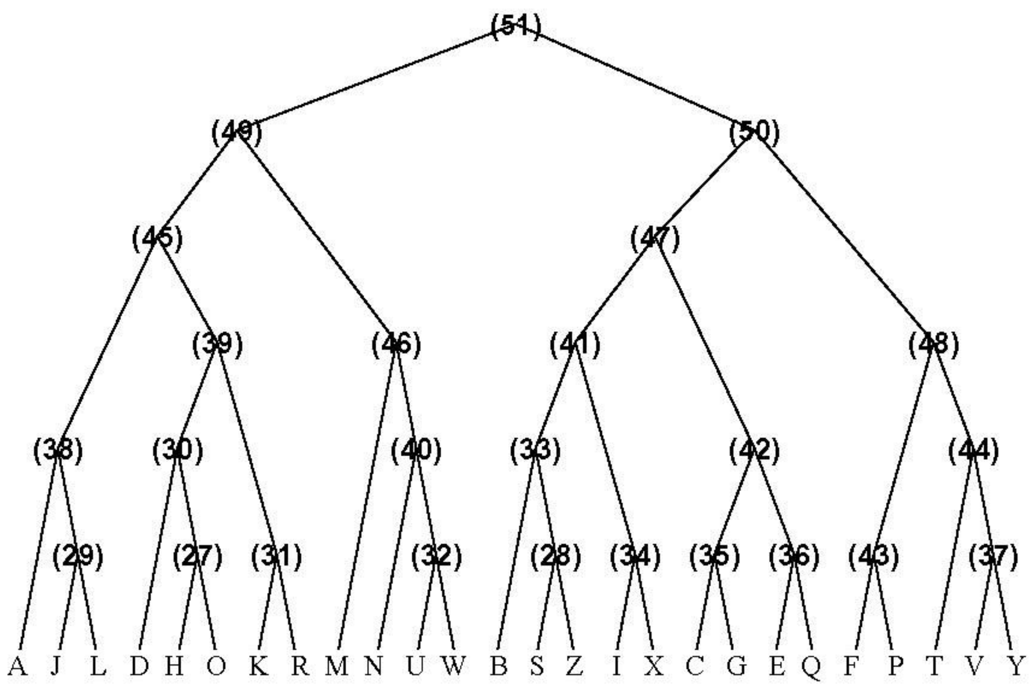

Figure 5.5: Building the BHC tree using GAMLS and TS-FS for the letter recognition data.

The average class accuracies are shown in Table 5.7 where the results from Section 4.3.4 are included for comparison, and the greatest accuracies achieved from the various methods are highlighted. The TS algorithms, again, were able to classify the individual classes with greater consistency (and with smaller standard deviations) than the BHC or BHC with feature selection.

Class	BHC	BHC FS	BHC TS-FS	TS Build
A	85.26	86.54	87.82	90.38
B	43.38	26.47	61.03	77.94
C	71.83	74.65	80.99	77.46
D	80.24	34.13	73.65	79.64
E	52.63	53.29	58.55	52.63
F	71.90	71.90	78.43	78.43
G	39.63	68.29	70.73	73.17
H	47.68	33.77	63.58	46.36
I	73.94	70.30	83.03	85.45
J	77.03	77.70	81.08	80.41
K	60.96	18.49	50.00	63.01
L	73.25	79.62	77.71	80.89
M	85.42	86.81	94.44	93.75
N	87.35	74.10	88.55	84.94
O	43.17	49.64	69.06	74.82
P	70.83	76.79	79.76	76.79
Q	50.60	71.43	67.86	69.64
R	54.66	57.76	56.52	59.63
S	58.39	49.69	73.91	68.94
T	80.79	65.56	78.81	77.48
U	74.40	46.43	81.55	76.79
V	88.97	77.21	86.76	85.29
W	85.61	86.33	89.93	89.93
X	73.58	67.30	76.10	72.33
Y	80.69	55.86	80.69	81.38
Z	77.22	60.13	92.41	91.14
Average	**68.82**	**62.31**	**76.27**	**76.49**
Standard Deviation	**15.26**	**18.83**	**11.47**	**11.44**

Table 5.7: BHC, BHC FS, BHC TS-FS and TS Build classification accuracies (%) by letter for letter recognition data.

It is interesting to note that there are some dramatic differences. The use of feature selection never degrades the performance dramatically relative to the original set. However, it can improve results dramatically. Further, the problems with the greedy algorithm are clear – it has dramatically degraded results for D, K, U, and Y. E is difficult for all of the algorithms to classify, but benefits from the TS-FS. The letters B, O and Z benefit from TS relative to using all of the features and the greedy feature selection while classification of the letters B and O is significantly improved when using the TS Build. Both TS based algorithms have substantially reduced standard deviations of the classification accuracies.

5.6 CONCLUSIONS

Algorithm average execution times are displayed in Table 5.8 for comparison. The TS Build algorithm has increased execution times as related to the other algorithms due to its process: GAMLS is executed three times and TS-FS is executed twice for each node. Also, more candidate features lead to increased execution times as evidenced by comparing the ALI and Hyperion average execution times. As was noted in Section 4.4, the Hyperion BB TS Build suffers from the same matrix multiplication issues associated with the calculation of the best basis as the BB TS-FS, and this is reflected in the increased average algorithm execution time.

Algorithm	ALI	Hyperion	Hyperion BB
BHC	0.07707	0.65949	0.36346
BHC FS	0.12904	1.06725	0.76155
BHC TS-FS	1.23707	6.04634	15.38026
TS Build	2.68853	10.76789	65.55054

Table 5.8: Average algorithm execution times in minutes for BHC, BHC FS, BHC TS-FS and TS Build.

The TS feature selection was used within the BHC algorithm to reduce the feature space in an effort to allow GAMLS to make a better partition at every multiclass node. In addition, parameter estimates used for performing the classification may benefit from TS Build. Its impact was generally positive: classification accuracies of many classes were improved, and the standard deviations of accuracies were consistently reduced. Once constructed, the class hierarchy is static and has no opportunity for recourse. The possibility for recourse arises by allowing the rearrangement of the nodes (classes) within the class hierarchy structure. In order to investigate this, a new algorithm was developed. This new method, referred to as the Tabu Search Tree Rearrangement Algorithm (TSTRA), is discussed in the following chapter.

Chapter 6

Binary Hierarchical Classifier Tree Rearrangement Using Tabu Search

Once the BHC class hierarchy is constructed, the original BHC framework does not provide any possibility of recourse. No recovery is possible if a bad decision was made in the partitioning phase of any of the metaclasses. The tree rearrangement algorithm described in this chapter performs as a post-processor that uses the BHC tree output as its incumbent solution.

6.1 TABU SEARCH TREE REARRANGEMENT

The tabu search tree arrangement algorithm (TSTRA) uses the same classifier as the BHC and the same training data that were used to construct the original BHC class hierarchy. Using the BHC tree as the TSTRA initial solution, the TSTRA move neighborhood is defined as any neighboring tree resulting from an adjacent insertion of any leaf node to every other nonadjacent leaf node. For example, a BHC tree for a problem with five classes is pictured in Figure 6.1(a). Figures 6.1(b), (c), (d) show the alternate trees when Class 1 is inserted in its other possible positions. This complete neighborhood would include the results of all insertions of classes 2, 3, 4, and 5.

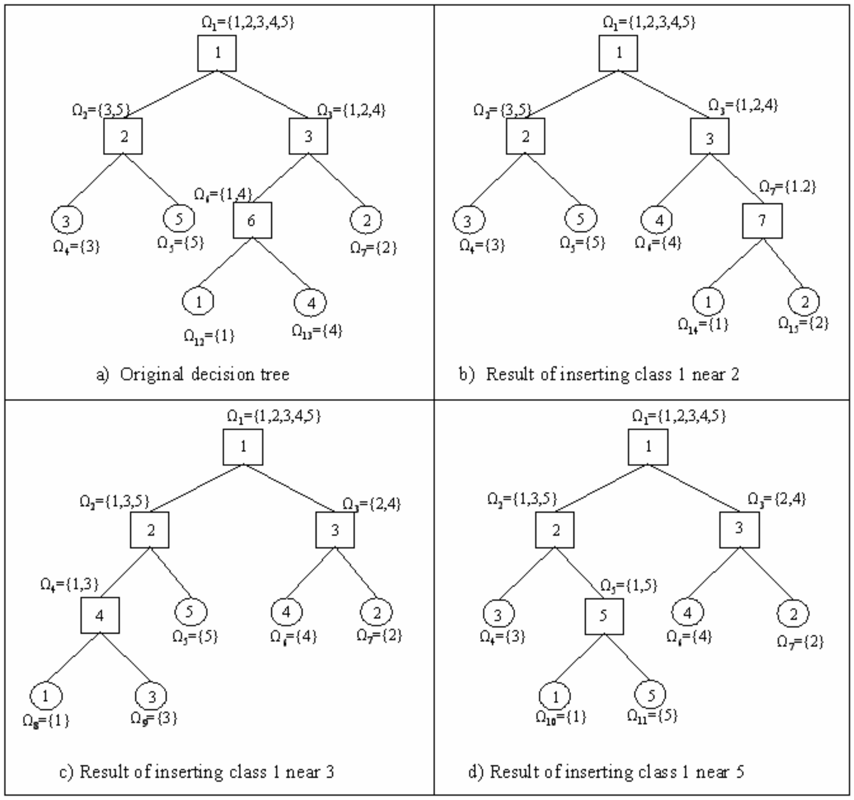

Figure 6.1: Example of neighboring tree structures.

In this application of TS methodology, the tabu list begins as a column vector of zeros with a row for each class. Once a class is selected for movement within the tree structure, the class is marked as tabu and the tabu list records the iteration number of the class into the list. The class cannot be moved again within the tabu tenure number of iterations unless moving the class results in finding a classification accuracy that is better than any found thus far, overruling the tabu status. As in the previous implementations of TS, the tabu tenure is adaptive between a user-defined maximum and minimum. Given an incumbent tree, the best non-tabu move within the move neighborhood is selected (unless a tabu status is overruled) for that iteration. The tree with the best accuracy found is maintained and updated (nodes merged, metaclass statistics and Fisher projections computed) as appropriate when a tree with an improved accuracy is found. Iterations continue until a user-specified number of iterations has been completed or no improving tree structures have been found within a specified number of iterations. Upon termination, the algorithm returns the best tree structure found for classification. Although a user-specified option exists that allows the tree-rearrangements to be level-restricted if prior knowledge of the problem suggests such a limitation would be beneficial, it was not implemented in this study. In addition, if implemented, this level restriction can be adaptive by adjusting the level of the tree considered for change depending on the ability, or inability, of the TSTRA to find improving solutions. Restrictions which prohibit changes to major partitions of the classes (for example, at the root node) intensify the search in the current solution space while movements allowing such effects diversify the search.

6.2 TSTRA RESULTS FOR ALI DATA

The Botswana ALI data were analyzed with the TSTRA. Because improving moves were consistently found during the early iterations with no improvements in the later iterations, the maximum number of iterations was limited to 20. Tabu tenure was maintained at 3, and execution was halted if 10 iterations were performed and no improving solution was found.

The TSTRA considers the current class hierarchy output by the BHC, and using the same training data and classifier, rearranges the tree structure to find better classification accuracies. In every experiment, the TSTRA was able to find improved class hierarchy structures and increased classification accuracies using the training data, and these improved class hierarchies, while not guaranteed to do so, improved or maintained the classification accuracies for the test and independent test data, as well. When compared to the BHC, the TSTRA averaged a 1.60% increase in the classification accuracies per experiment using the test data and a 2.15% average increase per experiment using the independent test data (see Table 6.1). In 8 of the 10 experiments, the TSTRA maintained the original partition of the root node; in the 2 experiments where the original partition is altered, the aquatic vegetation (class 21), which is usually grouped with classes 1, 2, 7 and 15, is moved to the other subset and grouped with the backswamp (class 16).

TS-FS (as described in Section 4.1) was performed as a post processing operation on the TSTRA resulting class hierarchies; while 7 of the 10 experiment overall accuracies were improved by the TS-FS for the test data (with an average increase of 2.43% per experiment), only one was improved using the independent test data. An average of 111.8 features per tree were selected by TS-FS. The standard deviation of accuracies is improved for both the TSTRA and the TSTRA TS-FS results, relative to the original BHC.

Experiment	BHC	TSTRA	TSTRA TS-FS
ALI1	88.72 / 72.82	89.23 / 74.49	89.06 / 70.39
ALI 2	87.20 / 71.71	90.27 / 74.92	89.41 / 72.17
ALI 3	86.29 / 69.69	88.85 / 71.47	90.14 / 72.36
ALI 4	86.60 / 70.96	87.76 / 74.16	91.53 / 69.69
ALI 5	88.33 / 73.33	88.80 / 75.40	90.75 / 71.01
ALI 6	87.64 / 73.63	89.23 / 75.13	89.75 / 69.69
ALI 7	86.86 / 70.96	87.81 / 74.65	88.93 / 69.56
ALI 8	85.82 / 75.03	90.10 / 75.03	89.23 / 71.15
ALI 9	87.25 / 69.61	88.54 / 73.09	89.02 / 71.07
ALI 10	88.98 / 72.68	89.06 / 73.60	90.14 / 71.34
Average	**87.37 / 72.04**	**88.97 / 74.19**	**89.80 / 70.84**
Standard Deviation	**1.05 / 1.76**	**0.83 / 1.20**	**0.85 / 1.00**

Table 6.1: BHC, TSTRA and TSTRA TS-FS overall experiment classification accuracies (%) for Botswana ALI testing/independent test data.

A majority of the individual ALI classes benefited from the TSTRA (see Table 6.2); 17 of the 23 classes increased in accuracies averaging a 1.80% increase in the individual class accuracies for the test data and a 2.24% increase for the independent test data. The south riparian (class 2) class accuracy was markedly decreased by the TSTRA. On 8 of the 10 resulting TSTRA trees, south riparian is grouped with acacia woodlands (class 7) whereas on the BHC trees, it is only found grouped with acacia woodlands on a single tree. The goal of the TSTRA is to find trees with increased overall classification accuracies; in its current implementation, it is not constrained from decreasing some class accuracies in its quest to do so, as is the case with the south riparian (class 2). At the same time, it is able to substantially improve the classification accuracies of some classes, for example, the island interior (class 13). Figure 6.2 is an example of the classification of the data subset from Figure 4.3.

Class #	Class	BHC	TSTRA	TSTRA TS-FS
1	north riparian	76.93 / 54.05	76.92 / 67.84	75.38 / 62.57
2	south riparian	88.12 / 85.08	87.09 / 72.95	91.98 / 74.01
3	short mopane	95.76 / 88.48	97.84 / 88.76	98.16 / 87.40
4	mopane (dense)	82.59 / 77.88	84.69 / 81.01	85.88 / 77.88
5	acacia mix	87.96 / 92.82	89.38 / 93.83	88.36 / 90.80
6	woodland mix	96.00 / 87.50	96.40 / 97.34	97.60 / 97.92
7	acacia woodlands	84.98 / 38.54	83.49 / 48.71	87.15 / 46.20
8	acacia shrublands	66.26 / 40.17	72.98 / 42.46	68.94 / 38.69
9	acacia grasslands	84.95 / 16.84	84.47 / 18.05	76.83 / 16.72
10	mopane/pechuel/grass mix	94.02 / 93.06	93.89 / 95.72	93.54 / 92.96
11	grass/pechuel mix	88.71 / 85.71	88.82 / 94.08	91.65 / 89.73
12	dry grasses	81.74 / 88.43	81.81 / 87.29	78.72 / 78.86
13	island interior	87.47 / 76.15	95.08 / 84.35	91.22 / 79.35
14	exposed soil	79.67 / 63.63	81.88 / 67.58	92.18 / 71.94
15	reeds1	93.34 / 95.03	95.31 / 95.97	93.15 / 90.64
16	backswamp	84.83 / 70.00	85.27 / 72.57	84.12 / 77.92
17	floodplain1	81.98 / 34.70	88.99 / 39.76	93.86 / 22.32
18	floodplain2	85.20 / 69.92	88.65 / 74.92	94.06 / 75.00
19	floodplain3	80.14 / 59.43	81.28 / 57.07	88.31 / 46.26
20	water	96.93 / 90.51	98.26 / 92.31	98.59 / 90.36
21	aquatic vegetation	82.67 / 82.22	86.12 / 68.47	90.55 / 82.37
22	firescar1	99.28 / 65.13	99.76 / 67.76	98.64 / 51.65
23	firescar2	98.31 / 99.20	98.08 / 97.16	95.90 / 86.30
	Average	**86.86 / 71.93**	**88.54 / 74.17**	**89.34 / 70.78**
	Standard Deviation	**4.27 / 5.27**	**3.58 / 5.06**	**4.17 / 6.22**

Table 6.2: BHC, TSTRA and TSTRA TS-FS average testing/independent test classification accuracies (%) by class for Botswana ALI data.

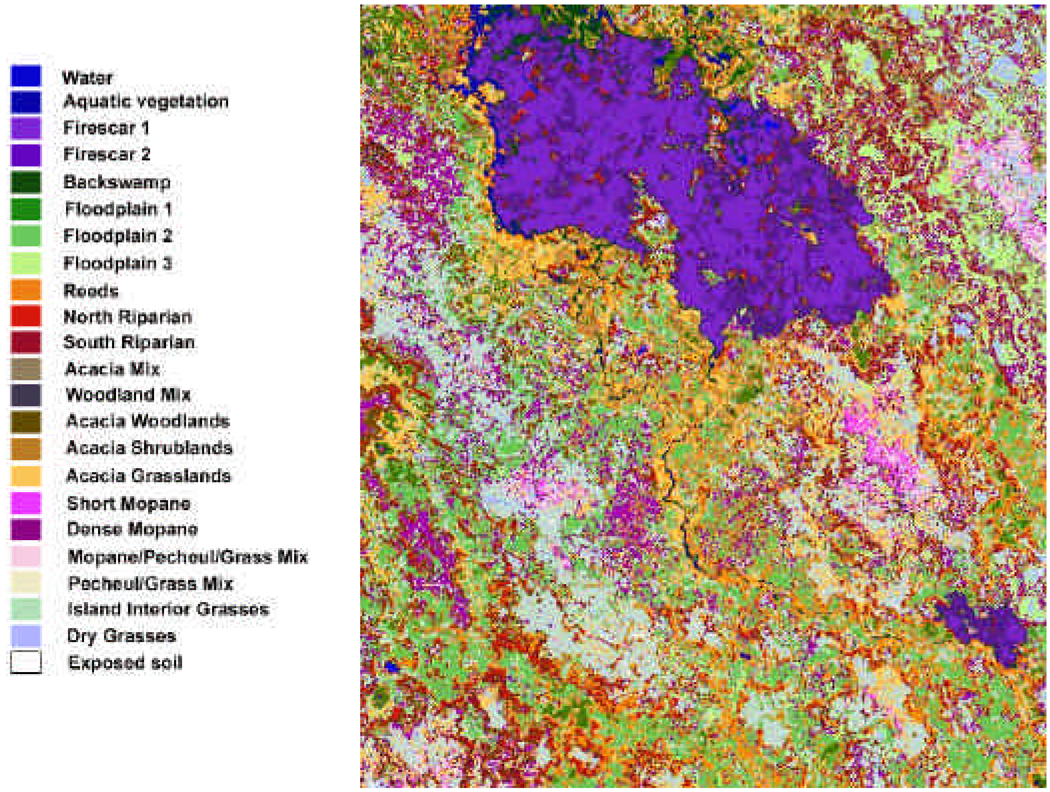

Figure 6.2: Example of a classified subset using the TSTRA classifier (experiment ALI7: test set accuracy 87.81%, independent test set accuracy 74.65%).

6.3 TSTRA RESULTS FOR HYPERION DATA

Due to the implementation of BB at each individual metaclass, whenever a single class is moved during the TSTRA, a new BB must be found at each metaclass that is affected by the move of the class. Because of the time involved for doing so, the TSTRA was not implemented for the Hyperion data using BB.

The parameter settings for the analysis of the Hyperion data were: tabu tenure 3, number of iterations 20 and halt execution after 10 iterations with no improvement. For the test data, the TSTRA improved the overall accuracies of the BHC in 8 of the 10 experiments, maintained the same class hierarchy in experiment HYP11 (it was unable to find a better hierarchical tree when classifying the training data), and decreased the overall accuracy in experiment HYP14. The TSTRA also decreased the standard deviation of the classification accuracies relative to the BHC: for the testing data, the TSTRA standard deviation was ~35% of that of the BHC, and for the independent test data, the TSTRA standard deviation was ~45% of that of the BHC. In terms of classification accuracies, addition of the TS-FS to the resulting TSTRA structures was advantageous for this data set, especially when applied to the independent test data (see Table 6.3), resulting in the highest average overall accuracy of 64.30% and an average increase of 3.06% per experiment relative to the BHC. The standard deviation of the accuracies for TSTRA TS-FS was smaller than that of the BHC, but increased relative to the TSTRA. When executed, the TS-FS chose an average of 65.6 features per tree.

Experiment	**BHC**	**TSTRA**	**TSTRA TS-FS**
HYP11	92.71 / 61.23	92.71 / 61.23	93.51 / 60.87
HYP12	88.76 / 56.87	92.53 / 62.07	92.96 / 66.00
HYP13	88.08 / 69.36	91.54 / 62.11	91.35 / 63.64
HYP14	91.91 / 60.07	91.66 / 63.64	91.97 / 61.83
HYP15	89.99 / 58.55	92.16 / 62.96	91.85 / 63.56
HYP16	91.85 / 60.63	92.90 / 61.19	90.67 / 68.64
HYP17	91.60 / 59.43	91.85 / 58.31	91.85 / 58.51
HYP18	91.91 / 60.59	93.14 / 62.76	92.77 / 61.87
HYP19	89.19 / 63.52	91.91 / 62.88	93.21 / 71.73
HYP20	90.67 / 62.07	92.09 / 63.52	91.91 / 66.32
Average	**90.67 / 61.23**	**92.25 / 62.07**	**92.21 / 64.30**
Standard Deviation	**1.58 / 3.39**	**0.54 / 1.57**	**0.89 / 3.93**

Table 6.3: BHC, TSTRA and TSTRA TS-FS overall experiment classification accuracies (%) for Botswana Hyperion testing/independent test data.

As mentioned in Section 4.3.2 and Section 5.3, in experiments HYP12 and HYP19, classes 6 and 9 were not grouped together and resulted in some of the lowest overall classification accuracies per experiment. The TSTRA results for these two experiments improved the classification accuracies for the test data, partly (as this was not the only change) by changing the partition at the root node and grouping classes 6 and 9 together at the bottom of the tree (only experiment HYP12 was improved for the independent test data). Although these changes aided the overall classification, they did not result in significant increases in individual accuracies for the classes (see Table 6.4). In addition to classes 6 and 9, class 14 (exposed soil) also changed subsets at the partition of the root node in experiments HYP13 and HYP17; as was noted in Section 4.3.2, feature selection, again, significantly increased the class 14 accuracy (see Table 6.4).

Class #	Class	BHC	TSTRA	TSTRA TS-FS
1	water	100.00 / 99.92	99.78 / 99.60	99.41 / 99.13
2	hippo grass	87.60 / 15.68	97.40 / 28.15	97.40 / 37.66
3	floodplain grasses1	95.12 / 81.39	96.00 / 87.28	95.44 / 59.94
4	floodplain grasses2	96.92 / 72.00	96.73 / 63.45	95.25 / 61.94
5	reeds1	86.03 / 48.93	88.35 / 52.38	86.35 / 59.17
6	riparian	80.09 / 60.76	82.60 / 60.29	84.85 / 63.18
7	firescar2	98.96 / 82.27	98.42 / 79.04	96.26 / 84.43
8	island interior	95.05 / 84.90	97.81 / 84.59	95.34 / 75.61
9	acacia woodlands	88.07 / 69.27	89.29 / 69.41	83.44 / 58.68
10	acacia shrublands	90.86 / 86.74	92.55 / 89.21	91.45 / 91.00
11	acacia grasslands	93.02 / 18.49	94.53 / 16.34	92.82 / 38.41
12	short mopane	87.66 / 66.67	87.11 / 65.95	92.67 / 77.06
13	mixed mopane	84.40 / 57.86	85.00 / 62.06	92.17 / 50.73
14	exposed soils	76.16 / 77.98	86.61 / 79.44	98.72 / 98.43
	Average	**90.00 / 65.92**	**92.30 / 66.94**	**92.97 / 68.24**
	Standard Deviation	**4.82 / 8.48**	**3.41 / 6.60**	**3.28 / 8.67**

Table 6.4: BHC, TSTRA and TSTRA TS-FS average testing/independent test classification accuracies (%) by class for Botswana Hyperion data.

6.4 TSTRA RESULTS FOR LETTER RECOGNITION DATA

The TSTRA was implemented on the BHC output for the letter recognition data with a dynamic tabu tenure of 3 to 5 (originally set at 3), a stopping criterion of 30 (or 10 iterations without any improvements). The TSTRA maintained the original root node partition, but resulted in an overall classification accuracy of 71.91% which was an improvement over the BHC accuracy of 68.82%. Twelve letter moves were made; the most noticeable of these involved bringing the letters B and E, the letters D and O, and the letters Q and X closer together (see Figure 6.3) increasing the individual class accuracies for 5 of these 6 letters (see Table 6.5). Executing TS-FS on the rearranged hierarchy (using the same parameter settings as outlined in Section 4.3.4) resulted in an accuracy of 76.01% (compared to 76.27% which was achieved by executing TS-FS on the BHC).

