Detecting and Ordering Salient Regions

Larry Shoemaker[1], Robert E. Banfield[1], Lawrence O. Hall[1], Kevin W. Bowyer[2], and W. Philip Kegelmeyer[3]

[1] Computer Science and Engineering, University of South Florida, Tampa, FL 33620-5399
Tel.: (813)974-3652, Fax: (813)974-5456
lwshoema@cse.usf.edu
rbanfiel@cse.usf.edu
hall@cse.usf.edu

[2] Computer Science and Engineering, University of Notre Dame, South Bend, IN 46556
kwb@cse.nd.edu

[3] Sandia National Labs, Computer and Information Sciences, PO Box 969, MS 9951,
Livermore, CA 94551
wpk@sandia.gov

Abstract. We describe an ensemble approach to learning salient regions from arbitrarily partitioned data. The partitioning comes from the distributed processing requirements of large-scale simulations. The volume of the data is such that classifiers can train only on data local to a given partition. Since the data partition reflects the needs of the simulation, the class statistics can vary from partition to partition. Some classes will likely be missing from some or even most partitions. We combine a fast ensemble learning algorithm with scaled probabilistic majority voting in order to learn an accurate classifier from such data. Since some simulations are difficult to model without a considerable number of false positive errors, and since we are essentially building a search engine for simulation data, we order predicted regions to increase the likelihood that most of the top-ranked predictions are correct (salient). Results from simulation runs of a canister being torn and from a casing being dropped show that regions of interest are successfully identified in spite of the class imbalance in the individual training sets. Lift curve analysis shows that the use of data driven ordering methods provides a statistically significant improvement over the use of the default, natural time step ordering. Significant time is saved for the end user by allowing an improved focus on areas of interest without the need to conventionally search all of the data.

Key words: Random forest · Saliency · Probabilistic voting · Imbalanced training data · Lift

1 Introduction

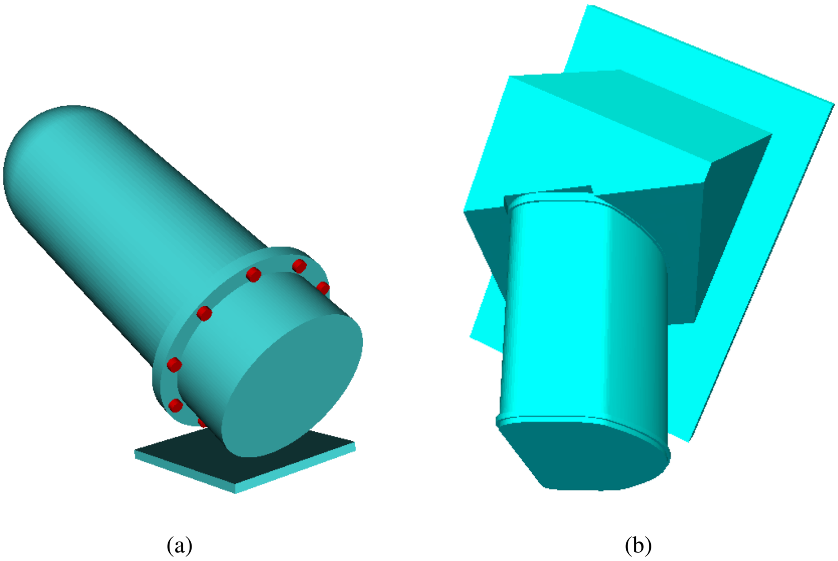
We consider the problem of dealing with datasets too large to fit in the memory of any one computer and too bandwidth-intensive to move between neighboring computers [20]. Such problems exist in the United States Department of Energy's Advanced Simulation and Computing (ASC) program [27, 1], wherein a supercomputer simulates a hypothetical real-world event. The simulation data is partitioned and distributed across separate disks, to facilitate parallel computation. It may be many terabytes to petabytes in size. The current state of the art is that developers spend time manually browsing for anomalies in order to develop the simulation, and domain experts spend similar time looking for salient events. We want to create a tool to let them manually mark a small number of examples and then automatically flag found examples throughout the rest of the dataset, or similar datasets for directed browsing. Because there will be false positives, we want to present predicted positives to the user in an order that increases the chances of true positives being presented early.

As a result of partitioning, the points of interest, or "salience", in some partitions may be limited to only a few nodes. Salient points, being few in number, exhibit a pathological minority class classification problem. The problems associated with imbalanced datasets and various strategies for dealing with those problems are described in [26] and [41]. Techniques include various forms of undersampling and oversampling [8], and cost-sensitive learning methods [13]. In the case of a partition having zero salient points, a single-class "classifier" will be learned. This motivates a scaling adjustment to the voting scheme used in [36] and [38], and developed in [37] to improve accuracy, as shown in Section 4. Facial region recognition experiments and analysis of nodal as well as regional accuracy were also included in [38]. A different, smaller dataset with only four partitions was used in both [36] and [38]. We first used the ordering of salient regions and the use of lift quality to measure the quality of the ordering for the casing simulation only in [37].

