An Experimental Assessment of Decision Trees and Decision Tree Ensembles

Student number: 100225776. Blackboard ID: gny17hvu

1 Introduction

A critical part of using Machine Learning algorithms is knowing how they are constructed and how their performance is evaluated. In this paper we seek to refute the null hypothesis that a tuned variant of the ID3CW Decision Tree classifier we have implemented shows no statistically significant performance improvement compared to its predecessor and an un-tuned variant over a range of problems. We will also investigate whether the TreeEnsemble ensemble shows a statistically significant improvement over other classifiers on our chosen data set and a specific case study data set.

We believe that we will see a statistically significant improvement in the performance of our tuned classifiers over the un-tuned version. In general, this has been understood for many years, Mu and Nandi (2007) demonstrated that automated parameter tuning improved breast cancer diagnosis accuracy by 4.9%, and Gholap (2012) found an increase of 1.41% in the accuracy of Weka's J48 classifier after tuning. We feel that although the TreeEnsemble will produce better results than ID3CW on its own (Budzik, 2019), when compared with more complex classifiers it will likely fall behind due to lacking the improvements other classifiers have made in the past 35 years of Machine Learning research.

We will first explain the data sets we will be using, then give some information on the classifier and ensemble before presenting the results of my experiments and discussing their meaning.

2 Data Description

We will be using two groups of data and a single case study to test our classifiers on. The first group is a selection of 23 multivariate, categorical (discrete) data sets from the UCI Machine Learning Repository (Dua and Graff, 2017), of which the characteristics are shown in Appendix A.1. The second is a group of 36 multivariate, real-valued (continuous) data sets also from the UCI Machine Learning Repository as described in Appendix A.2. My case-study data set is MiddlePhalanxOutlineCorrect, from The UCR/UEA TSC archive (Bagnall et al., 2021), detailed in Appendix A.3. The case study includes points of data from 891 xrays of a middle phalanx bone from a middle finger and the estimation from a previous algorithm of where a box drawn around that bone should be to contain the bone but not extend too far out. This task is to classify whether the bounds were drawn correctly based off of classification of these estimations by humans.

3 Classifier Description

The classifier designed for this paper is ID3CW, a variant of the ID3 classifier introduced by Quinlan (1986). One modification allows the user to select the attribute selection mechanism the classifier employs from Information Gain, Chi-Squared, Chi-Squared with Yates Correction and Gini Coefficient. These can be defined by using the command line argument `-C [criteria]` where `[criteria]` can be `ig`, `chisquared`, `chisquaredyates` or `gini`. The mechanism can also be set from within a Java program by passing the above command as an argument to the `setOptions` method of the ID3CW object before calling the `buildClassifier` method. By default, the classifier will use Information Gain as the attribute selection mechanism. Another

modification allows the classifier to handle continuous attributes by taking a random value from that set of attributes and splitting the attribute across that value, then storing the value for later classification on the test set instances.

We have also developed an ensemble classifier called TreeEnsemble which is a collection of ID3CW classifiers that generate predictions via majority vote. By default TreeEnsemble builds 50 ID3CW classifiers, where each classifier selects a random attribute selection mechanism and a random 50% subset of the attributes from the data set. All three of these settings can be modified with the `setNumClassifiers`, `setSplitMeasure` and `setSampledPercentage` methods on the TreeEnsemble object. Additionally, the `setGiveProbability` method can be set to `true` to return the probability the selected class is correct rather than the name of the class.

For testing, we have chosen a selection of different classifiers implemented in the Weka (Frank et al., 2009) toolkit:

- J48 - The Weka implementation of C4.5 (Quinlan, 2014), an extension of the ID3 algorithm which handles continuous attributes, missing attribute values and includes tree pruning.
- Bagging - First noted by Breiman (1996), this is an ensemble that fits random subsets of the data to a collection of REPTree classifiers and then collects their predictions through voting to form a final prediction.
- Random Forest - Outlined by Breiman (2001), this is an ensemble classifier which is formed of RandomTree classifiers, which randomly choose an attribute at each node and does not perform pruning.
- Rotation Forest - Proposed by Rodriguez et al. (2006), this is a complex ensemble classifier which splits the given training data into multiple subsets, then performs Principle Component Analysis (PCA) on each subset then using the features extracted from PCA to form a new feature set. The training data is then transformed (Rotated) into the new feature set, which a decision tree is trained on. Repeating this process with different splits improves diversity and accuracy.
- LogitBoost - Proposed by Friedman et al. (2000), this classifier uses a boosting ensemble, an algorithm which trains classifiers and then adds lower weights to trees that perform poorly to train a final classifier. Specifically, LogitBoost modifies AdaBoost (a common boosting algorithm) by using logistic regression as the cost function.
- DecisionStump - Rarely used on it's own but often part of a boosting ensemble, this classifier is a one-level decision tree which classifies data on only one attribute. We're not expecting this to perform very well.

4 Results

4.1 ID3CW Discrete

In the first experiment, we tested whether there was any difference in average accuracy of the ID3CW classifier on the UCI Discrete group of data when tuning the attribute selection method, and compared those classifiers to the accuracy of default ID3 and J48 classifiers. We ran each experiment five times and averaged the accuracy values between those five runs. We expect that at least one of the tuned versions of ID3CW should perform slightly better overall than the default Information Gain (IG) version and that ID3CW with IG should perform almost exactly the same as ID3. We also expect J48 to perform significantly better than all others in this experiment. The results of this experiment are recorded in Appendix B.1. We can see from this table that in general, all four versions of ID3CW performed almost identically on each dataset

with the only differences being in molecular-promoters (variation in accuracy of 0.125), nursery (variation of 0.097) and zoo (variation of 0.167). This was surprising as we expected at least one classifier to perform better, or worse, than the others. We were also surprised to see that the ID3 algorithm performs significantly better or worse than our tuned ID3CW despite being almost exactly the same as ID3CW splitting on Information Gain. Noteworthy data sets here include balance-scale (where ID3CW-IG showed a 28.0% increase in average accuracy over ID3), chess-krvk (where ID3 shows a 51.7% improvement over ID3CW-IG), led-display (where ID3 leads by 63.4%), nursery (ID3 leads again by 27.2%), semeion (where ID3 leads by 69.2% ahead of ID3CW-IG) and zoo, where ID3 beat our ID3CW-IG classifier by 31.2%. After seeing these results we checked our test environment to ensure there were no abnormalities in the processing of the data but could not understand why these datasets produced such large differences between these two classifiers. As expected, the J48 classifier performed either as well as, or significantly better than all ID3CW variants as well as ID3.