Class	BHC	TSTRA	TSTRA TS-FS
A	85.26	85.90	87.18
B	43.38	66.18	73.53
C	71.83	73.24	79.58
D	80.24	83.83	73.65
E	52.63	79.61	84.87
F	71.90	77.12	78.43
G	39.63	53.66	69.51
H	47.68	51.66	51.66
I	73.94	80.00	83.03
J	77.03	75.00	80.41
K	60.96	53.42	39.73
L	73.25	75.16	80.25
M	85.42	87.50	93.75
N	87.35	79.52	85.54
O	43.17	59.71	68.35
P	70.83	67.86	79.17
Q	50.60	52.98	67.86
R	54.66	68.94	65.84
S	58.39	47.83	60.25
T	80.79	80.13	80.79
U	74.40	76.19	77.98
V	88.97	89.71	86.03
W	85.61	88.49	96.40
X	73.58	69.18	71.70
Y	80.69	76.55	77.93
Z	77.22	70.25	82.91
Average	**68.82**	**71.91**	**76.01**
Standard Deviation	**15.26**	**12.25**	**12.29**

Table 6.5: BHC, TSTRA and TSTRA TS-FS classification accuracies (%) by letter for the letter recognition dataset.

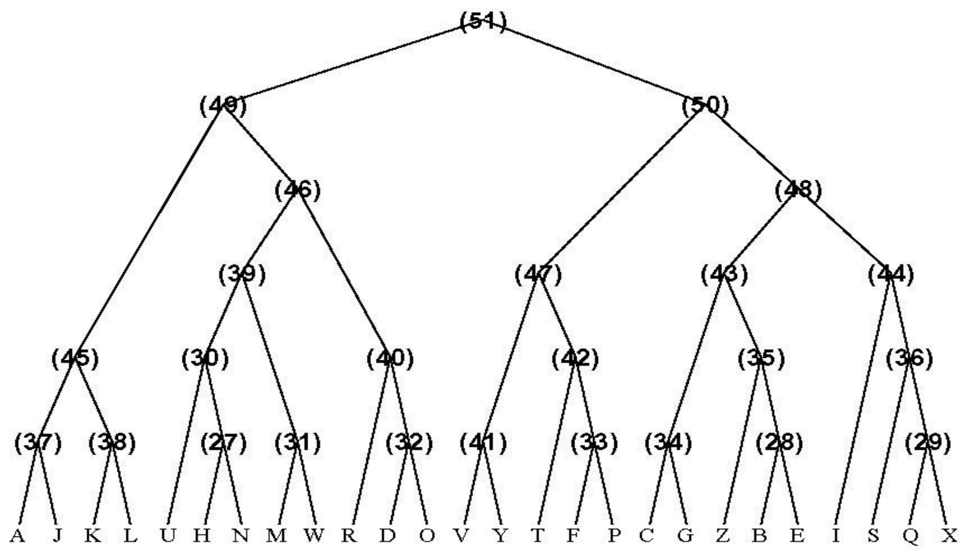

Figure 6.3: TSTRA class hierarchy for the letter recognition data.

6.5 CONCLUSIONS

Average execution times for the BHC and TSTRA are displayed in Table 6.6. The average TSTRA execution times are somewhat elevated; in order to interface with existing MATLAB code, an additional step was required to restructure the tree after every node movement. This additional step renumbered the tree nodes into the sequence that the code expects, but may be unnecessary if this inefficiency could be corrected through alternate coding.

Algorithm	ALI	Hyperion	Hyperion BB
BHC	0.07707	0.65949	0.36346
TSTRA	9.23284	22.84715	--

Table 6.6: Average algorithm execution times in minutes for BHC and TSTRA.

The TSTRA was successful in finding better class hierarchies for the training data as compared to the BHC when the objective was to increase overall classification accuracy. These TSTRA hierarchies translated into increased classification accuracies for the test data and independent test data in a majority of the experiments. In addition, the TSTRA consistently reduced the standard deviations for all of the datasets over that of the BHC. Upon executing the TS-FS on the resulting TSTRA trees, the overall average classification accuracy was increased from 88.97% to 89.80% for the ALI data and decreased from 92.25% to 92.21% for the Hyperion data.

The move neighborhood selected for and currently implemented in the TSTRA is extremely limited. Further research into the structure of the move neighborhoods, the classifiers used and the overall objective of the improvements when comparing the hierarchies may enhance the future use of the TSTRA.

Chapter 7

Conclusions

Classification methods and techniques are becoming increasingly utilized as new emerging technologies acquire masses of data and the demand for their use in new applications increases. High levels of accuracy are desirable (and often required) to accommodate the varied fields that utilize these methodologies in today's fast-paced data-driven society. Results obtained from traditional classification algorithms can often be improved by integrating new techniques within their structures.

7.1 SUMMARY OF CONTRIBUTIONS

This research focused on the incorporation of the metaheuristic Tabu Search for feature selection within the multiclassifier system of the BHC. In addition, a tree rearrangement algorithm using Tabu Search was developed.

7.1.1 Tabu Search Feature Selection

Input space reduction is often a necessity when classification algorithms are faced with an input space of high dimensionality. Feature selection reduces the input space by eliminating those features that are useless or redundant (but fully exploits the information that the full set of features provides) and allows for improved parameter estimation for classification. In addition, feature selection preserves domain knowledge and interpretability of the input space, particularly relative to feature extraction methods that project the data into new spaces.

Feature selection was explored for use within the framework of the BHC supervised classification algorithm in a variety of ways. The Tabu Search metaheuristic was first utilized to solve the combinatorial optimization problem of feature selection as a post-processor of the class hierarchy, in place of the greedy feature selection that is currently being employed. The ability of TS to efficiently search the solution space and to enhance the performance of the classifiers was demonstrated by the reduction of the input space, the increased classification accuracies, and the decreased standard deviations of the accuracies that were attained.

An enhancement to the BHC algorithm, which uses TS-FS in the construction of the class hierarchy, TS Build, was also developed. This algorithm demonstrated that applying feature selection in the construction of the class hierarchy is significantly useful compared to only applying feature selection as a post-processing step for classification. This incorporation of TS-FS in the building of the class hierarchies was another novel contribution of the study.

7.1.2 Classification Tree Rearrangement

The implementation of the TSTRA demonstrated the potential for recourse after a class hierarchy is built. This algorithm allows for recovery should a less-than-optimal partition be made at a multiclass node in the hierarchy-building process. Utilizing the same classifier and partition of training and test data, the TSTRA constructed alternate class hierarchies whose accuracies were increased when classifying the training data. While not guaranteed to also increase the overall classification accuracies of the test data, it achieved increased accuracies when classifying the test data in all but one instance (where the accuracy was not significantly degraded).

7.2 FUTURE WORK

While research in the field of classification has been ongoing for over forty years, it remains a difficult and intensely studied area. Data with a large number of inputs and outputs are now being acquired in multiple application areas that will require specialized techniques for classification and information extraction in order to utilize the data to their fullest potential. This current work can be extended in a variety of ways to meet this growing need.

7.2.1 The Classifier and Feature Selection

The classifier used within the BHC algorithm was not altered in this research. The same classifier was utilized here for comparison purposes to assess the effects of the TS-FS on the classification accuracies. An alternative classifier may be more appropriate for use with TS-FS; this approach needs to be investigated with other methodologies. Preliminary results of using TS-FS with the Bayesian Pairwise Classifier [48] are promising. In addition, the investigation of more advantageous measures of goodness for inclusion and exclusion of features could facilitate the feature selection in the identification of more meaningful feature subsets.

7.2.2 Best Bases

When aggregating the bands using BB, a correlation threshold of .90 was implemented in this study without consideration for the amount of training data that is present to estimate the class parameters. An alternative threshold presented in [49], takes into account the possibility of limited training data, and dictates that the (number of training data observations)/(number of features) should be greater than 5, thus aggregating the bands until this threshold is reached. The addition of this check and threshold should be considered for use in the algorithms described, especially for datasets possessing a great number of features like the Hyperion data.

7.2.3 Tabu Search

The Tabu Search (TS) metaheuristic is ever evolving; continued research has brought about a multitude of new, innovative techniques in its implementation and new problems for its application. The move neighborhoods implemented for the feature selection in this study were limited to swaps and inserts, while the tree rearrangement was extremely limited in that it only paired classes at the leaf node level. These TS algorithms may benefit from the addition or total replacement of the move neighborhoods used. In addition, further research could provide alternative parameters, attributes, adaptive methods, starting solutions, and techniques that would aid classification algorithms, especially those with large numbers of inputs and outputs, such as hyperspectral data. The sensitivity of the TS starting solution in the TS Build using the Hyperion data was briefly studied by randomly choosing a subset of features for the second metaclass partition, as opposed to using those features with the highest Fisher weights (see Figure 5.1). The resulting classification accuracies were comparable, indicating a lack of sensitivity to the incumbent solution and warranting further research in this area.

Additionally, further study into the sensitivity of the current TS parameters may yield a more extensive search of the solution space.

Other techniques coupled with TS, such as maintaining ensembles of the best feature subsets for each metaclass identified by TS may prove to be helpful. Subjective evaluation of the feature subsets by subject matter experts may provide better classifiers as opposed to focusing on the classification accuracy of the test data as the primary measure of goodness. Another technique often applied to TS is the use of a candidate move list as opposed to searching the entire move neighborhood, which can be huge when the number of candidate features (i.e. hyperspectral data) is large.

In the present implementation of TS-FS, a node is "skipped" and TS-FS is not performed when the classes at the current node are classified with an accuracy of 100%. While this approach saves computation time, it may miss an opportunity to refine a subset of features. TS-FS could still be implemented to search for subsets of decreased cardinality while maintaining (or possibly slightly reducing) the classification accuracy at the current node.

7.2.4 Tree Rearrangement

The TSTRA can potentially enhance any binary tree structure, and with modifications, could be applied to other types of decision trees. Research into different measures of goodness, classifiers and tree structures may, as a composite, yield better hierarchies for classification. Feature selection could also be incorporated into the TSTRA.

7.2.5 A Grove of Trees

This research showed that many different class hierarchies are constructed from the very same data: the only difference being the way that the data were partitioned into training and testing sets. Each hierarchy has its strengths, but at the same time, each has its weaknesses. Further research could identify a way to group these differing hierarchy structures to exploit their strengths collectively while limiting the negative effects of their individual weaknesses.

APPENDIX A

Selected ALI Data Class Hierarchies and Confusion Matrices

A.1 EXPERIMENT ALI2

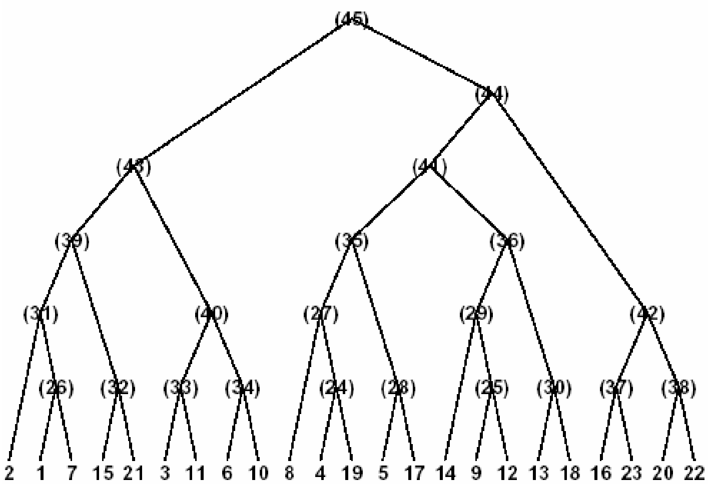

Figure A.1: Experiment ALI2 BHC class hierarchy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	63	9	0	0	0	1	7	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	75.0
2	6	85	0	0	1	1	2	1	0	0	0	0	0	2	0	0	0	0	0	0	1	0	0	85.9
3	1	0	143	0	5	0	0	0	0	3	1	0	0	1	0	0	0	0	0	0	0	0	0	92.9
4	0	0	0	104	3	0	0	5	1	0	1	1	0	0	0	0	0	0	21	0	0	0	0	76.5
5	0	0	0	0	108	0	1	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	93.9
6	0	0	0	0	0	97	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	98.0
7	7	0	0	0	0	0	61	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	87.1
8	0	2	0	6	4	0	2	48	2	0	0	0	0	3	0	0	8	24	1	0	0	0	0	48.0
9	0	0	0	3	0	1	0	3	74	0	0	1	8	0	0	0	0	1	1	0	0	0	0	80.4
10	0	0	6	0	1	0	0	0	0	78	1	1	0	2	0	0	0	0	0	0	0	0	0	87.6
11	0	0	2	0	3	0	0	0	0	1	77	0	0	0	0	0	0	0	0	0	0	0	0	92.8
12	0	0	0	0	0	0	0	0	6	0	0	107	3	1	0	0	0	0	13	0	0	0	0	82.3
13	0	0	0	0	0	0	0	0	0	0	0	10	72	2	0	0	0	0	3	0	0	0	0	82.8
14	0	0	0	0	0	0	0	0	0	0	2	2	0	47	0	0	0	0	0	0	0	0	0	92.2
15	1	0	0	0	0	0	0	0	0	0	0	0	0	0	93	0	0	1	0	0	1	0	0	96.9
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	101	0	0	0	3	9	0	0	89.4
17	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	3	77	0	0	0	0	0	0	93.9
18	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	14	70	0	0	0	0	0	81.4
19	0	0	0	11	2	0	0	0	0	0	3	4	0	0	0	0	0	0	131	0	0	0	0	86.8
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	117	0	0	0	100.0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	63	0	0	85.1
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	123	0	100.0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0	0	0	0	1	78	95.1
	80.8	88.5	94.7	83.9	85.0	97.0	82.4	71.6	87.1	95.1	90.6	84.9	86.7	79.7	96.9	87.1	76.2	72.9	77.1	97.5	84.0	99.2	100.0	
Average Accuracy: 87.20%																								

Table A.1: Experiment ALI2 BHC confusion matrix.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	60	10	0	0	0	0	5	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	76.9
2	3	78	0	0	0	0	4	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0	88.6
3	0	1	137	0	5	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	93.8
4	0	0	0	103	5	0	0	6	3	0	0	6	5	0	0	0	0	3	11	0	0	0	0	72.5
5	0	0	0	1	108	0	0	2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	4	93.1
6	0	0	5	0	0	97	1	5	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	85.8
7	10	5	0	0	0	0	64	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	79.0
8	0	2	0	10	2	1	0	47	5	0	0	0	0	0	0	0	8	0	1	0	0	0	0	61.8
9	0	0	0	1	0	0	0	2	65	0	0	1	0	1	1	0	0	0	2	0	0	0	0	89.0
10	0	0	9	0	0	2	0	0	3	74	0	0	0	1	0	0	0	0	0	0	0	0	0	83.1
11	0	0	0	0	4	0	0	0	0	4	78	0	0	1	0	0	0	0	0	0	0	0	0	89.7
12	0	0	0	1	0	0	0	0	6	0	0	106	5	4	1	0	1	0	7	0	0	0	0	80.9
13	0	0	0	0	0	0	0	0	0	0	0	4	68	0	0	0	0	2	6	0	0	0	0	85.0
14	0	0	0	0	0	0	0	0	0	0	6	1	0	52	3	0	0	0	0	0	0	0	0	83.9
15	3	0	0	0	0	0	0	0	0	0	0	0	0	0	89	0	0	1	0	0	1	0	0	94.7
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	84	0	0	0	1	1	0	0	97.7
17	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	13	85	0	0	0	0	0	5	80.2
18	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	3	53	0	0	0	0	0	91.4
19	0	0	0	8	3	0	0	0	0	0	0	8	4	0	0	0	2	37	143	0	0	0	0	69.8
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	119	0	0	0	100.0
21	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	72	0	0	80.0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	122	0	100.0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	69	97.2
	76.9	81.2	90.7	83.1	85.0	97.0	86.5	70.1	76.5	90.2	91.8	84.1	81.9	88.1	92.7	72.4	84.2	55.2	84.1	99.2	96.0	98.4	88.5	
Average Accuracy: 85.30%																								

Table A.2: Experiment ALI2 BHC FS confusion matrix.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	63	5	0	0	0	0	9	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	80.8
2	2	86	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	95.6
3	0	1	146	0	6	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	94.2
4	0	0	0	107	2	0	0	4	1	0	0	0	6	0	0	0	0	5	10	0	0	0	0	79.3
5	0	0	0	0	111	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	98.2
6	0	0	3	0	0	97	1	3	2	4	0	1	0	0	0	0	0	0	0	0	0	0	0	87.4
7	11	2	0	0	0	0	62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	82.7
8	0	2	0	9	2	1	1	52	5	0	0	0	1	0	0	0	1	1	1	0	0	0	0	68.4
9	0	0	0	1	0	0	0	3	73	1	0	1	0	0	1	0	0	1	0	0	0	0	0	90.1
10	0	0	2	0	0	2	0	0	0	75	0	0	0	2	0	0	0	0	0	0	0	0	0	92.6
11	0	0	0	0	3	0	0	0	0	1	77	0	0	2	0	0	0	0	0	0	0	0	0	92.8
12	0	0	0	0	0	0	0	0	4	0	0	111	2	0	0	0	0	1	9	0	0	0	0	87.4
13	0	0	0	0	0	0	0	0	0	0	0	7	69	0	0	0	0	2	0	0	0	0	0	88.5
14	0	0	0	0	0	0	0	0	0	0	7	1	1	55	3	0	0	0	1	0	0	0	0	80.9
15	2	0	0	0	0	0	0	0	0	0	0	0	0	0	91	11	0	1	0	0	3	0	0	84.3
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96	4	0	0	2	6	0	0	88.9
17	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	1	88	0	0	0	0	0	0	96.7
18	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	7	85	0	0	0	0	0	91.4
19	0	0	0	7	3	0	0	0	0	0	0	5	3	0	0	0	0	0	149	0	0	0	2	88.2
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	118	0	0	0	99.2
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	66	0	0	90.4
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	122	0	100.0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	2	76	96.2
	80.8	89.6	96.7	86.3	87.4	97.0	83.8	77.6	85.9	91.5	90.6	88.1	83.1	93.2	94.8	82.8	87.1	88.5	87.6	98.3	88.0	98.4	97.4	
Average Accuracy: 89.71%																								

Table A.3: Experiment ALI2 BHC TS-FS confusion matrix.

A.2 EXPERIMENT ALI3

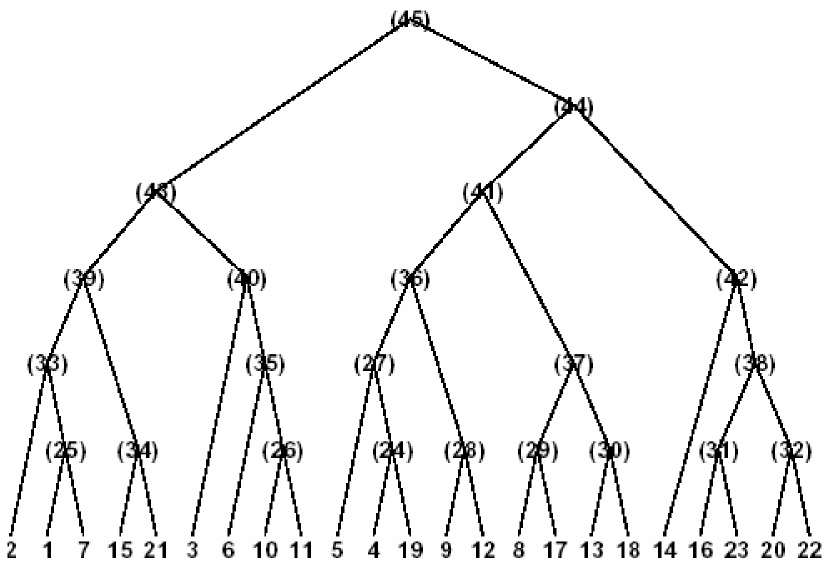

Figure A.2: Experiment ALI3 BHC class hierarchy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	61	10	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	79.2
2	2	83	0	0	3	0	2	2	0	0	0	0	0	2	0	0	0	0	0	0	1	0	0	87.4
3	3	1	142	0	8	1	0	0	0	5	11	0	0	1	0	0	0	0	0	0	0	0	0	82.6
4	0	0	0	92	1	0	0	15	3	0	0	1	0	0	0	0	0	0	23	0	0	0	0	68.1
5	0	0	0	1	111	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	98.2
6	0	0	2	0	4	97	0	0	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	89.8
7	9	1	0	0	0	1	62	2	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	80.5
8	0	1	0	8	0	1	3	40	6	0	0	0	0	0	0	0	0	1	3	0	0	0	0	63.5
9	0	0	0	1	0	0	0	3	69	0	0	3	1	0	0	0	0	1	0	0	0	0	0	88.5
10	0	0	6	0	0	0	0	0	0	75	0	1	0	2	0	0	0	0	0	0	0	0	0	89.3
11	0	0	1	0	0	0	0	0	0	0	64	0	0	1	0	0	0	0	0	0	0	0	0	97.0
12	0	0	0	1	0	0	0	0	2	0	1	114	7	13	0	0	0	0	2	0	0	0	0	81.4
13	0	0	0	8	0	0	0	2	1	0	0	0	73	0	0	0	0	0	3	0	0	0	0	83.9
14	0	0	0	0	0	0	0	0	0	0	9	1	1	39	0	0	0	0	9	0	0	0	0	66.1
15	2	0	0	0	0	0	1	0	0	0	0	0	0	1	91	0	0	2	0	0	0	0	0	93.8
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	85	0	0	0	4	10	0	0	85.9
17	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	12	99	3	0	0	0	0	1	85.3
18	0	0	0	7	0	0	0	0	0	0	0	0	0	0	1	1	1	89	0	0	0	0	0	89.9
19	0	0	0	6	0	0	0	1	1	0	0	6	1	0	0	0	0	0	130	0	0	0	0	89.7
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	116	0	0	0	100.0
21	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	18	0	0	0	0	64	0	0	75.3
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	124	1	99.2
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	76	98.7
	78.2	86.5	94.0	74.2	87.4	97.0	83.8	59.7	81.2	91.5	75.3	90.5	88.0	66.1	94.8	73.3	98.0	92.7	76.5	96.7	85.3	100.0	97.4	
Average Accuracy: 86.29%																								

Table A.4: Experiment ALI3 BHC confusion matrix.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	56	5	0	0	0	0	12	0	0	0	0	0	0	0	17	1	0	0	0	0	0	0	0	61.5
2	1	83	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.5
3	0	0	135	0	0	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	97.8
4	0	0	0	107	3	0	0	6	3	0	0	0	0	0	0	0	0	0	17	0	0	0	0	78.7
5	0	0	0	0	109	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	98.2
6	0	0	3	0	11	95	1	5	5	2	0	0	0	0	0	0	0	0	0	0	0	0	0	77.9
7	12	7	0	0	0	0	53	1	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	68.8
8	0	1	0	5	1	0	2	38	2	0	0	0	0	0	0	0	5	3	2	0	0	0	0	64.4
9	0	0	0	2	0	1	0	6	70	0	0	5	1	0	1	0	0	0	0	0	0	0	0	81.4
10	0	0	4	0	0	2	0	0	1	77	1	0	0	1	0	0	0	0	0	0	0	0	0	89.5
11	0	0	9	0	2	0	0	0	0	0	77	0	0	1	1	0	0	0	0	0	0	0	0	85.6
12	0	0	0	1	0	0	0	1	0	2	0	106	9	1	0	0	0	0	6	0	0	0	0	84.1
13	0	0	0	4	0	0	0	2	3	0	1	3	73	0	0	0	0	0	1	0	0	0	0	83.9
14	0	0	0	0	0	0	0	0	0	0	4	7	0	55	0	0	0	0	0	0	0	0	0	83.3
15	4	0	0	0	0	0	3	0	0	0	0	0	0	0	73	0	0	1	0	0	0	0	0	90.1
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	92	0	0	0	2	1	0	0	96.8
17	0	0	0	0	0	0	1	4	0	0	0	0	0	0	1	0	78	1	0	0	0	0	1	90.7
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	91	0	0	0	0	0	97.8
19	0	0	0	5	1	0	0	2	1	0	2	5	0	1	0	0	0	0	144	0	0	0	0	89.4
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	118	0	0	0	100.0
21	5	0	0	0	0	0	0	0	0	0	0	0	0	0	1	19	0	0	0	0	74	0	0	74.7
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	123	0	100.0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	0	0	0	0	1	77	80.2
	71.8	86.5	89.4	86.3	85.8	95.0	71.6	56.7	82.4	93.9	90.6	84.1	88.0	93.2	76.0	79.3	77.2	94.8	84.7	98.3	98.7	99.2	98.7	
Average Accuracy: 86.64%																								

Table A.5: Experiment ALI3 BHC FS confusion matrix.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	54	7	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	79.4
2	4	81	0	0	2	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90.0
3	0	0	149	0	1	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	98.0
4	0	0	0	112	2	0	0	4	1	0	0	1	0	0	0	0	0	0	18	0	0	0	0	81.2
5	0	0	0	0	108	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	98.2
6	1	0	0	0	6	99	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90.8
7	13	7	0	0	0	0	63	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	75.0
8	0	1	0	6	1	0	2	45	2	0	0	0	0	1	0	0	1	0	3	0	0	0	0	72.6
9	0	0	0	1	0	1	0	9	75	0	0	2	0	0	0	0	0	0	0	0	0	0	0	85.2
10	0	0	2	0	1	0	0	0	2	78	1	0	0	2	0	0	0	0	0	0	0	0	0	90.7
11	0	0	0	0	4	0	0	0	0	1	76	1	0	2	0	0	0	0	0	0	0	0	0	90.5
12	0	0	0	0	0	0	0	0	1	1	0	115	8	2	0	0	0	0	4	0	0	0	0	87.8
13	0	0	0	3	0	0	0	1	2	0	0	3	75	0	0	0	0	0	1	0	0	0	0	88.2
14	0	0	0	0	0	0	0	0	0	0	8	1	0	51	2	0	0	0	0	0	0	0	0	82.3
15	3	0	0	0	0	0	0	0	0	0	0	0	0	0	91	16	0	2	0	0	0	0	0	81.3
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	93	0	0	0	4	11	0	0	86.1
17	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	99	1	0	0	0	0	0	97.1
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	93	0	0	0	0	0	100.0
19	0	0	0	1	1	0	0	4	0	0	0	3	0	1	0	0	0	0	143	0	0	0	0	93.5
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	116	0	0	0	100.0
21	3	0	0	0	0	0	0	0	0	0	0	0	0	0	2	6	0	0	0	0	64	0	0	85.3
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	123	2	98.4
23	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	76	95.0
	69.2	84.4	98.7	90.3	85.0	99.0	85.1	67.2	88.2	95.1	89.4	91.3	90.4	86.4	94.8	80.2	98.0	96.9	84.1	96.7	85.3	99.2	97.4	
Average Accuracy: 89.88%																								

Table A.6: Experiment ALI3 BHC TS-FS confusion matrix.

