Confidence in Predictions from Random Tree Ensembles

Siddhartha Bhattacharyya
Information and Decision Sciences, College of Business Administration, University of Illinois, Chicago
sidb@uic.edu

***Abstract*—** Obtaining an indication of confidence of predictions is desirable for many data mining applications. Predictions complemented with confidence levels can inform on the certainty or extent of reliability that may be associated with the prediction. This can be useful in varied application contexts where model outputs form the basis for potentially costly decisions, and in general across risk sensitive applications. The conformal prediction framework presents a novel approach for obtaining valid confidence measures associated with predictions from machine learning algorithms. Confidence levels are obtained from the underlying algorithm, using a non-conformity measure which indicates how 'atypical' a given example set is. The non-conformity measure is key to determining the usefulness and efficiency of the approach. This paper considers inductive conformal prediction in the context of random tree ensembles like random forests, which have been noted to perform favorably across problems. Focusing on classification tasks, and considering realistic data contexts including class imbalance, we develop non-conformity measures for assessing the confidence of predicted class labels from random forests. We examine the performance of these measures on multiple datasets. Results demonstrate the usefulness and validity of the measures, their relative differences, and highlight the effectiveness of conformal prediction random forests for obtaining predictions with associated confidence.

Keywords- prediction confidence; random forests; conformal prediction; classification; data mining.

1. INTRODUCTION

Obtaining an indication of confidence of predictions is desirable for many data mining applications. Predictions complemented with confidence levels can inform on the certainty or extent of reliability that may be associated with the prediction. This can be useful, for example, where model outputs form the basis for potentially costly decisions, where decision makers use predictions in determining actions that involve allocation of limited resources, and in general for risk sensitive applications. Here, one may focus on high confidence predictions, or seek alternate strategies to deal with lower confidence cases. This can be beneficial in various applications, ranging from direct marketing, customer attrition and retention efforts, fraud prediction in credit card, insurance, healthcare, etc., network intrusion, to churn or bankruptcy predictions and medical diagnosis. The focus in this paper is on confidence values for classification problems such as these, where the dependent variable specifies a class label. In many of the aforementioned problems, examples of the 'positive' class of interest - fraud, response, bankruptcy, etc. - is often in a minority, since data typically carries fewer such cases. This requires care in handling unbalanced data, and is an issue considered in our study.

The PAC (Probably Approximately Correct) learning framework in machine learning provides error bounds on predictions, assuming only an i.i.d. distribution of the data. These bounds, however, have been noted to be often overly loose with practical data; further, such bounds apply to the overall error rate rather than for predictions on specific cases. Bayesian methods can also be applied to obtain confidence of predictions, but make strong assumptions on the data and can be problematic when these are violated [15]. In regression settings, estimating uncertainty around predictions is standard practice for linear regression, and has been examined for more general settings in neural networks [10, 18]. An interesting recent study utilizes the dyadic structure of some data for determining certainty of predictions [7].

The conformal prediction (CP) framework [17, 22] presents a relatively new and novel approach for complementing predictions with valid confidence measures. Separate confidence levels are obtained for individual

predictions, and the only assumption is that the training data and examples to be predicted be independently drawn from the same distribution. It provides a powerful means for 'hedging' of individual predictions with valid confidence levels [10]. One obtains predictions that can be considered correct with a probability at least as high as the corresponding confidence. Alternately, for a specified confidence level or threshold, it provides a prediction set that includes the class labels predicted at that confidence. Prediction sets can be empty, indicating lack of a prediction at the desired confidence, or can include multiple labels when the training data and developed model do not allow predicting a single label with certainty. Along with confidence, CP also provides a measure of 'credibility' for each prediction, indicating the extent to which the data can be considered adequate for making the prediction.

CP is based on the underlying learning algorithm, and uses a non-conformity measure which indicates how 'atypical' a given example set is. For an example to be predicted, all potential class labels are attempted, and the 'strangeness' of each in the context of the set of training data examples then measured; this is used to obtain the likelihood of different labels being accurate, and a confidence value thereby obtained. It thus provides confidence values associated with different potential prediction labels. The development of conformal prediction for k-nearest neighbors and support vector machines as underlying learning algorithms has been described in [22]. It has been applied with neural networks [16], regression, and recent work considers its use with random forests [8, 25, 26]. Most of this work uses CP in a transductive inference setting, which requires model re-learning for each new example to be predicted, and can be computationally burdensome. Here, we consider inductive conformal prediction [16], where a model is first developed from training data, and a separate calibration dataset is then used for estimating confidence.

This paper considers the use of inductive conformal predictions with random decision tree ensembles as in random forests. Random forests [3] have been popular in application in recent years, with outstanding performance across problems, and have been noted to perform favorably in comparison with the strongest machine learning and statistical techniques [5,19,2,23]. Recent work shows their application to large data [1] and to broader data mining problems like feature induction [20]. They are conceptually simple, computationally efficient, robust to noise, and readily applicable, and do not require onerous parameter optimization. The notion of proximity given by random forests (RF) provides a natural measure of similarity between cases, and has shown some interesting applications for clustering, outlier analysis and visualization [4]. We propose two new proximity based measures of non-conformity for estimating confidence of predictions from random forests. For performance comparison, two measures from the literature are also included.

Performance of random forest based conformal predictions is shown for five datasets from different domains. Three of these have been used in various machine learning studies, a fourth is a marketing dataset from the Direct Marketing Educational Foundation, and the fifth is from a real life credit card fraud detection context. We focus on these five here to show details that help bring out performance differences among the nonconformity measures. Experimental results demonstrate the usefulness and validity of the proposed non-conformity measures, their relative differences, and highlight the effectiveness of conformal prediction random forests for obtaining predictions with associated confidence levels.

The next section provides a background on conformal prediction, including inductive conformal prediction. Section 3 begins with an overview of random forests and then details the non-conformity measures developed. Section 4 then describes the data used for evaluation and experimental results. The final section provides a summary and directions for future research.

2. CONFORMAL PREDICTION

This section provides a brief overview of the conformal prediction mechanism, and details the inductive conformal prediction procedure used. Details on conformal prediction can be found in [22,17]. Inductive conformal prediction is described in [16].

Consider a set of examples $z_i=(x_i, y_i)$, $i = 1,..n-1$, where $x_i \in R^d$ is the vector of attributes (independent variables) and $y_i \in Y$ is the class label (dependent variable). Let x_n be a new example whose class label we want to predict. The essential idea of conformal prediction is to try each class label $c \in Y$ as the prediction for x_n and measure the 'randomness' of the sequence

$z_1=(x_1, y_1), \ldots, z_{n-1}=(x_{n-1}, y_{n-1}), z_n=(x_n, c)$;

that is, how well the sequence conforms with the i.i.d assumption for the data, or, how likely it is that the sequence is drawn independently from the same distribution. Measuring the 'typicalness' of a sequence, considering a predicted label c for x_n is akin to determining the likelihood that this is the true label for x_n, since all the other (x_i,y_i) in the sequence, besides (x_n, c), are from the given data.

The test of randomness for a sequence is based on a non-conformity measure which determines how atypical an example in a sequence is, relative to the other examples. It is defined as a set of functions $A_n: Z^{n-1} \times Z \rightarrow R$ and assigns a non-conformity score for each example in the sequence:

$\alpha_i = A_n(\{z_1, .., z_{i-1}, z_{i+1},..,z_n\}, z_i)$,

where {..} denotes a bag, rather than a set, to signify that order of examples in the sequence should not matter to the calculation. The non-conformity scores are then used to obtain a p-value for the sequence as

$p(z_1, ..,z_n) = |i: i=1..n: \alpha_i \geq \alpha_n| / n$.

A smoothed function is noted to have better properties [18], and is defined as

$(|i:i=1..n, \alpha_i > \alpha_n| + \eta|i:i=1..n, \alpha_i = \alpha_n|)/n$,

where $\eta \in [0,1]$ is uniformly randomly obtained (the smoothed function differs only in terms of randomly breaking ties $\alpha_i = \alpha_n$)

For every potential label y for an example to be predicted, the p-value indicates the extent to which this label conforms with other examples in the sequence. These p-values provide, for any i.i.d. distribution P and every significance level ϵ,

$P\{ p(c) \leq \epsilon \} \leq \epsilon$.