4.2 ID3CW Continuous

We then used the same set of classifiers against the continuous data set (except ID3, as it is unable to handle continuous data) to see how well it performs over an average of 5 tests where we split on different random instances each time. Given the results above we now believe that all versions of ID3CW will perform similarly to each other, but J48 will perform equally or much better than them in all cases. The results of this experiment are recorded in Appendix B.2. We can see from the table that our assumptions this time are correct, that there are only two datasets where one version of ID3CW performed significantly better than the others - using Chi-Squared as the attribute selection mechanism on oocytes-trisopterus-states-5b and on steel-plates (performing 5.0% and 14.3% better respectively than Information Gain, the second closest). These results alone, however, are not enough to say that Chi-Squared is definitively a better attribute selection mechanism than any other. Once again, J48 performed as well or better than all variations of ID3CW on each data set. This is despite the normalisation criteria (splitting randomly on a given value for each attribute) being the same for both ID3CW and J48.

4.3 TreeEnsemble with Train-Test Split

Our third experiment tested variations of TreeEnsemble on the Discrete group of data and compared their average accuracies across 5 tests with the data previously collected on the IG variant of ID3CW. We expect to see all variants of TreeEnsemble to show equivalent or better average accuracy than ID3CW across all datasets. We are also testing if there is any significant difference between the different attribute selection mechanisms. We expect that TreeEnsemble using a random attribute selection mechanism should have an average accuracy as a mean of the set of accuracies from the other selection mechanisms. The results we retrieved can be found in table 10. As expected, the TreeEnsemble variants performed as good or better than ID3CW by itself. However, there were interesting differences to note. Firstly, TreeEnsemble performed far better than ID3CW across the board on the habermans-survival dataset. This is possibly because one of the three attributes in that set is not relevant to the class and causing the base classifier to often mis-classify entries, and where TreeEnsemble removes attributes on each tree, this attribute is probably being missed out of a majority of classifiers and therefore the majority vote can find the correct classification. We could possibly find the same with balance-scale, fertility, molecular-splice and pendigits, all of which showed significant improvement across all TreeEnsemble variants compared to ID3CW. There is one single dataset that performed significantly worse across the board compared to ID3CW, and that is Monks-1. We assume for this dataset that there are a couple attributes that are fundamental to the class and that by leaving out one or more of those attributes, the accuracy of the classifier is lost. There

was an unusual result in the zoo data set, with the Information Gain version of TreeEnsemble registering an average far below other classifiers, but from what we can see this is the result of a large variance in results on that dataset across the board. This could be down to the fact that zoo is very small dataset with only 100 entries, and therefore there was not enough training data to build an accurate model for the test data. In regards to adjusting the attribute selection mechanisms, there doesn't appear to be a large variation between them, with Chi-Squared again coming in slightly ahead of the other mechanisms on most data sets, particularly on the fertility, molecular-splice, optdigits, semeion and zoo data sets. Gini appears a little worse across the board, but not by a statistically significant margin.

We wanted to tune the parameters of our TreeEnsemble a little to see if that would improve the accuracy of the results. Our first tuning was to set the attribute selection percentage to 100% to see how this would change the result. Our expectation is that this will cause TreeEnsemble to perform almost the same as ID3CW as there would be little to differentiate each classifier within the ensemble. The results of this tuning can be found in table 12 of Appendix B. As expected, there was an overall reduction in the accuracy of the models when all attributes are selected on each classifier. However, there are two main points of interest in the data. The first is that although the accuracy of TreeEnsemble has dropped slightly on habermans-survival, it is still far ahead of the performance of ID3CW. This is possibly due to a rounding error in working out how many attributes to include, which has possibly caused an attribute to be dropped. For the fertility dataset, too, we are seeing a small reduction in the accuracy of TreeEnsemble over default settings, but an increase in accuracy compared to ID3CW. The nursery and zoo datasets continue to give wildly fluctuating results which doesn't seem to be affected by the tuning of the classifier.

Exploring tuning further, we decided to continue to use 50% as our attribute selection percentage but increase the amount of classifiers in the ensemble by five times, to 250. Although this would increase the processing time by a large amount, we hope that this would give us more accurate predictions. The results of this experiment can be found in 14. As we can see from the results, there are no statistically significant gains to accuracy from running more than 50 classifiers in a TreeEnsemble. The only dataset that seemed benefit across the board was hayes-roth, however this is likely a fluke due to the small number of instances (132), considering the variation we observed in the sets of results.

To illustrate whether there is any advantage to using one of these classifiers over another, we have constructed a Critical Difference diagram to illustrate the difference in the sum rank averages between each of these classifiers, as found in figure 1. From this we can see that although Chi-Squared performed the best out of all the classifiers, they are all within the same clique meaning there is no statistically significant difference between them. Indeed, a Friedman Test (Stangroom, n.d.) on this data returns a p-value of 0.0911, which is below our α of 0.05, meaning there is no statistical significance.