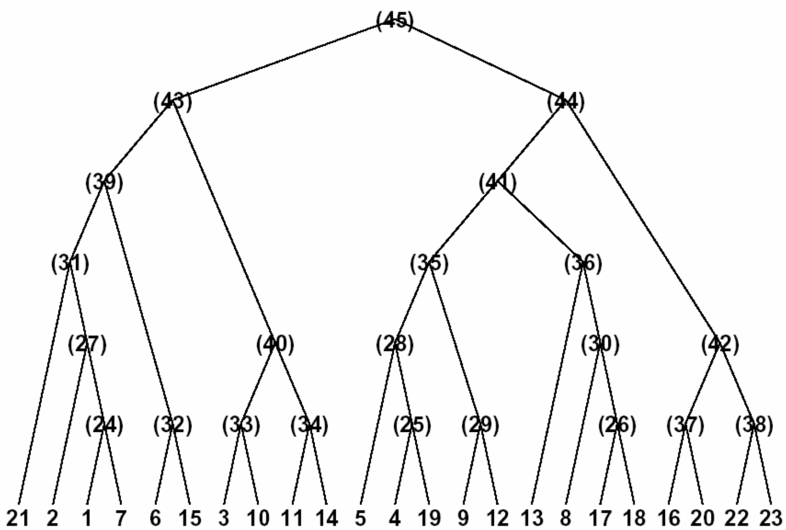

Figure A.3: Experiment ALI3 TS Build class hierarchy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	55	6	0	0	0	0	7	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	77.5
2	4	79	0	0	0	0	2	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	90.8
3	0	0	146	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	99.3
4	0	0	0	112	2	0	0	4	1	0	0	1	0	0	0	0	0	0	17	0	0	0	0	81.8
5	0	2	0	0	109	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.5
6	1	1	4	0	7	95	0	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	84.1
7	12	7	0	0	0	1	60	0	0	0	0	0	0	0	1	4	0	0	0	0	0	0	0	70.6
8	0	1	0	7	3	0	4	42	2	0	0	0	0	0	0	0	0	0	2	0	0	0	0	68.9
9	0	0	0	1	0	1	0	9	77	0	0	3	1	0	0	0	0	0	0	0	0	0	0	83.7
10	0	0	0	0	0	1	0	0	1	81	0	0	0	0	0	0	0	0	0	0	0	0	0	97.6
11	0	0	1	0	4	0	0	0	0	0	83	0	0	0	0	0	0	0	0	0	0	0	0	94.3
12	0	0	0	0	0	0	0	0	1	0	0	114	8	2	0	0	0	0	4	0	0	0	0	88.4
13	0	0	0	0	0	0	0	0	2	0	0	2	74	0	0	0	0	0	1	0	0	0	0	93.7
14	0	0	0	0	1	2	0	0	0	0	2	3	0	57	3	0	0	0	2	0	0	0	0	81.4
15	2	0	0	0	0	0	1	0	0	0	0	0	0	0	89	0	0	3	0	0	0	0	0	93.7
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0	0	19	0	0	83.2
17	1	0	0	2	0	0	0	2	0	0	0	0	0	0	0	1	100	0	1	0	0	0	0	93.5
18	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	93	0	0	0	0	0	97.9
19	0	0	0	1	1	0	0	4	0	0	0	3	0	0	0	0	0	0	143	0	0	0	0	94.1
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0	100.0
21	3	0	0	0	0	0	0	0	0	0	0	0	0	0	1	12	0	0	0	0	56	0	0	77.8
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	123	0	100.0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	78	96.3
	70.5	82.3	96.7	90.3	85.8	95.0	81.1	62.7	90.6	98.8	97.6	90.5	89.2	96.6	92.7	81.0	99.0	96.9	84.1	100.0	74.7	99.2	100.0	
Average Accuracy: 89.93%																								

Table A.7: Experiment ALI3 TS Build confusion matrix.

A.3 EXPERIMENT ALI4

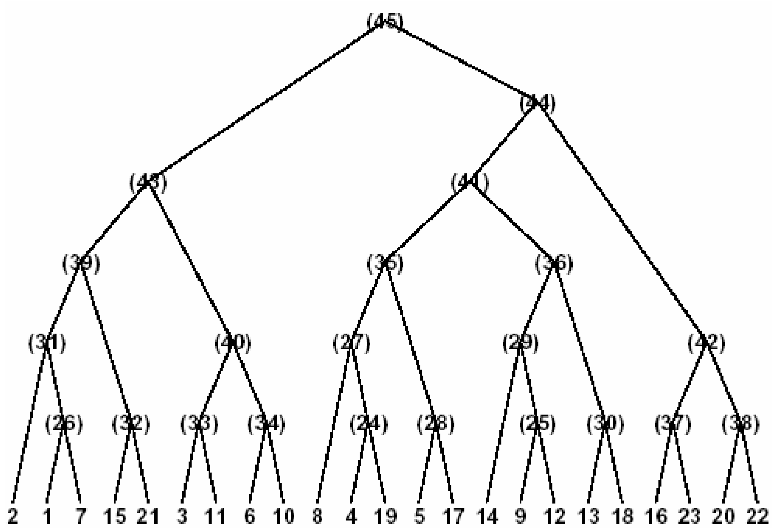

Figure A.4: Experiment ALI4 BHC class hierarchy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	60	8	0	0	0	1	6	2	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	76.9
2	6	84	1	0	0	1	5	2	0	0	0	0	0	2	3	0	0	0	0	0	2	0	0	79.2
3	0	1	142	0	9	1	0	0	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	90.4
4	0	0	0	106	3	0	0	5	0	0	0	1	1	0	0	0	0	0	18	0	0	0	0	79.1
5	0	0	0	2	111	0	1	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	92.5
6	0	0	0	0	0	97	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97.0
7	11	1	0	0	0	0	59	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	80.8
8	0	2	0	5	1	0	2	44	0	0	0	0	0	0	0	0	8	17	2	0	0	0	0	54.3
9	0	0	0	0	0	0	0	1	68	1	0	4	11	0	0	0	0	0	1	0	0	0	0	79.1
10	0	0	7	0	0	0	0	0	1	77	0	1	0	1	0	0	0	0	0	0	0	0	0	88.5
11	0	0	1	0	2	0	0	0	0	1	76	0	0	1	0	0	0	0	0	0	0	0	0	93.8
12	0	0	0	1	0	0	0	0	12	0	0	96	5	1	0	0	0	0	8	0	0	0	0	78.0
13	0	0	0	0	0	0	0	2	0	0	0	10	66	1	0	0	1	0	1	0	0	0	0	81.5
14	0	0	0	0	0	0	0	0	1	0	6	3	0	50	0	0	0	0	0	0	0	0	0	83.3
15	1	0	0	0	0	0	1	0	0	0	0	0	0	0	90	0	0	2	0	0	0	0	0	95.7
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	106	0	0	0	3	8	0	0	90.6
17	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	2	72	0	0	0	0	0	0	92.3
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	77	0	0	0	0	0	87.5
19	0	0	0	10	1	0	0	1	0	0	2	11	0	3	0	0	0	0	140	0	0	0	0	83.3
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	117	0	0	0	100.0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	7	0	0	0	0	65	0	0	89.0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	124	2	98.4
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	76	89.4
	76.9	87.5	94.0	85.5	87.4	97.0	79.7	65.7	80.0	93.9	89.4	76.2	79.5	84.7	93.8	91.4	71.3	80.2	82.4	97.5	86.7	100.0	97.4	
Average Accuracy: 86.60%																								

Table A.8: Experiment ALI4 BHC confusion matrix.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	55	4	0	0	0	0	8	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	79.7
2	4	88	0	0	0	0	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90.7
3	0	0	145	0	8	0	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	92.4
4	0	0	0	104	1	0	0	7	7	0	0	16	0	0	0	0	0	7	14	0	0	0	1	66.2
5	0	0	0	0	116	0	0	4	0	0	0	0	0	0	0	0	1	3	0	0	0	0	0	93.5
6	0	0	0	0	1	96	0	3	8	4	0	1	0	0	0	0	0	0	0	0	0	0	0	85.0
7	15	2	0	0	0	1	64	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	75.3
8	0	2	0	10	0	0	0	43	6	0	0	0	0	0	0	0	0	7	3	0	0	0	0	60.6
9	0	0	0	0	0	0	0	1	44	0	0	0	0	0	0	0	0	0	2	0	0	0	0	93.6
10	0	0	6	0	0	3	0	0	0	70	0	0	0	0	0	0	0	0	0	0	0	0	0	88.6
11	0	0	0	0	0	0	0	0	0	2	76	0	0	0	0	0	0	0	0	0	0	0	0	97.4
12	0	0	0	0	0	0	0	0	17	3	0	82	2	0	0	0	0	0	5	0	0	0	0	75.2
13	0	0	0	0	0	0	0	0	0	0	0	9	80	0	0	0	0	0	6	0	0	0	0	84.2
14	0	0	0	0	0	0	0	0	1	1	7	7	0	59	3	0	0	0	2	0	0	0	0	73.8
15	3	0	0	0	0	0	0	0	0	0	0	0	0	0	89	1	2	1	0	0	0	0	0	92.7
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	91	0	0	0	4	1	0	0	94.8
17	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	5	98	6	0	0	0	0	2	86.0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	68	0	0	0	0	0	100.0
19	0	0	0	10	1	0	0	1	2	0	0	11	1	0	0	0	0	4	138	0	0	0	5	79.8
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	116	0	0	0	100.0
21	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	19	0	0	0	0	74	0	0	77.9
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	123	0	100.0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	70	98.6
	70.5	91.7	96.0	83.9	91.3	96.0	86.5	64.2	51.8	85.4	89.4	65.1	96.4	100.0	92.7	78.4	97.0	70.8	81.2	96.7	98.7	99.2	89.7	
Average Accuracy: 85.99%																								

Table A.9: Experiment ALI4 BHC FS confusion matrix.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	61	0	0	0	0	0	5	1	0	0	0	0	0	0	2	1	0	0	0	0	0	0	0	87.1
2	3	90	0	0	1	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	91.8
3	0	0	144	0	3	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	94.1
4	0	0	0	112	1	0	0	4	3	0	0	1	0	0	0	0	0	4	11	0	0	0	0	82.4
5	0	0	0	0	119	0	0	3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	96.7
6	0	0	0	0	1	99	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	95.2
7	10	4	0	0	0	0	66	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	80.5
8	0	1	0	5	0	0	1	52	1	0	0	0	1	0	0	0	0	1	2	0	0	0	0	81.3
9	0	0	0	0	0	0	0	1	64	1	0	1	0	0	0	0	0	0	2	0	0	0	0	92.8
10	0	0	5	0	0	0	0	0	0	74	0	0	0	1	0	0	0	0	0	0	0	0	0	92.5
11	0	0	2	0	1	1	0	0	0	5	73	0	0	1	0	0	0	0	0	0	0	0	0	88.0
12	0	0	0	0	0	0	0	0	9	2	0	98	1	0	0	0	1	0	5	0	0	0	0	84.5
13	0	0	0	0	0	0	0	0	0	0	0	8	66	0	0	0	0	0	0	0	0	0	0	89.2
14	0	0	0	0	0	0	0	0	1	0	6	4	0	57	2	0	0	0	1	0	0	0	0	80.3
15	4	0	0	0	0	0	0	0	0	0	0	0	0	0	90	6	0	2	0	0	0	0	0	88.2
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	102	0	0	0	4	7	0	0	90.3
17	0	1	0	0	0	0	0	2	0	0	0	0	0	0	0	1	94	0	0	0	0	0	1	94.9
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	89	0	0	0	0	0	95.7
19	0	0	0	7	1	0	0	1	3	0	0	14	15	0	0	0	0	0	149	0	0	0	1	78.0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	116	0	0	0	100.0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	6	0	0	0	0	68	0	0	90.7
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	123	0	100.0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	76	97.4
	78.2	93.8	95.4	90.3	93.7	99.0	89.2	77.6	75.3	90.2	85.9	77.8	79.5	96.6	93.8	87.9	93.1	92.7	87.6	96.7	90.7	99.2	97.4	
Average Accuracy: 90.01%																								

Table A.10: Experiment ALI4 BHC TS-FS confusion matrix.

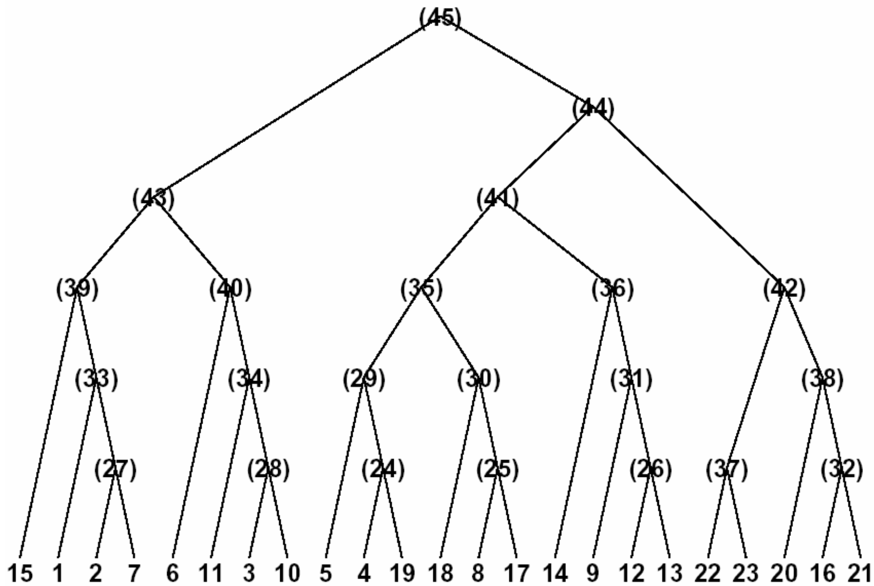

Figure A.5: Experiment ALI4 TSTRA class hierarchy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	56	5	0	0	0	0	3	0	0	0	0	0	0	2	2	0	0	0	0	0	6	0	0	75.7
2	5	80	1	0	0	1	3	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	87.0
3	0	0	148	0	0	2	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	96.7
4	0	0	0	110	1	0	0	6	1	0	0	1	1	0	0	0	0	0	18	0	0	0	0	79.7
5	0	0	0	0	114	0	0	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0	1	96.6
6	0	0	0	0	7	96	0	0	4	2	0	1	0	3	0	0	0	0	0	0	0	0	0	85.0
7	15	3	0	0	0	1	61	3	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	72.6
8	0	7	0	2	3	0	7	55	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	69.6
9	0	0	0	0	0	0	0	0	62	1	0	5	1	0	0	0	0	2	1	0	0	0	0	86.1
10	0	0	1	0	0	0	0	0	1	75	0	0	0	0	0	0	0	0	0	0	0	0	0	97.4
11	0	0	1	0	1	0	0	0	0	2	66	0	0	2	0	0	0	0	0	0	0	0	0	91.7
12	0	0	0	0	0	0	0	0	16	0	0	93	1	0	0	0	0	0	7	0	0	0	0	79.5
13	0	0	0	0	0	0	0	0	0	0	0	0	77	0	0	0	1	2	1	0	0	0	0	95.1
14	0	0	0	0	0	0	0	0	1	0	16	4	0	47	0	0	0	0	0	0	0	0	0	69.1
15	1	0	0	0	0	0	0	0	0	0	0	0	0	1	93	0	0	2	0	0	4	0	0	92.1
16	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	105	3	0	0	0	1	0	0	94.6
17	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	4	79	0	0	0	0	0	0	94.0
18	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	2	6	90	0	0	0	0	0	90.9
19	0	0	0	11	1	0	0	1	0	0	0	22	3	3	0	0	0	0	138	0	0	0	0	77.1
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0	100.0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	64	0	0	94.1
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	124	0	100.0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	77	86.5
	71.8	83.3	98.0	88.7	89.8	96.0	82.4	82.1	72.9	91.5	77.6	73.8	92.8	79.7	96.9	90.5	78.2	93.8	81.2	100.0	85.3	100.0	98.7	
Average Accuracy: 87.76%																								

Table A.11: Experiment ALI4 TSTRA confusion matrix.

A.4 EXPERIMENT ALI5

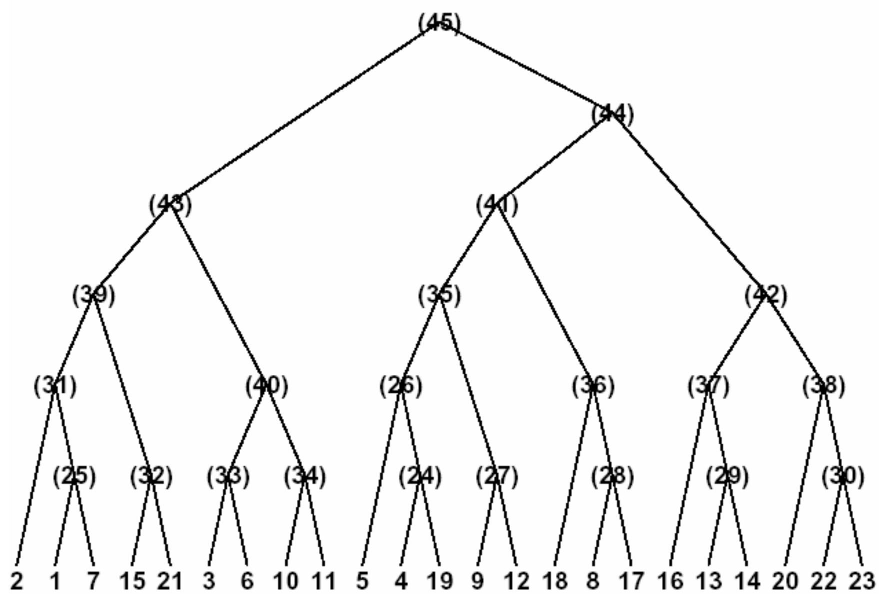

Figure A.6: Experiment ALI5 TS Build class hierarchy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	62	12	0	0	0	0	5	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	76.5
2	2	81	0	0	0	2	1	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	89.0
3	0	0	148	0	8	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	92.5
4	0	0	0	106	3	0	0	7	0	0	0	0	1	0	0	0	0	0	10	0	0	0	0	83.5
5	0	0	0	0	108	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
6	0	0	0	0	2	95	0	0	4	3	0	0	0	0	0	0	0	0	0	0	0	0	0	91.3
7	10	2	0	0	0	0	66	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	82.5
8	0	1	0	10	3	0	2	44	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	71.0
9	0	0	0	3	0	1	0	7	74	0	0	4	18	0	0	0	0	0	1	0	0	0	0	68.5
10	0	0	2	0	0	2	0	0	3	74	0	0	0	0	0	0	0	0	0	0	0	0	0	91.4
11	0	0	1	0	2	0	0	0	0	0	81	0	0	1	0	0	0	0	0	0	0	0	0	95.3
12	0	0	0	0	0	0	0	0	2	0	0	106	18	0	0	0	0	0	4	0	0	0	0	81.5
13	0	0	0	0	0	0	0	0	0	0	0	4	42	0	0	0	0	0	1	0	0	0	0	89.4
14	0	0	0	0	0	0	0	0	0	1	4	8	0	58	1	2	0	0	6	0	0	0	8	65.9
15	3	0	0	0	0	0	0	0	0	0	0	0	0	0	92	8	0	1	0	0	2	0	0	86.8
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90	0	0	0	1	2	0	0	96.8
17	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	100	3	0	0	0	0	1	94.3
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	92	0	0	0	0	0	100.0
19	0	0	0	5	1	0	0	1	1	0	0	4	4	0	0	0	0	0	147	0	0	0	0	90.2
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	118	2	0	0	90.1
21	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	4	0	0	0	0	69	0	0	92.0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	123	0	100.0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	69	95.8
	79.5	84.4	98.0	85.5	85.0	95.0	89.2	65.7	87.1	90.2	95.3	84.1	50.6	98.3	95.8	77.6	99.0	95.8	86.5	98.3	92.0	99.2	88.5	
Average Accuracy: 88.41%																								

Table A.12: Experiment ALI5 TS Build confusion matrix.

A.5 EXPERIMENT ALI6

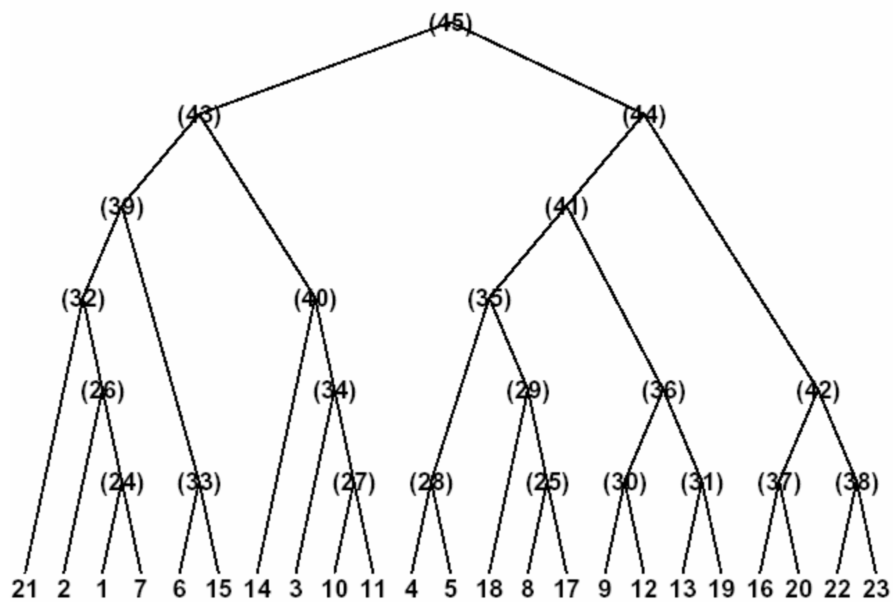

Figure A.7: Experiment ALI6 TS Build class hierarchy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	60	4	0	0	0	1	8	0	0	0	0	0	0	0	4	7	3	0	0	0	1	0	0	68.2
2	3	88	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94.6
3	0	0	146	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	97.3
4	0	0	0	100	1	0	0	8	0	0	0	0	0	0	0	0	1	0	14	0	0	0	0	80.6
5	0	1	0	1	110	0	0	3	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	94.8
6	0	0	1	0	8	92	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	87.6
7	10	2	0	0	0	1	60	1	0	0	0	0	0	0	1	6	0	0	0	0	0	0	0	74.1
8	0	0	0	3	0	0	4	48	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	84.2
9	0	0	0	3	0	2	0	4	77	1	0	2	0	0	0	0	0	0	2	0	0	0	0	84.6
10	0	0	2	0	0	4	0	0	0	77	1	0	0	0	0	0	0	0	0	0	0	0	0	91.7
11	0	0	2	0	7	0	0	0	0	0	83	0	0	1	2	0	0	0	4	0	0	0	0	83.8
12	0	0	0	0	0	0	0	0	1	0	0	112	2	0	0	0	0	0	4	0	0	0	0	94.1
13	0	0	0	0	0	0	0	0	1	0	0	3	76	0	0	0	0	0	2	0	0	0	0	92.7
14	0	0	0	0	0	0	0	0	0	0	1	6	5	58	0	0	0	0	0	0	0	0	0	82.9
15	4	0	0	0	0	0	0	0	0	0	0	0	0	0	89	0	0	4	0	0	0	0	0	91.8
16	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	102	0	0	0	1	10	0	0	88.7
17	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	97	0	0	0	0	0	0	99.0
18	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	92	0	0	0	0	0	98.9
19	0	0	0	16	0	0	0	2	1	0	0	3	0	0	0	0	0	0	142	0	0	0	1	86.1
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	119	0	0	0	100.0
21	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	64	0	0	97.0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	122	1	99.2
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	76	97.4
	76.9	91.7	96.7	80.6	86.6	92.0	81.1	71.6	90.6	93.9	97.6	88.9	91.6	98.3	92.7	87.9	96.0	95.8	83.5	99.2	85.3	98.4	97.4	
Average Accuracy: 90.36%																								

Table A.13: Experiment ALI6 TS Build confusion matrix.