Thus, if the p-value $p(c)$ for a prediction $c \in Y$, is less than or equal to ϵ, then either c is not the accurate label, or the sequence including c is a rare occurrence with probability at most ϵ. The predicted label for an example is taken as the class label $c \in Y$ corresponding to the largest p-value. The second largest p-value from among the potential labels in Y gives the highest probability of any label other than that corresponding to the maximum p-value. The *confidence* for the predicted label is thus taken as 1 minus the second largest p-value. The highest p-value is also taken as the *credibility* for the prediction, and gives an indication of adequacy of the training data for making the prediction. A low credibility value would mean that either the training data is non-random, or the predicted example is not representative with the training data. One typically seeks high confidence predictions that also carry a credibility that is not too low.

The p-values can also be used to obtain prediction sets $\Gamma^{\epsilon} \subseteq Y$ at defined significance levels ϵ; a prediction set gives the labels $\{c:p(c)>\epsilon\}$. This gives the set of possible labels at confidence 1-ϵ. Note that with $\epsilon_1 \geq \epsilon_2$, $\Gamma^{\epsilon 1} \subseteq \Gamma^{\epsilon 2}$ [9]. An empty prediction set means that no predictions are possible at that confidence, while multiple labels in the prediction set imply uncertainty with regard to the class label. A single label in the prediction set indicates certain prediction at the given confidence level. The extent of certain predictions obtained gives the efficiency of conformal prediction, and can vary with the non-conformity measures used.

CP in a transductive setting, as originally developed, requires the underlying algorithm to be invoked for every new example, and for each class label that can be predicted. This is computationally inefficient, and can be infeasible for many data mining problems and where the underlying algorithm is compute-intensive. Inductive Conformal Prediction [16] addresses this inefficiency by splitting the training data into a separate (proper) training

set and a calibration set. A model is first developed from the proper training data, and then applied to the calibration data together with the new example to be predicted, to obtain a p-values for each potential label for the example. Let $z_1,..z_t$ be the proper training data and $z_{t+1},..z_{t+s}$ be the calibration data. Inductive conformal prediction then operates as follows:

1. Develop a model (random forest, in our case) from the training data $z_1,..z_t$.
2. Determine the conformity scores $\alpha_{t+1}, .., \alpha_{t+s}$ for all examples in the calibration set.
3. Given a new example x_p for prediction:
   For each potential label $c \in Y$:
   Determine α_p^c for (x_p, c)
   Compute the p-value for the sequence $z_{t+1},\ldots,z_{t+s}$, (x_p, c) and corresponding $\alpha_{t+1}, .., \alpha_{t+s}$, α_p^c as

$$p(c) = (|\, i = t+1,\ldots,t+s, x_p : \alpha_i \geq \alpha_p^c \,|)/(s+1).$$

The class label corresponding to the largest p-value is output as the prediction. Alternately, one may apply a confidence threshold T_{conf} and output the prediction region (that is, the predicted labels at the threshold confidence). A credibility threshold T_{cred} can also similarly be applied. One seeks predictions with high confidence and credibility that is not lower than, say, 5% [10].

3. RANDOM FORESTS AND NON-CONFORMITY MEASURES

3.1 Random Forests

Random forests (RF) [3] combine the concepts of random subspace and bagging to build a set of classification or regression trees from bootstrapped samples of the data. A random subset of attributes is considered in tree induction; typically, a subset of $\sqrt{A}$ attributes is considered at every step, where A is the total number of attributes in the data. Predictions are made by aggregating the scores of individual trees in the ensemble. RFs are computationally efficient since each tree is built independently of the others. With a large number of trees in the ensemble, they are also noted to be robust to overfitting and noise in the data. They have been popular in application in recent years, with demonstrated strong performance across domains from marketing, image classification, and fraud detection to various bio-medical problems. A recent survey of applications is given in [21].

Proximity between a pair of cases is determined as the count of trees where the two cases occur at a common leaf node, divided by the number of trees in the ensemble. If $r_{i,j}$ represents proximity between two cases i and j in the data, $(1-r_{i,j})$ gives a measure of distance between the cases.. This is a distance measure between cases in a high-dimensional Euclidean space. Given the separation of cases achieved by the random forest ensemble, this distance measure has been found to be effective in various ways – for imputing missing values in the data, clustering, and outlier detection [3,4]. A scatter plot of the cases on the principal components of a related distance matrix can be used for obtaining a multi-dimensional scaling plot, useful for visualizing the separation between the data. In this paper, we utilize this novel measure of closeness for defining non-conformity.

Conformal prediction with random forests is of recent interest. In [26], non-conformity is defined similarly to how random forests can be used for detecting outliers [4]; a nonconformity measure that considers how distant or outlying an example is from other examples of the same class is used, and a variant where non-conformity is compared only with other cases of the same class in obtaining p-values. Considering outlyingness of an example, in this way, relative to all cases within a class, however, does not make adequate use of local neighborhood patterns (where, arguably, tree based models derive much of their advantage from). The authors, in another paper [25], propose a nonconformity measure defined similarly to k-nearest neighbor conformal predictors [17, 22], but with distance determined by inverse RF proximity. Reference [8] suggests a simple and intuitive measure of non-conformity based on the proportion of trees that correctly classify an example, and compares this with the k-

proximity measure of [25]. Both these measures, however, may not perform well when there in class imbalance in the data.

3.2 Random Forest Non-comformity Measures

This section gives the details of the non-conformity measures designed for use with random forests. Two new non-conformity measures based on RF proximities are defined. For comparison, two other non-conformity measures are also included: a simple measure based on RF classification as suggested in [8], and a k-nn based measure using proximity for distance as in [25]. Experiments evaluate performance of the proposed new non-conformity measures, and their advantages relative to the others.

Consider a random forest model, *rf*, developed from the training data. Assume a certain label for the new example to be predicted, and consider the extended set of calibration examples together with the new example. Each example here thus carries an actual class label (for the new example to be predicted, the 'actual' class label is the assumed label; as noted above, p-values are computed assuming every possible class label in turn for the new example). To simplify notation, we refer to this extended set as F and generically refer to an example in F as (x_f, c) where c is its actual class label for x_f; we also use $class(x_f)$ to denote the actual class label. This set of examples is evaluated using the *rf* model, that is, passed through each tree in the ensemble. Every tree thus predicts a class label for each example, and proximities between each pair of examples are also available. These are used in the non-conformity measures defined here.

Proximity values give a measure of closeness of examples in the high dimensional proximity space, and can be used in a manner analogous to k-nearest neighbor based non-conformity measures suggested in [8,20]. There, non-conformity scores are defined as the ratio of the sum of k-nearest neighbor distances with examples of the same class to the sum of k-nearest neighbor distances with examples of the other classes:

$$\alpha_i = \frac{\text{distance of k closest cases of the same class}}{\text{distance of k closest cases of different class}}.$$

Where an example is close to other examples of the same class, and far from examples of other classes, the non-conformity score will be small; and it will be large where an example is closer to examples of other classes than it is to examples of the same class. Applying the same intuition with proximities, and noting that proximity is the reverse of distance, non-conformity can be defined as the reciprocal of the ratio of average k-nearest proximity considering examples of the same class, to the average k-nearest proximity considering examples of the other classes. In other words, non-conformity of an example is taken as the ratio of its closeness with examples of other classes to its closeness with examples of the same class. Closeness is measured by the average values of proximity to its k-nearest-proximity examples.

$$\alpha_i = \frac{\text{average proximity of k closest cases of different class}}{\text{average proximity of k closest cases of the same class}}.$$

1)Random Forest Proportion (rfP)

This is a simple measure of non-conformity, as in [8], based on the proportion of trees in the ensemble that vote for the actual class of examples in F. The non-conformity score for an example x_f is 1 minus the proportion of trees that vote for the actual class label, c, of x_f. Let a_f^c denote this non-conformity score for an example x_f with actual class c. The p-value is then calculated as:

$$p(x_f, c) = (|\, a_i^c \geq a_f^c : \mathrm{x_i} \in \mathrm{F \text{ where } } class(x_i) = c\,|) / |F| \qquad (2)$$

We use the smoothed value as shown in the Section 2.