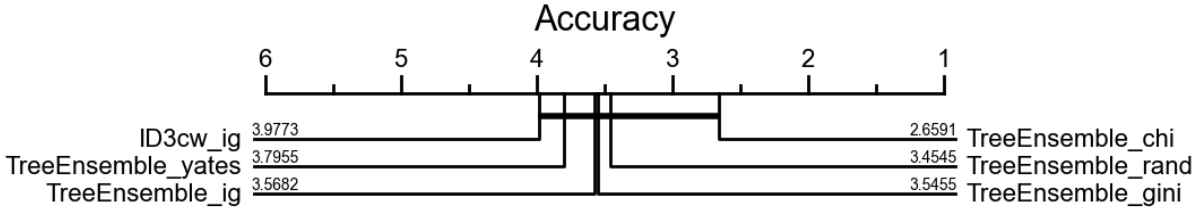

Figure 1: A Critical Difference diagram comparing different versions of TreeEnsemble to ID3CW

Accuracy is only one measurement, however. To find how good this ensemble is, we must take multiple measurements in a measured way and find other performance measures to compare it to other algorithms.

4.4 TreeEnsemble with Cross Validation

Cross validation is a process where a dataset is split into v different subsets (in our case ten) and for each fold, one subset is held back for testing while the model is trained on the remaining *v-1* subsets, up to fold v. This allows for higher accuracy for any classifier which is unable to guarantee a perfect model (Browne, 2000). For this experiment, we will compare the Chi-Squared variant of TreeEnsemble with J48, Bagging, Random Forest, Rotation Forest, LogitBoost and DecisionStump on the UCI Discrete group of data. Due to the complexity of these other classifiers, we expect TreeEnsemble to only beat DecisionStump, though we feel it should likely rank close to Bagging and LogitBoost. Rather than present these results in a table, we have constructed a Critical Difference Diagram as shown in figure 2. Here we can see that TreeEnsemble performed better than DecisionStump but sadly still within the same clique. Surprisingly, Bagging and LogitBoost were not in the same clique as TreeEnsemble - in fact LogitBoost was in the same clique as RotationForest, the highest ranked classifier in this experiment.

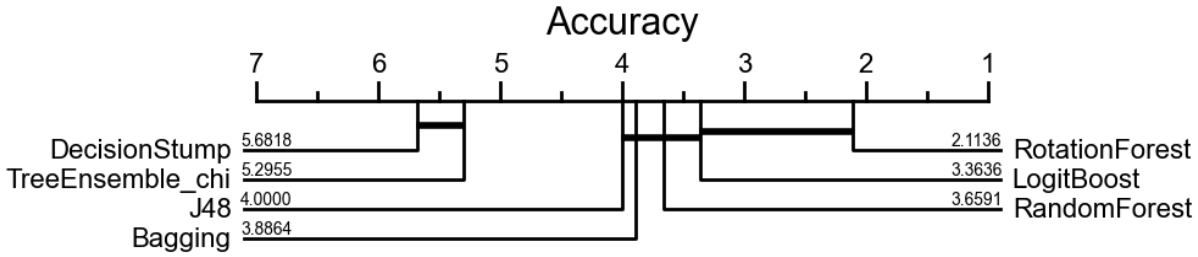

Figure 2: A Critical Difference diagram comparing TreeEnsemble-Chi to other existing classifiers

Looking at the data in depth, it would be tempting to remove fertility, habermans-survival and spect-heart from the test data as the variation between each classifier is below 10% between the best and worst performing classifier, meaning we're not gaining any meaningful information from these data. A test on this new dataset did show an improvement in the accuracy of TreeEnsemble, however this would only artificially improve the score of our classifier as we're specifically picking datasets that help to show our classifier in the best light.

4.5 Classification of the Case Study

Using the above classifiers, we passed in the data from our case study. Due to the nature of this problem, we're looking for the maximum true positives and true negatives, indicating our classifier can recognise both x-rays that have been bounded correctly, as well as ones that are incorrectly bounded. Here, we are interested in the Balanced Accuracy. Where Accuracy takes into account only the predictions we got correct, Balanced Accuracy takes an average which includes the predictions we got incorrect, too. This allows us to account for the fact that there were more cases given to us where the bounds were drawn correctly (554 vs 337). The relevant performance measures across the tested classifiers can be found on Appendix 16. Here, we find that TreeEnsemble appears to perform in line compared to the other classifiers aside from DecisionStump, which was to be expected. A Negative Log Likelyhood (NLL) of 0.9339 tells us that the model is quite confident when it is wrong, as much as DecisionStump is, which is wrong far more often. An AUROC value of 0.749 tells us that in general, the classifier is finding more true positives than it is finding false positives.

Generally speaking, to find if any of these differences between the classifier models are statistically significant, we must take the balanced accuracy from each test for each classifier and perform a Friedman Test against an α of 0.05. However, as we are performing tests only on a single dataset, we must use the individual folds for testing which are not independent and

therefore we can not draw any strong conclusions from our results. Indeed, the test gives an incredibly small p-value of ¡0.0001, which would indicate major statistical significance between the classifiers, which is likely true if you compare DecisionStump to RotationForest. We decided to create a Critical Difference diagram on this finding, which can be found in figure 3, below. Indeed, we have two cliques that largely overlap, but for this experiment at least, TreeEnsemble performed in line with most of our other classifiers. The poor performance of TreeEnsemble on continuous data is expected after discussing the performance of ID3CW earlier in this paper.