In this paper, which is revised and expanded from an earlier four-page version by the authors in [37], we give new examples of learning from four simulation runs of a canister being torn, and expand on learning from one run of a casing being dropped. These are relatively small simulations, used in initial investigations in the ASC program, but large enough to show the utility and advantages of our approach. A visualization of a casing being dropped is shown in Figure 1a. An illustration of the canister tear simulation model appears in Figure 1b. We have evaluated how well our approach can detect connected groups of salient nodes. Also, we have measured the quality of our ordering of salient region predictions, as discussed in Section 5. We show that it is possible to obtain an accurate prediction and a useful ordering of salient regions, even when the data is broken up arbitrarily in 3D space with no particular relation to feature space. Results on the canister tear and casing datasets indicate that experts working with much larger simulations can benefit from the predictive guidance obtained from only a small amount of relevant data.

As a final piece of evidence for the utility of this approach, one of the authors (Kegelmeyer) of this paper assisted with a real-world example of a much larger simulation that involved 162 runs of 876 gigabytes of data in each run. The original data is classified. The structural safety simulation was a very complicated model with many layers that developed cracks and tears, which sometimes constituted a breach all the way from the outside to the inside of the model. After 180 man hours were spent manually marking only the tears in 12 runs, a faster approach was needed. A distributed classification approach similar to that proposed in this paper was used to train on the 12 marked runs and test the remaining 150 runs to find all tears and evaluate breaches in a cumulative 75 man hours.

The main contributions of this paper are to show that accurate regions of interest can be learned even when a dataset is distributed in disjoint partitions and that the regions can be ordered so that the actual interesting regions are presented first. Individual predictions may be somewhat noisy because data is distributed in ways that are not helpful to the learning algorithm, but they get smoothed into good regions. The true positive regions can reliably be presented first in a ranked list. This approach can be used on problems with minority classes where finding regions of interest is the goal and training noise is less than about 10%. For example, recognizing abnormal regions in medical images or finding supernovae in astronomy are potential applications.


(a)

(b)

Fig. 1: Visualizations of the casing and canister tear simulations. For the casing simulation, ground truth salient (bolt) regions are the smaller, darker regions. a Casing simulation, b canister tear simulation

2 Related Work

We discuss the most relevant research in the related areas of incremental learning, distributed learning, and ranking problems. Incremental learning [15, 30, 40, 24], where the model changes as training data becomes available over time, provides a potential approach for creating a model from a very large training dataset. The model could be built on one set of data and then moved to another processor for continued learning on a second set of data, etc. Incremental learning models that require the storage of previous training examples, such as instance-based learning approaches [30], and decision tree approaches [14] are time consuming for very large datasets. Also, we could find no work evaluating their performance on very large datasets. Alternatively, data mining of streaming data [2, 17] has been developed precisely for **endless** streams of data. The datasets considered in this paper could be treated as a stream, although they lack a natural ordering principle. Our empirical experiments show statistically different results depending on how the partitions are ordered.

There are distributed learning algorithms, such as distributed boosting [10], that could be applied to this problem. The authors of [28] evaluate several distributed boosting algorithms, one of which deals specifically with learning from homogeneous distributions of data scattered across different sites. They consider the problem from the standpoint of data privacy, where data examples may not be propagated to other computers. In this algorithm, they compute statistics on the data such as mean and covariance in order to calculate the Mahalanobis distance between sites. Sites containing similar distributions employ the authors' distributed boosting algorithm, while those without similarity use standard boosting.

In the distributed boosting algorithm, a boosted classifier was built in each partition and broadcast to the other partitions. Using this ensemble of classifiers, the weight of each example was updated. A global weight array stores the sum of the updated weights for each individual site, thus providing information on how difficult it is to learn at any one site, and weighting that partition accordingly for the next iteration. The authors showed that this algorithm was at least as accurate as standard boosting on the centralized data base. The only spatially disjoint sets used in [28] were two very small synthetic datasets with three equal size classes, two physical dimensions, and no time dimension. In contrast, our much larger datasets consist of physics simulations of real world events with unequal size classes, three physical dimensions, a time dimension, and different partition schemes that present unique data mining challenges.

Distributed learning models have been shown to be able to provide classification performance that is competitive with that obtained on all of the data [9]. There is some work that indicates it is possible to do effective distributed learning with cost sensitive data [18]. Further, any approach that builds independent classifiers or models and combines them could potentially be applied [35]. Of the work discussed here, only in [28] were spatially disjoint datasets used, with significant differences from our work as mentioned above. In addition, we have developed smoothing and thresholding methods to obtain regional predictions.