2) K-nearest neighbor Proximity (kProx)

This non-conformity measure, used in [25], follows the standard k-nearest neighbor based measure for non-conformity [9, 22] as shown above. Considering proximity as inverse-distance, non-conformity can be taken as:

$$a_f^C = \sum_{i=1}^{K} prox(x_f^C, x_i^{\neg C}) / \sum_{j=1}^{K} prox(x_f^C, x_i^C).$$

Here, $prox(x_f^C, x_j^{\neg C})$ is the proximity of a case (x_f, C) with a case x_i of a non-C class. Based on the a_f^C, the p-values are calculated in the usual way (2).

3) Relative neighborhood proximity (relProx)

This measure of non-conformity is also based on the average k-nearest neighbor proximity of an example. It, however, considers the fact that examples of different classes can show different patterns of dispersion in proximity space, as shown in Figure 1a; this can be especially so where the classes are not equally present in the data, as in unbalanced data that is typical to many data mining problems. In such situations, the average proximity between examples of one class may be very different from that for examples of another class. To help counter such differences that can hinder accurate estimation, we normalize the average k-neighbor proximity for an example belonging to a class by dividing it by the average k-neighbor proximity for all examples of that class. The K-neighborhood proximity to examples of the same class is given by

$$r_K^c(x_f) = \sum_{i=1}^{K} r(x_f^C, x_i^C)$$

for K-closest proximity cases x_i of x_f of the same class C. The overall average of k-neighbor class proximities for class c is then

$$\overline{r^c} = \sum_f r_f^c / n_c,$$

where the summation is over examples f where $class(x_f)=c$, and n_c gives the number of such examples in F. The relative k-neighbor proximity for an example is then given by

$$r^c(x_f) = r_k^c(x_f) / (\overline{r_k^c}).$$

The non-conformity scores are obtained as

$$a_f^c = r^{\neg c}(x_f) / r^c(x_f),$$

where $\neg c$ indicates examples of classes other than c. The p-values are then calculated as in (2) for the measure above.

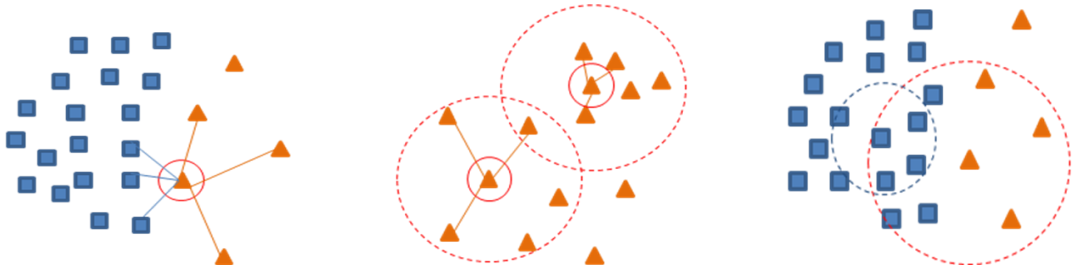
4) Number of close proximity examples (numProx)

The rationale for this measure comes from observing that while effectiveness of random forests for classification is seen in the separation of examples achieved in proximity space, this separation may vary across regions of this space. While some regions might have a larger number of examples, other areas may carry fewer examples and which may be relatively more apart, as illustrated in Figure 1b. This can give rise to inconsistency in non-conformity scores across examples in different regions. Taking the count of close proximity cases of the same class, without considering their actual 'distances' can then be beneficial.

Here, rather than the actual proximity values that show how close the examples are, we instead use the count of examples that are within a certain proximity range. Non-conformity is then defined as 1 minus the number of examples of the same class that are within a certain proximity range. For an example x_f whose actual class is c,

$$a_f^c = 1 - (|i : r(x_i, x_f) > \theta, class(x_j) = c|).$$

The p-value is then obtained using (2). The parameter θ acts as a minimum proximity threshold for examples to be considered in the count. Since proximity between two cases denotes the proportion of trees in the ensemble where these cases occur in the same leaf node, a high threshold will consider only those examples that are very similar in the context of the ensemble; high values of the parameter can result in low counts for many of the examples, leading to unreliable behavior. A value of 0.1 implies that only those examples that occur together in 10% of the trees in the ensemble are considered. In general, different thresholds can be applied for the different classes (Figure 1c). These will need to be set with careful consideration of specific data.



1(a): Two classes, dispersed differently

1(b): Same class, different dispersion

1(c) Two classes, different thresholds

Figure 1: Measuring non-conformity from differently dispersed data in proximity space

4. EXPERIMENTS AND RESULTS

4.1 Data

The non-conformity measures above have been evaluated on various datasets. This section examines their performance and usefulness in application to five datasets from different domains. Given space considerations, we restrict attention here to these five datasets to help bring out performance details and differences. Three of these are from the UCI Machine Learning Repository and have been used in numerous studies. Another, dmef, is an extract from the publicly available Direct Marketing Educational Foundation datasets. The fifth comes from a real-life dataset on credit card fraud, a problem where random forests have shown strong performance in recent studies [2, 23]. These have been chosen here to cover a range of application areas, and considering datasets for classification problems which are of at least moderate size to allow for separate training, calibration and test subsets. Table 1 gives basic information of these.

Considering binary classification, the dependent variable in the splice and waveform datasets are recoded. For waveform, the classes 1 and 2 are combined. For splice, E1 is taken as the positive class of interest, to give greater imbalance among the two classes. 30% of the data is separated out for the test set, and the remaining used for training. In applying inductive conformal prediction, we use a 70:30 split of the training data for the proper training and calibration subsets.

TABLE 1 DATASETS USED

Dataset	Size	% 'positive' examples	# attributes
spambase	4601	39.4%	58
splice	3190	24%	61
waveform	5000	33%	41
dmef	14,500	27%	58
ccfraud	98,498	5%	20

The dmef and ccfraud data are samples from a larger original dataset, which suffice for our experiments. For dmef, 58 attributes were taken after routine processing, and out of the sample of 14,500 cases, 2500 were used as a test set (the original data has 99,200 cases). The ccfraud data is a sample from a larger dataset from an international credit card operation, and after processing, has 20 derived attributes. Since the full data carried very few fraudulent transactions, a sample of non-fraud cases was taken together with all known fraud cases. From this, datasets with different fraud rate can be obtained for experimentation. In the sample considered, the test data has 14,477 cases with 5% fraud rate. A proper training data with higher fraud rate of 25% was used, and the fraud rate for the calibration data was kept at 5% to match that in the test data; both the proper training and calibration sets were obtained from sampling the training dataset of 84,021 cases.

4.2 Experimental Results

Experiments consider the effectiveness of conformal prediction with random forests, using the four non-conformity measures. We examine performance at different confidence levels (equivalently, significance levels). Performance is considered at significance levels of 0.01, 0.025, .05, 0.1 and 0.2, corresponding to confidence levels of 99%, 97.5%, 95%, 90% and 80% respectively. As noted in [9], a high-confidence prediction can be trusted if its accompanying credibility is not lower than around 5%. We report results for credibility of 5%; at much higher levels of credibility, as expected, fewer cases are predicted. At high confidence levels, we expect fewer errors; all examples, though, may not receive a definite prediction. With lower confidence, more of the data should be predictable. The efficiency of conformal prediction, as given by the proportion of examples that receive a unique class prediction, is recorded for different confidence levels. We also consider the error rate (err) at different levels of confidence, and the proportion of overall positive and negative examples that are correctly predicted with specified confidence (corr+ and corr- respectively).

Classification performance among the cases that receive a single prediction is examined, at different levels of confidence, through standard measures – accuracy (acc), true positive rate (tpr), true negative rate (tnr), precision (prec), and true negative accuracy (tna). These follow usual definitions: accuracies of the positive and the negative class examples as given by true positive rate (tpr) and true negative rate (tnr); precision (prec) shows the proportion of predicted true cases that are actually true; and the true negative accuracy (tna) gives the proportion of predicted negative cases that are actually negative. Performance is reported as percentages, except for the error rate, which is indicated as a simple proportion, to help relate with the significance level.

For each dataset, we also show performance for the regular random forest model. This helps in evaluating the relative performance of conformal prediction at different levels of confidence. Note that conformal prediction is not intended to outperform the underlying random forest classifiers. Rather, its advantage arises from being able to establish confidence levels in predictions. Our interest is thus more in observing how effective the different non-conformity measures are, in terms of how many cases are predicted with confidence, the accuracy on these, and general classification performance, at different given levels of confidence. Given this focus, the simple measures of classification performance noted above suffice, and we thus do not show values for other measures like auc, F-measure, etc.