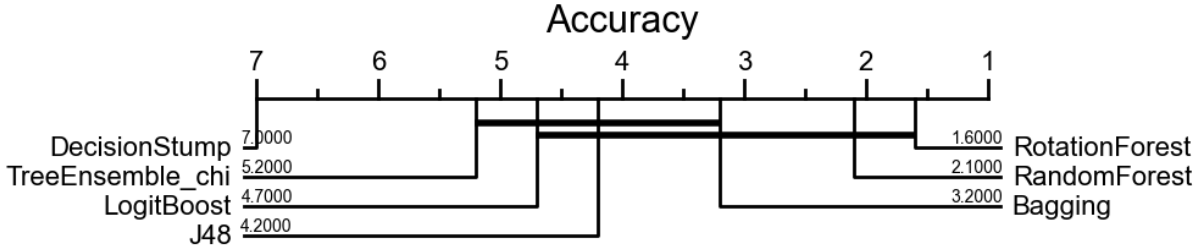

Figure 3: A Critical Difference diagram comparing our chosen classifiers on the MiddlePhalanxOutlineCorrect dataset

4.6 Case Study with Discretized Filter

For our final experiment, we hypothesized that if we Discretize the data into a binary split before use, and use the same split on all classifiers, we will see a reduction in the performance of the other classifiers to be more in line with TreeEnsemble. The results from this experiment are presented in table 18. Once again we were surprised by the results - not only did the discretization reduce the performance measures of the competing classifiers, but the quality of TreeEnsemble rose by a reasonable margin to become the best performing classifier. The increase in Balanced Accuracy to be comparable to Accuracy shows us that TreeEnsemble has improved performance on problems with many classes or large class imbalances. The reduction in NLL shows us that it is producing very good probability estimates, and the high AUROC shows it is much better at ranking, now. A Critical Difference diagram on this data is shown in figure 4 below. Although the margin is not enough to say TreeEnsemble is the best performing classifier, it is in the same clique as RotationForest and Bagging, which also performed well. This improvement in performance is likely due to the selection criteria of Discretize being of higher quality than the one implemented in TreeEnsemble.

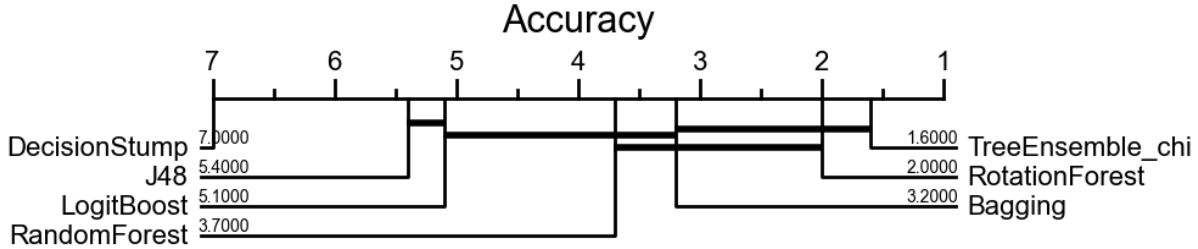

Figure 4: A Critical Difference diagram comparing our chosen classifiers on the Discretized MiddlePhalanxOutlineCorrect dataset

5 Conclusions

Ideally we would have spent more time on tuning our classifiers, perhaps by trying different combinations of classifier counts and lower attribute selection percentages to see if that could give

a better result. The results from Discretization at the end of our paper open up the possibility of revising this algorithm at a later date to use the filter rather than our own implementation. Although we wanted our developed classifiers to do well, we feel we refrained from trying to artificially increase their prediction quality in order to avoid bias in the results.

A case has been made by Benavoli et al. (2017) that it might be worth moving away from Null Hypothesis Significance Tests and over to Bayesian tests instead, which were shown to be a better fit for Machine Learning classifiers, but that is outside the scope of this particular paper.

We could also explore more robust ways to split data into individual classifiers within TreeEnsemble. A random selection of x% of the attributes runs the risk of selecting a subset of attributes multiple times, which could bias the results to those chosen attributes. This is obviously offset by running the test multiple times, but it may be worth adding a flag option to TreeEnsemble which, when set, will specify to select all subsets of data. The downside of this is that this would require creating 2^n classifiers, where n is the number of attributes in the set. For datasets of 50,000+ entries beyond 11 attributes, this begins to take a gigantic amount of time to process. Another approach is to find all subsets of attributes at a given number of selections (k), which would result in a selection of only $\binom{n}{k}$, which is much more reasonable, though still unwieldy for sets such as molecular-splice, which could end up with $\binom{60}{30}$ at its largest selection, or 1.18×10^{17}, clearly not feasible for any sized dataset! A hybrid system that could choose a number for k based off of the number of attributes in the dataset could be an option, which then opens up another path of research, if it's better to create 495 trees of 4 attributes each from a set of 12 or 495 trees of 8 attributes from that same set? Would there be a difference in results at all?

References

Bagnall, A., Lines, J., Vickers, W., and Keogh, E. (2021). The uea & ucr time series classification repository. **URL:** http://timeseriesclassification.com/.

Benavoli, A., Corani, G., Demšar, J., and Zaffalon, M. (2017). Time for a change: a tutorial for comparing multiple classifiers through bayesian analysis. *The Journal of Machine Learning Research*, 18(1):2653–2688.

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

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Browne, M. W. (2000). Cross-validation methods. *Journal of mathematical psychology*, 44(1):108–132.

Budzik, J. (2019). Many heads are better than one: The case for ensemble learning. *KDnuggets, September*.

Dua, D. and Graff, C. (2017). Uci machine learning repository. **URL:** http://archive.ics.uci.edu/ml.

Frank, E., Hall, M., Holmes, G., Kirkby, R., Pfahringer, B., Witten, I. H., and Trigg, L. (2009). Weka-a machine learning workbench for data mining. In *Data mining and knowledge discovery handbook*, pages 1269–1277. Springer.

Friedman, J., Hastie, T., Tibshirani, R., et al. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *Annals of statistics*, 28(2):337–407.

Gholap, J. (2012). Performance tuning of j48 algorithm for prediction of soil fertility. *arXiv preprint arXiv:1208.3943*.

Mu, T. and Nandi, A. K. (2007). Breast cancer detection from fna using svm with different parameter tuning systems and som–rbf classifier. *Journal of the Franklin Institute*, 344(3-4):285–311.