Many variants of ranking problems in machine learning exist. One approach is to learn an order of items based on a pairwise score function [11]. Bucket orders, i.e., total orders with ties are considered in [19]. The Spearman rank correlation metric is minimized by using a simple weighted voting procedure [22], and by active learning of label ranking functions [7]. The authors of [39] use an order consistency metric to measure how similar the predicted order is to the true order of recommendation on item graphs. Since we are concerned only with whether our predictions satisfy an overlap requirement, we use lift quality as developed for database marketing [29, 33].

3 Experimental Datasets and Procedures

In this section, the canister tear simulation and casing simulation datasets are described, including physical and spatial characteristics, as well as their respective train and test datasets.

3.1 Canister Tear Simulation

In the canister tear simulation, a canister is dropped on a strike plate as shown in Figure 1a. The canister appears at the top, over the strike plate. The canister is made of one material for the sides and of a second material for the top and bottom. Simulated welds join the top and bottom to the sides. The collision of the canister with the strike plate causes compression faults in the canister shell at the point of impact and rupture faults (tears) in the canister shell farthest from impact. In our experiments, depending on the particular run, we observed 11 to 31 time steps for the simulated event. The baseline run was designated run 1. In runs 2 and 3, variables associated with the two canister materials were given values different from the baseline values. In run 4, the shell height of the middle region of the canister shell was increased from 1 to 2 and the refined weld surface and thickness were each reduced from 2 to 1.

Physical and Spatial Characteristics. In the four different instances of the canister tear simulation provided to us by the Department of Energy, all in the EXODUS II format, nodes and finite elements of the simulation model are embedded in a mesh framework [34]. Nine physical variables are stored for each node within each of the time steps. They are the displacement on the X, Y, and Z axes; velocity on the X, Y, and Z axes; and acceleration on the X, Y, and Z axes. In addition, 17 variables are stored for each finite element of eight nodes. We converted to a purely nodal representation by averaging all values of the corresponding elemental variables that contain the node. We then used only nodal variables for learning. Table 1 shows the parameter settings for each run. Table 13 in the Appendix shows the different ranges taken on by the features available in each run.

Table 1: Physical and spatial characteristics for the canister tear simulation runs

Tear Run	1	2	3	4
# of nodes per time step	140,293	140,293	140,293	81,465
# of time steps	11	11	11	31
Total # of nodes	1,543,223	1,543,223	1,543,223	2,525,415
% of salient nodes in training time step	0.79	4.94	3.25	2.28
% of salient nodes in remaining time steps	2.17	3.73	4.50	2.38
% of salient nodes in all time steps	2.05	3.84	4.38	2.30

There were 37 nodal variables for each run, including 26 that changed

Train and Test Sets. To create labeled data for every time step, those pieces of the canister that have deformed so as to possibly indicate a tear in the canister wall were marked as salient by manual editing of the data via a custom plug-in to the open source visualization tool ParaView [21]. At the beginning of the simulation there are no salient nodes within the mesh. As time progresses and the canister deforms, more and more nodes were marked salient.

The marking of the salient nodes within the mesh can in principle be as precise as desired, but more precision requires greater effort in manual marking. In actual ASC work, the scientists use a tool that permits them to quickly mark coarsely shaped regions, or to laboriously mark detailed regions. Since they invariably choose the fast but coarse option, we have done the same, allowing noise in the class labels by marking areas rather than individual nodes in the simulation models. Smoothing of the output to create regions may reduce the noise in predictions created by imprecise labeling, as we shall see.

The data for the middle time step of each canister tear simulation run was divided spatially according to the computer to which it is assigned and used for training classifiers and/or ensembles. The partitioning divided the canister into 14 disjoint spatial partitions of roughly equal size, as shown in Figure 2. Table 2 shows the number of salient nodes in each canister tear partition of the training time step. The training data in eight of the 14 training partitions of both runs 1 and 3 have no salient examples. The training data in seven of the 14 training partitions of both runs 2 and 4 have no salient examples. In addition, two other partitions of run 1 each contain only two salient examples. The high number of one-class partitions was deliberately chosen to illustrate the advantages of scaled probabilistic majority voting. In reality, the partitioning would be arbitrary and not user selectable.

In each time step and in each partition, saliency was designated as described. Every node not designated salient received the label “unknown”, rather than “not salient”, to reflect the fact that, in general, the users will indicate only salient regions. An ensemble of classifiers was trained on each of the 14 partitions of the training time step. Testing was done either on all of the remaining time steps of the same run, or on all time steps of each different run. Figures 3 and 4 show a view of the training time step and the final time step of all four canister tear runs.