Results shown are averages over 10 runs. A neighborhood size K=10 was used for the *relProx* measure for all datasets; the proximity threshold value θ for the *numProx* measure was set at 0.1, except for the splice dataset where setting θ to 0.0 was found more effective. These parameter values were experimentally determined for reasonable performance. In the results below, numbers in the column labeled 'sig' are the significance levels, where confidence equals 1 minus the significance. All numbers shown, except for sig and err, are in percentage.

We also include results for the *kProx* measure for the ccfraud data; as expected, and noted in Section 3.2.1, this measure may be problematic with imbalanced classes in the data; our findings confirm this, and given space limitations, we do not present details on this measure for other datasets.

1) Efficiency of conformal prediction

Efficiency of conformal prediction is measured by the proportion of total examples that receive a single prediction. It is also important to consider the proportions of total cases of the two classes that are correctly

predicted, especially with data that shows imbalance between examples of two classes. Table 2 shows the percentage of total cases that receive a definite (single) prediction, at different levels of confidence, for the different datasets. The proportion of predicted cases at different confidence levels is also in the graphs in Figure 1 (relProx is labeled relX in graphs). As expected, we find that the proportion of total cases predicted with a single label rises as the confidence level is lowered from 99%, but can begin to decrease after a point. The confidence level at which a maximum number of predictions is obtained varies by dataset; it is generally at around 95% for the spambase, splice and ccfraud data, and lower, around 90% for the waveform data; for the dmef data, fewer high confidence predictions are obtained with any of the non-conformity measures. The relative efficiency of different confidence measures is found to be data dependent. At 99%, except for dmef data where few examples are predicted with 99% confidence, the efficiency of prediction varies by non-conformity measure. The rfP and kProx measures obtain somewhat higher number of definite predictions at 99% confidence. Overall, all the measures are found effective at predicting confidence, and a large proportion of data obtains single predictions.

Table 3 gives the proportions of total cases of the 'positive' and 'negative' classes that are correctly predicted at different significance levels. As seen here, the higher predictions with the rfP and kProx measures can arise from predicting more of the majority class examples. This is noticed for the waveform, splice and ccfraud datasets, and especially for the dmef data, as seen in the graphs in Figure 2. On the dmef data, rfP is unable to predict any of the positive class cases with high confidence. This is not surprising, since the rfP measure is directly derived from random forest votes for different classes. The performance of rfP follows that of the underlying random forest model – notice from the bottom row of Table 3 that the random forest models show higher accuracy on the more numerous negative class examples. With the kProx measure too, predictions are biased towards examples of the majority class. It predicts very few cases with high confidence on the dmef data, while the two new measures, relProx and numProx, are able to predict more of the positive class cases with confidence. On the ccFraud data, the kProx measure is again seen to predict fewer 'positive' cases than the other measures. This is also evident from the graph for ccFraud in Figure 3, where the number of positive class examples that are correctly predicted is much lower for *kProx*, compared with other measures

The relProx measure, which explicitly seeks balance among the classes by normalizing the proximities against the average proximity of different classes, is found to be effective at predicting positive cases with confidence. This is noticed especially for the waveform, splice and dmef datasets (see Figure 2), where it correctly predicts a greater proportion of positive cases than negative cases. On the splice data, the numProx measure also predicts more of the positive than negative class cases accurately. The numProx measure generally does well in predicting a large proportion of cases at around the 95% and 90% confidence levels, and it does so for both classes. At the highest confidence level of 99%, however, it predicts fewer cases.

2) Validity of conformal prediction

The values of error (err) in Table 4 indicate how many of the predicted cases are in error, as a fraction of the total number of cases. Comparing these with significance values in the same row, we find that observed error rates are close to or lower than the significance levels corresponding to the prediction confidence. This shows the validity of inductive conformal prediction with the proposed random forest measures. Error rate at all the 80% to 99% confidence levels considered are noted to be lower than that for regular random forests. In certain cases, the error rate is seen to drop at lower confidence levels – for instance, lower error at significance of 0.2 than for 0.1. This occurs due to fewer cases that receive a single prediction at lower confidence; error rate here is thus calculated out of fewer predicted cases.

3) Classification effectivenes

Tables 5 show values for different measures of classification performance obtained using the non-conformity measures on the different datasets. These pertain to cases that are predicted with a single value at different confidence levels. At high confidence, less of the data is predicted, but classification performance is seen to be better. For data where the positive class examples are fewer in number, the precision is noticed to be generally better at higher confidence levels – this arises from greater prediction accuracy at high confidence levels.

Compared to regular random forests, random forest conformal prediction gives better classification performance at higher confidence, across datasets.

Performance is seen to vary with the different non-conformity measures. The graphs in Figure 4 highlight some differences in performance amongst the measures. These plot the accuracy, and true positive and true negative rates. Performance of the rfP measure is closer to that of regular random forests than for the proximity based measures. As noted earlier, this is only to be expected, since the rfP measure of non-conformity is based on the proportion of trees that vote for the predicted class of an example, and is thereby more directly related to random forest classifications. The relProx and numProx measures generally obtain higher true positive rates. The weakness of rfP on true positive rates is seen in the Figure 4 graphs for waveform, spambase, and especially for the dmef data where it largely fails to accurately predict the minority class. The kProx measure also shows poor performance for the positive class, arising from its being overwhelmed by the majority class, as evident from the graphs for dmef and ccFraud data.

Results for the dmef data present an interesting comparison of the different non-conformity measures. Regular random forests do not perform well on the minority (positive) class cases (tpr of 26.9%). Here, the rfP measure is able to predict only very few positive class cases with confidence (none at 99% and 95% confidence); at 80% confidence, the tpr is comparable with random forests. The kProx measure also shows similar performance with respect to accurate positive class prediction. The proposed numProx and relProx measures, designed to take into account potentially different patterns of separation of the two classes in proximity space, show much better performance on the minority positive class. The relProx measure shows somewhat better performance on the dmef data, compared with numProx. The difficulty with the dmef data may be due to the relatively small sample considered; nonetheless, it serves a useful purpose here in illustrating differences among the proposed non-conformity measures. It also serves to highlight the fact that with inadequate data, fewer predictions are possible at high confidence.

Table 2: Percentage of definite (single) predictions of different non-conformity measures

Data/ Method	sig	spambase	waveform	splice	dmef	ccFraud
	0.01	79.47	65.37	93.34	5.08	88.32
	0.025	92.04	75.34	95.60	13.18	96.03
	0.05	98.01	84.51	95.60	24.47	98.47
	0.1	93.14	97.43	90.79	48.30	91.55
rfP	0.2	82.28	84.42	80.99	88.36	80.39
	0.01	76.37	64.84	93.43	8.40	95.15
	0.025	91.17	73.14	95.94	18.20	96.15
	0.05	98.14	82.19	95.94	31.43	96.15
	0.1	92.94	95.94	90.59	52.97	90.56
kProx	0.2	82.29	85.43	81.31	84.63	80.87
	0.01	63.22	30.71	79.87	6.83	43.95
	0.025	89.80	71.35	86.50	15.99	81.34
	0.05	97.95	82.75	92.17	32.79	99.07
	0.1	90.44	97.63	97.67	54.61	90.11
numProx	0.2	72.21	79.86	83.27	81.05	75.15
	0.01	75.06	47.19	72.21	7.29	62.47
	0.025	90.43	59.53	91.28	13.97	95.05
	0.05	98.31	71.26	98.99	21.63	98.31
	0.1	93.03	87.75	93.67	34.88	91.91
relProx	0.2	82.71	91.99	81.48	57.68	81.64

Table 3: Accuracy on positive and negative classes for different non-conformity measures