A.6 EXPERIMENT ALI7

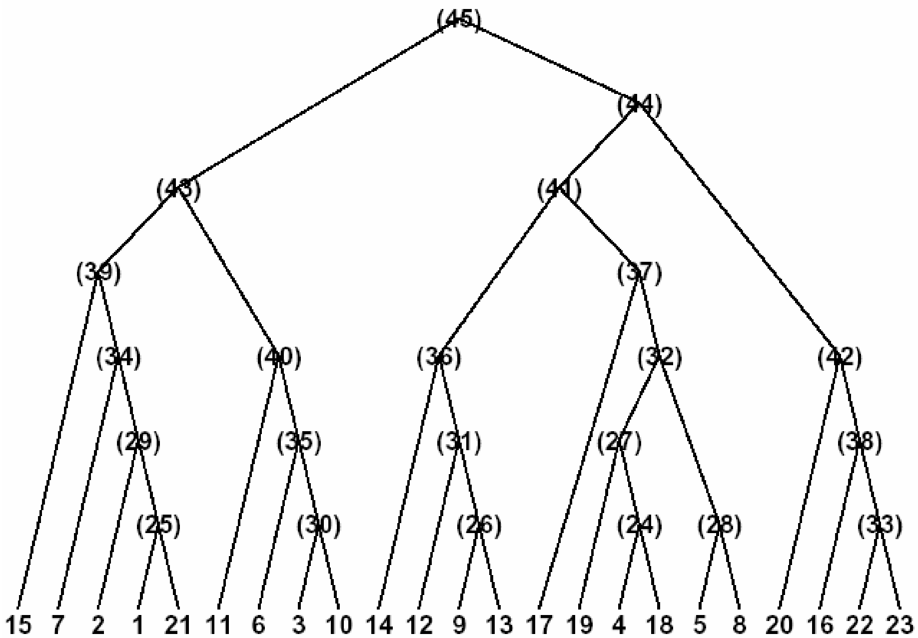

Figure A.8: Experiment ALI7 TSTRA class hierarchy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	59	13	0	0	0	0	10	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	70.2
2	3	80	0	0	1	0	6	2	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	85.1
3	0	0	147	0	1	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	97.4
4	0	0	0	96	1	1	0	6	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	82.8
5	0	0	0	2	118	0	0	2	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	95.2
6	2	0	0	0	2	98	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	92.5
7	14	3	0	0	0	0	55	0	0	0	0	0	0	0	0	7	0	0	0	0	1	0	0	68.8
8	0	0	0	6	0	0	3	45	0	0	0	0	0	0	0	0	2	0	1	0	0	0	0	78.9
9	0	0	0	0	0	1	0	1	71	0	0	10	0	0	0	0	0	0	1	0	0	0	0	84.5
10	0	0	3	0	0	0	0	0	0	78	0	2	0	1	0	0	0	0	0	0	0	0	0	92.9
11	0	0	1	0	4	0	0	0	0	1	75	0	0	0	0	0	0	0	0	0	0	0	0	92.6
12	0	0	0	1	0	0	0	0	7	0	1	96	8	0	0	0	0	0	11	0	0	0	0	77.4
13	0	0	0	0	0	0	0	0	1	0	0	1	75	0	0	0	0	11	0	0	0	0	0	85.2
14	0	0	0	0	0	0	0	0	1	0	8	5	0	55	0	0	0	0	0	0	0	0	0	79.7
15	0	0	0	0	0	0	0	0	0	0	0	0	0	2	94	0	0	3	0	0	1	0	0	94.0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	102	0	0	0	4	6	0	0	91.1
17	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	4	85	1	0	0	0	0	0	87.6
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	80	0	0	0	0	0	97.6
19	0	0	0	19	0	0	0	3	1	0	0	12	0	1	0	0	0	1	143	0	0	0	0	79.4
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	116	4	0	0	96.7
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	61	0	0	96.8
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	124	0	100.0
23	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	12	0	1	0	0	0	78	83.9
	75.6	83.3	97.4	77.4	92.9	98.0	74.3	67.2	83.5	95.1	88.2	76.2	90.4	93.2	97.9	87.9	84.2	83.3	84.1	96.7	81.3	100.0	100.0	
Average Accuracy: 87.81%																								

Table A.14: Experiment ALI7 TSTRA confusion matrix.

A.7 EXPERIMENT ALI8

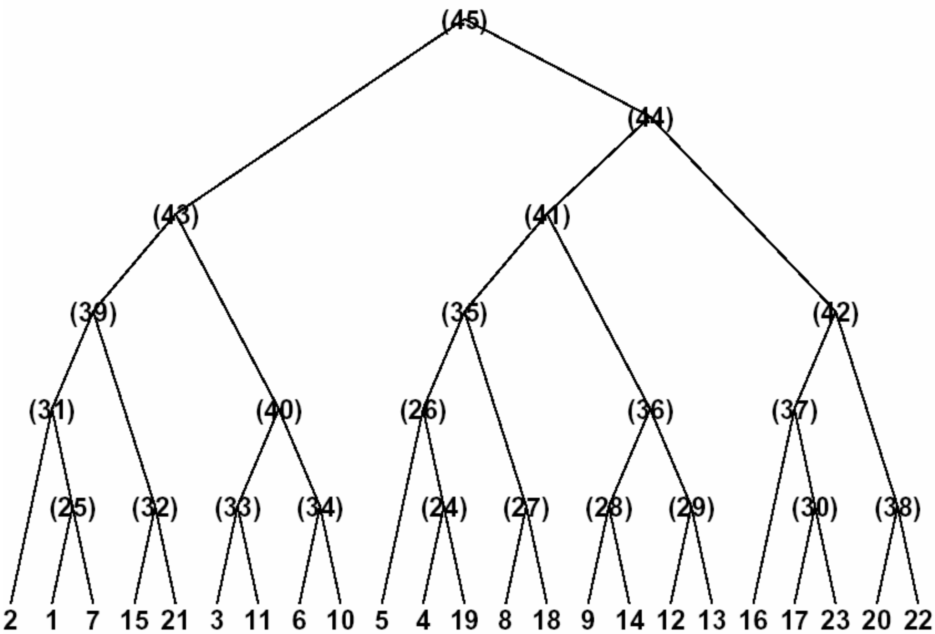

Figure A.9: Experiment ALI8 BHC class hierarchy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	59	6	0	0	0	0	6	0	0	0	0	0	0	0	1	0	0	0	0	0	5	0	0	76.6
2	5	86	0	0	0	1	0	1	0	0	0	0	0	2	1	0	0	0	0	0	1	0	0	88.7
3	0	1	143	0	9	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	91.7
4	0	0	0	105	1	0	0	7	3	0	0	0	1	0	0	0	0	0	24	0	0	0	0	74.5
5	0	0	0	1	111	0	0	1	0	0	2	0	0	0	0	0	1	0	0	0	0	0	0	95.7
6	0	0	0	0	1	97	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94.2
7	11	2	0	0	0	0	66	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	80.5
8	0	1	0	8	2	1	2	53	1	0	0	0	0	0	0	0	29	0	0	0	0	0	0	54.6
9	0	0	0	1	0	1	0	0	66	0	0	27	1	0	0	0	0	0	2	0	0	0	0	67.3
10	0	0	5	0	0	0	0	0	1	78	0	2	0	0	0	0	0	0	0	0	0	0	0	90.7
11	0	0	3	0	2	0	0	0	0	2	77	0	0	2	0	0	0	0	0	0	0	0	0	89.5
12	0	0	0	0	0	0	0	0	7	0	0	85	2	5	0	0	0	0	8	0	0	0	0	79.4
13	0	0	0	1	0	0	0	1	2	0	0	1	79	0	1	0	0	11	0	0	0	0	0	82.3
14	0	0	0	0	0	0	0	0	0	0	5	6	0	44	0	0	0	0	0	0	0	0	0	80.0
15	2	0	0	0	0	0	0	0	0	0	0	0	0	0	88	0	0	0	0	0	1	0	0	96.7
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	103	0	0	0	0	11	0	0	90.4
17	0	0	0	0	0	0	0	3	0	0	0	1	0	4	0	1	51	0	0	0	0	0	0	85.0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	85	0	0	0	0	0	81.0
19	0	0	0	8	1	0	0	0	0	0	0	4	0	1	0	0	0	0	136	0	0	0	0	90.7
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	1	0	0	99.2
21	1	0	0	0	0	0	0	0	0	0	0	0	0	0	3	11	0	0	0	0	56	0	0	78.9
22	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	121	2	97.6
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	3	76	95.0
	75.6	89.6	94.7	84.7	87.4	97.0	89.2	79.1	77.6	95.1	90.6	67.5	95.2	74.6	91.7	88.8	50.5	88.5	80.0	100.0	74.7	97.6	97.4	
Average Accuracy: 85.82%																								

Table A.15: Experiment ALI8 BHC confusion matrix.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	63	4	0	0	0	0	6	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	81.8
2	3	86	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	93.5
3	0	0	134	0	2	0	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	96.4
4	0	0	0	100	2	1	0	7	7	0	0	2	0	0	0	0	0	0	12	0	0	0	0	76.3
5	0	0	0	0	104	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	99.0
6	0	1	2	0	6	97	0	6	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	78.9
7	10	4	0	0	0	0	66	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	82.5
8	0	0	0	16	4	1	1	50	3	0	0	0	0	0	0	0	16	0	1	0	0	0	0	54.3
9	0	0	0	0	0	0	0	0	43	0	0	8	1	0	0	0	0	5	3	0	0	0	0	71.7
10	0	0	13	0	0	1	0	0	1	80	0	1	0	1	0	0	0	0	0	0	0	0	0	82.5
11	0	0	2	0	2	0	0	0	0	0	73	0	0	0	0	0	0	0	0	0	0	0	0	94.8
12	0	0	0	0	0	0	0	0	16	0	0	80	4	2	0	0	0	0	10	0	0	0	0	71.4
13	0	0	0	0	0	0	0	0	2	0	0	5	73	0	0	0	0	0	9	0	0	0	0	82.0
14	0	0	0	0	0	0	0	0	0	0	12	19	0	56	2	0	14	1	2	0	0	0	0	52.8
15	2	0	0	0	0	0	0	0	0	0	0	0	0	0	88	0	2	0	0	0	0	0	0	95.7
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	93	0	0	0	0	0	0	0	100.0
17	0	1	0	0	6	0	0	1	0	0	0	0	0	0	0	0	57	0	0	0	0	0	0	87.7
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	90	0	0	0	0	0	88.2
19	0	0	0	8	1	0	0	0	1	0	0	11	5	0	0	0	0	0	133	0	0	0	0	83.6
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	120	1	0	0	96.0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	19	0	0	0	0	74	0	0	78.7
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	124	8	93.9
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	70	98.6
	80.8	89.6	88.7	80.6	81.9	97.0	89.2	74.6	50.6	97.6	85.9	63.5	88.0	94.9	91.7	80.2	56.4	93.8	78.2	100.0	98.7	100.0	89.7	
Average Accuracy: 84.48%																								

Table A.16: Experiment ALI8 BHC FS confusion matrix.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	63	5	0	0	0	0	4	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	86.3
2	2	85	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	95.5
3	0	0	142	0	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	97.9
4	0	0	0	100	1	1	0	6	3	0	0	3	1	0	0	0	0	0	13	0	0	0	0	78.1
5	0	0	0	1	107	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	97.3
6	0	1	2	0	5	97	0	3	5	0	0	1	0	0	0	0	0	0	0	0	0	0	0	85.1
7	11	4	0	0	0	0	68	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	81.9
8	0	1	0	15	6	1	1	49	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	64.5
9	0	0	0	1	0	0	0	1	46	1	0	4	0	0	0	0	0	1	4	0	0	0	0	79.3
10	0	0	5	0	0	1	0	0	0	78	1	2	0	2	0	0	0	0	0	0	0	0	0	87.6
11	0	0	2	0	1	0	0	0	0	2	75	0	0	0	0	0	0	0	0	0	0	0	0	93.8
12	0	0	0	0	0	0	0	1	23	0	0	91	2	0	0	0	0	0	3	0	0	0	0	75.8
13	0	0	0	0	0	0	0	1	6	0	0	1	78	0	0	0	0	1	5	0	0	0	0	84.8
14	0	0	0	0	0	0	0	0	0	0	8	13	0	57	1	0	0	1	3	0	0	0	0	68.7
15	2	0	0	0	0	0	0	0	0	0	0	0	0	0	93	11	0	0	0	0	1	0	0	86.9
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	3	8	0	0	90.1
17	0	0	0	0	2	0	0	4	0	0	0	0	0	0	0	0	98	0	0	0	0	0	0	94.2
18	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	2	93	0	0	0	0	0	96.9
19	0	0	0	6	1	0	0	1	0	0	0	11	1	0	0	0	1	0	141	0	0	0	0	87.0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	117	1	0	0	99.2
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	5	0	0	0	0	65	0	0	91.5
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	124	3	97.6
23	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	75	97.4
	80.8	88.5	94.0	80.6	84.3	97.0	91.9	73.1	54.1	95.1	88.2	72.2	94.0	96.6	96.9	86.2	97.0	96.9	82.9	97.5	86.7	100.0	96.2	
Average Accuracy: 88.28%																								

Table A.17: Experiment ALI8 BHC TS-FS confusion matrix.

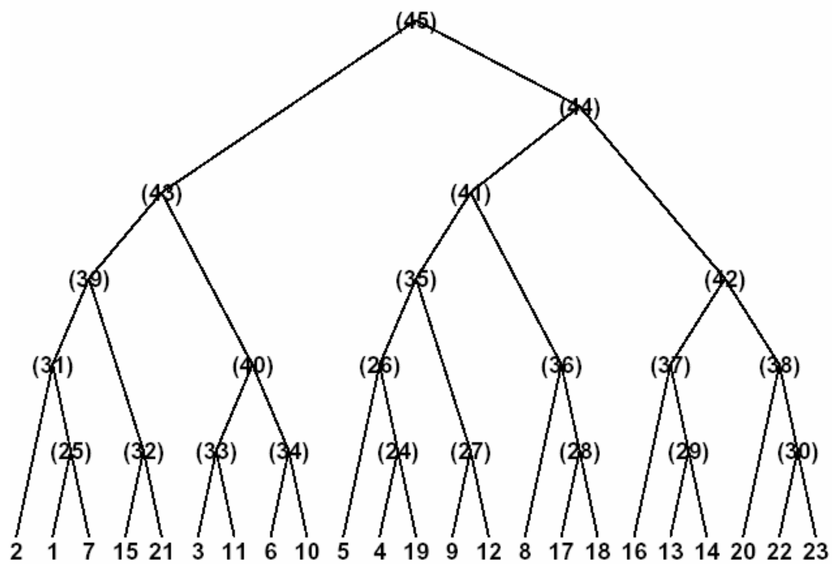

Figure A.10: Experiment ALI8 TS Build class hierarchy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	63	2	0	0	0	0	5	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	85.1
2	1	87	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97.8
3	0	0	143	0	2	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	97.3
4	0	0	0	107	2	0	0	10	2	0	0	1	1	0	0	0	0	0	11	0	0	0	0	79.9
5	0	0	0	0	113	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	99.1
6	0	1	0	0	5	98	0	3	5	0	0	2	0	0	0	0	0	0	0	0	0	0	0	86.0
7	13	5	0	0	0	0	68	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	79.1
8	0	0	0	5	2	0	1	41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	83.7
9	0	0	0	1	0	2	0	5	76	1	0	3	10	0	0	0	0	0	2	0	0	0	0	76.0
10	0	0	7	0	0	0	0	0	0	78	1	0	0	2	0	0	0	0	0	0	0	0	0	88.6
11	0	0	1	0	1	0	0	0	0	1	75	0	0	0	0	0	0	0	0	0	0	0	0	96.2
12	0	0	0	0	0	0	0	1	2	0	0	94	24	2	0	0	0	0	8	0	0	0	0	71.8
13	0	0	0	0	0	0	0	0	0	0	0	8	48	0	0	0	0	0	3	0	0	0	0	81.4
14	0	0	0	0	0	0	0	0	0	0	9	14	0	55	1	0	0	0	12	0	0	0	4	57.9
15	1	0	0	0	0	0	0	0	0	0	0	0	0	0	90	6	0	0	0	0	1	0	0	91.8
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	86	0	0	0	0	8	0	0	91.5
17	0	1	0	0	1	0	0	4	0	0	0	0	0	0	0	1	100	0	0	0	0	0	3	90.9
18	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	1	96	0	0	0	0	0	95.0
19	0	0	0	7	1	0	0	1	0	0	0	4	0	0	0	0	0	0	134	0	0	0	0	91.2
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	118	1	0	0	90.8
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	10	0	0	0	0	65	0	0	85.5
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	124	0	100.0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	2	0	0	71	94.7
	80.8	90.6	94.7	86.3	89.0	98.0	91.9	61.2	89.4	95.1	88.2	74.6	57.8	93.2	93.8	74.1	99.0	100.0	78.8	98.3	86.7	100.0	91.0	
Average Accuracy: 87.76%																								

Table A.18: Experiment ALI8 TS Build confusion matrix.

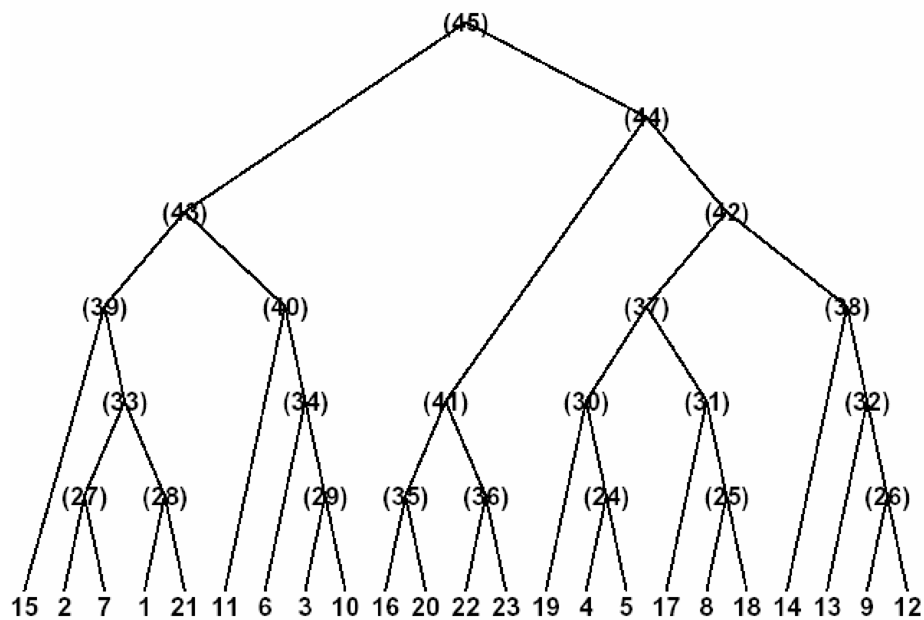

Figure A.11: Experiment ALI8 TSTRA class hierarchy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	61	6	0	0	0	0	10	0	0	0	0	0	0	0	1	2	0	0	0	0	1	0	0	75.3
2	4	83	0	0	0	1	0	0	0	0	0	0	0	2	0	2	0	0	0	0	0	0	0	90.2
3	0	1	149	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	97.4
4	0	0	0	109	1	1	0	6	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	80.1
5	0	0	0	0	108	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	99.1
6	0	0	0	0	1	97	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	93.3
7	11	5	0	0	0	0	63	3	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	75.9
8	0	1	0	5	5	0	1	50	1	0	0	0	0	0	0	0	3	0	1	0	0	0	0	74.6
9	0	0	0	1	0	1	0	1	74	0	0	7	0	0	0	0	0	0	2	0	0	0	0	86.0
10	0	0	1	0	0	0	0	0	0	79	0	2	0	0	0	0	0	0	0	0	0	0	0	96.3
11	0	0	1	0	10	0	0	0	0	0	78	0	0	2	0	0	0	0	0	0	0	0	0	85.7
12	0	0	0	0	0	0	0	0	2	0	0	103	0	0	0	0	0	0	10	0	0	0	0	89.6
13	0	0	0	0	0	0	0	0	2	0	0	6	83	0	0	0	0	10	1	0	0	0	0	81.4
14	0	0	0	0	0	0	0	0	0	0	5	5	0	54	0	0	0	0	0	0	0	0	0	84.4
15	1	0	0	0	0	0	0	0	0	0	0	0	0	0	92	0	0	0	0	0	0	0	0	98.9
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	104	0	0	0	1	11	0	0	89.7
17	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	2	92	0	0	0	0	0	0	93.9
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	86	0	0	0	0	0	94.5
19	0	0	0	9	2	0	0	1	0	0	2	3	0	1	0	0	0	0	137	0	0	0	1	87.8
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	119	1	0	0	99.2
21	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	6	0	0	0	0	62	0	0	88.6
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	124	0	100.0
23	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	77	96.3
	78.2	86.5	98.7	87.9	85.0	97.0	85.1	74.6	87.1	96.3	91.8	81.7	100.0	91.5	95.8	89.7	91.1	89.6	80.6	99.2	82.7	100.0	98.7	
Average Accuracy: 90.10%																								

Table A.19: Experiment ALI8 TSTRA confusion matrix.

A.8 EXPERIMENT ALI9

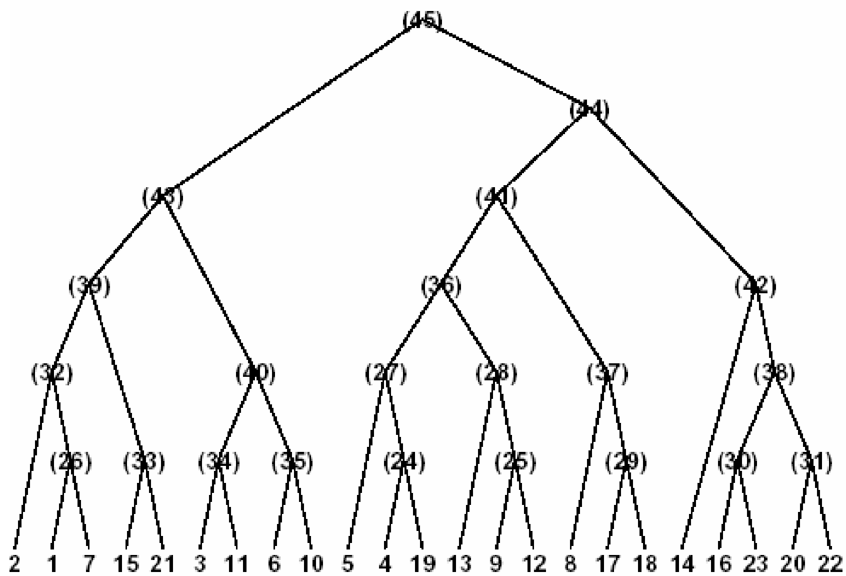

Figure A.12: Experiment ALI9 BHC class hierarchy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	62	10	0	0	0	0	6	0	0	0	0	0	0	0	3	0	0	0	0	0	2	0	0	74.7
2	1	85	1	0	0	2	5	1	0	0	0	0	0	2	1	0	0	0	0	0	3	0	0	84.2
3	1	0	149	0	6	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	92.5
4	0	0	0	105	1	0	0	16	1	0	0	1	0	0	0	0	0	0	15	0	0	0	0	75.5
5	0	0	0	0	115	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97.5
6	0	0	0	0	0	98	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	98.0
7	10	0	0	0	0	0	61	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	84.7
8	0	0	0	3	2	0	2	35	0	0	0	0	0	0	0	0	2	3	1	0	0	0	0	72.9
9	0	0	0	0	0	0	0	1	78	0	0	3	2	0	0	0	0	0	4	0	0	0	0	88.6
10	0	0	0	0	0	0	0	0	1	76	0	1	0	2	0	0	0	0	0	0	0	0	0	95.0
11	0	0	1	0	2	0	0	0	0	1	73	0	0	1	0	0	0	0	0	0	0	0	0	93.6
12	0	0	0	0	0	0	0	0	1	0	1	109	3	8	0	0	0	0	4	0	0	0	0	86.5
13	0	0	0	0	0	0	0	0	1	0	0	4	75	0	0	0	0	2	0	0	0	0	0	91.5
14	0	0	0	0	0	0	0	0	0	0	11	4	1	45	0	0	0	0	16	0	0	0	0	58.4
15	1	0	0	0	0	0	0	0	0	0	0	0	0	1	86	0	0	2	0	0	0	0	0	95.6
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	79	0	0	0	3	13	0	0	83.2
17	0	0	0	1	0	0	0	6	0	0	0	0	0	0	0	22	95	1	0	0	0	0	0	76.0
18	0	1	0	2	0	0	0	3	0	0	0	0	1	0	1	0	0	88	0	0	0	0	0	91.7
19	0	0	0	13	1	0	0	2	1	0	0	4	1	0	0	0	0	0	130	0	0	0	0	85.5
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	117	0	0	0	100.0
21	3	0	0	0	0	0	0	0	0	0	0	0	0	0	4	15	0	0	0	0	57	0	0	72.2
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	123	1	99.2
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	1	77	93.9
	79.5	88.5	98.7	84.7	90.6	98.0	82.4	52.2	91.8	92.7	85.9	86.5	90.4	76.3	89.6	68.1	94.1	91.7	76.5	97.5	76.0	99.2	98.7	
Average Accuracy: 87.25%																								