The classifiers predicted each test example based on the attributes associated with that example. The votes of each partition were combined using a scaled probabilistic combination of the votes (to be reviewed in Section 4). We obtained *region*-based results by smoothing and thresholding the point-based predictions. Smoothing occurs by averaging saliency values of nodes within a specified distance and subsequently binarizing the saliency using the Otsu automatic thresholding algorithm [32]. We focus on the accuracy of region detection, not node detection, because it is regions that are presented to the user, and assessed for their utility.

3.2 Casing Simulation

In this dataset, also used in [37] and [25], a casing was dropped on the ground as shown in Figure 1a. The casing is composed of four main sections: the nose cone, the body tube, the coupler, and the tail. The coupler connects the body tube and tail through a

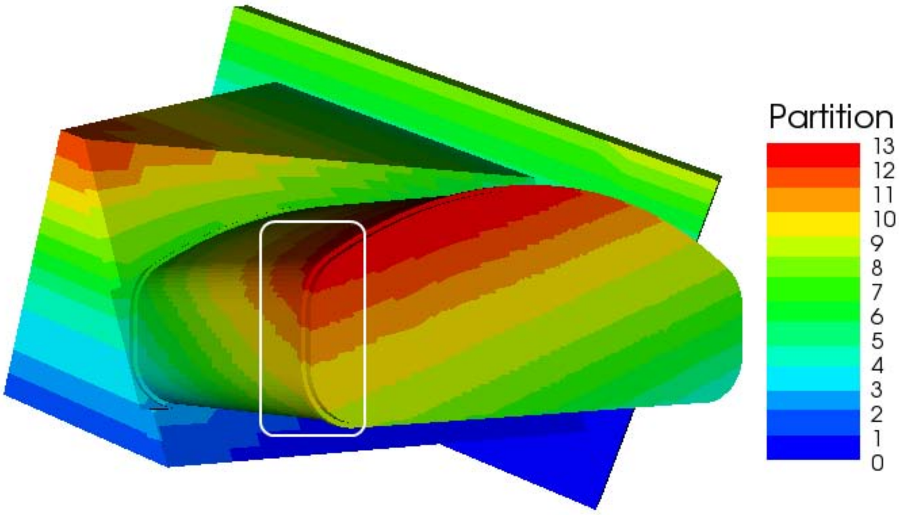


Fig. 2: A visualization of the 14 canister tear simulation partitions, with the tear area (seen in later time steps) inside the white outline

Table 2: Salient class statistics by partition for the canister tear simulation runs

Partition	# of training nodes		# of salient training nodes			
	Run 1, 2, or 3	Run 4	Run 1	Run 2	Run 3	Run 4
0	5041	5041	0	0	0	0
1	6800	6800	0	0	0	0
2	6271	6271	0	0	0	0
3	7471	7388	0	0	0	0
4	12,980	7441	0	0	0	0
5	10,672	5720	0	0	0	0
6	12,257	6642	0	0	0	0
7	10,759	5972	2	64	0	32
8	11,651	5823	166	1076	226	183
9	12,653	6560	471	1679	1415	337
10	10,938	6371	2	622	1095	389
11	8258	4406	0	1226	685	289
12	8693	3780	37	1192	537	368
13	15,849	3250	433	1071	599	262
all	140,293	81,465	1111	6930	4557	1860

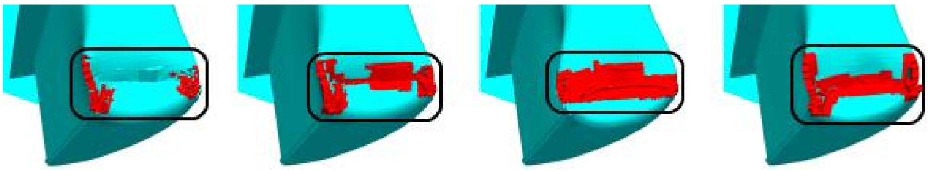


Fig. 3: Training time step in the canister tear simulation run 1, run 2, run 3, and run 4 (left to right). Ground truth salient regions are darker than unknown regions. The tear area view in Figure 2 has been rotated 90 °cw

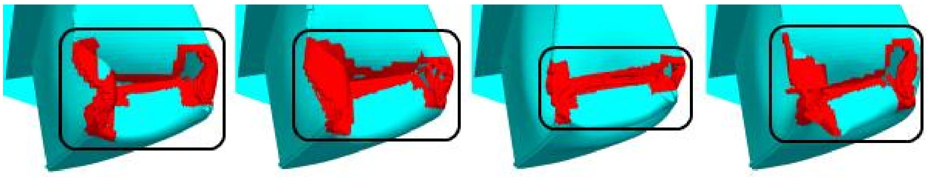


Fig. 4: Final time step in the canister tear simulation run 1, run 2, run 3, and run 4 (left to right). Ground truth salient regions are darker than unknown regions. The tear area view in Figure 2 has been rotated 90 °cw

series of ten bolts. The ground has also been modeled. The casing was dropped from a short height and landed on the tail at an angle. This simulation recorded the stresses across the entire device as might be found were it to be accidentally dropped during transport, storage, etc.