Data/		spambase		waveform		splice		dmef		ccFraud	
Method	sig	corr+	corr-	corr+	corr-	corr+	corr-	corr+	corr-	corr+	corr-
	0.01	71.97	82.36	45.87	73.16	80.09	96.62	0.00	5.45	64.08	88.46
	0.025	84.28	92.64	55.48	80.56	84.60	97.39	0.00	14.48	79.34	94.27
	0.05	89.36	95.94	63.45	86.38	84.60	97.39	0.04	27.05	84.77	95.60
	0.1	85.35	93.50	74.40	92.52	73.88	95.67	1.94	53.16	70.39	91.10
rfP	0.2	74.52	85.10	63.51	86.32	47.54	91.29	17.99	87.90	47.55	81.62
	0.01	69.61	79.03	51.30	69.80	82.41	95.90	0.14	10.16	36.86	97.21
	0.025	84.21	91.26	58.26	75.91	87.72	96.95	1.11	21.44	44.96	97.62
	0.05	89.26	95.90	65.08	82.04	87.72	96.95	3.54	35.45	44.96	97.62
	0.1	85.40	92.89	75.25	89.56	76.16	94.77	9.33	55.97	15.73	93.87
kProx	0.2	75.85	84.23	67.67	83.99	51.74	90.43	21.94	80.83	7.36	84.26
	0.01	76.05	72.72	58.76	39.99	92.90	64.02	7.16	5.44	61.16	61.51
	0.025	86.27	88.90	68.37	50.80	98.84	85.78	15.38	9.10	79.39	93.23
	0.05	91.10	94.90	77.60	59.70	99.64	94.20	22.10	13.46	83.09	95.62
	0.1	87.39	91.58	87.23	70.27	96.56	89.97	31.99	20.85	72.70	90.94
relProx	0.2	81.36	81.02	89.11	72.96	91.56	76.53	47.04	31.92	65.87	81.02
	0.01	65.96	59.47	29.90	29.28	88.26	75.97	5.56	4.92	60.88	41.96
	0.025	85.33	88.19	64.05	70.39	95.09	80.45	11.23	12.39	79.26	78.75
	0.05	90.11	94.94	72.83	78.01	99.06	83.89	23.16	25.69	89.04	93.88
	0.1	85.51	89.06	82.13	86.11	99.96	87.35	34.11	41.52	83.44	86.92
numProx	0.2	74.94	68.05	70.56	76.18	92.32	78.29	51.82	54.31	75.70	72.85
Random forest		90.51	97.00	76.40	93.41	90.58	90.58	26.93	93.36	87.52	96.31

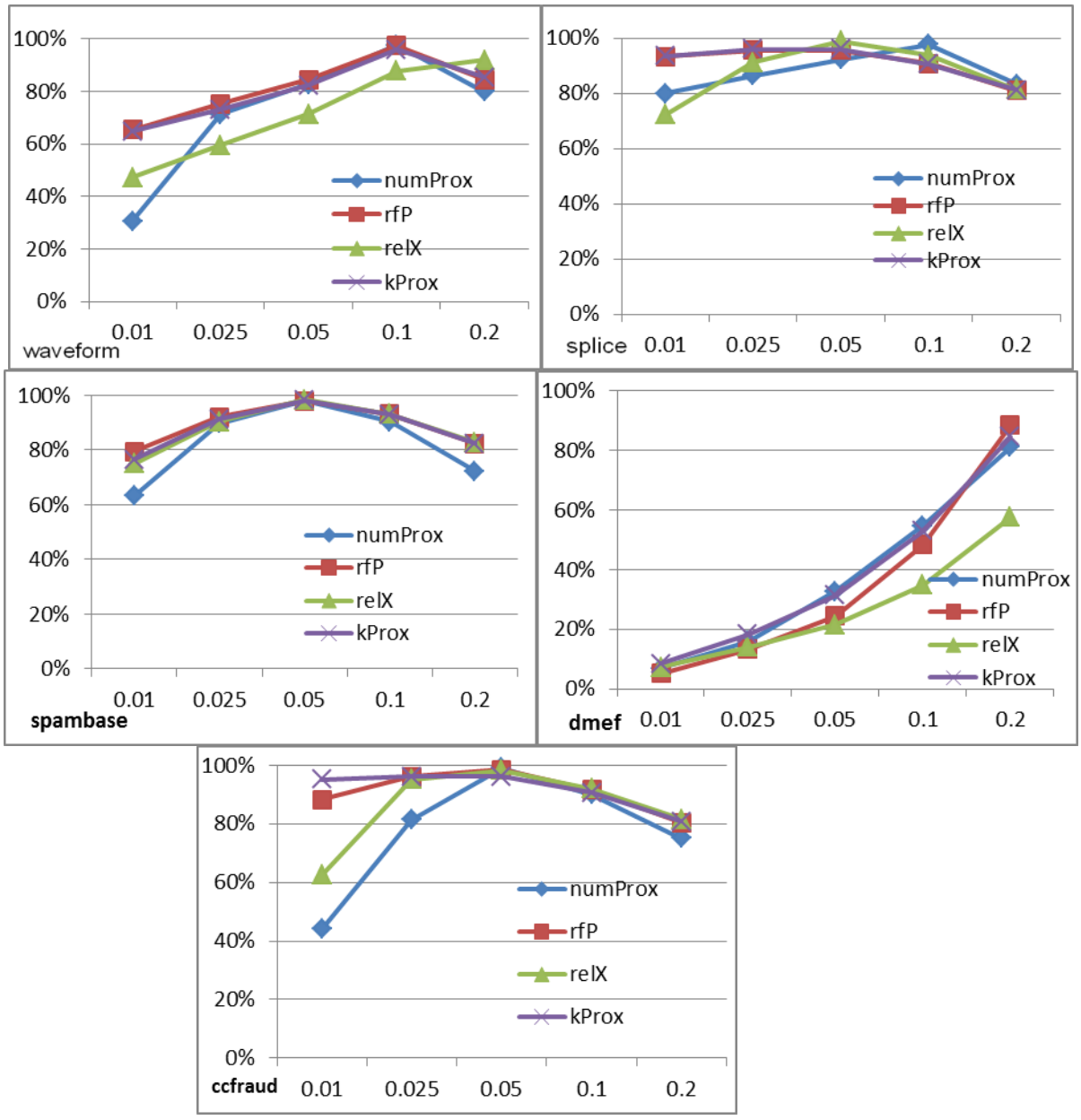
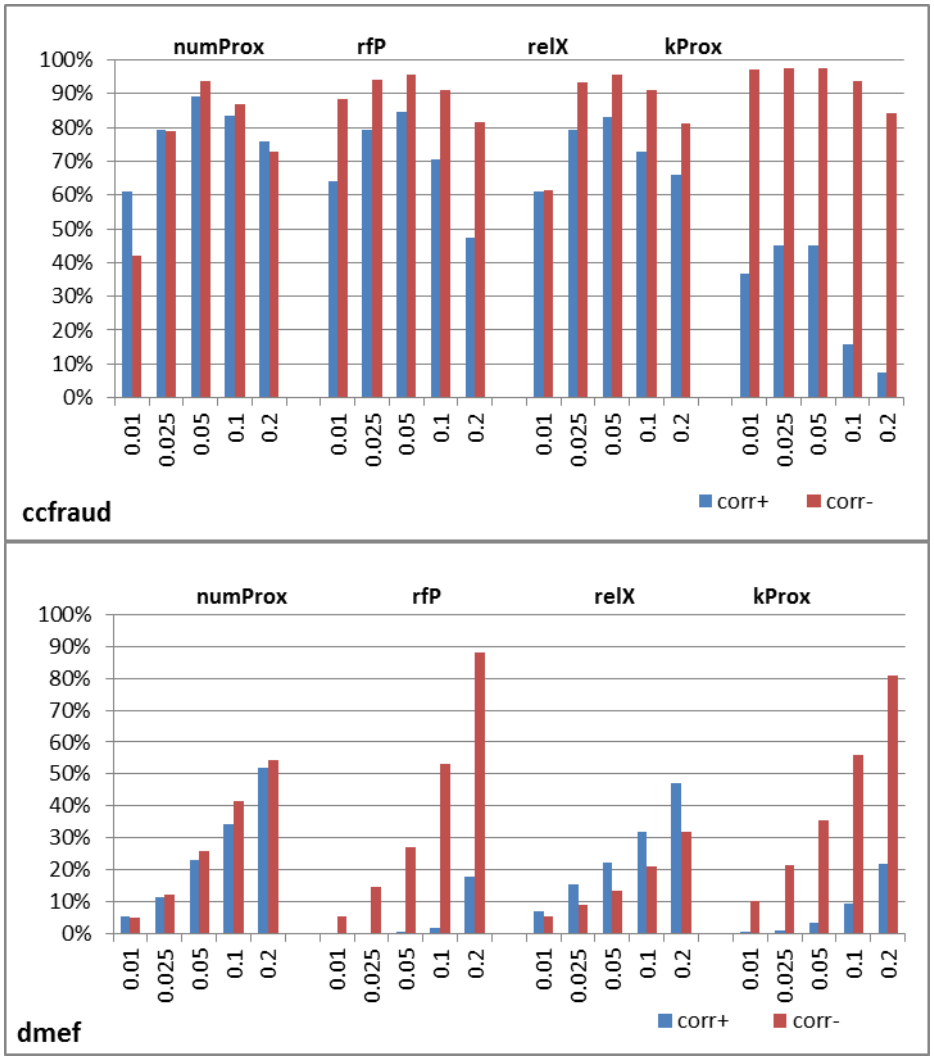