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

Quinlan, J. R. (2014). *C4.5: Programs for Machine Learning*. Elsevier.

Rodriguez, J. J., Kuncheva, L. I., and Alonso, C. J. (2006). Rotation forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1619–1630.

Stangroom, J. (n.d.). The friedman test for repeated-measures. **URL:** https://www.socscistatistics.com/tests/friedman/default.aspx.

A Appendix A - Datasets Used in Evaluation

A.1 UCI Discrete Group

Data Set	Instances/ Attributes/ Classes	Class Distribution	Train/Test Split
balance-scale	625 / 4 / 3	288 Left, 49 Balanced, 288 Right	437/188
chess-krvk	28056 / 6 / 18	2796 Draw, 78 One, 246 Two, 81 Three, 198 Four, 471 Five, 592 Six, 683 Seven, 1433 Eight, 1712 Nine, 1985 Ten, 2854 Eleven, 3597 Twelve, 4194 Thirteen, 4553 Fourteen, 2166 Fifteen, 390 Sixteen	19639/8417
chess-krvkp	3196 / 36 / 2	1669 Won, 1527 Nowin	2237/959
connect-4	67557 / 42 / 3	44473 Win, 16635 Loss, 6449 Draw	47289/20268
contraceptive-method	1473 / 9 / 3	629 No-use, 333 Long-term, 511 Short-term	1031/442
fertility	100 / 9 / 2	88 Normal, 12 Altered	70/30
habermans-survival	306 / 3 / 2	225 Survived, 81 Died	214/92
hayes-roth	132 / 4 / 3	51 Class-1, 51 Class-2, 30 No-Class	
led-display	500 / 7 / 10	45 Zero, 37 One, 51 Two, 57 Three, 52 Four, 52 Five, 47 Six, 57 Seven, 53 Eight, 49 Nine	350/150
lymphography	148 / 18 / 4	2 Normal-find, 81 Metastases, 61 Malign-lymph, 4 Fibrosis	103/45
molecular-promoters	106 / 57 / 2	53 Promoters, 53 Non-Promoters	74/32
molecular-splice	3190 / 60 / 3	767 EI, 768 IE, 1655 N	2233/957
monks-1	556 / 6 / 2	278 Zero, 278 One	389/167
monks-2	601 / 6 / 2	395 Zero, 206 One	420/181
monks-3	554 / 6 / 2	266 Zero, 288 One	387/167
nursery	12960 / 8 / 5	4320 not_recom, 2 recommend, 328 very_recom, 4266 priority, 4044 spec_prior	9072/3888
optdigits	5620 / 75 / 10	557 Zero, 571 One, 577 Two, 572 Three, 568 Four, 558 Five, 558 Six, 566 Seven, 554 Eight, 562 Nine	3934/1686
pendigits	10992 / 24 / 10	1143 Zero, 1143 One, 1144 Two, 1055 Three, 1144 Four, 1055 Five, 1056 Six, 1142 Seven, 1055 Eight, 1055 Nine	7694/3298
semeion	1593 / 257 / 10	161 Zero, 162 One, 159 Two, 159 Three, 161 Four, 159 Five, 161 Six, 158 Seven, 155 Eight, 158 Nine	1115/478
spect-heart	267 / 22 / 2	157 Zero, 110 One	186/80
tic-tac-toe	958 / 9 / 2	626 Positive, 332 Negative	670/288
zoo	101 / 17 / 7	41 Class-1, 20 Class-2, 5 Class-3, 13 Class-4, 4 Class-5, 8 Class-6, 10 Class-7	70/31

Table 1: Data characteristics of the UCI Discrete group of datasets

A.2 UCI Continuous Group

Data Set	Instances/ Attributes/ Classes	Class Distribution	Train/Test Split
bank	4521 / 16 / 2	4000 Zero, 521 One	3164/1357
blood	748 / 4 / 2	570 Zero, 178 One	523/225
breast-cancer-wisc-diag	569 / 30 / 2	357 Zero, 212 One	398/171
breast-tissue	106 / 9 / 6	21 Zero, 15 One, 18 Two, 16 Three, 14 Four, 22 Five	74/32
cardiotocography-10clases	2126 / 21 / 10	384 Zero, 579 One, 53 Two, 81 Three, 72 Four, 332 Five, 252 Six, 107 Seven, 69 Eight, 197 Nine	1488/638
ecoli	336 / 7 / 8	143 Zero, 77 One, 52 Two, 35 Three, 20 Four 5 Five, 2 Six, 2 Seven	70/30
glass	214 / 9 / 6	70 Zero, 76 One, 17 Two, 13 Three, 9 Four, 29 Five	149/65
hill-valley	1212 / 100 / 2	605 Zero, 606 One	848/364
image-segmentation	2310 / 18 / 7	330 Zero, 330 One, 330 Two, 330 Three, 330 Four, 330 Five, 330 Six	1617/693
ionosphere	351 / 33 / 2	126 Zero, 225 One	245/106
iris	150 / 4 / 3	50 Zero, 50 One, 50 Two	105/45
libras	360 / 90 / 15	24 Zero, 24 One, 24 Two, 24 Three, 24 Four, 24 Five, 24 Six, 24 Seven, 24 Eight, 24 Nine, 24 Ten, 24, Eleven, 24 Twelve, 24 Thirteen, 24 Fourteen	252/108
musk-2	6598 / 166 / 2	5580 Zero, 1018 One	4618/1980
oocytes_merluccius_nucleus_4d	1022 / 41 / 2	337 Zero, 685 One	715/307
oocytes_trisopterus_states_5b	912 / 32 / 3	525 Zero, 14 One, 373 Two	638/274
optical	5620 / 62 / 10	554 Zero, 571 One, 557 Two, 572 Three, 568 Four, 558 Five, 558 Six, 566 Seven, 554 Eight, 562 Nine	3934/1686
ozone	3536 / 72 / 2	2463 Zero, 73 One	1775/761
page-blocks	5473 / 10 / 5	4913 Zero, 329 One, 28 Two, 88 Three, 115 Four	3831/1642
parkinsons	195 / 22 / 2	48 Zero, 147 One	136/59
pendigits	10992 / 16 / 10	1143 Zero, 1143 One, 1144 Two, 1055 Three, 1144 Four, 1055 Five, 1056 Six, 1142 Seven, 1055 Eight, 1055 Nine	7694/3298
planning	182 / 12 / 2	130 Zero, 52 One	127/55
post-operative	90 / 8 / 3	64 Zero, 2 One, 24 Two	63/27
ringnorm	7400 / 20 / 2	3664 Zero, 3736 One	5180/2220
seeds	210 / 7 / 3	70 Zero, 70 One, 70 Two	147/63
spambase	4601 / 57 / 2	2788 Zero, 1813 One	3220/1381
statlog-image	2310 / 18 / 7	330 Zero, 330 One, 330 Two, 330 Three, 330 Four, 330 Five, 330 Six	1617/693