Physical and Spatial Characteristics. The goal using this dataset is to discover which nodes in the simulation belong to bolts. When dropped at an angle on the tail, one group of bolts experiences a tensile force, while the other group of bolts experiences a compressive force. Each was also subject to sheer forces. These forces were expressed in many other sections of the casing as well. The physical characteristics of the individual nodes modeling the bolts are not substantially different from those modeling the rest of the casing. In other words, there is no underlying feature of "boltness" which would make this an easy problem without using additional block node identification or location geometry. This additional information was only used for the initial labeling of bolt nodes as salient, and not as one of the features for improving test accuracy, as discussed later in Section 3.2.

The physical and spatial characteristics are provided in Table 3. Dataset attributes include the motion variables of displacement, velocity, and acceleration as well as several interaction variables such as contact force, total internal force, total external force, and reaction force. The different ranges for each of these attributes are shown in Table 14 in the Appendix. A time step showing the ground truth data is shown in Figure 1a. The bolts are the smaller, darker regions and represent the salient nodes in this simulation.

There are several important differences between this dataset and the canister tear dataset. There is not a large change in the structure of the casing data as the simulation runs through time. The change in the structure occurs mostly at the end of the simulation after some amount of shear has taken place. Since the structural changes are more subtle, the deformation of the casing simulation turns out to be more difficult than the canister simulation to accurately predict. Instead, for this dataset it is considered sufficient merely to identify the bolts.

Table 3: Physical and spatial characteristics for the casing simulation

# nodal variables	21	Total # non-bolt nodes	1,450,533
# time steps	21	Total # bolt nodes	119,280
# non-bolt nodes per time step	69,073	Total # nodes	1,569,813
# bolt nodes per time step	5680	Total % of bolt nodes	7.6%
Total # nodes per time step	74,753		

The properties of the partitioning for the casing dataset are shown in Table 4. Figure 5 shows the partitioning of the actual simulation. The partitioning was performed lengthwise in 12 pieces across the cylindrical body so as to distribute the bolts across computers. We purposefully partitioned the data so that four of the partitions do not contain any node from a bolt. This created four one-class classifiers, which were processed accordingly by the voting algorithm during classification. Two of the remaining partitions each contain a complete bolt and parts of two other bolts. The six remaining partitions each contain only a part of each of two bolts. The ground section was also partitioned and used for training in case its data became relevant in later time steps.

Train and Test Sets. During the simulation, nodes belonging to all of the bolts were specifically designated as their own substructure or block within the simulation. Therefore, labeling of those points was a matter of setting all those nodes as salient, and hence the training and test sets are labeled perfectly. This block node identification was deliberately not used as one of the features for improving test accuracy in order to establish a legitimate machine learning challenge for our methods. Recall that in the canister tear simulation the ground truth was subject to the inaccuracies inherent in the tools available to designate saliency.

Data from time steps zero to six were combined for each of the 12 partitions to form 12 sets of training data. The test set consisted of all of the data in the remaining time steps, 7 to 20. A classifier or an ensemble of classifiers was trained on the training data of each of the 12 partitions. Testing was performed using a scaled probabilistic combination of those 12 votes (to be reviewed in Section 4). The classifiers predicted each test example based on the attributes associated with that example. We obtained region-based results by smoothing and thresholding the point-based predictions.

Table 4: Partitioning characteristics for the casing simulation

Partition	# of training nodes	# of salient training nodes	% of salient training nodes
0	53,529	6818	12.74
1	59,654	5110	8.57
2	37,625	0	0.00
3	29,617	980	3.31
4	40,467	6972	17.23
5	29,183	0	0.00
6	43,106	0	0.00
7	29,155	3374	11.57
8	30,254	4578	15.13
9	54,488	0	0.00
10	49,728	3374	6.78
11	66,465	8554	12.87
all	523,271	39,760	7.60