Figure 2: % cases predicted at different significance (confidence) levels


numProx
rfP
relX
kProx
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
0.01
0.025
0.05
0.1
0.2
spambase
corr+
corr-
numProx
rfP
relX
kProx
waveform
corr+
corr-
numProx
rfP
relX
kProx
splice
corr+
corr-



Figures 3: % of positive and negative class cases accurately predicted at different significance (confidence) levels

Table 4: Error rate of different non-conformity measures

Data/ Method	sig	spambase	waveform	splice	dmef	ccFraud
	0.01	0.012	0.012	0.010	0.017	0.010
	0.025	0.027	0.031	0.026	0.039	0.026
	0.05	0.049	0.065	0.047	0.078	0.054
	0.1	0.028	0.129	0.074	0.152	0.034
numProx	0.2	0.014	0.056	0.017	0.275	0.022
	0.01	0.012	0.016	0.006	0.012	0.011
	0.025	0.027	0.034	0.012	0.028	0.025
	0.05	0.047	0.060	0.012	0.051	0.034
	0.1	0.028	0.111	0.003	0.098	0.015
rfP	0.2	0.013	0.059	0.000	0.204	0.005
	0.01	0.010	0.007	0.014	0.014	0.010
	0.025	0.026	0.027	0.024	0.031	0.025
	0.05	0.049	0.054	0.035	0.057	0.033
	0.1	0.031	0.116	0.021	0.108	0.019
relProx	0.2	0.016	0.135	0.014	0.214	0.014
	0.01	0.011	0.014	0.007	0.011	0.010
	0.025	0.027	0.033	0.012	0.026	0.012
	0.05	0.049	0.060	0.012	0.051	0.012
	0.1	0.030	0.113	0.002	0.103	0.006
kProx	0.2	0.014	0.070	0.000	0.206	0.005
Random forest		0.056	0.124	0.034	0.256	0.041

Tables 5 a. & b CF performance on (a) spambase data, (b) waveform data

		spambase					waveform				
ccfraud	sig	acc	tpr	prec	tnr	Tna	acc	tpr	prec	tnr	Tna
	0.01	98.15	97.25	98.28	98.78	98.05	93.23	91.45	89.48	93.97	95.14
	0.025	96.96	95.49	96.71	97.91	97.12	95.64	94.52	92.33	96.18	97.31
	0.05	94.99	92.71	94.38	96.45	95.37	92.15	90.15	86.99	93.17	94.91
	0.1	96.93	95.53	96.57	97.82	97.16	86.82	84.29	78.82	88.14	91.47
numProx	0.2	98.01	97.11	98.15	98.67	97.90	93.00	91.35	88.09	93.83	95.59
	0.01	98.49	96.80	99.08	99.47	98.16	97.57	91.65	99.09	99.70	97.08
	0.025	97.10	94.37	97.98	98.79	96.58	95.49	86.66	97.69	99.14	94.73
	0.05	95.24	91.60	96.09	97.59	94.74	92.89	82.55	94.03	97.59	92.47
	0.1	96.95	94.17	97.79	98.68	96.45	88.57	77.59	87.28	94.19	89.15
rfP	0.2	98.36	96.58	98.98	99.41	98.02	92.98	82.72	94.20	97.66	92.53
	0.01	98.65	97.96	98.71	99.08	98.63	98.42	99.97	96.50	97.20	99.98
	0.025	97.1	96.00	96.75	97.92	97.45	95.50	99.66	90.08	92.71	99.77
	0.05	95.02	92.98	94.26	96.34	95.50	92.41	99.28	83.73	88.22	99.52
	0.1	96.67	95.2	96.19	97.59	96.97	86.73	96.39	74.09	81.39	97.67
relProx	0.2	98.12	97.37	97.91	98.60	98.27	85.39	95.42	72.05	80.01	97.04
	0.01	98.64	97.19	99.10	99.47	98.40	97.86	94.29	98.25	99.30	97.72
	0.025	97.06	95.10	97.16	98.28	97.00	95.50	89.22	95.86	98.28	95.36
kProx	0.05	95.05	91.71	95.47	97.20	94.80	92.74	85.09	91.65	96.34	93.21

	0.1	96.77	94.63	96.88	98.11	96.71	88.24	79.72	84.67	92.59	89.92
	0.2	98.35	96.76	98.81	99.29	98.08	91.78	83.68	90.39	95.69	92.38
Random forest		94.44	90.51	95.15	97.00	94.02	87.56	76.40	85.89	93.41	88.30

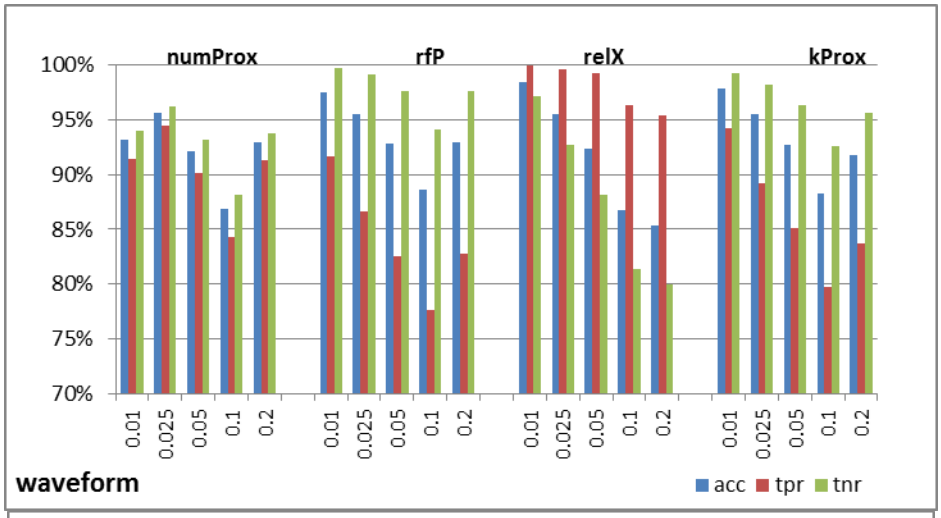
Table 5 c & d. CF performance on (c) splice and (d) dmef data