Table A.20: Experiment ALI9 BHC confusion matrix.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	44	8	0	0	0	0	9	0	0	0	0	0	0	0	4	3	0	0	0	0	2	0	0	62.9
2	2	85	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94.4
3	0	0	139	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	97.2
4	0	0	0	102	1	0	0	17	0	0	0	2	0	0	0	0	0	0	15	0	0	0	0	74.5
5	0	0	0	0	113	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	97.4
6	0	0	5	0	4	98	0	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	87.5
7	15	2	0	0	0	0	60	0	0	0	0	0	0	0	1	6	0	0	0	0	0	0	0	71.4
8	0	0	0	5	4	0	3	39	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	73.6
9	0	0	0	4	0	0	0	1	72	1	0	16	0	0	0	0	0	1	4	0	0	0	0	72.7
10	0	0	7	0	0	1	0	0	2	76	0	0	0	1	0	0	0	0	0	0	0	0	0	87.4
11	0	0	0	0	3	0	0	0	0	3	78	0	0	1	0	0	0	0	0	0	0	0	0	91.8
12	0	0	0	0	0	0	0	0	6	1	0	89	3	3	0	0	0	0	7	0	0	0	0	81.7
13	0	0	0	0	0	0	0	1	2	0	0	1	72	0	0	0	0	1	2	0	0	0	0	91.1
14	0	0	0	0	0	0	0	0	0	0	7	9	3	54	0	0	0	0	0	0	0	0	0	74.0
15	11	0	0	0	0	0	0	0	0	0	0	0	0	0	84	0	0	1	0	0	0	0	0	87.5
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	92	0	0	0	3	3	0	0	93.9
17	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	6	92	0	0	0	0	0	0	92.0
18	0	1	0	6	0	0	0	1	0	0	0	0	1	0	5	0	5	93	0	0	0	0	0	83.0
19	0	0	0	6	1	0	0	1	0	0	0	8	3	0	0	0	0	0	139	0	0	0	0	88.0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	117	1	0	0	99.2
21	6	0	0	0	0	0	0	0	0	0	0	0	0	0	2	9	0	0	0	0	69	0	0	80.2
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	122	0	100.0
23	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	2	0	3	0	0	2	77	89.5
	56.4	88.5	92.1	82.3	89.0	98.0	81.1	58.2	84.7	92.7	91.8	70.6	86.7	91.5	87.5	79.3	91.1	96.9	81.8	97.5	92.0	98.4	98.7	
Average Accuracy: 86.73%																								

Table A.21: Experiment ALI9 BHC FS confusion matrix.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	53	10	0	0	0	1	5	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	74.6
2	5	85	0	0	1	0	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	88.5
3	0	0	149	0	3	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	97.4
4	0	0	0	104	0	0	0	8	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	83.2
5	0	0	0	0	113	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
6	0	0	0	0	0	98	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97.0
7	12	0	0	0	0	0	64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	84.2
8	0	0	0	3	6	0	2	42	0	0	0	0	0	0	0	0	2	0	1	0	0	0	0	75.0
9	0	0	0	1	0	0	0	7	75	0	0	3	0	0	0	0	0	0	1	0	0	0	0	86.2
10	0	0	2	0	0	1	0	0	2	77	0	0	0	1	0	0	0	0	0	0	0	0	0	92.8
11	0	0	0	0	2	0	0	0	0	3	77	0	0	3	0	0	0	0	0	0	0	0	0	90.6
12	0	0	0	0	0	0	0	0	4	1	0	109	5	2	0	0	0	1	6	0	0	0	0	85.2
13	0	0	0	0	0	0	0	1	1	0	0	1	78	0	0	0	0	0	0	0	0	0	0	96.3
14	0	0	0	0	0	0	0	0	0	0	8	6	0	53	1	0	0	0	0	0	0	0	0	77.9
15	6	0	0	0	0	0	1	0	0	0	0	0	0	0	90	0	0	1	0	0	0	0	0	91.8
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96	0	0	0	3	11	0	0	87.3
17	0	1	0	1	1	0	0	2	0	0	0	0	0	0	0	2	96	3	0	0	0	0	0	90.6
18	0	0	0	4	0	0	0	1	0	0	0	0	0	0	1	0	2	90	0	0	0	0	0	91.8
19	0	0	0	11	1	0	0	3	0	0	0	7	0	0	0	0	0	1	144	0	0	0	0	86.2
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	117	0	0	0	100.0
21	2	0	0	0	0	0	0	0	0	0	0	0	0	0	2	18	0	0	0	0	64	0	0	74.4
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	122	0	100.0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	5	0	0	2	78	90.7
	67.9	88.5	98.7	83.9	89.0	98.0	86.5	62.7	88.2	93.9	90.6	86.5	94.0	89.8	93.8	82.8	95.0	93.8	84.7	97.5	85.3	98.4	100.0	
Average Accuracy: 89.67%																								

Table A.22: Experiment ALI9 BHC TS-FS confusion matrix.

Appendix B

Selected Hyperion Data Class Hierarchies and Confusion Matrices

B.1 Experiment HYP12

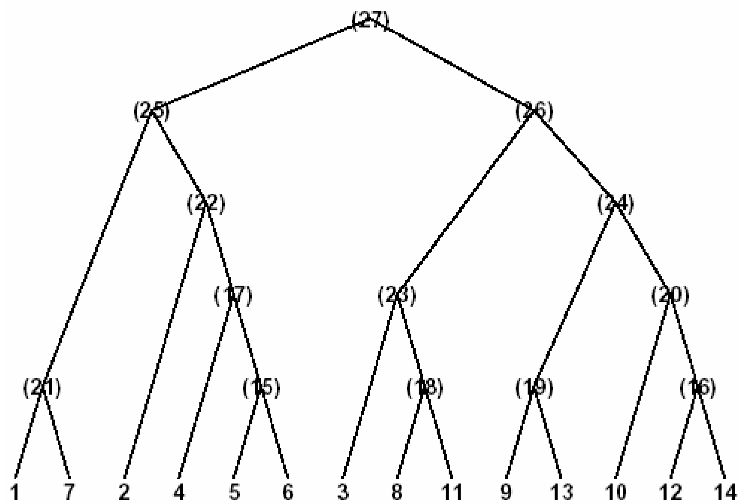

Figure B.1: Experiment HYP12 BHC class hierarchy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	135	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	0	50	0	0	1	0	0	0	0	0	0	0	0	0	98.0
3	0	0	122	0	0	0	1	1	0	0	0	0	0	0	98.4
4	0	0	1	107	2	0	0	0	2	0	0	0	0	0	95.5
5	0	0	0	0	121	19	0	1	3	0	0	0	0	0	84.0
6	0	0	0	0	10	90	0	0	26	0	0	0	1	0	70.9
7	0	0	0	0	0	0	127	0	0	0	0	0	0	1	99.2
8	0	0	1	0	0	0	0	95	0	0	0	0	0	2	96.9
9	0	0	0	0	0	12	1	0	117	6	0	1	0	9	80.1
10	0	0	1	0	0	1	0	1	6	107	7	1	12	0	78.7
11	0	0	0	0	0	0	0	3	0	6	143	0	1	2	92.3
12	0	0	0	0	0	0	0	0	0	0	1	79	6	3	88.8
13	0	0	0	0	0	12	0	0	3	5	0	9	114	0	79.7
14	0	0	0	0	0	0	0	0	0	0	1	0	0	30	96.8
	100.0	100.0	97.6	100.0	90.3	67.2	98.4	94.1	74.5	86.3	94.1	87.8	85.1	63.8	
Average Accuracy: 88.76%															

Table B.1: Experiment HYP12 BHC confusion matrix.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	135	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	0	49	0	0	1	0	0	0	0	0	0	0	0	0	98.0
3	0	0	122	1	0	0	0	0	0	6	0	0	0	0	94.6
4	0	0	1	104	4	0	1	0	0	0	0	0	0	0	94.5
5	0	1	1	1	120	18	4	0	0	0	0	0	0	0	82.8
6	0	0	0	0	3	54	0	0	9	0	0	4	0	0	77.1
7	0	0	0	0	1	0	123	0	0	0	0	0	0	0	99.2
8	0	0	1	1	0	0	0	95	0	2	1	0	0	1	94.1
9	0	0	0	0	5	62	0	0	144	5	0	0	0	0	66.7
10	0	0	0	0	0	0	1	0	0	71	7	0	0	0	89.9
11	0	0	0	0	0	0	0	1	0	37	141	0	0	0	78.8
12	0	0	0	0	0	0	0	4	0	0	0	78	11	0	83.9
13	0	0	0	0	0	0	0	0	4	3	0	8	119	0	88.8
14	0	0	0	0	0	0	0	1	0	0	3	0	4	46	85.2
	100.0	98.0	97.6	97.2	89.6	40.3	95.3	94.1	91.7	57.3	92.8	86.7	88.8	97.9	
Average Accuracy: 86.53%															

Table B.2: Experiment HYP12 BHC FS confusion matrix.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	135	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	0	49	0	0	1	0	0	0	0	0	0	0	0	0	98.0
3	0	0	123	5	0	0	0	0	0	2	0	0	0	0	94.6
4	0	0	0	100	2	0	2	0	0	0	0	0	0	0	96.2
5	0	1	0	1	123	19	5	0	1	0	0	1	0	0	81.5
6	0	0	0	0	3	94	0	0	21	0	0	2	0	0	78.3
7	0	0	0	0	0	0	121	0	0	0	0	0	0	0	100.0
8	0	0	1	1	1	0	0	98	0	2	1	0	0	2	92.5
9	0	0	0	0	1	21	0	0	131	2	0	0	0	0	84.5
10	0	0	1	0	1	0	1	0	0	93	10	0	0	0	87.7
11	0	0	0	0	1	0	0	1	0	23	140	0	0	2	83.8
12	0	0	0	0	1	0	0	1	0	0	0	81	8	0	89.0
13	0	0	0	0	0	0	0	0	4	2	1	6	122	0	90.4
14	0	0	0	0	0	0	0	1	0	0	0	0	4	43	89.6
	100.0	98.0	98.4	93.5	91.8	70.1	93.8	97.0	83.4	75.0	92.1	90.0	91.0	91.5	
Average Accuracy: 89.75%															

Table B.3: Experiment HYP12 BHC TS-FS confusion matrix.

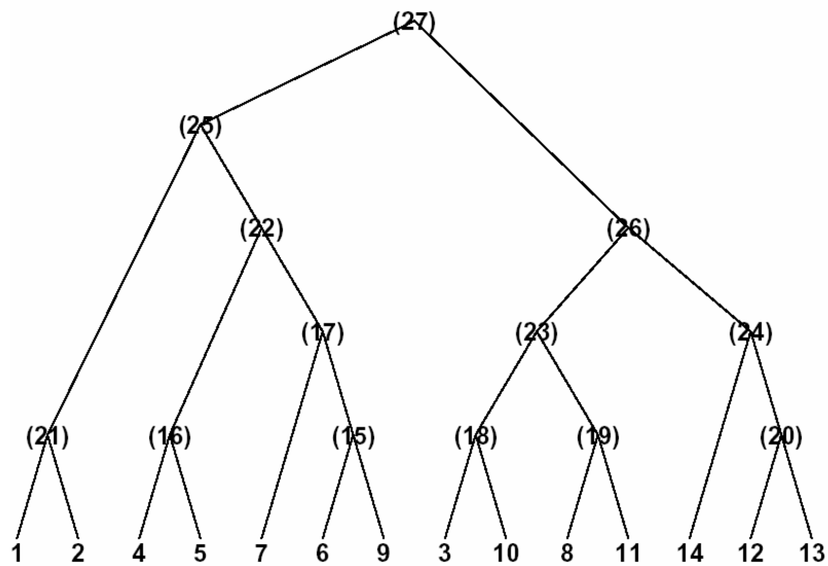

Figure B.2: Experiment HYP12 TS Build class hierarchy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	135	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	0	50	0	0	4	0	0	0	0	0	0	0	0	0	92.6
3	0	0	123	1	0	1	0	0	0	5	0	0	0	0	94.6
4	0	0	0	103	2	0	0	0	0	0	0	0	0	0	98.1
5	0	0	0	2	117	21	0	0	1	0	0	0	0	0	83.0
6	0	0	0	0	9	100	1	0	17	0	0	0	0	0	78.7
7	0	0	1	1	0	1	128	0	0	0	0	0	0	0	97.7
8	0	0	0	0	0	0	0	98	0	10	0	0	1	0	89.9
9	0	0	0	0	0	4	0	0	136	0	0	0	0	0	97.1
10	0	0	1	0	0	1	0	0	1	101	8	0	2	0	88.6
11	0	0	0	0	1	0	0	2	0	7	144	0	0	0	93.5
12	0	0	0	0	1	1	0	0	0	0	0	88	2	1	94.6
13	0	0	0	0	0	5	0	1	2	1	0	2	128	5	88.9
14	0	0	0	0	0	0	0	0	0	0	0	0	1	41	97.6
	100.0	100.0	98.4	96.3	87.3	74.6	99.2	97.0	86.6	81.5	94.7	97.8	95.5	87.2	
Average Accuracy: 92.16%															

Table B.4: Experiment HYP12 TS Build confusion matrix.

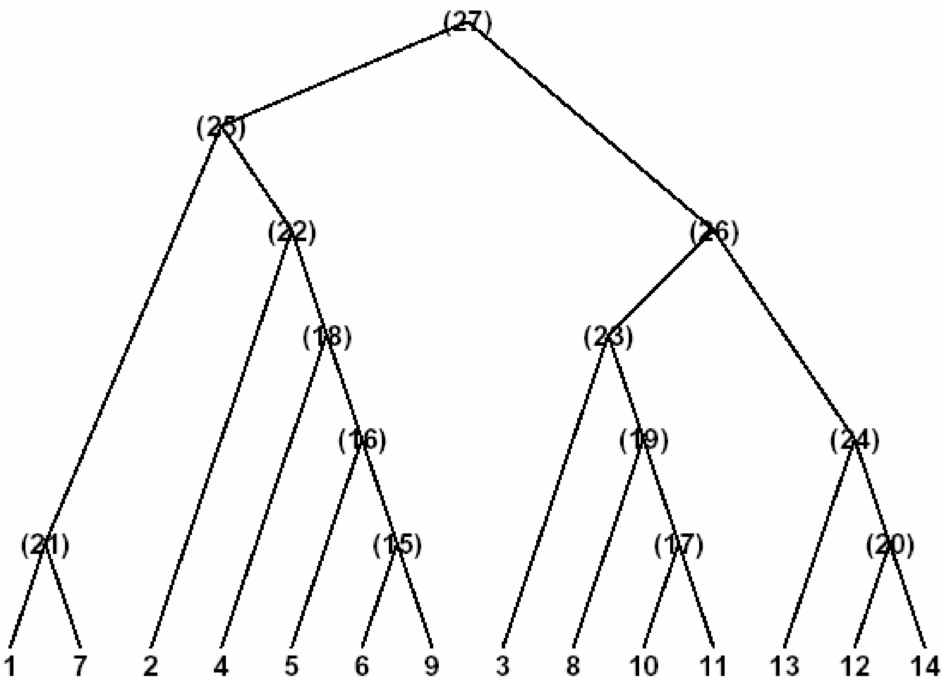

Figure B.3: Experiment HYP12 TSTRA class hierarchy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	135	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	0	50	0	0	1	0	0	0	0	0	0	0	0	0	98.0
3	0	0	120	0	0	0	1	0	0	0	0	0	0	0	99.2
4	0	0	1	106	3	0	0	0	0	0	0	0	0	0	96.4
5	0	0	1	1	128	11	1	0	2	0	0	0	0	0	88.9
6	0	0	0	0	2	109	0	0	9	0	0	0	2	0	89.3
7	0	0	0	0	0	0	127	0	0	0	0	0	0	6	95.5
8	0	0	1	0	0	0	0	99	0	0	0	0	0	3	96.1
9	0	0	1	0	0	8	0	0	138	1	0	0	0	0	93.2
10	0	0	1	0	0	0	0	1	1	114	1	0	1	0	95.8
11	0	0	0	0	0	0	0	1	0	9	149	0	1	0	93.1
12	0	0	0	0	0	2	0	0	0	0	0	68	5	4	86.1
13	0	0	0	0	0	4	0	0	7	0	2	22	125	4	76.2
14	0	0	0	0	0	0	0	0	0	0	0	0	0	30	100.0
	100.0	100.0	96.0	99.1	95.5	81.3	98.4	98.0	87.9	91.9	98.0	75.6	93.3	63.8	
Average Accuracy: 92.53%															

Table B.5: Experiment HYP12 TSTRA confusion matrix.

B.2 Experiment HYP13

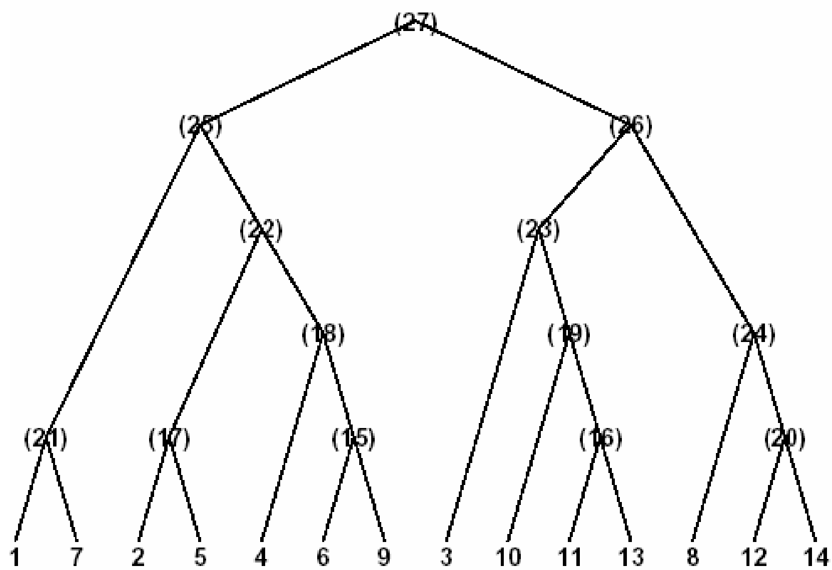

Figure B.4: Experiment HYP13 BHC class hierarchy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	135	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	0	42	0	0	9	1	0	0	0	0	0	0	0	0	80.8
3	0	0	114	1	0	0	0	3	0	0	0	0	0	0	96.6
4	0	0	8	103	4	0	0	0	0	0	0	0	0	0	89.6
5	0	8	0	3	114	7	0	0	1	0	0	2	0	1	83.8
6	0	0	0	0	7	110	0	0	5	0	0	2	1	0	88.0
7	0	0	0	0	0	0	128	0	0	0	0	0	0	11	92.1
8	0	0	2	0	0	1	0	97	0	1	2	1	2	1	90.7
9	0	0	1	0	0	8	1	0	144	0	0	2	3	0	90.6
10	0	0	0	0	0	1	0	0	5	113	7	1	9	10	77.4
11	0	0	0	0	0	0	0	1	0	9	140	0	2	0	92.1
12	0	0	0	0	0	0	0	0	0	0	2	65	8	4	82.3
13	0	0	0	0	0	6	0	0	2	1	1	17	109	8	75.7
14	0	0	0	0	0	0	0	0	0	0	0	0	0	12	100.0
	100.0	84.0	91.2	96.3	85.1	82.1	99.2	96.0	91.7	91.1	92.1	72.2	81.3	25.5	
Average Accuracy: 88.08%															

Table B.6: Experiment HYP13 BHC confusion matrix.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	135	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	0	50	0	0	3	0	0	0	0	0	0	0	0	0	94.3
3	0	0	113	1	0	0	0	0	0	0	0	0	0	0	99.1
4	0	0	8	105	6	0	0	0	0	0	0	0	0	0	88.2
5	0	0	0	0	116	9	1	0	0	0	0	0	0	0	92.1
6	0	0	0	0	8	96	0	0	14	0	0	0	0	0	81.4
7	0	0	0	0	0	0	127	0	0	1	0	0	0	0	99.2
8	0	0	0	0	0	0	0	91	0	0	1	0	1	0	97.8
9	0	0	1	0	0	21	1	0	122	8	0	0	1	0	79.2
10	0	0	1	0	0	0	0	3	7	103	0	0	0	0	90.4
11	0	0	2	0	0	0	0	5	0	9	151	0	0	0	90.4
12	0	0	0	1	0	2	0	2	0	0	0	78	1	1	91.8
13	0	0	0	0	1	6	0	0	14	3	0	12	129	0	78.2
14	0	0	0	0	0	0	0	0	0	0	0	0	2	46	95.8
	100.0	100.0	90.4	98.1	86.6	71.6	98.4	90.1	77.7	83.1	99.3	86.7	96.3	97.9	
Average Accuracy: 90.30%															

Table B.7: Experiment HYP13 BHC FS confusion matrix.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	135	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	0	50	0	0	1	0	0	0	0	0	0	0	0	0	98.0
3	0	0	122	2	0	0	0	0	0	3	0	0	1	0	95.3
4	0	0	2	104	2	0	1	0	0	0	0	0	0	0	95.4
5	0	0	0	0	120	9	2	0	0	0	0	1	0	0	90.9
6	0	0	0	0	10	112	0	0	13	0	0	0	0	0	83.0
7	0	0	0	1	0	0	125	0	0	0	0	0	0	0	99.2
8	0	0	0	0	0	0	0	95	0	0	2	0	2	0	96.0
9	0	0	0	0	0	7	0	0	141	4	0	0	1	0	92.2
10	0	0	1	0	0	0	0	1	1	107	1	0	1	0	95.5
11	0	0	0	0	0	0	1	3	0	7	147	0	0	0	93.0
12	0	0	0	0	1	0	0	2	0	0	0	75	5	1	89.3
13	0	0	0	0	0	6	0	0	2	3	0	14	123	0	83.1
14	0	0	0	0	0	0	0	0	0	0	2	0	1	46	93.9
	100.0	100.0	97.6	97.2	89.6	83.6	96.9	94.1	89.8	86.3	96.7	83.3	91.8	97.9	
Average Accuracy: 92.77%															

Table B.8: Experiment HYP13 BHC TS-FS confusion matrix.

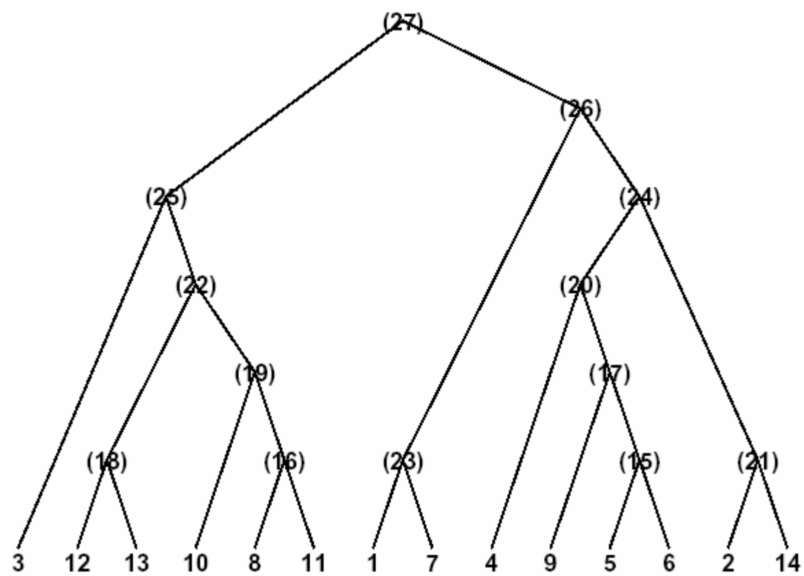

Figure B.5: Experiment HYP13 TSTRA class hierarchy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	135	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	0	49	0	0	0	0	0	0	0	0	0	0	0	0	100.0
3	0	0	122	1	0	1	0	1	0	0	1	0	0	0	96.8
4	0	0	2	104	2	0	0	0	0	0	0	0	0	0	96.3
5	0	1	0	2	118	13	1	1	0	0	0	0	0	2	85.5
6	0	0	0	0	13	104	0	0	10	0	0	0	4	0	79.4
7	0	0	0	0	0	0	127	0	0	0	0	0	0	0	100.0
8	0	0	1	0	0	0	0	95	0	1	4	0	0	0	94.1
9	0	0	0	0	1	8	1	0	139	0	0	0	1	0	92.7
10	0	0	0	0	0	0	0	1	2	114	5	0	0	0	93.4
11	0	0	0	0	0	0	0	3	0	9	141	0	0	1	91.6
12	0	0	0	0	0	3	0	0	2	0	0	79	18	0	77.5
13	0	0	0	0	0	5	0	0	4	0	1	11	111	0	84.1
14	0	0	0	0	0	0	0	0	0	0	0	0	0	44	100.0
	100.0	98.0	97.6	97.2	88.1	77.6	98.4	94.1	88.5	91.9	92.8	87.8	82.8	93.6	
Average Accuracy: 91.54%															

Table B.9: Experiment HYP13 TSTRA confusion matrix.