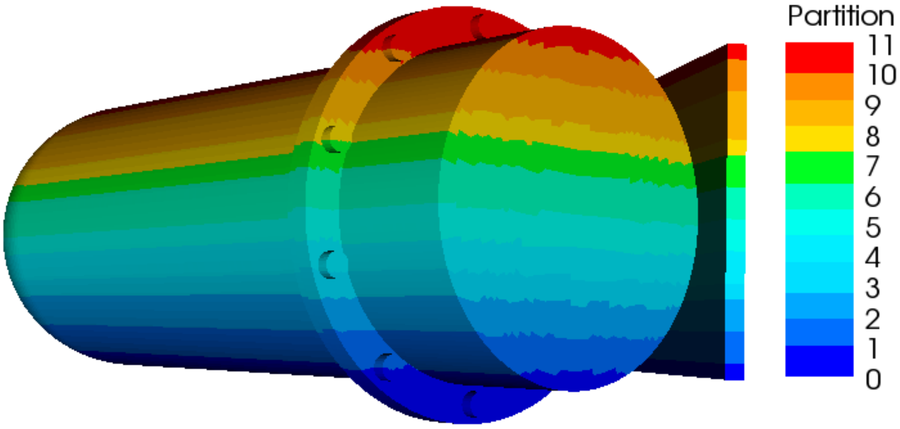


Fig. 5: A visualization of the 12 casing simulation partitions. Four of the eight bolts that are each contained in more than one partition are visible

4 Predicting and Ordering Salient Regions

Initially, a classifier or ensemble of classifiers was constructed using the labeled, spatially disjoint, training data local to each partition. Each of these classifiers or ensembles was then transferred to a test partition from either the same or similar simulations. Once there, each classifier or ensemble of classifiers was used to predict the class of each instance of test data local to that computer. Due to possible class imbalances, a scaled probabilistic majority vote of all class predictions was used to determine the consensus class of each instance of test data. Because regional predictions are the ultimate goal, connected-component regions of the predicted data were constructed, smoothed, and thresholded for better accuracy. For evaluation purposes, these predicted regions were compared to the labeled ground truth test regions, using different overlap thresholds to determine the quality of each result.

First, to establish a baseline for each partition we used a single default pruned C4.5 release 8, decision tree (DT) with a certainty factor of 25, trained on the data at that partition. Then we used Breiman's random forest (RF) algorithm [6], with 250 unpruned trees per partition with both unweighted (RF) and weighted (RFW) predictions. The accuracy of random forests was evaluated in [5] and shown to be comparable with or better than other well-known ensemble generation techniques. The number of random features chosen at each decision tree node was $log_2 n + 1$ given n features. The values for the same set of features for each individual node is presented to each DT or RF, but RF randomly selects only some of the features for use internally. RF predictions produce a single class vote for the forest, while RFW predictions are based on the percentage of trees that vote for a class. The motivation for using this ensemble technique stems from the inherent speed benefit of analyzing only a few possible attributes from which a test is selected at an internal tree node.

Classification of a test point within the simulation involves prediction by each partition's ensemble of decision trees. Because our algorithms need to work when only a few computers have salient examples, a simple majority vote algorithm may fail to classify any points as salient. In a large-scale simulation it is likely that there will be nodes which have no salient examples in training. If many individual classifiers are unable to predict a node as salient because there are no salient examples in the individual training sets, then it may be impossible for a majority vote to predict a node as salient. Therefore we must consider the prior probability that any given node contained salient examples during training and therefore is capable of producing a classifier that can predict an example as salient. A breakdown of this algorithm as presented in [4] is as follows:

$$\begin{aligned} p(w_1|x) &= \text{\% of ensembles voting for class } w_1 \text{ for example } x \\ P(w_1) &= \text{\% of ensembles capable of predicting class } w_1 \end{aligned}$$

$$\begin{aligned} \text{Classify as } w_1 \text{ if} : \frac{p(w_1|x)}{P(w_1)} &> \frac{p(w_2|x)}{P(w_2)} \\ \text{Classify as } w_2 \text{ if} : \frac{p(w_1|x)}{P(w_1)} &< \frac{p(w_2|x)}{P(w_2)} \end{aligned}$$

Thus, a probabilistic majority vote can be applied for a two-class problem. For instance, suppose there are five training partitions, including two partitions with both unknown (w_1 class) and salient (w_2 class) examples and three partitions with only unknown (w_1 class) examples. Therefore $p(w_1) = 5$ and $p(w_2) = 2$. If the first two partitions each vote salient for example x , and since the final three partitions can only vote unknown for example x, the overall vote would be salient since $\frac{3}{5} < \frac{2}{2}$.

This algorithm does not differentiate between ensembles trained on data with a very different number of examples by class. In order to further improve accuracy, we modified the input to the above algorithm by first multiplying each partition's ensemble vote for each class by the percentage of examples of that class in the corresponding partition, compared to the number of examples of that class in all partitions. After this additional step, the modified class votes were totaled, and the above algorithm applied. We call this implementation a scaled probabilistic majority vote (spmv).