continued from overleaf			
Data Set	**Instances/ Attributes/ Classes**	**Class Distribution**	**Train/Test Split**
statlog-landsat	6435 / 36 / 6	1533 Zero, 703 One, 1358 Two, 626 Three, 707 Four, 1508 Five	4504/1931
statlog-shuttle	58000 / 9 / 7	45586 Zero, 50 One, 171 Two, 8903 Three, 3267 Four, 10 Five, 13 Six	40600/17400
steel-plates	1941 / 27 / 7	158 Zero, 190 One, 391 Two, 72 Three, 55 Four, 402 Five, 673 Six	1358/583
synthetic-control	600 / 60 / 6	100 Zero, 100 One, 100 Two, 100 Three, 100 Four, 100 Five	420/180
twonorm	7400 / 20 / 2	3703 Zero, 3697 One	5180/2220
vertebral-column-3clases	310 / 6 / 2	60 Zero, 100 One, 150 Two	217/93
wall-following	5456 / 24 / 4	2205 Zero, 826 One, 2097 Two, 328 Three	3819/1637
waveform-noise	5000 / 40 / 3	1692 Zero, 1653 One, 1655 Two	3500/1500
wine-quality-white	4898 / 11 / 7	20 Zero, 163 One, 1457 Two, 2198 Three, 880 Four, 175 Five, 5 Six	3428/1470
yeast	1484 / 8 / 10	463 Zero, 429 One, 244 Two, 163 Three, 51 Four, 44 Five, 35 Six, 30 Seven, 20 Eight, 5 Nine	1038/446

Table 3: Data characteristics of the UCI Continuous group of datasets

A.3 MiddlePhalanxOutlineCorrect Dataset

Data Set	**Instances/ Attributes/ Classes**	**Class Distribution**	**Train/Test Split**
MiddlePhalanx-OutlineCorrect	891 / 80 / 2	337 Zero, 554 One	623/268

Table 4: Data characteristics of the MiddlePhalanxOutlineCorrect dataset

B Appendix B - Evaluation Results

B.1 Testing ID3CW variants with UCI Discrete Data against ID3 and J48

Data Set	ID3CW-IG	ID3CW-Chi	ID3CW-Yates	ID3CW-Gini	J48	ID3
balance-scale	0.6310	0.6182	0.6374	0.6535	0.6545	0.4545
chess-krvk	0.1808	0.2121	0.1798	0.1788	0.5339	0.3740
chess-krvkp	0.9933	0.9941	0.9956	0.9937	0.9913	0.9939
connect-4	0.7379	0.7381	0.7386	0.7399	0.8001	0.7307
contraceptive-method	0.4579	0.4344	0.4385	0.4502	0.4796	0.3756
fertility	0.7067	0.6867	0.7133	0.7134	0.8533	0.7267
habermans-survival	0.3544	0.3696	0.3522	0.3804	0.7543	0.3652
hayes-roth	0.6000	0.5500	0.5650	0.5650	0.7050	0.6700
led-display	0.2520	0.2547	0.2466	0.2520	0.7000	0.6880
lymphography	0.5455	0.5364	0.5591	0.5227	0.7682	0.6818
molecular-promoters	0.7562	0.7062	0.6312	0.6437	0.8375	0.7562
molecular-splice	0.7198	0.7181	0.7101	0.6959	0.9381	0.8863
monks-1	0.9796	0.9748	0.9641	0.9605	0.9581	0.9832
monks-2	0.6000	0.6089	0.5889	0.6089	0.6489	0.6156
monks-3	0.9627	0.9675	0.9603	0.9651	0.9904	0.9687
nursery	0.7073	0.6105	0.6347	0.6321	0.9630	0.9713
optdigits	0.2694	0.2708	0.2599	0.2501	0.6045	0.4513
pendigits	0.3182	0.3482	0.3240	0.3292	0.5900	0.3605
semeion	0.2276	0.2569	0.2121	0.2310	0.7477	0.7402
spect-heart	0.6450	0.6575	0.6150	0.6600	0.7150	0.6875
tic-tac-toe	0.8425	0.8349	0.8153	0.8397	0.8502	0.8334
zoo	0.6333	0.7267	0.5600	0.6333	0.9067	0.9200

Table 6: Average accuracies from five runs of the given classifiers against the UCI Discrete group of data