B.3 EXPERIMENT HYP17

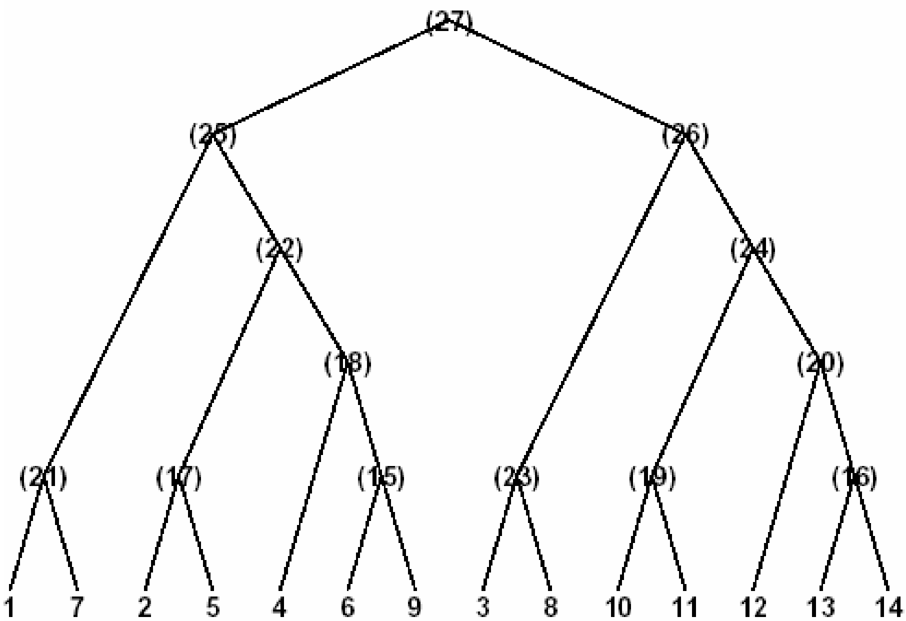

Figure B.6: Experiment HYP17 BHC class hierarchy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	135	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	0	39	0	0	0	0	0	0	0	0	0	0	0	0	100.0
3	0	0	119	1	0	0	0	1	0	0	0	0	0	0	98.3
4	0	0	4	105	6	0	0	0	0	0	0	0	0	0	91.3
5	0	11	0	1	117	10	0	0	1	0	0	0	0	0	83.6
6	0	0	0	0	11	108	1	0	8	0	0	0	1	0	83.7
7	0	0	0	0	0	0	127	0	0	0	0	0	0	0	100.0
8	0	0	0	0	0	0	0	99	0	0	0	0	0	0	100.0
9	0	0	1	0	0	11	1	0	142	1	0	0	1	0	90.4
10	0	0	1	0	0	0	0	1	1	116	11	0	2	1	87.2
11	0	0	0	0	0	0	0	0	0	6	140	0	0	1	95.2
12	0	0	0	0	0	1	0	0	0	0	1	80	13	3	81.6
13	0	0	0	0	0	4	0	0	5	1	0	10	117	3	83.6
14	0	0	0	0	0	0	0	0	0	0	0	0	0	39	100.0
	100.0	78.0	95.2	98.1	87.3	80.6	98.4	98.0	90.4	93.5	92.1	88.9	87.3	83.0	
Average Accuracy: 91.60%															

Table B.10: Experiment HYP17 BHC confusion matrix.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	135	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	0	50	0	0	0	0	0	0	0	0	0	0	0	0	100.0
3	0	0	90	0	0	0	0	0	0	19	0	0	0	0	82.6
4	0	0	30	104	5	0	2	0	0	0	0	0	0	0	73.8
5	0	0	0	1	79	20	1	0	0	0	0	0	0	0	78.2
6	0	0	1	0	18	104	2	0	9	0	0	0	3	0	75.9
7	0	0	1	0	0	0	123	0	0	0	0	0	0	0	99.2
8	0	0	1	0	0	0	0	96	0	1	4	0	0	0	94.1
9	0	0	2	0	1	1	0	0	140	7	0	0	1	0	92.1
10	0	0	0	0	0	0	0	2	5	90	8	0	5	0	81.8
11	0	0	0	0	0	0	1	0	0	5	140	0	0	0	95.9
12	0	0	0	0	1	3	0	0	0	0	0	86	12	0	84.3
13	0	0	0	0	10	5	0	0	1	2	0	4	113	0	83.7
14	0	0	0	2	20	1	0	3	2	0	0	0	0	47	62.7
	100.0	100.0	72.0	97.2	59.0	77.6	95.3	95.0	89.2	72.6	92.1	95.6	84.3	100.0	
Average Accuracy: 86.29%															

Table B.11: Experiment HYP17 BHC FS confusion matrix.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	135	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	0	50	0	0	0	0	0	0	0	0	0	0	0	0	100.0
3	0	0	122	1	0	0	0	0	0	8	0	0	0	0	93.1
4	0	0	1	105	2	0	0	0	0	0	0	0	0	0	97.2
5	0	0	0	1	111	9	1	0	0	0	0	0	0	0	91.0
6	0	0	0	0	19	121	0	0	11	0	0	0	1	0	79.6
7	0	0	0	0	0	0	128	0	0	0	0	0	0	0	100.0
8	0	0	1	0	0	0	0	95	0	1	3	0	0	0	95.0
9	0	0	0	0	0	2	0	0	141	6	0	0	0	0	94.6
10	0	0	0	0	0	0	0	1	3	101	9	0	6	0	84.2
11	0	0	1	0	0	0	0	0	0	8	140	0	0	0	94.0
12	0	0	0	0	0	0	0	0	0	0	0	89	8	0	91.8
13	0	0	0	0	0	2	0	0	1	0	0	1	119	0	96.7
14	0	0	0	0	2	0	0	5	1	0	0	0	0	47	85.5
	100.0	100.0	97.6	98.1	82.8	90.3	99.2	94.1	89.8	81.5	92.1	98.9	88.8	100.0	
Average Accuracy: 92.90%															

Table B.12: Experiment HYP17 BHC TS-FS confusion matrix.

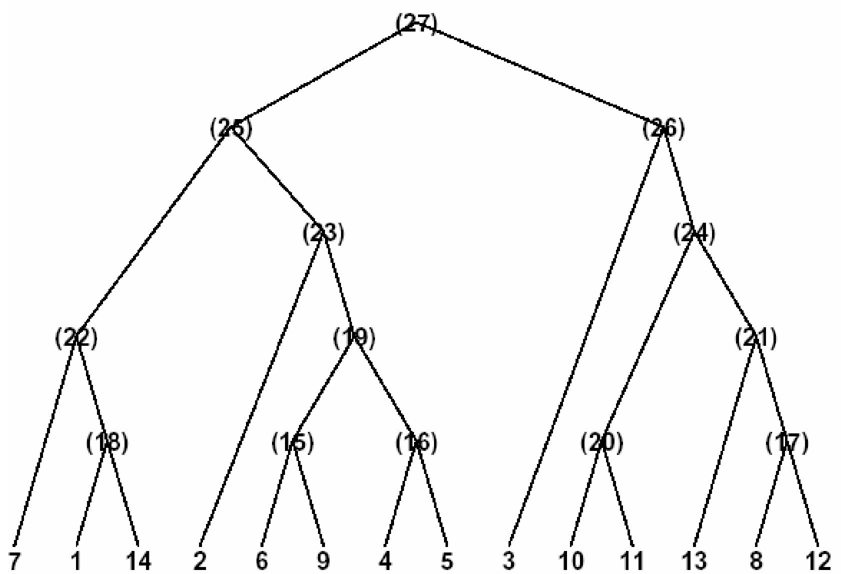

Figure B.7: Experiment HYP17 TSTRA class hierarchy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	133	0	0	0	0	0	1	0	0	0	0	0	0	0	99.3
2	0	49	0	0	0	0	0	0	0	0	0	0	0	0	100.0
3	0	0	122	1	0	0	0	0	0	0	0	0	1	0	98.4
4	0	0	0	99	2	0	0	1	0	0	0	0	0	0	97.1
5	0	1	0	5	124	7	2	0	0	0	0	0	0	0	89.2
6	0	0	0	0	8	111	0	0	8	0	0	0	1	1	86.0
7	2	0	0	1	0	0	126	0	1	0	0	0	0	7	92.0
8	0	0	0	0	0	1	0	97	0	0	1	0	0	0	98.0
9	0	0	1	0	0	10	0	0	143	2	0	1	2	0	89.9
10	0	0	2	0	0	0	0	3	2	116	11	0	3	0	84.7
11	0	0	0	0	0	0	0	0	0	5	139	0	0	3	94.6
12	0	0	0	0	0	0	0	0	0	0	1	80	10	2	86.0
13	0	0	0	1	0	5	0	0	3	1	0	9	117	3	84.2
14	0	0	0	0	0	0	0	0	0	0	0	0	0	31	100.0
	98.5	98.0	97.6	92.5	92.5	82.8	97.7	96.0	91.1	93.5	91.4	88.9	87.3	66.0	
Average Accuracy: 91.85%															

Table B.13: Experiment HYP17 TSTRA confusion matrix.

B.4 EXPERIMENT HYP18

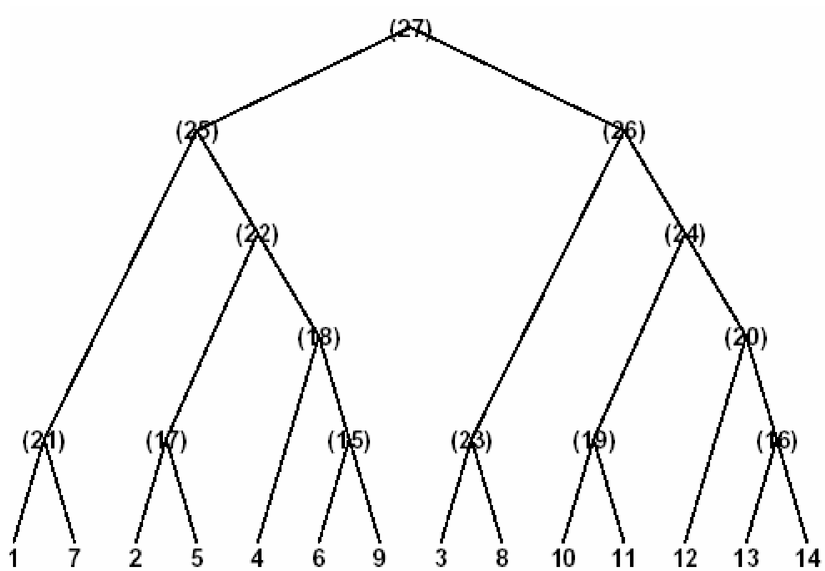

Figure B.8: Experiment HYP18 BHC class hierarchy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	135	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	0	40	0	0	2	0	0	0	0	0	0	0	0	0	95.2
3	0	0	116	0	0	0	1	0	0	0	0	0	0	0	99.1
4	0	0	6	106	8	0	0	0	0	0	0	0	0	0	88.3
5	0	10	0	1	112	5	1	0	2	0	0	0	0	0	85.5
6	0	0	0	0	11	114	0	0	5	0	0	0	2	0	86.4
7	0	0	0	0	0	0	127	0	0	0	0	0	0	0	100.0
8	0	0	2	0	1	0	0	101	0	0	0	1	0	1	95.3
9	0	0	0	0	0	12	0	0	142	0	0	0	1	0	91.6
10	0	0	0	0	0	0	0	0	4	114	5	0	4	1	89.1
11	0	0	1	0	0	0	0	0	0	9	146	0	0	1	93.0
12	0	0	0	0	0	0	0	0	0	0	0	84	18	2	80.8
13	0	0	0	0	0	2	0	0	3	1	1	5	109	0	90.1
14	0	0	0	0	0	1	0	0	1	0	0	0	0	42	95.5
	100.0	80.0	92.8	99.1	83.6	85.1	98.4	100.0	90.4	91.9	96.1	93.3	81.3	89.4	
Average Accuracy: 91.91%															

Table B.14: Experiment HYP18 BHC confusion matrix.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	135	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	0	50	0	0	4	0	1	0	0	0	0	0	0	0	90.9
3	0	0	116	0	0	0	0	0	0	16	0	0	0	0	87.9
4	0	0	8	107	7	0	1	0	0	0	0	0	0	0	87.0
5	0	0	0	0	111	8	0	0	0	0	0	0	0	0	93.3
6	0	0	0	0	9	112	0	0	5	0	0	0	0	0	88.9
7	0	0	0	0	1	0	123	0	0	0	0	0	0	0	99.2
8	0	0	1	0	0	0	0	99	0	1	1	0	0	1	96.1
9	0	0	0	0	1	12	4	0	131	8	0	0	0	0	84.0
10	0	0	0	0	0	0	0	0	1	91	7	0	5	0	87.5
11	0	0	0	0	0	0	0	0	0	8	144	0	1	0	94.1
12	0	0	0	0	0	0	0	0	0	0	0	85	7	0	92.4
13	0	0	0	0	1	2	0	2	20	0	0	5	120	0	80.0
14	0	0	0	0	0	0	0	0	0	0	0	0	1	46	97.9
	100.0	100.0	92.8	100.0	82.8	83.6	95.3	98.0	83.4	73.4	94.7	94.4	89.6	97.9	
Average Accuracy: 90.80%															

Table B.15: Experiment HYP18 BHC FS confusion matrix.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	135	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	0	49	0	0	1	0	0	0	0	0	0	0	0	0	98.0
3	0	0	120	0	0	0	0	0	0	3	0	0	0	0	97.6
4	0	0	3	103	7	0	2	0	0	0	0	0	0	0	89.6
5	0	1	0	4	113	4	0	0	1	0	0	0	0	0	91.9
6	0	0	0	0	10	121	1	0	7	0	0	0	7	0	82.9
7	0	0	0	0	0	0	122	0	0	0	0	0	0	0	100.0
8	0	0	1	0	0	0	0	97	0	0	0	0	0	0	99.0
9	0	0	0	0	2	7	2	0	143	7	0	0	1	0	88.3
10	0	0	1	0	0	0	0	0	5	101	3	0	4	0	88.6
11	0	0	0	0	0	0	0	0	0	10	149	0	1	0	93.1
12	0	0	0	0	0	0	0	1	0	0	0	82	8	0	90.1
13	0	0	0	0	0	2	0	0	1	2	0	8	113	0	89.7
14	0	0	0	0	1	0	2	3	0	1	0	0	0	47	87.0
	100.0	98.0	96.0	96.3	84.3	90.3	94.6	96.0	91.1	81.5	98.0	91.1	84.3	100.0	
Average Accuracy: 92.34%															

Table B.16: Experiment HYP18 BHC TS-FS confusion matrix.

B.5 EXPERIMENT HYP19

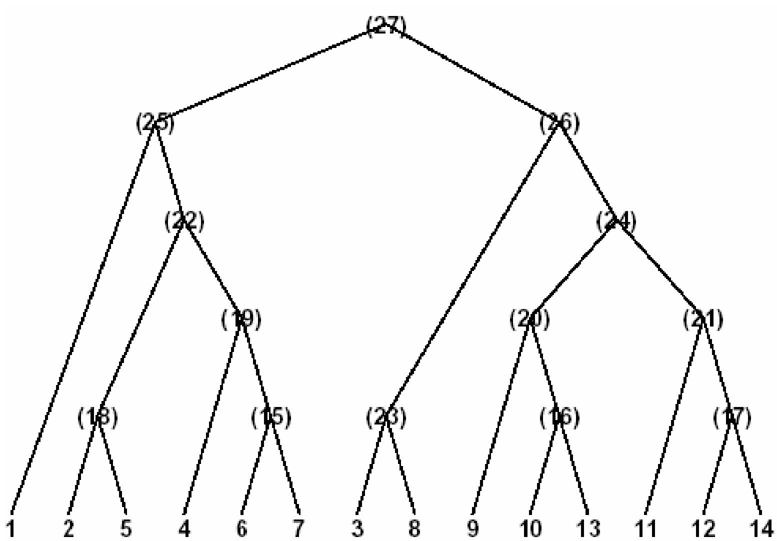

Figure B.9: Experiment HYP19 BHC class hierarchy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	135	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	0	43	0	0	1	0	0	0	0	0	0	0	0	0	97.7
3	0	0	118	0	0	1	0	0	0	0	0	0	0	0	99.2
4	0	0	5	104	3	0	0	0	0	0	0	0	0	0	92.9
5	0	7	0	1	109	5	0	0	0	0	0	0	0	6	85.2
6	0	0	1	1	20	98	0	0	27	0	0	0	2	0	65.8
7	0	0	1	1	0	0	128	0	0	0	0	0	0	0	98.5
8	0	0	0	0	0	0	0	100	0	0	0	0	0	1	99.0
9	0	0	0	0	1	21	1	0	127	0	0	0	1	0	84.1
10	0	0	0	0	0	0	0	0	2	115	14	0	5	0	84.6
11	0	0	0	0	0	0	0	1	0	8	137	3	0	0	91.9
12	0	0	0	0	0	0	0	0	0	0	0	79	8	7	84.0
13	0	0	0	0	0	9	0	0	1	1	1	7	118	0	86.1
14	0	0	0	0	0	0	0	0	0	0	0	1	0	33	97.1
	100.0	86.0	94.4	97.2	81.3	73.1	99.2	99.0	80.9	92.7	90.1	87.8	88.1	70.2	
Average Accuracy: 89.19%															

Table B.17: Experiment HYP19 BHC confusion matrix.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	134	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	1	47	0	0	2	0	0	0	0	0	0	0	0	0	94.0
3	0	0	121	0	0	0	0	0	0	8	0	0	0	0	93.8
4	0	0	3	102	2	0	0	0	0	0	0	0	0	0	95.3
5	0	3	0	4	92	19	0	0	0	0	0	2	0	0	76.7
6	0	0	0	0	29	43	0	0	9	0	0	0	0	0	53.1
7	0	0	1	0	1	0	128	0	0	0	0	0	0	0	98.5
8	0	0	0	1	0	0	0	97	0	5	3	0	0	2	89.8
9	0	0	0	0	8	67	0	0	143	1	0	0	1	0	65.0
10	0	0	0	0	0	0	1	0	0	102	29	0	0	0	77.3
11	0	0	0	0	0	0	0	2	0	7	120	0	0	0	93.0
12	0	0	0	0	0	0	0	1	0	0	0	83	12	0	86.5
13	0	0	0	0	0	5	0	1	5	1	0	5	120	0	87.6
14	0	0	0	0	0	0	0	0	0	0	0	0	1	45	97.8
	99.3	94.0	96.8	95.3	68.7	32.1	99.2	96.0	91.1	82.3	78.9	92.2	89.6	95.7	
Average Accuracy: 85.05%															

Table B.18: Experiment HYP19 BHC FS confusion matrix.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	134	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	0	50	0	0	1	0	0	0	0	0	0	0	0	0	98.0
3	0	0	122	7	0	0	0	0	0	1	0	0	0	0	93.8
4	0	0	2	97	1	0	0	0	0	0	0	0	0	0	97.0
5	0	0	0	2	112	14	0	0	0	0	0	0	0	0	87.5
6	0	0	0	0	14	101	1	1	38	0	0	0	0	0	65.2
7	1	0	1	1	2	0	128	0	0	0	0	0	0	0	96.2
8	0	0	0	0	0	0	0	94	0	1	0	0	0	0	98.9
9	0	0	0	0	0	13	0	0	114	1	0	0	0	0	89.1
10	0	0	0	0	0	0	0	1	0	115	27	0	2	0	79.3
11	0	0	0	0	0	0	0	0	0	2	124	0	0	0	98.4
12	0	0	0	0	0	0	0	3	0	1	1	82	8	0	86.3
13	0	0	0	0	4	6	0	0	5	3	0	8	123	0	82.6
14	0	0	0	0	0	0	0	2	0	0	0	0	1	47	94.0
	99.3	100.0	97.6	90.7	83.6	75.4	99.2	93.1	72.6	92.7	81.6	91.1	91.8	100.0	
Average Accuracy: 89.13%															

Table B.19: Experiment HYP19 BHC TS-FS confusion matrix.

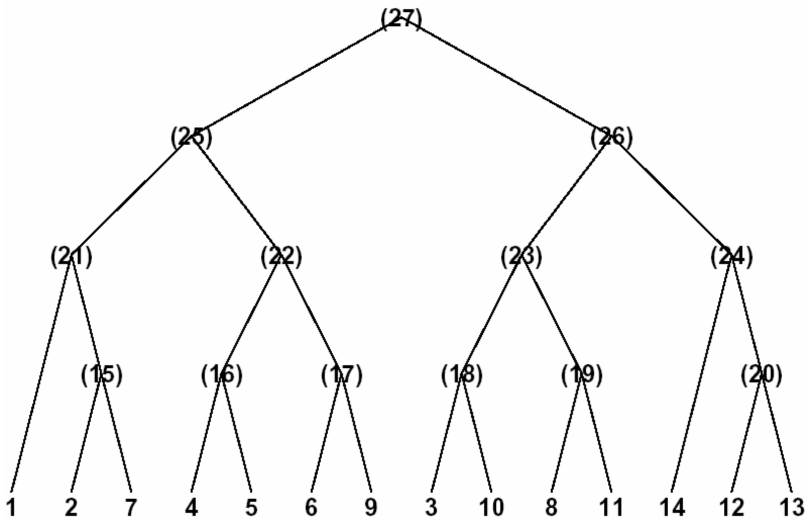

Figure B.10: Experiment HYP19 TS Build class hierarchy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	130	1	0	0	0	0	0	0	0	0	0	0	0	0	99.2
2	5	44	0	0	3	0	0	0	0	0	0	0	0	0	84.6
3	0	0	119	0	0	0	0	0	0	5	0	0	1	0	95.2
4	0	0	2	100	6	0	1	0	1	0	0	0	0	0	90.9
5	0	4	0	3	107	9	0	0	0	0	0	0	0	0	87.0
6	0	0	0	0	13	117	0	0	17	0	0	0	2	0	78.5
7	0	1	0	4	3	0	128	0	0	0	0	0	1	0	93.4
8	0	0	0	0	1	0	0	99	0	5	2	0	1	0	91.7
9	0	0	2	0	0	3	0	0	136	0	0	0	3	0	94.4
10	0	0	2	0	0	1	0	0	2	110	10	0	1	0	87.3
11	0	0	0	0	0	0	0	1	0	4	140	0	0	0	96.6
12	0	0	0	0	0	0	0	0	0	0	0	83	3	0	96.5
13	0	0	0	0	1	4	0	1	1	0	0	7	121	0	89.6
14	0	0	0	0	0	0	0	0	0	0	0	0	1	47	97.9
	96.3	88.0	95.2	93.5	79.9	87.3	99.2	98.0	86.6	88.7	92.1	92.2	90.3	100.0	
Average Accuracy: 91.48%															

Table B.20: Experiment HYP19 TS Build confusion matrix.

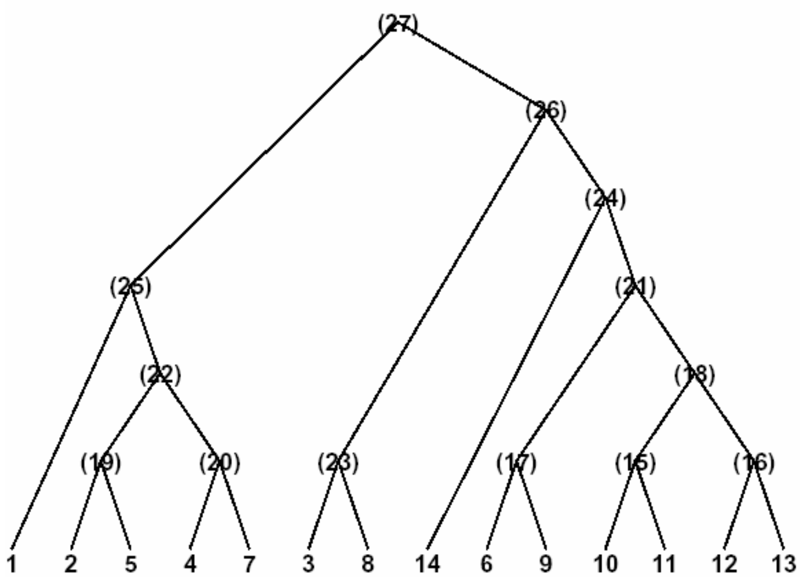

Figure B.11: Experiment HYP19 TSTRA class hierarchy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	135	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	0	43	0	0	1	0	0	0	0	0	0	0	0	0	97.7
3	0	0	118	0	0	0	2	0	0	0	0	0	0	0	98.3
4	0	0	7	106	4	0	0	0	0	0	0	0	0	0	90.6
5	0	7	0	1	112	8	2	0	0	0	0	0	0	0	86.2
6	0	0	0	0	17	118	0	0	14	0	0	0	3	0	77.6
7	0	0	0	0	0	0	123	0	0	0	0	0	0	0	100.0
8	0	0	0	0	0	0	0	100	0	0	0	0	0	0	100.0
9	0	0	0	0	0	6	2	0	138	0	0	0	2	0	93.2
10	0	0	0	0	0	0	0	0	2	119	10	0	1	0	90.2
11	0	0	0	0	0	0	0	1	0	5	141	0	0	0	95.9
12	0	0	0	0	0	1	0	0	2	0	0	71	11	0	83.5
13	0	0	0	0	0	1	0	0	1	0	1	19	117	0	84.2
14	0	0	0	0	0	0	0	0	0	0	0	0	0	47	100.0
	100.0	86.0	94.4	99.1	83.6	88.1	95.3	99.0	87.9	96.0	92.8	78.9	87.3	100.0	
Average Accuracy: 91.91%															

Table B.21: Experiment HYP19 TSTRA confusion matrix.