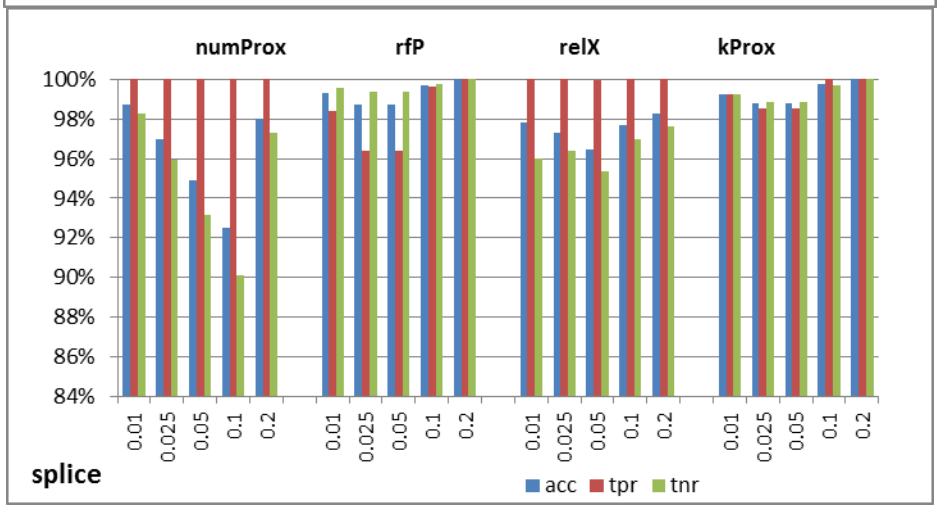
	splice						dmef				
ccfraud	sig	acc	tpr	prec	tnr	Tna	acc	tpr	prec	tnr	Tna
	0.01	98.74	100.0	95.41	98.30	100.0	74.47	73.12	58.77	75.29	85.88
	0.025	97.00	100.0	89.63	95.95	100.0	75.36	57.76	67.75	84.93	78.79
	0.05	94.91	100.0	83.31	93.19	100.0	76.14	63.42	62.50	82.16	82.70
	0.1	92.48	100.0	76.26	90.09	100.0	72.14	61.39	51.85	76.57	82.83
numProx	0.2	98.00	100.0	92.91	97.29	100.0	66.13	63.35	44.00	67.24	81.89
	0.01	99.34	98.38	98.51	99.60	99.56	76.32	0.00	0.00	100.0	76.32
	0.025	98.72	96.37	97.74	99.38	98.99	78.34	0.00	0.00	100.0	78.34
	0.05	98.72	96.37	97.74	99.38	98.99	78.95	0.19	0.00	100.0	78.94
	0.1	99.72	99.64	98.93	99.74	99.91	79.74	5.35	93.48	99.89	79.57
rfP	0.2	100.0	100.0	100.0	100.0	100.0	76.86	22.22	67.71	96.23	77.72
	0.01	97.82	100.0	94.06	96.01	100.0	81.47	77.33	73.85	83.44	86.64
	0.025	97.32	100.0	90.56	96.39	100.0	77.90	81.67	68.00	75.09	86.66
	0.05	96.46	99.96	87.08	95.37	99.99	73.59	80.49	60.36	69.43	86.13
	0.1	97.72	100.0	91.51	96.98	100.0	68.88	78.99	52.18	63.75	85.88
relProx	0.2	98.28	100.0	94.12	97.63	100.0	62.84	76.17	43.90	56.95	84.42
	0.01	99.26	99.25	97.41	99.27	99.79	86.77	2.87	0	99.86	86.81
	0.025	98.79	98.57	96.02	98.85	99.59	85.87	10.77	88.16	99.68	85.85
	0.05	98.79	98.57	96.02	98.85	99.59	83.78	17.08	79.99	99.01	83.94
	0.1	99.78	100.0	98.94	99.73	100.0	80.50	22.27	71.98	97.45	81.15
kProx	0.2	100.0	100.0	100.0	100.0	100.0	75.62	27.42	60.96	93.48	77.66
Random forest		96.59	90.58	94.73	98.45	97.14	74.37	26.93	61.94	93.36	76.14

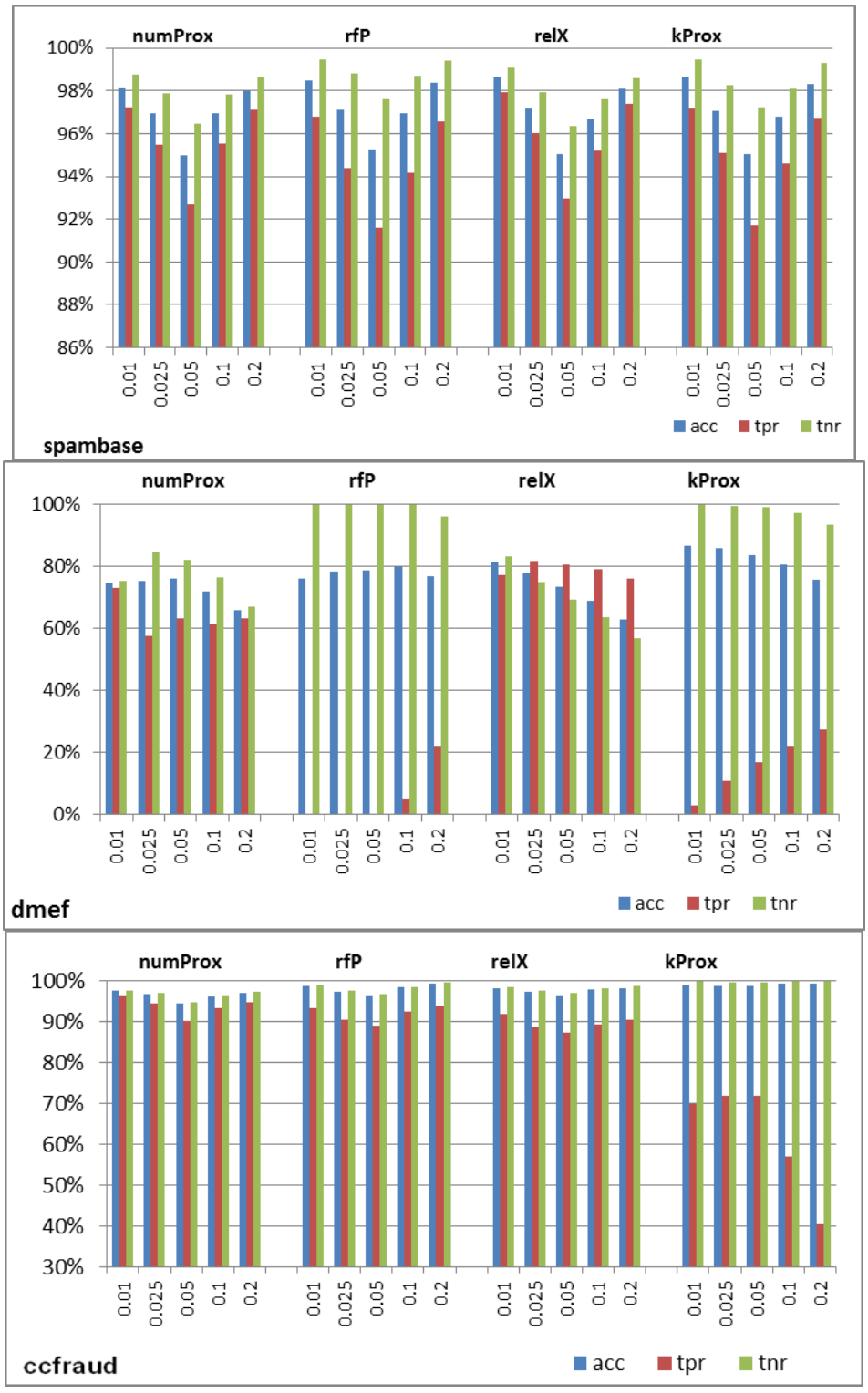
Table 5e. CF performance on ccfraud data

ccfraud	sig	acc	tpr	prec	tnr	Tna
	0.01	97.57	96.37	76.82	97.66	99.70
	0.025	96.84	94.47	62.99	96.97	99.69
	0.05	94.52	90.26	47.46	94.74	99.46
	0.1	96.27	93.44	57.77	96.42	99.65
numProx	0.2	97.13	94.76	66.29	97.27	99.70
	0.01	98.77	93.25	78.96	98.99	99.72
	0.025	97.38	90.53	65.48	97.71	99.54
	0.05	96.53	89.08	59.84	96.91	99.43
	0.1	98.37	92.44	74.67	98.62	99.67
rfP	0.2	99.40	93.94	88.13	99.58	99.80
	0.01	98.19	91.99	81.24	98.51	99.54
	0.025	97.35	88.70	66.51	97.78	99.43
	0.05	96.63	87.32	60.82	97.10	99.34
	0.1	97.95	89.37	72.12	98.34	99.50
relProx	0.2	98.31	90.55	77.75	98.67	99.56
kProx	0.01	98.98	69.74	91.60	99.81	99.15

	0.025	98.78	71.89	88.69	99.69	99.06
	0.05	98.78	71.89	88.69	99.69	99.06
	0.1	99.33	57.08	97.12	99.97	99.35
	0.2	99.41	40.33	98.37	99.99	99.42
Random forest		95.87	87.52	55.63	96.31	99.32







Figures 4: Accuracy, true-positive and true negative rates among definite predictions at different significance (confidence) levels

5. SUMMARY AND FUTURE WORK

Being able to provide confidence levels with individual predictions can bring significant advantages for many data mining problems. Conformal prediction presents a framework for developing such capability, and this paper considers predictions with confidence using random forests. Random forests, with demonstrated strong performance across domains, are increasingly in use for predictive modeling. The work in this paper considers inductive conformal prediction, and describes new non-conformity measures that are effective in realistic data contexts, and will be useful for complementing predictions from random forest models with confidence values.