B.2 Testing ID3CW variants with UCI Continuous data against J48

Data Set	ID3CW-IG	ID3CW-Chi	ID3CW-Yates	ID3CW-Gini	J48
bank	0.8541	0.8572	0.8515	0.8517	0.8886
blood	0.7937	0.7688	0.7625	0.7812	0.7732
breast-cancer-wisc-diag	0.9497	0.9380	0.9322	0.9415	0.9404
breast-tissue	0.3750	0.3750	0.3500	0.3625	0.6500
cardiotocography-10clases	0.3724	0.4661	0.3972	0.3373	0.8254
ecoli	0.4198	0.5069	0.4514	0.4633	0.8079
glass	0.3969	0.4469	0.4375	0.4250	0.6688

continued from overleaf					
Data Set	**ID3CW-IG**	**ID3CW-Chi**	**ID3CW-Yates**	**ID3CW-Gini**	**J48**
hill-valley	0.5363	0.5033	0.5088	0.5242	0.4742
image-segmentation	0.2880	0.3714	0.3463	0.3174	0.9431
ionosphere	0.8381	0.8876	0.8057	0.8381	0.8743
iris	0.7111	0.6800	0.7022	0.6578	0.9244
libras	0.0907	0.0981	0.0963	0.0963	0.6019
musk-2	0.9543	0.9564	0.9528	0.9539	0.9596
oocytes-merluccius-nucleus-4d	0.6964	0.6932	0.6918	0.6840	0.7505
oocytes-trisopterus-states-5b	0.7423	0.7810	0.7380	0.6912	0.8803
optical	0.2363	0.2572	0.2419	0.2480	0.8923
ozone	0.9469	0.9485	0.9516	0.9490	0.9622
page-blocks	0.8953	0.9056	0.8972	0.8984	0.9681
parkinsons	0.8966	0.8517	0.8448	0.8414	0.8517
pendigits	0.2710	0.2897	0.2420	0.2674	0.9550
planning	0.6218	0.6037	0.6473	0.6400	0.7309
post-operative	0.6444	0.6074	0.6148	0.6741	0.6889
ringnorm	0.8137	0.8049	0.8195	0.8234	0.9076
seeds	0.6254	0.6794	0.7079	0.6635	0.9270
spambase	0.9185	0.9126	0.9104	0.9077	0.9242
statlog-image	0.2750	0.2958	0.2811	0.2975	0.9544
statlog-landsat	0.5410	0.5126	0.4928	0.5056	0.8564
statlog-shuttle	0.7872	0.7974	0.7885	0.7863	0.9994
steel-plates	0.4333	0.5058	0.3914	0.4179	0.7326
synthetic-control	0.3400	0.3189	0.3433	0.3289	0.9045
twonorm	0.8497	0.8422	0.8506	0.8494	0.8446
vertebral-column-3clases	0.7850	0.7699	0.7785	0.7871	0.7936
wall-following	0.4730	0.4484	0.5070	0.5309	0.9956
waveform-noise	0.6523	0.6643	0.6618	0.6497	0.7557
wine-quality-white	0.4554	0.4696	0.4519	0.4475	0.5724
yeast	0.3083	0.3272	0.3061	0.3196	0.5600

Table 8: Average accuracies from five runs of the given classifiers against the UCI Continuous group of data

B.3 Testing TreeEnsemble with Discrete data

Data Set	**ID3CW**	**Tree-Ensemble-Random**	**Tree-Ensemble-IG**	**Tree-Ensemble-Chi**	**Tree-Ensemble-Yates**	**Tree-Ensemble-Gini**
balance-scale	0.6310	0.8021	0.7904	0.8053	0.8022	0.7925
chess-krvk	0.1808	0.1987	0.1963	0.2096	0.1830	0.1886
chess-krvkp	0.9933	0.9610	0.9585	0.9583	0.9610	0.9545
connect-4	0.7379	0.7409	0.7383	0.7387	0.7389	0.7396
contraceptive-method	0.4579	0.4991	0.5113	0.5000	0.5027	0.4932
fertility	0.7067	0.8267	0.8400	0.9200	0.8600	0.8600
habermans-survival	0.3544	0.7544	0.7478	0.7283	0.7696	0.7457
hayes-roth	0.6000	0.5950	0.5700	0.5900	0.6100	0.5250
led-display	0.2520	0.2853	0.3387	0.3240	0.3867	0.2720
lymphography	0.5455	0.5864	0.5545	0.5318	0.6045	0.5682
molecular-promoters	0.7562	0.8188	0.8000	0.8188	0.8375	0.8688
molecular-splice	0.7198	0.8571	0.8201	0.8640	0.8182	0.8295
monks-1	0.9796	0.7222	0.7653	0.7784	0.6982	0.7341
monks-2	0.6000	0.6711	0.6433	0.6745	0.6844	0.6434
monks-3	0.9627	0.9361	0.9132	0.9060	0.9277	0.8988
nursery	0.7073	0.6250	0.6692	0.6308	0.7100	0.5885
optdigits	0.2694	0.3239	0.2741	0.3510	0.2944	0.2884
pendigits	0.3182	0.4979	0.4241	0.4445	0.3985	0.4231
semeion	0.2276	0.2469	0.2523	0.2870	0.2213	0.2389
spect-heart	0.6450	0.6925	0.7250	0.7200	0.7175	0.6825
tic-tac-toe	0.8425	0.8000	0.7575	0.7819	0.7722	0.7909
zoo	0.6333	0.5800	0.4333	0.6933	0.5933	0.5400

Table 10: Average accuracies from 5 runs of the given classifiers with default settings on the UCI discrete group of data

Data Set	**ID3CW**	**Tree-Ensemble-Random**	**Tree-Ensemble-IG**	**Tree-Ensemble-Chi**	**Tree-Ensemble-Yates**	**Tree-Ensemble-Gini**
balance-scale	0.6310	0.8021	0.8043	0.7872	0.7936	0.8043
chess-krvk	0.1808	0.1991	0.1885	0.2060	0.1929	0.1929
chess-krvkp	0.9933	0.9691	0.9712	0.9670	0.9696	0.9764
connect-4	0.7379	0.7348	0.7370	0.7366	0.7309	0.7368
contraceptive-method	0.4579	0.4991	0.5149	0.5086	0.5104	0.5091
fertility	0.7067	0.9067	0.8467	0.8733	0.8867	0.8800
habermans-survival	0.3544	0.7435	0.6848	0.7348	0.7326	0.7087
hayes-roth	0.6000	0.6350	0.6450	0.7100	0.6850	0.6200
led-display	0.2520	0.2600	0.2920	0.3066	0.2907	0.2720
lymphography	0.5455	0.5955	0.6455	0.5409	0.5409	0.5273