An n-class problem's class votes would be similarly modified, and the algorithm below would then be applied [4]:

$$\text{Classify as } w_n : \ argmax_n(\frac{p(w_n|x)}{P(w_n)})$$

In the case of a tie vote, the unknown class was predicted, since a definite salient vote has not been determined. We are interested in directing people to salient regions so, presumably, missing a few salient points that are tied in a vote will not be important for region recognition.

Casing simulation ground truth salient regions (bolts) are constant in size, while salient regions in the canister tear simulations generally grow larger with each time step. Different methods were explored in an attempt to order true salient regions before false positive regions. Predicted regions were ordered by their size in number of nodes, with largest regions first. This method assumes that very small predicted regions are less likely to meet overlap threshold requirements for true positives. Another method ordered the predicted regions closest to the mean size of all predicted regions first. This technique assumes false positive regions are more likely to be very small or very large. Regions were also ordered by the mean of the salient margins of the scaled probabilistic majority votes by ensembles for nodes in each region before smoothing. In this case, salient margin is computed by salient votes minus unknown votes. Regions with higher means are ordered first since the ensemble voting shows more confidence in a salient classification. In addition, using domain knowledge, predicted regions for casing experiments were ordered by how closely their number of nodes compared to the number of nodes (568) in each ground truth bolt. The goal is to point the user to actual ground truth regions first, and false positive regions last, in those cases where perfect accuracy cannot be obtained.

5 Experimental Results

First, the basic experimental steps for both the casing and the canister tear experiments are described. Next, the metrics used to evaluate the results of predicting and ordering salient regions are explained. Finally, the results are presented and analyzed.

5.1 Experiments

For the casing experiments, training was performed on the data contained in each of the 12 partitions of time steps 0 to 6 to create both a single pruned decision tree and a 250-tree random forest ensemble for each partition. The decision tree classifier or the random forest ensemble of each training partition returned a single prediction (or a weighted prediction in the case of random forests weighted) for each test example in test time steps 7 to 20. The 12 predictions from those classifiers or ensembles were combined into a single prediction for each test example using the scaled probabilistic majority vote (see Section 4).

For each of the canister tear simulation experiments, training was performed on the data contained in each of the 14 partitions of the training time step of a single run. For each of runs 1, 2, and 3, this time step was 5 of 0 to 11, and for run 4, this time step was 15 of 0 to 31. Both a single pruned decision tree and a 250-tree random forest ensemble were created for each partition of the training data in a run. The decision tree classifier or the random forest ensemble of each partition returned a single prediction (or a weighted prediction in the case of random forests weighted) for each test example of either the remaining time steps of the same run, or of all time steps of a different run. The 14 predictions from those classifiers or ensembles were combined into a single prediction for each test example using the scaled probabilistic majority vote (see Section 4).

The salient regions of the data were marked using the region-based tools of the ParaView application [21]. The ensembles of classifiers used to classify the test data often produced smaller salient clusters of nodes or even individual isolated salient nodes, which do not correspond well to the larger marked, ground truth regions. In order to improve the regional accuracy of these ensembles, we employed some of the regional tools in the Feature Characterization Library (FCLib-1.2.0) toolkit [23] to process the ensemble prediction data. The numeric class label (0.5 for unknown, 1.0 for salient) of all nodes within a physical radius of three units of each node (found by testing on the training data) was averaged in a smoothing operation. We expect that smoothing at three units will erase smaller dimension regions without degrading larger regions.

After smoothing, nodes had numeric class labels in the range from [0.5,1]. These values were binarized using the Otsu automatic thresholding algorithm [32]. Predicted regions were created from connected components of salient nodes after smoothing. Smoothing tended to remove the smaller salient regions and the isolated salient nodes. All pairs of salient regions separated by no more than the maximum edge distance between nodes for the casing simulation, or by no more than an edge distance of two units between nodes for the canister tear simulation runs, were assigned the same region label. Another tool was used to generate overlap matrices of connected component ground truth and predicted regions. Predicted salient regions were finally ordered by the various size and voting confidence methods as described in the previous section.

5.2 Evaluation Metrics

Our previous approach [36] did not consider the actual node intersection percentage between ground truth and predicted salient regions. We extended that approach [38, 37] by establishing thresholds for the overlap percentage of the nodes in a ground truth salient region and a predicted salient region for the prediction to be counted as a true positive. The overlap required for a true positive at a given threshold was applied separately to both the ground truth region and to the predicted region. If no predicted salient regions sufficiently overlapped a ground truth salient region, a false negative was registered for the failure to adequately predict the ground truth region.