Appendix C

Selected Hyperion Best Bases Data Class Hierarchies and Confusion Matrices

C.1 EXPERIMENT HYP12

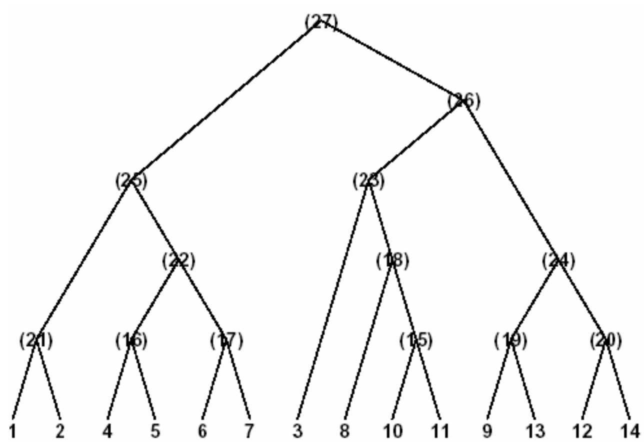

Figure C.1: Experiment HYP12 BHC BB class hierarchy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	135	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	0	50	0	0	2	0	0	0	0	0	0	0	0	0	96.2
3	0	0	124	0	0	1	1	1	0	0	0	1	0	0	96.9
4	0	0	1	106	0	0	0	0	2	0	0	0	0	0	97.2
5	0	0	0	1	126	14	0	1	2	0	0	0	0	0	87.5
6	0	0	0	0	6	95	0	0	25	0	0	0	1	0	74.8
7	0	0	0	0	0	0	127	0	0	0	0	0	0	1	99.2
8	0	0	0	0	0	0	0	97	0	0	0	0	0	3	97.0
9	0	0	0	0	0	18	1	0	126	1	0	0	0	0	86.3
10	0	0	0	0	0	0	0	2	0	115	12	0	0	0	89.1
11	0	0	0	0	0	0	0	0	0	5	137	1	1	1	94.5
12	0	0	0	0	0	0	0	0	0	0	0	84	8	5	86.6
13	0	0	0	0	0	6	0	0	2	3	3	4	123	0	87.2
14	0	0	0	0	0	0	0	0	0	0	0	0	1	37	97.4
	100.0	100.0	99.2	99.1	94.0	70.9	98.4	96.0	80.3	92.7	90.1	93.3	91.8	78.7	
Average Accuracy: 91.54%															

Table C.1: Experiment HYP12 BHC BB confusion matrix.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	135	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	0	47	0	0	2	8	0	0	0	0	0	0	0	0	82.5
3	0	0	121	1	0	0	0	0	0	2	0	0	0	0	97.6
4	0	0	0	102	3	0	14	0	0	0	0	0	0	0	85.7
5	0	1	0	4	117	25	0	0	1	0	0	8	0	0	75.0
6	0	2	0	0	7	32	0	0	2	0	0	0	0	0	74.4
7	0	0	2	0	0	1	114	0	0	0	0	0	0	0	97.4
8	0	0	0	0	0	0	0	98	0	5	0	0	0	0	95.1
9	0	0	0	0	5	66	0	0	146	0	0	0	1	0	67.0
10	0	0	2	0	0	0	0	0	2	109	13	0	3	0	84.5
11	0	0	0	0	0	0	0	1	1	6	138	0	1	1	93.2
12	0	0	0	0	0	0	0	1	0	0	0	77	4	0	93.9
13	0	0	0	0	0	2	1	1	5	2	0	5	121	0	88.3
14	0	0	0	0	0	0	0	0	0	0	1	0	4	46	90.2
	100.0	94.0	96.8	95.3	87.3	23.9	88.4	97.0	93.0	87.9	90.8	85.6	90.3	97.9	
Average Accuracy: 86.66%															

Table C.2: Experiment HYP12 BHC BB FS confusion matrix.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	135	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	0	50	0	0	1	0	0	0	0	0	0	0	0	0	98.0
3	0	0	121	2	0	0	0	0	0	1	0	0	0	0	97.6
4	0	0	3	100	1	0	1	0	0	0	0	0	0	0	95.2
5	0	0	0	2	126	21	0	1	1	0	0	0	0	0	83.4
6	0	0	0	0	3	89	1	1	22	0	0	0	3	0	74.8
7	0	0	0	2	0	0	126	0	0	1	0	0	0	0	97.7
8	0	0	0	0	0	0	0	96	0	2	1	0	0	0	97.0
9	0	0	0	0	0	18	0	0	125	1	0	0	1	0	86.2
10	0	0	1	1	1	0	0	0	1	112	4	0	0	0	93.3
11	0	0	0	0	1	0	1	2	0	5	146	0	0	0	94.2
12	0	0	0	0	0	0	0	0	0	0	0	78	5	0	94.0
13	0	0	0	0	1	6	0	1	8	2	1	12	121	0	79.6
14	0	0	0	0	0	0	0	0	0	0	0	0	4	47	92.2
	100.0	100.0	96.8	93.5	94.0	66.4	97.7	95.0	79.6	90.3	96.1	86.7	90.3	100.0	
Average Accuracy: 90.92%															

Table C.3: Experiment HYP12 BHC BB TS-FS confusion matrix.

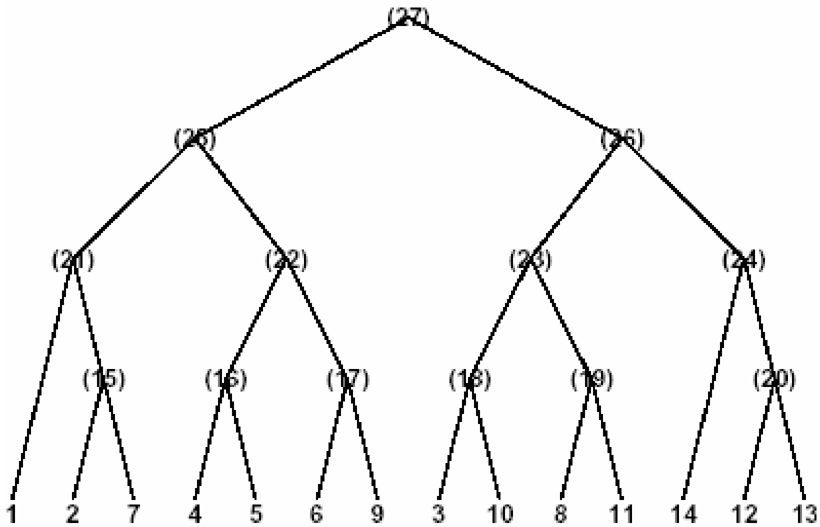

Figure C.2: Experiment HYP12 TS Build BB class hierarchy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	135	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	0	50	0	0	2	0	0	0	0	0	0	0	0	0	96.2
3	0	0	123	0	0	0	0	0	0	3	0	0	0	0	97.6
4	0	0	1	105	3	1	0	0	0	0	0	0	0	0	95.5
5	0	0	0	1	121	13	2	0	1	0	0	0	0	0	87.7
6	0	0	0	0	6	100	0	0	21	0	0	0	0	0	78.7
7	0	0	0	1	0	0	127	0	0	0	0	0	0	0	99.2
8	0	0	0	0	1	0	0	95	0	8	1	0	0	0	90.5
9	0	0	0	0	0	10	0	0	133	0	0	0	3	0	91.1
10	0	0	1	0	0	0	0	3	1	109	4	0	3	0	90.1
11	0	0	0	0	0	0	0	2	0	4	147	0	1	0	95.5
12	0	0	0	0	0	1	0	0	0	0	0	83	7	0	91.2
13	0	0	0	0	1	9	0	1	1	0	0	6	120	2	85.7
14	0	0	0	0	0	0	0	0	0	0	0	1	0	45	97.8
	100.0	100.0	98.4	98.1	90.3	74.6	98.4	94.1	84.7	87.9	96.7	92.2	89.6	95.7	
Average Accuracy: 92.22%															

Table C.4: Experiment HYP12 TS Build BB confusion matrix.

C.2 EXPERIMENT HYP16

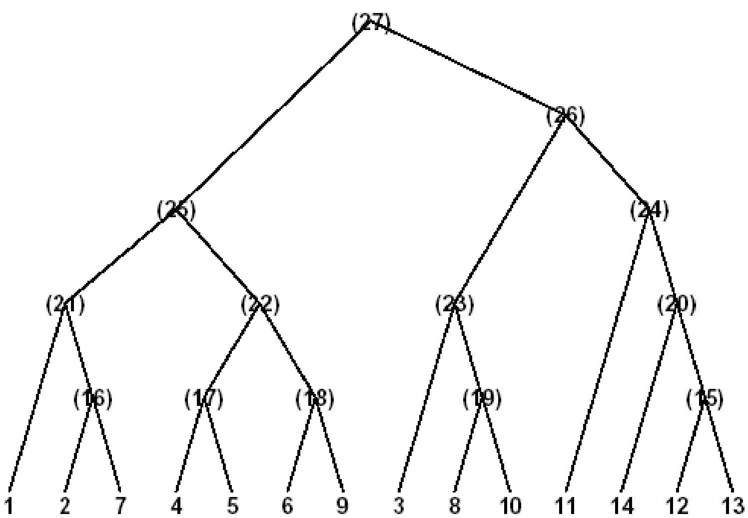

Figure C.3: Experiment HYP16 BHC BB class hierarchy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	135	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	0	49	0	0	1	0	0	0	0	0	0	0	0	0	98.0
3	0	0	120	1	0	0	0	0	1	1	0	0	0	0	97.6
4	0	0	5	104	4	0	1	0	0	0	0	0	0	0	91.2
5	0	1	0	1	120	8	0	0	0	0	0	0	0	0	92.3
6	0	0	0	0	9	111	0	0	5	0	0	0	2	0	87.4
7	0	0	0	0	0	0	127	0	0	0	0	0	0	0	100.0
8	0	0	0	1	0	0	0	96	0	1	2	0	0	1	95.0
9	0	0	0	0	0	11	0	0	145	2	0	0	1	0	91.2
10	0	0	0	0	0	3	1	1	6	102	15	0	4	0	77.3
11	0	0	0	0	0	0	0	2	0	17	134	0	0	2	86.5
12	0	0	0	0	0	0	0	0	0	0	0	83	13	2	84.7
13	0	0	0	0	0	1	0	0	0	1	1	7	114	0	91.9
14	0	0	0	0	0	0	0	2	0	0	0	0	0	42	95.5
	100.0	98.0	96.0	97.2	89.6	82.8	98.4	95.0	92.4	82.3	88.2	92.2	85.1	89.4	
Average Accuracy: 91.54%															

Table C.5: Experiment HYP16 BHC BB confusion matrix.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	133	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	2	49	0	0	4	0	0	0	0	0	0	0	0	0	89.1
3	0	0	115	1	0	0	4	0	0	0	0	0	0	0	95.8
4	0	0	6	97	5	0	0	0	0	0	0	0	0	0	89.8
5	0	0	1	5	99	7	1	0	0	0	0	0	1	0	86.8
6	0	0	0	0	14	121	0	0	10	0	0	0	8	0	79.1
7	0	1	0	1	5	0	100	0	0	0	0	0	0	0	93.5
8	0	0	0	0	0	0	0	97	0	0	1	0	2	1	96.0
9	0	0	1	0	0	5	0	0	135	0	0	0	14	0	87.1
10	0	0	2	3	0	0	0	1	4	113	12	0	1	0	83.1
11	0	0	0	0	0	0	0	0	0	9	136	0	0	0	93.8
12	0	0	0	0	0	1	16	0	0	0	0	86	7	0	78.2
13	0	0	0	0	7	0	6	0	8	2	2	4	100	0	77.5
14	0	0	0	0	0	0	2	3	0	0	1	0	1	46	86.8
	98.5	98.0	92.0	90.7	73.9	90.3	77.5	96.0	86.0	91.1	89.5	95.6	74.6	97.9	
Average Accuracy: 88.14%															

Table C.6: Experiment HYP16 BHC BB FS confusion matrix.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	134	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	1	48	0	0	3	0	0	0	0	0	0	0	0	0	92.3
3	0	0	121	0	0	0	1	0	0	0	0	0	0	0	99.2
4	0	0	1	103	1	0	1	0	0	0	0	0	0	0	97.2
5	0	1	0	2	119	8	0	1	0	0	0	0	0	0	90.8
6	0	0	0	0	8	115	0	0	8	0	0	1	4	0	84.6
7	0	1	0	2	1	0	127	0	0	0	0	0	0	0	96.9
8	0	0	0	0	1	0	0	97	0	0	0	0	1	1	97.0
9	0	0	0	0	0	5	0	0	140	4	0	0	0	0	94.0
10	0	0	3	0	0	0	0	1	3	111	8	0	0	0	88.1
11	0	0	0	0	0	0	0	1	0	9	141	0	0	0	93.4
12	0	0	0	0	0	0	0	0	0	0	0	86	4	0	95.6
13	0	0	0	0	1	6	0	0	6	0	2	3	124	0	87.3
14	0	0	0	0	0	0	0	1	0	0	1	0	1	46	93.9
	99.3	96.0	96.8	96.3	88.8	85.8	98.4	96.0	89.2	89.5	92.8	95.6	92.5	97.9	
Average Accuracy: 93.39%															

Table C.7: Experiment HYP16 BHC BB TS-FS confusion matrix.

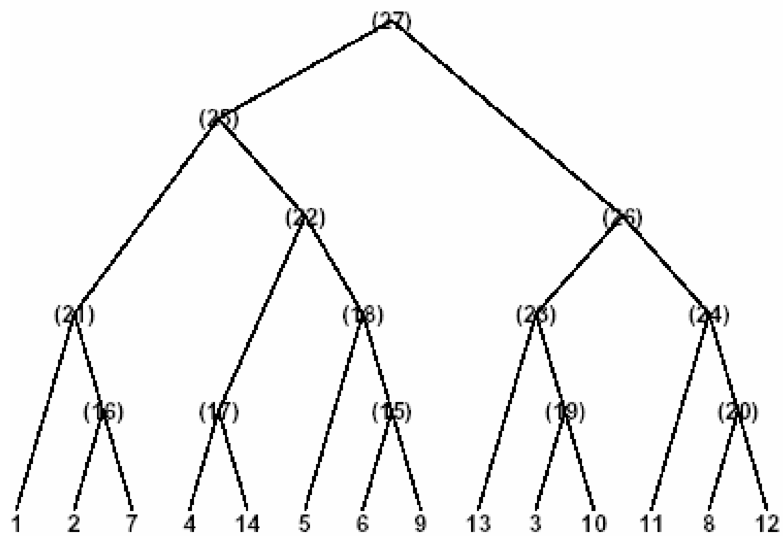

Figure C.4: Experiment HYP16 TS Build BB class hierarchy.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	134	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	1	48	0	0	3	0	0	0	0	0	0	0	0	0	92.3
3	0	0	121	0	0	0	0	0	0	0	0	0	0	0	100.0
4	0	0	0	95	0	0	3	0	0	0	0	0	0	0	96.9
5	0	1	0	8	122	8	0	0	0	0	0	0	0	0	87.8
6	0	0	0	0	9	112	0	0	10	0	0	0	0	0	85.5
7	0	1	0	3	0	0	126	0	0	0	0	0	0	0	96.9
8	0	0	0	0	0	0	0	99	0	2	1	0	3	1	93.4
9	0	0	0	0	0	6	0	0	134	0	0	0	0	0	95.7
10	0	0	4	0	0	0	0	1	9	121	5	0	0	0	86.4
11	0	0	0	0	0	0	0	0	0	1	142	0	0	0	99.3
12	0	0	0	0	0	1	0	0	0	0	0	85	5	0	93.4
13	0	0	0	1	0	7	0	0	4	0	0	5	126	0	88.1
14	0	0	0	0	0	0	0	1	0	0	4	0	0	46	90.2
	99.3	96.0	96.8	88.8	91.0	83.6	97.7	98.0	85.4	97.6	93.4	94.4	94.0	97.9	
Average Accuracy: 93.33%															

Table C.8: Experiment HYP16 TS Build BB confusion matrix.

Appendix D

Letter Recognition Data Confusion Matrices

Letter	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	
A	133	0	0	0	0	0	1	2	0	0	0	0	6	0	5	1	6	1	0	0	0	0	0	0	1	1	84.7
B	0	59	0	11	0	1	3	2	2	1	4	0	1	0	3	4	2	15	7	0	0	3	0	1	0	3	48.4
C	0	1	102	0	8	1	8	0	2	0	18	0	0	0	7	0	1	0	0	0	0	0	0	0	0	0	68.9
D	0	3	0	134	0	5	2	9	2	5	3	5	0	5	7	11	1	6	0	0	1	0	0	3	1	1	65.7
E	2	21	4	0	80	2	3	0	1	0	8	9	0	0	0	0	3	2	13	1	0	1	0	2	0	1	52.3
F	0	0	0	0	3	110	1	2	5	1	0	0	0	0	0	13	1	0	2	4	0	2	0	0	1	3	74.3
G	0	0	6	0	8	1	65	3	0	0	2	4	0	0	6	2	14	19	5	3	0	0	0	1	0	1	46.4
H	0	5	1	1	0	0	3	72	0	1	3	2	3	5	24	1	1	10	0	2	4	1	6	0	0	0	49.7
I	0	5	0	0	0	1	0	1	122	13	0	0	0	0	0	3	1	0	1	0	0	0	0	4	0	2	79.7
J	0	0	0	1	0	0	0	0	10	114	0	0	0	0	3	0	7	0	0	0	0	0	0	1	0	0	83.8
K	4	1	12	1	1	0	10	5	0	0	89	0	0	2	0	0	0	7	0	2	0	0	0	10	0	0	61.8
L	0	0	5	3	5	0	15	1	2	0	0	115	0	0	5	0	11	0	11	1	2	0	0	2	0	2	63.9
M	0	0	1	1	0	0	1	3	0	0	1	0	123	2	0	0	0	1	0	0	7	0	7	0	0	0	83.7
N	1	0	0	1	0	6	0	14	1	1	0	1	4	145	0	0	0	5	0	0	5	0	3	0	0	0	77.5
O	4	2	3	1	1	1	3	10	0	1	1	0	2	0	60	0	11	4	2	1	8	1	0	0	0	0	51.7
P	0	0	0	0	0	1	0	0	0	0	0	0	0	1	4	119	2	0	0	0	0	2	0	0	0	0	92.2
Q	0	3	1	0	0	0	32	0	0	0	1	8	0	0	3	2	85	0	4	0	0	0	0	0	1	0	60.7
R	0	8	1	7	1	0	6	6	6	0	7	2	0	0	6	0	4	88	3	2	0	0	0	3	0	0	58.7
S	2	22	1	1	8	2	4	0	4	1	0	0	0	0	0	0	12	1	94	1	0	0	0	5	3	17	52.8
T	0	0	2	0	1	14	0	0	0	0	0	0	0	0	1	0	1	0	1	122	12	0	0	1	7	2	74.4
U	3	0	2	0	0	0	0	8	0	0	4	1	0	3	0	0	0	0	0	1	125	0	0	4	3	1	80.6
V	0	0	0	0	0	0	6	8	0	0	1	0	0	2	1	0	3	0	0	0	3	121	4	1	8	0	76.6
W	0	0	1	0	0	2	0	0	0	0	0	0	5	1	4	1	1	0	0	0	1	4	119	0	0	0	85.6
X	2	6	0	1	10	3	1	4	6	6	4	9	0	0	0	0	0	2	2	3	0	1	0	117	2	2	64.6
Y	3	0	0	0	0	2	0	1	0	0	0	0	0	0	0	11	1	0	0	3	0	0	0	4	117	0	82.4
Z	2	0	0	4	26	1	0	0	2	4	0	1	0	0	0	0	0	0	16	5	0	0	0	0	1	122	66.3
	85.3	43.4	71.8	80.2	52.6	71.9	39.6	47.7	73.9	77.0	61.0	73.2	85.4	87.3	43.2	70.8	50.6	54.7	58.4	80.8	74.4	89.0	85.6	73.6	80.7	77.2	68.8
Average Accuracy: 68.82%																											

Table D.1: Letter Recognition BHC confusion matrix.

Letter	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	
A	135	0	0	1	0	0	0	1	0	1	0	0	4	0	5	0	6	0	1	0	0	0	0	0	2	4	84.4
B	1	36	0	15	1	1	1	1	1	1	1	3	0	0	0	2	1	9	8	1	0	0	0	5	0	2	40.0
C	0	0	106	1	4	0	23	0	0	0	6	0	0	0	1	1	0	0	1	0	12	0	0	0	0	0	68.4
D	0	0	0	57	0	0	0	9	0	0	5	0	0	6	10	1	2	4	0	0	2	0	0	0	0	0	59.4
E	4	7	4	6	81	2	2	5	3	0	9	7	2	3	0	1	0	2	8	3	1	1	0	2	0	3	51.9
F	0	1	2	0	1	110	1	2	11	2	2	0	0	0	0	7	0	0	4	20	6	2	1	10	4	4	57.9
G	0	7	11	2	12	2	112	9	1	0	9	5	0	0	14	6	3	5	0	4	8	3	0	9	3	0	49.8
H	2	1	5	0	1	4	0	51	0	1	30	2	0	19	5	0	3	14	0	3	15	0	0	1	0	0	32.5
I	0	2	0	0	6	0	0	0	116	3	0	0	0	0	0	1	0	0	7	0	0	0	0	0	0	0	85.9
J	2	0	0	6	0	0	0	0	14	115	0	0	0	0	4	0	2	0	1	1	0	0	0	1	1	0	78.2
K	0	4	0	1	0	0	2	7	1	0	27	1	0	0	5	0	1	3	0	3	2	0	0	6	0	1	42.2
L	0	0	0	1	0	0	3	0	1	2	1	125	0	0	1	0	6	0	15	0	0	0	0	0	0	0	80.6
M	0	0	1	0	0	0	2	2	0	0	1	0	125	3	0	0	0	1	0	0	6	0	8	0	0	0	83.9
N	1	0	4	6	0	1	0	9	0	0	4	0	5	123	0	0	0	0	0	0	12	4	2	0	1	0	71.5
O	1	0	5	0	0	0	0	2	0	1	10	1	0	6	69	0	5	3	0	0	9	0	1	0	0	0	61.1
P	0	0	0	6	0	3	0	0	1	4	0	1	0	0	3	129	1	8	0	0	0	8	0	0	1	1	77.7
Q	1	0	0	13	1	0	5	4	1	1	2	0	0	0	14	5	120	9	3	1	1	1	0	8	8	4	59.4
R	0	10	0	33	2	2	5	13	1	2	11	0	4	2	1	0	0	93	0	0	0	2	1	5	0	1	49.5
S	4	58	2	4	26	9	5	3	12	14	6	8	0	0	0	2	14	6	80	7	0	0	0	1	1	32	27.2
T	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	4	99	0	0	0	1	24	3	72.8
U	0	0	0	0	0	0	0	4	0	0	1	0	1	0	2	0	0	0	0	0	78	0	0	1	3	0	86.7
V	0	0	0	0	0	0	2	2	0	0	0	0	0	1	0	1	1	0	0	3	5	105	4	0	14	0	76.1
W	0	0	2	0	0	1	0	0	0	0	0	0	3	3	5	1	2	0	0	0	5	7	120	0	2	0	79.5
X	2	9	0	11	7	3	0	22	1	1	19	4	0	0	0	1	0	4	1	1	0	0	0	107	0	0	55.4
Y	3	1	0	3	0	10	1	5	1	0	2	0	0	0	0	10	0	0	9	4	6	3	2	2	81	8	53.6
Z	0	0	0	1	10	0	0	0	0	0	0	0	0	0	0	0	1	0	19	1	0	0	0	0	0	95	74.8
	86.5	26.5	74.6	34.1	53.3	71.9	68.3	33.8	70.3	77.7	18.5	79.6	86.8	74.1	49.6	76.8	71.4	57.8	49.7	65.6	46.4	77.2	86.3	67.3	55.9	60.1	62.3
Average Accuracy: 62.31%																											

Table D.2: Letter Recognition BHC FS confusion matrix.