Continued from overleaf

Data Set	ID3CW	Tree-Ensemble-Random	Tree-Ensemble-IG	Tree-Ensemble-Chi	Tree-Ensemble-Yates	Tree-Ensemble-Gini
molecular-promoters	0.7562	0.8375	0.8687	0.8625	0.8437	0.8438
molecular-splice	0.7198	0.8495	0.8238	0.8711	0.8399	0.8391
monks-1	0.9796	0.7258	0.7293	0.7497	0.7461	0.7449
monks-2	0.6000	0.6422	0.6333	0.6600	0.6633	0.6522
monks-3	0.9627	0.8735	0.9024	0.9157	0.9337	0.9157
nursery	0.7073	0.6055	0.6504	0.6671	0.6568	0.6046
optdigits	0.2694	0.3050	0.2677	0.3448	0.2841	0.2699
pendigits	0.3182	0.4907	0.4375	0.4198	0.4224	0.4819
semeion	0.2276	0.2255	0.2376	0.2971	0.2034	0.2272
spect-heart	0.6450	0.6950	0.6750	0.6950	0.7275	0.7175
tic-tac-toe	0.8425	0.7958	0.8084	0.7819	0.7770	0.7826
zoo	0.6333	0.5333	0.4933	0.7733	0.5800	0.5067

Table 12: Average accuracies from 5 runs of the given classifiers with setSampledPercentage set to 100 on the UCI discrete group of data

Data Set	ID3CW	Tree-Ensemble-Random	Tree-Ensemble-IG	Tree-Ensemble-Chi	Tree-Ensemble-Yates	Tree-Ensemble-Gini
balance-scale	0.6310	0.8021	0.8043	0.7872	0.7936	0.8043
chess-krvk	0.1808	0.1991	0.1885	0.2060	0.1929	0.1929
chess-krvkp	0.9933	0.9691	0.9712	0.9670	0.9696	0.9764
connect-4	0.7379	0.7348	0.7370	0.7366	0.7309	0.7368
contraceptive-method	0.4579	0.4991	0.5149	0.5086	0.5104	0.5091
fertility	0.7067	0.9067	0.8467	0.8733	0.8867	0.8800
habermans-survival	0.3544	0.7435	0.6848	0.7348	0.7326	0.7087
hayes-roth	0.6000	0.6350	0.6450	0.7100	0.6850	0.6200
led-display	0.2520	0.2600	0.2920	0.3066	0.2907	0.2720
lymphography	0.5455	0.5955	0.6455	0.5409	0.5409	0.5273
molecular-promoters	0.7562	0.8375	0.8687	0.8625	0.8437	0.8438
molecular-splice	0.7198	0.8495	0.8238	0.8711	0.8399	0.8391
monks-1	0.9796	0.7258	0.7293	0.7497	0.7461	0.7449
monks-2	0.6000	0.6422	0.6333	0.6600	0.6633	0.6522
monks-3	0.9627	0.8735	0.9024	0.9157	0.9337	0.9157
nursery	0.7073	0.6055	0.6504	0.6671	0.6568	0.6046
optdigits	0.2694	0.3050	0.2677	0.3448	0.2841	0.2699
pendigits	0.3182	0.4907	0.4375	0.4198	0.4224	0.4819
semeion	0.2276	0.2255	0.2376	0.2971	0.2034	0.2272
spect-heart	0.6450	0.6950	0.6750	0.6950	0.7275	0.7175

Continued from overleaf

Data Set	ID3CW	Tree-Ensemble-Random	Tree-Ensemble-IG	Tree-Ensemble-Chi	Tree-Ensemble-Yates	Tree-Ensemble-Gini
tic-tac-toe	0.8425	0.7958	0.8084	0.7819	0.7770	0.7826
zoo	0.6333	0.5333	0.4933	0.7733	0.5800	0.5067

Table 14: Average accuracies from 5 runs of the given classifiers with setNumClassifiers set to 250 on the UCI discrete group of data

B.4 Testing Ensemble classifiers against the Case Study dataset

Classifier	Accuracy	Balanced Accuracy	Negative Log Likelyhood	AUROC
TreeEnsemble	0.7565	0.7052	0.9339	0.7490
J48	0.7487	0.7261	1.4766	0.7093
Bagging	0.7835	0.7539	0.6647	0.8580
RandomForest	0.7992	0.7769	0.6509	0.8552
RotationForest	0.8216	0.8028	0.5661	0.8952
LogitBoost	0.7543	0.7166	0.7720	0.7951
DecisionStump	0.6533	0.5388	0.9610	0.5305

Table 16: Average performance measures from ten-fold cross validation on the given classifiers on the MiddlePhalanxOutlineCorrect dataset

Classifier	Accuracy	Balanced Accuracy	Negative Log Likelyhood	AUROC
TreeEnsemble	0.8194	0.7902	0.5972	0.8742
J48	0.7240	0.6996	1.3687	0.7148
Bagging	0.7801	0.7494	0.6787	0.8447
RandomForest	0.7722	0.7344	0.6671	0.8388
RotationForest	0.7947	0.7656	0.6161	0.8704
LogitBoost	0.7465	0.6985	0.7765	0.7696
DecisionStump	0.5724	0.5118	0.9804	0.5183

Table 18: Average performance measures from ten-fold cross validation on the given classifiers on the discretized MiddlePhalanxOutlineCorrect dataset