Multiple non-conformity measures for use with random forests are examined. Two new measures, based on the useful concept of proximity available from random forests, are designed in consideration of class imbalance that may occur in data, and the uneven distributions of data points in proximity space that commonly arise. For comparison, two non-conformity measures suggested in the literature are also included. Experiments with multiple datasets confirm the effectiveness of the proposed measures, advantages they bring over current measures, and help evaluate performance differences. Overall, results show that random forest based conformal prediction is valid and useful. Performance observed at different confidence levels indicates that high confidence predictions can be obtained with adequate efficiency (efficiency measured by the proportion of total examples that receive a single prediction). The extent of data that can be predicted with high confidence is data dependent.

Performance comparisons reveal the advantages of the proposed random forest non-conformity measures. These are found to be particularly effective in the presence of class imbalance in the data; here, while the existing measures tend to be overwhelmed by the majority class data, the numProx and relProx measures show better ability to accurately capture minority class information in assessing prediction confidence. Such unbalanced data is common in many data mining contexts, and effective prediction of the minority class is usually of key concern. Random forests based conformal prediction using the new operators will be useful in this regard. In future work, the performance of different non-conformity measures with modified RF approaches for unbalanced data as in [6] will be interesting. Incorporation of mechanisms for addressing imbalanced data as in [24] into conformal prediction presents another avenue for future research.

Two recent studies have reported strong performance of random forests for fraud identification [23,2]. The effectiveness of random forest conformal prediction on the ccFraud data is encouraging, in terms of its potential for this important data-mining area. High confidence in predictions will be useful for flagging fraudulent transactions. Where predictions cannot be made with confidence, whether for fraud or otherwise, alternate checks may be prudent. Very low credibility of predictions on specific data implies an inadequacy in the data and given model for predicting such cases; it points to the need for obtaining more training data on similar cases. The use of conformal predictors and utilizing confidence and credibility of predictions for fraud detection is a valuable topic for further investigation.

Predictions complemented by confidence values present new opportunities in data mining. Any application where predictions form the basis for cost or resource sensitive decisions will find value from confidence and credibility of predictions – low values call for a need for caution in using model predictions, and alternate characterizations of risk from decisions. It can enable novel problem formulations and ways to incorporate model outputs, enhanced by confidence and credibility, in decision making. Approaches for handling high and lower confidence predictions in model implementation, and identifying data for increased confidence of predictions remain useful areas for further research.

Various data mining areas involve costly data acquisition, and active learning approaches seek the most informative training examples to obtain labels for. Considering low credibility for specific cases as indicative of inadequate data, future research can explore approaches to identify the most useful additional training data examples to enhance prediction credibility and confidence.

While the focus in this paper was on classification problems, exploring conformal predictions for random forest regressions is also a useful topic of investigation. Confidence and credibility of predictions in text mining applications where models are developed on often sparse data with a large number of word vector attributes, will also be interesting.

REFERENCES

[1] Basilico, J.D., M. A. Munson, T. G. Kolda, K. R. Dixon and W. P. Kegelmeyer. 2011. COMET: A Recipe for Learning and Using Large Ensembles on Massive Data, *Proceedings of the 2011 IEEE International Conference on Data Mining* (ICDM 2011), 41-50.

[2] Bhattacharyya,S., Jha,S., Tharakunnel,K., & Westland,J.C. 2011.Data mining for credit card fraud: A comparative study. *Decision Support Systems, 50(3),602-613.*

[3] Breiman,L. RandomForests.2001. *Mach.Learn.*45,5–32.

[4] Breiman, L.and Cutler, A. 2005. Random Forest website: http://www.math.usu.edu/~adele/forests.

[5] Caruana,R., Karampatziakis,R. & Yessenalina,A.. 2008. An empirical evaluation of supervised learning in high dimensions. In *Proceedings of the 25th International conference on Machine learning* (ICML '08), 96-103.

[6] Chen,C., Liaw,A., Breiman,L.2004. Using Random Forest to Learn Imbalanced Data, Technical Report 666, Statistics Department, University of California at Berkeley, 2004.

[7] Deodhar,M. & J.Ghosh. 2009. Mining for the most certain predictions from dyadic data. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (KDD '09),249-258.

[8] Devetyarov, D. and I. Nouretdinov, "Prediction with Confidence Based on a Random Forest Classifier", in Proc. AIAI, 2010, pp.37-44.

[9] Dietterich, T. G.2002.Ensemble Learning. *The Handbook of Brain Theory and Neural Networks*, 2nd ed., (M.A. Arbib, Ed.), Cambridge, MA: The MIT Press.

[10] Gammerman,A., Vovk.V.2007.Hedging predictions in machine learning. *Comput. J.* 50, 2 (March 2007), 151-163.

[11] T. Heskes. 1997. Practical confidence and prediction intervals. In *Advances in Neural Information Processing Systems 9* (NIPS'97), 176–82.

[12] Hulse,J.V.,Khoshgoftaar,T.M.,Napolitano,A.2007.Experimental Perspectives on Learning from Imbalanced Data, In *Proceedings of the 24th international conference on Machine learning* (ICML '07), 935-942.

[13] Lambrou, A., Papadopoulos, H., Gammerman, A. 2011. Reliable Confidence Measures for Medical Diagnosis with Evolutionary Algorithms. *IEEE Trans. Information Technology in Biomedicine,* 15(1), (Jan. 2011), 93-99.

[14] Laxhammar,R.,Falkman,G. 2010. Conformal prediction for distribution-independent anomaly detection in streaming vessel data. In Proceedings of the First International *Workshop on Novel Data Stream Pattern Mining Techniques (StreamKDD '10).* 47-55.

[15] Melluish, T., Saunders, C., Nouretdinov, I., Vovk, V. 2001. Comparing the Bayes and Typicalness frameworks, In *Proceedings of the 12th European Conference on Machine Learning* (EMCL '01), 360-371.

[16] Papadopoulos, H., Vovk, V., Gammerman, A. 2007. Conformal prediction with neural networks. In *Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence,*Vol. 2, 388-395.

[17] Shafer,G., Vovk.V. 2008. A tutorial on conformal prediction. *J.Mach.Learn.Res*.,9:371-421,2008.

[18] Shrestha, D. and D. Solomatine. 2006. Machine learning approaches for estimation of prediction interval for the model output. *Neural Networks*, 19(2):225–235.

[19] Statnikov,A., Wang,L., Aliferis,C.F., 2008. A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification, *BMC Bioinformatics*, July 22 (2008) 9-319.

[20] Vens, C. & Costa, F. 2011. Random Forest Based Feature Induction. *2011 IEEE 11th International Conference on Data Mining*(ICDM, 2011), 744-753

[21] Verikas, A. Gelzinis, and M. Bacauskiene. 2011. Mining data with random forests: A survey and results of new tests. *Pattern Recogn.* 44, 2 (February 2011), 330-349.

[22] Vovk, V., Gammerman, A. ,Shafer, G. 2005. *Algorithmic Learning in a Random World*, Springer, New York.

[23] Whitrow,C., Hand,D.J., Juszczak,P., Weston,D., Adams,N.M. 2009. Transaction Aggregation as a Strategy for Credit Card Fraud Detection, *Data Mining and Knowledge Discovery*, 18(1) (2009), 30-55.

[24] Wang, B.and N. Japkowicz. 2010. Boosting support vector machines for imbalanced data sets. *Knowledge and Information Systems*, 25(1), 1-20.

[25] Wang,H., Lin,C., Yang,F., Hu.X. 2009. Hedged predictions for traditional Chinese chronic gastritis diagnosis with confidence machine. *Comput. Biol. Med.* 39, 5 (May 2009), 425-432.

[26] Yang, F., H. Wang, H. Mi, C. Lin, and W. Cai, "Using random forest for reliable classification and cost-sensitive learning for medical diagnosis", BMC Bioinformatics, 10 (Suppl 1), 2009.