Letter	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	
A	137	0	0	1	0	0	0	1	0	0	0	4	1	0	6	0	5	1	0	1	0	0	0	0	1	1	86.2
B	0	83	0	3	0	1	4	3	1	0	2	2	1	0	0	3	0	6	2	0	0	2	0	0	0	0	73.5
C	0	0	115	0	5	1	7	1	0	0	15	3	0	0	5	0	0	1	1	0	5	0	0	0	0	0	72.3
D	4	1	0	123	0	3	0	9	1	1	3	1	0	5	5	3	2	7	0	0	1	0	0	0	1	0	72.4
E	0	3	3	2	89	2	3	1	2	0	5	3	0	0	0	0	0	3	7	3	1	0	0	1	0	0	69.5
F	1	0	1	4	0	120	1	1	4	0	2	0	0	0	0	7	0	0	4	6	0	0	0	2	0	2	77.4
G	0	4	5	1	9	0	116	5	0	0	2	1	0	0	6	1	1	7	3	2	2	0	0	0	0	0	70.3
H	2	3	1	0	1	1	2	96	0	2	25	4	2	4	2	2	6	12	1	6	0	4	2	6	2	2	51.1
I	0	9	0	1	0	1	1	0	137	5	0	0	0	0	1	2	0	3	1	2	0	0	0	3	0	0	82.5
J	0	0	1	2	0	0	0	0	4	120	0	1	0	0	1	0	1	0	1	0	0	0	0	1	0	0	90.9
K	1	3	6	0	2	0	6	4	2	0	73	2	0	1	1	1	3	3	8	0	3	1	0	13	0	0	54.9
L	0	0	0	1	2	0	3	0	0	0	0	122	0	0	0	0	4	0	6	0	0	0	0	0	0	0	88.4
M	1	0	3	1	0	0	2	2	0	0	0	0	136	3	0	0	0	1	0	0	4	0	8	0	0	0	84.5
N	0	0	1	1	0	7	0	1	1	1	0	0	1	147	0	0	0	4	0	0	0	0	1	0	0	0	89.1
O	1	3	4	2	0	0	3	4	0	0	0	1	0	4	96	5	20	1	0	0	4	1	1	0	0	0	64.0
P	0	0	0	5	0	4	1	1	0	0	1	1	0	0	4	134	0	12	0	3	0	0	0	2	1	0	79.3
Q	0	0	0	0	0	1	2	2	3	4	1	6	0	0	4	1	114	0	0	1	0	0	0	1	6	1	77.6
R	0	9	0	5	0	1	3	1	0	0	1	0	2	0	1	0	3	91	1	0	0	1	0	3	0	0	74.6
S	4	16	0	9	16	7	4	2	7	6	2	1	0	0	2	0	2	5	119	3	0	0	0	1	4	4	55.6
T	0	0	0	3	0	4	0	0	0	0	0	0	0	0	0	0	4	0	0	119	0	0	0	1	3	0	88.8
U	2	0	0	0	0	0	0	4	0	0	4	0	0	1	1	0	0	1	0	0	137	0	1	2	0	1	89.0
V	1	1	1	0	0	0	4	2	0	0	0	3	1	1	0	0	0	0	0	1	2	118	0	0	8	0	82.5
W	0	0	1	0	0	0	0	1	0	0	0	0	0	0	3	0	0	0	0	0	9	7	125	0	2	0	84.5
X	0	0	0	1	4	0	1	6	1	2	10	2	0	0	1	0	0	3	0	1	0	0	0	121	0	0	79.1
Y	2	0	0	2	0	0	1	0	0	0	0	0	0	0	0	9	3	0	0	1	0	2	1	2	117	1	83.0
Z	0	1	0	0	24	0	0	4	2	7	0	0	0	0	0	0	0	0	7	2	0	0	0	0	0	146	75.6
	87.8	61.0	81.0	73.7	58.6	78.4	70.7	63.6	83.0	81.1	50.0	77.7	94.4	88.6	69.1	79.8	67.9	56.5	73.9	78.8	81.5	86.8	89.9	76.1	80.7	92.4	76.3
Average Accuracy: 76.27%																											

Table D.3: Letter Recognition BHC TS-FS confusion matrix.

Letter	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	
A	141	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	6	0	3	1	0	0	0	0	3	0	88.1
B	0	106	0	10	1	0	1	2	0	0	1	1	0	0	0	6	0	23	3	0	0	2	0	2	0	0	67.1
C	0	0	110	0	5	2	7	2	0	0	13	3	0	0	6	0	0	0	2	0	3	0	0	0	0	0	71.9
D	3	1	0	133	0	2	0	8	3	2	0	0	0	3	5	2	2	6	2	0	1	0	0	4	1	0	74.7
E	0	1	6	0	80	2	2	2	2	2	6	3	0	0	0	0	0	2	7	2	1	0	0	0	0	2	66.7
F	0	0	1	0	0	120	0	0	4	5	0	0	0	0	0	5	0	0	3	8	0	1	0	4	2	1	77.9
G	0	1	6	0	8	3	120	5	0	0	2	2	0	0	0	1	0	1	4	3	2	1	0	0	0	0	75.5
H	0	1	0	0	1	1	2	70	0	2	4	1	2	2	1	2	7	5	2	1	4	0	1	3	0	1	61.9
I	0	2	0	1	6	1	0	0	141	6	0	0	0	0	0	3	1	0	5	0	0	0	0	3	0	1	82.9
J	0	0	0	0	0	0	0	0	1	119	0	0	0	0	1	0	1	0	1	0	0	0	0	3	0	0	94.4
K	1	3	12	0	2	1	10	13	1	0	92	5	0	5	1	1	11	4	1	2	6	1	0	13	0	1	49.5
L	0	0	0	0	0	0	0	0	1	0	0	127	0	0	0	0	2	2	7	0	0	0	0	0	0	0	91.4
M	0	0	1	1	0	0	1	3	1	0	0	0	135	1	0	0	0	0	0	0	2	0	10	0	0	0	87.1
N	0	0	0	0	0	6	0	5	0	0	0	0	2	141	0	0	0	3	0	0	0	0	1	0	0	0	89.2
O	1	3	3	2	0	0	3	4	0	0	0	1	1	4	104	5	13	1	0	0	10	1	1	0	0	0	66.2
P	0	0	0	0	0	4	0	0	0	0	0	0	0	0	2	129	0	9	0	2	0	0	0	1	1	0	87.2
Q	0	0	0	5	1	0	3	1	2	4	2	7	0	0	3	3	117	0	0	0	0	0	0	3	4	3	74.1
R	0	10	0	4	2	1	7	5	0	0	8	1	1	2	0	1	2	96	1	2	0	1	0	0	0	0	66.7
S	4	2	0	2	12	3	2	1	7	3	5	3	0	0	2	0	1	4	111	7	0	0	0	2	2	2	63.4
T	0	0	0	1	1	2	0	0	0	0	0	0	0	0	0	1	0	0	1	117	1	0	0	0	2	1	92.1
U	3	0	2	2	0	1	0	13	0	0	6	0	1	2	1	0	0	1	0	2	129	0	0	4	1	0	76.8
V	0	1	0	0	0	0	3	1	0	0	0	0	1	1	1	0	0	0	0	1	4	116	0	0	8	0	84.7
W	0	0	1	0	0	0	0	1	0	0	0	0	0	4	3	0	0	1	0	0	4	11	125	0	3	0	81.7
X	1	4	0	3	16	1	3	10	1	1	7	3	0	0	3	3	0	3	2	1	1	0	0	115	0	1	64.2
Y	2	0	0	1	0	2	0	3	0	0	0	0	1	1	0	6	5	0	0	1	0	2	1	1	118	1	81.4
Z	0	1	0	2	17	1	0	2	1	4	0	0	0	0	0	0	0	0	6	1	0	0	0	1	0	144	80.0
	90.4	77.9	77.5	79.6	52.6	78.4	73.2	46.4	85.5	80.4	63.0	80.9	93.8	84.9	74.8	76.8	69.6	59.6	68.9	77.5	76.8	85.3	89.9	72.3	81.4	91.1	76.5
Average Accuracy: 76.49%																											

Table D.4: Letter Recognition TS Build confusion matrix.

Letter	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	
A	134	0	1	1	0	0	1	2	1	0	1	0	6	0	7	1	7	1	0	0	0	0	0	0	3	1	80.2
B	1	90	0	1	6	4	2	4	2	0	10	1	0	0	0	4	11	13	14	0	0	2	0	3	0	1	53.3
C	0	0	104	0	3	0	9	1	2	0	14	2	0	0	5	0	0	0	0	0	0	0	0	3	0	0	72.7
D	0	0	0	140	0	5	1	16	2	4	6	6	1	4	4	3	1	2	0	1	2	0	0	3	1	2	68.6
E	0	1	6	0	121	6	3	0	4	2	5	7	0	1	0	2	5	5	10	1	0	1	0	3	0	1	65.8
F	0	2	1	0	0	118	0	1	4	0	0	0	1	0	0	12	0	0	3	0	1	2	0	0	3	1	79.2
G	0	0	4	0	7	1	88	0	0	0	0	3	0	0	2	3	10	15	1	6	0	0	0	0	0	1	62.4
H	1	1	2	0	0	0	2	78	0	1	4	0	1	4	20	0	3	2	0	2	6	0	0	0	0	0	61.4
I	0	6	0	1	0	1	0	1	132	15	0	0	0	0	0	2	0	0	1	0	0	0	0	2	1	10	76.7
J	0	0	0	2	0	0	0	0	3	111	0	0	0	0	3	0	6	0	1	0	0	0	0	0	0	0	88.1
K	3	0	13	0	0	0	7	2	0	0	78	1	1	0	0	0	0	2	0	0	0	0	0	11	0	0	66.1
L	0	0	4	1	3	0	16	1	1	0	0	118	0	0	3	0	8	1	11	0	2	0	0	2	0	0	69.0
M	0	0	1	0	0	0	1	3	0	0	0	0	126	13	0	0	0	0	0	0	2	0	9	0	1	0	80.8
N	2	0	1	2	0	5	0	7	0	0	0	0	2	132	0	0	0	5	0	0	7	0	3	0	0	0	79.5
O	1	3	2	6	0	0	3	8	1	0	0	0	1	3	83	10	14	2	1	0	1	0	0	0	0	0	59.7
P	0	0	0	3	0	1	0	0	0	0	0	0	0	1	2	114	0	0	0	0	0	1	0	0	2	0	91.9
Q	0	0	0	0	2	0	11	0	1	2	1	6	0	0	4	0	89	0	3	0	0	0	0	1	3	4	70.1
R	1	17	1	2	4	0	7	8	0	0	19	2	2	1	1	1	1	111	3	1	0	3	0	6	0	0	58.1
S	3	8	0	1	1	3	4	1	2	2	0	2	0	0	0	1	0	0	77	4	0	0	0	5	4	17	57.0
T	0	0	1	0	1	3	0	0	0	0	0	0	0	0	0	3	0	0	2	121	1	0	0	1	8	7	81.8
U	1	0	0	0	0	0	0	4	0	1	3	0	0	3	0	0	0	0	0	4	128	0	0	3	0	1	86.5
V	0	0	1	0	0	0	6	8	0	0	3	0	0	1	1	1	5	0	0	1	9	122	4	2	8	0	70.9
W	1	0	0	0	0	2	1	0	0	0	0	0	3	3	4	1	1	0	0	0	9	5	123	0	0	0	80.4
X	2	7	0	2	2	2	2	4	8	8	2	9	0	0	0	0	0	2	3	1	0	0	0	110	0	1	66.7
Y	3	0	0	0	0	2	0	2	1	0	0	0	0	0	0	10	3	0	4	4	0	0	0	4	111	0	77.1
Z	3	1	0	5	2	0	0	0	1	2	0	0	0	0	0	0	4	0	27	5	0	0	0	0	0	111	68.9
	85.9	66.2	73.2	83.8	79.6	77.1	53.7	51.7	80.0	75.0	53.4	75.2	87.5	79.5	59.7	67.9	53.0	68.9	47.8	80.1	76.2	89.7	88.5	69.2	76.6	70.3	71.9
Average Accuracy: 71.91%																											

Table D.5: Letter Recognition TSTRA confusion matrix.

Letter	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	
A	136	0	0	1	0	0	0	1	0	0	0	0	1	0	7	0	6	0	0	1	0	0	0	0	0	1	88.3
B	0	100	0	17	0	4	6	3	4	1	2	2	0	0	1	6	0	18	3	0	0	3	0	1	0	0	58.5
C	0	0	113	0	4	0	7	0	0	0	8	1	0	0	7	0	0	0	1	0	2	0	0	1	0	0	78.5
D	4	2	0	123	0	0	1	14	1	0	2	1	0	5	2	3	2	1	2	1	2	0	0	0	0	1	73.7
E	0	1	4	0	129	5	2	2	2	0	15	7	0	0	0	3	3	5	9	9	1	0	0	3	0	1	64.2
F	0	1	2	0	1	120	0	1	4	5	0	0	0	0	0	5	0	0	3	1	2	0	0	3	1	2	79.5
G	0	2	6	0	5	1	114	5	0	0	2	1	0	0	4	0	1	1	1	3	4	0	0	0	0	0	76.0
H	0	3	0	3	1	1	4	78	1	3	26	4	1	5	7	4	8	3	1	5	3	2	1	5	5	1	44.6
I	0	4	0	2	0	0	0	0	137	6	0	0	0	0	0	2	1	6	2	2	0	0	0	2	0	0	83.5
J	1	0	2	1	0	0	0	0	3	119	0	0	0	0	1	0	1	0	1	0	0	0	0	3	0	0	90.2
K	0	1	6	0	0	1	3	2	2	0	58	1	0	0	0	1	1	2	3	0	0	0	0	13	0	1	61.1
L	1	0	0	1	2	0	1	1	0	0	0	126	0	0	1	0	6	1	10	0	0	0	0	0	0	0	84.0
M	3	0	2	0	0	2	3	3	0	0	1	0	135	4	1	0	0	1	0	0	4	3	1	0	0	0	82.8
N	2	0	1	0	0	7	0	2	0	0	2	2	2	142	0	0	0	5	0	0	1	0	1	0	0	0	85.0
O	0	1	3	0	0	0	5	13	0	1	2	1	0	4	95	3	19	0	0	0	6	2	0	0	0	0	61.3
P	0	1	0	0	0	5	0	0	0	0	0	0	1	0	3	133	0	9	0	1	0	0	0	0	5	0	84.2
Q	2	2	0	3	1	0	5	2	1	10	1	5	0	0	5	1	114	1	2	0	0	0	0	0	4	6	69.1
R	1	13	2	3	1	1	4	4	0	0	17	1	0	2	0	1	0	106	2	1	1	0	0	2	0	0	65.4
S	3	2	0	1	2	1	4	0	5	1	0	1	0	0	0	0	1	0	97	0	0	0	0	4	2	10	72.4
T	0	0	0	2	0	4	0	0	0	0	0	0	1	0	0	0	0	0	1	122	1	0	0	1	5	0	89.1
U	0	0	1	0	0	0	0	2	0	0	0	0	0	1	0	0	0	0	0	0	131	0	1	2	0	0	94.9
V	0	0	0	0	0	0	3	2	0	0	0	0	0	1	0	0	0	0	0	1	2	117	0	0	8	0	87.3
W	0	0	0	0	0	0	1	2	0	0	0	0	3	1	4	0	0	1	0	0	7	7	134	0	1	0	83.2
X	1	2	0	9	4	0	1	11	4	1	9	4	0	0	1	2	0	1	5	1	1	0	0	114	1	3	65.1
Y	2	0	0	0	0	1	0	3	1	0	0	0	0	1	0	4	5	0	1	3	0	2	1	2	113	1	80.7
Z	0	1	0	1	2	0	0	0	0	1	1	0	0	0	0	0	0	0	17	0	0	0	0	3	0	131	83.4
	87.2	73.5	79.6	73.7	84.9	78.4	69.5	51.7	83.0	80.4	39.7	80.3	93.8	85.5	68.3	79.2	67.9	65.8	60.2	80.8	78.0	86.0	96.4	71.7	77.9	82.9	76.0
Average Accuracy: 76.01%																											

Table D.6: Letter Recognition TSTRA TS-FS confusion matrix.

References

[1] M. R. Garey and D. S. Johnson, *Computers and Intractability, A Guide to the Theory of NP-Completeness*. San Francisco: W.H. Freeman and Company, 1979.

[2] S. Kumar, *Modular learning through output space decomposition*, Ph.D. dissertation, The University of Texas at Austin, 2000.

[3] S. Kumar, J. Ghosh, and M. Crawford, "A Hierarchical Multiclassifier System for Hyperspectral Data Analysis," *Lecture Notes in Computer Science*, F. Roli and J. Kittler, Eds., vol. 1857, pp. 270-279, 2000.

[4] A. Henneguelle, *Feature Extraction for Hyperspectral Data Analysis*, Masters thesis, The University of Texas at Austin, 2002.

[5] H. Liu and H. Motoda, Eds. *Feature selection for knowledge discovery and data mining*. Boston: Kluwer Academic Publishers, 1998a.

[6] M. Ben-Bassat, "Use of distance measures, information measures and error bounds in feature evaluation," in *Handbook of statistics*, vol. 2, P. R. Krishnaiah and L. N. Kanal, Eds. Amsterdam: North-Holland Publishing Company, 1982, pp. 773-791.

[7] P. A. Devijver and J. Kittler, *Pattern recognition: A statistical approach*, London: Prentice/Hall International, Inc., 1982.

[8] J. A. Richards and X. Jia, *Remote sensing digital image analysis: An introduction*, 3rd ed. Berlin: Springer-Verlag, 1999.

[9] T. Chenoweth and Z. Obradovic, "A multi-component nonlinear prediction system for the S&P 500 Index," *Neurocomputing*, vol. 10, issue 3, pp. 275-290, Apr. 1996.

[10] R. A. Johnson and D. W. Wichern, *Applied Multivariate Statistical Analysis.* 4th ed. New Jersey: Prentice Hall, 1999.

[11] D. A. Landgrebe, *Signal Theory Methods in Multispectral Remote Sensing*, J. A. Kong, Ed. Hoboken, New Jersey: Wiley Series in Remote Sensing, 2003.

[12] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley-Interscience Publication, 1973.

[13] P. Langley, "Selection of relevant features in machine learning," in *Proceedings of AAAI Fall Symposium on Relevance*, New Orleans, LA: AAAI Press, 1994, pp. 127-131.

[14] P. Langley, *Elements of Machine Learning*. San Francisco: Morgan Kaufmann, 1995.

[15] B. S. Serpico and L. Bruzzone, "A new search algorithm for feature selection in hyperspectral remote sensing images," *IEEE Transactiosn on Geoscience and Remote Sensing,* vol. 39, issue 7, pp. 1360-1367, Jul. 2001.

[16] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant features and the subset selection problem," in *Proceedings of the Eleventh International Conference on Machine Learning*, 1994, pp. 121-129.

[17] H. Liu and H. Motoda, Eds. *Feature extraction, construction and selection: A data mining perspective.* Boston: Kluwer Academic Publishers, 1998.

[18] P. M. Narendra and K. Fukunaga, "A branch and bound algorithm for feature subset selection," *IEEE Transactions on Computers*, vol. C-26, number 9, pp. 917-922, Sep. 1977.

[19] H. Liu, H. Motoda, and M. Dash, "A monotonic measure for optimal feature selection," in *European Conference on Machine Learning*, 1998, pp. 101-106.

[20] W. Siedlecki and J. Sklansky, "On automatic feature selection," *International Journal of Pattern Recognition and Artificial Intelligence,* vol. 2, issue 2, pp. 197-220, 1988.

[21] M. Kudo and J. Sklansky, "Comparison of algorithms that select features for pattern classifiers," *Pattern Recognition*, vol. 33, issue 1, pp. 25-41, Jan. 2000.

[22] W. Siedlecki and J. Sklansky, "A note on genetic algorithms for large-scale feature selection," *Pattern Recognition Letters*, vol. 10, pp. 335-347, Nov. 1989.

[23] H. Zhang and G. Sun, "Feature Selection Using Tabu Search Method," *Pattern Recognition*, vol. 35, issue 3, pp. 701-711, Mar. 2002.

[24] F. Glover, "Heuristics for integer programming using surrogate constraints," *Decision Sciences*, vol. 8, issue 1, pp. 156-166, Jan. 1977.

[25] J. W. Barnes and J. B. Chambers, "Solving the job shop scheduling problem using tabu search," *IIE Transactions*, vol. 27, pp. 257-263, Apr. 1995.

[26] W. Nanry and J. W. Barnes, "Solving the pickup and delivery problem with time windows using reactive tabu search," The University of Texas as Austin, Graduate Program in Operations Research Technical Report Series ORP98-03, 1998.

[27] J. Xu, S. Y. Chiu, and F. Glover, "Using tabu search to solve the Steiner tree-star problem in telecommunications network design," *Telecommunication Systems*, vol. 6, pp. 117-125, 1996.

[28] J. Xu, S. Y. Chiu, and F. Glover, "Tabu search for dynamic routing communications network design," *Telecommunication Systems*, vol. 8, issue 1, pp. 55-77, Jan. 1997.

[29] B. W. Colletti, *Group Theory and Metaheuristics*, Ph.D. dissertation, The University of Texas at Austin, 1999.

[30] B. Colletti and J. W. Barnes, "Local search structure in the symmetric traveling salesperson problem under a general class of rearrangement neighborhoods," *Applied Mathematical Letters*, vol. 14, issue 1, pp. 105-108, Jan. 2001.

[31] V. Wiley, *The Aerial Fleet Refueling Problem*, Ph.D. dissertation, The University of Texas at Austin, 2001.

[32] J. W. Barnes, V. Wiley, J. Moore, and D. Ryer, "Solving the Aerial Fleet Refueling Problem using Group Theoretic Tabu Search," unpublished.

[33] J. R. Crino, *A group theoretic tabu search methodology for solving the theater distribution vehicle routing and scheduling problem*, Ph.D. dissertation, Air Force Institute of Technology, Wright-Patterson AFB, OH, 2002.

[34] J. Crino, J. T. Moore, J. W. Barnes, and W. P. Nanry, "Solving the theater distribution vehicle routing and scheduling problem using group theoretic tabu search," Air Force Institute of Technology, University of Texas at Austin, and Office of the Army G-8, Pentagon, 2002.

[35] T. Combs, *A combined adaptive tabu search and set partitioning approach for the crew scheduling problem with an air tanker crew application*, Ph.D. dissertation, Air Force Institute of Technology, Wright-Patterson AFB, OH, 2002.

[36] K. P. Bennett and J. A. Blue, "An extreme point tabu search method for data mining," Rensselaer Polytechnic Institute, Troy, NY, R.P.I. Math Report No. 228, 1996.

[37] M. Dam and M. Zachariasen, *Tabu search on the Geometric Traveling Salesman Problem*, Masters thesis, The University of Copenhagen, 1994.

[38] J. C. Bezdek, J. Keller, R. Krisnapuram, and N. Pal, *Fuzzy models and algorithms for pattern recognition and image processing.* Boston: Kluwer Academic Publishers, 1999.

[39] T. G. Dietterich, "Ensemble methods in machine learning," in *Lecture Notes in Computer Science,* vol. 1857, F. Roli & J. Kittler, Eds. Germany: Springer, 2000, pp. 1-15.

[40] J. Ghosh, "Multiclassifier systems: back to the future," in *Proceedings of the Third International Workshop, MCS 2002,* F. Roli and J. Kittler, Eds. Germany: Springer-Verlag Lecture Notes in Computer Science (#2364), pp. 1-15, 2002.

[41] S. Kumar, J. Ghosh, and M. Crawford, "Best-bases feature extraction algorithms for classification of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 39, issue 7, pp. 1368-1379, Jul. 2001.

[42] J. T. Morgan, A. Henneguelle, M. M.Crawford, J. Ghosh, and A. Neuenschwander, "Adaptive feature spaces for land cover classification with limited ground truth," in *Proceedings of the Third International Workshop, MCS 2002,* F. Roli and J. Kittler, Eds. Germany: Springer-Verlag Lecture Notes in Computer Science (#2364), pp. 189-200, 2002.

[43] S. Kumar and J. Ghosh, “GAMLS: A generalized framework for associative modular learning systems,” in *Proceedings of the Applications and Science of Computational Intelligence II, Vol. 3722*, 1999, pp. 24-34.

[44] J. T. Morgan, A. Henneguelle, J. Ham, M. M. Crawford, and J. Ghosh, "Adaptive feature spaces for land cover classification with limited ground truth data,” *International Journal of Pattern Recognition and Artificial Intellligence*, in press.

[45] A. Neuenschwander, M. M. Crawford, and S. Ringrose, “Results from the EO-1 experiment – Use of Earth Observing-1 Advanced Land Imager (ALI) data to assess the vegetational response to flooding in the Okavango Delta, Botswana,” *International Journal of Remote Sensing*, to be published.

[46] J. S. Pearlman, P. S. Barry, C. Segal, J. Shepanski, D. Beiso, and S. Carman, “Hyperion, a space-based imaging spectrometer,” *IEEE Transactions on Geoscience and Remote Sensing*, Special Issue on EO-1, in press, 2003.

[47] C. L. Blake and C. J. Merz, UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science, 2002.

[48] M.M. Crawford, S. Kumar, M.R. Ricard, J.C. Gibeaut, and A.L. Neuenschwander, "Fusion of airborne polarimetric and interferometric SAR data for classification of coastal environments," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, issue 3, pp. 1306-1315, May 1999.

[49] J. T. Morgan, *Adaptive Hierarchical Classification with Limited Training Data*, Ph.D. dissertation, The University of Texas at Austin, 2002.

Vita

Donna Kay Korycinski was born in Louisville, Kentucky on November 26, 1964, the daughter of Phyllis Ann Cecil and Joseph Louis Cecil. A 1982 graduate of Western High School, Louisville, Kentucky, she entered Morehead State University in Morehead, Kentucky. She received the degree of Bachelor of Science from Morehead State University in May 1986 and was commissioned as an officer in the United States Army. Her military service has been continuous to the present day, and she has held positions as an aviator and Operations Research, Systems Analyst. She currently holds the rank of Lieutenant Colonel. She received the degree of Masters of Science in Engineering from the University of Texas, Austin, Texas in May 1996. In June 2000 she began doctoral studies in the Operations Research, Industrial Engineering Program at the University of Texas, Austin, Texas.

Permanent address: 3003 South Crums Lane, Louisville, KY 40216

This dissertation was typed by the author.