A false positive was recorded for each predicted region that did not sufficiently overlap any ground truth region. This may have resulted in more total predicted regions than actual regions. It is possible that more than one predicted salient region satisfied a given overlap threshold for intersection with a labeled salient region. We counted this as a single discovery of the ground truth region (true positive or TP), with the remaining prediction(s) counted as false positive(s). For the purposes of people searching for interesting events, this appears sensible because they would be directed to the region. If one predicted region sufficiently overlapped more than one labeled salient region, the only true positive counted was the one with the most overlap with the predicted region.

Recall, precision, and the traditional F-measure, which weights false positives (FP) and false negatives (FN) equally, provide measures of regional accuracy, as shown below [42].

$$\begin{aligned}\text{recall} &= \frac{TP}{TP+FN}\\ \text{precision} &= \frac{TP}{TP+FP}\\ \text{F-measure} &= \frac{2\cdot TP}{2\cdot TP+FP+FN}\end{aligned}$$

For many users, a small overlap threshold is an appropriate regional metric, since coarsely pointing those users to suspicious regions for further investigation is the main goal. From a machine learning viewpoint, a smaller overlap does not address the case where a very large region is always predicted salient. As long as this region minimally overlaps a given ground truth region, a true positive is counted. By increasing the overlap requirement to 10%, or even 50% for example, a more precise match is obtained. The stricter requirements also provide useful discrimination between classifier methods that would not be possible with a minimal overlap requirement.

While the F-measure indicates how well salient regions are found, it does not measure the quality of ordering predicted salient regions so that correct predictions are selected before false predictions. Lift is a measure often used in database marketing for this purpose and is defined as the percent of all targets (hits) in the first $p\%$ of the marketing list sorted by decreasing score of the model, divided by p. The authors of [29] and [33] previously addressed the measurement of lift quality in database marketing. The authors of [29] introduced a lift index that used a weighted sum of the items in the lift table. That index converged to the ratio of the area under the cumulative lift curve. The L-quality measure described in [33] is similar to the lift index, and ranges from -1 (worst case), to 0 (random case), to 1 (optimal case). We applied the same basic formula for calculating L-quality as shown below.

$$\text{L-quality}(M) = \frac{\text{SumCPH}(M) - \text{SumCPH}(R)}{\text{SumCPH}(B) - \text{SumCPH}(R)}$$

The term CPH denotes Cumulative Percent Hits, which is defined as lift multiplied by *p*% as explained above. The term SumCPH(*M*) is defined as the area under the CPH curve for the model *M*. The terms SumCPH(*R*) and SumCPH(*B*) are defined as the area under the CPH curve for the random model and for the optimal model respectively. The optimal case occurs when all targets are grouped at the beginning of the list. In our application, we sort a list of predicted regions instead of a list of potential customers. Instead of counting cumulative targets or hits, we count the number of cumulative ground truth salient regions that meet given overlap threshold requirements with a unique predicted region. While database applications involve numbers of potential customers large enough for practical expression as percentages on cumulative lift charts, the number of predicted regions is small enough to show on our charts as actual numbers.

Two cases that usually do not occur in database marketing applications would result in undefined L-qualities. First, the number of predicted regions may exactly equal the number of ground truth regions. In this case, all possible orders of the predicted regions have equal quality. Second, no ground truth region may be correctly predicted if the overlap threshold requirement is sufficiently high. Since all orderings are equivalent in these cases we cannot evaluate the L-quality. Hence, it will be undefined.

5.3 Results

Casing Simulation Regional Results Table 5 shows the regional results for the casing experiments evaluated with 10% and 50% overlap thresholds. These experiments used 12 partitions of training data, each from the first seven time steps. As discussed in Section 4, predicted regions were ordered by the ratio of region size to ground truth bolt size (GT ratio), by the ratio of region size to mean region size (size ratio), by region size from highest to lowest (size), and by the mean of the salient margins of the scaled probabilistic majority votes by ensembles for nodes in each region before smoothing (smm). Each method resulted in high L-qualities, which indicates that the ordering greatly improves the user experience compared to random ordering, by pointing to the most salient regions first, and causing most false positives to lie near the end of the list. Regions were also ordered naturally, by timestep (ts), from low to high, then by region number within each timestep from low to high. The corresponding natural L-qualities are lower, but still above zero, which a random ordering would produce. Figure 6 shows the cumulative lift curve for the model trained using 250 random forests unweighted trees for each of the 12 training partitions of data, using a 10% overlap threshold. Predicted regions were ordered by how closely their number of nodes compared to the number of nodes (568) in each ground truth bolt. For reference, the ideal, random, natural (ordered by timestep), and worst case lift curves are also shown.

Figure 7 shows the cumulative lift curves for a single decision tree (DT), and for 250 random forests unweighted (RF) and weighted (RFW) trees, for each of the 12 training partitions of data, using an overlap threshold of 10%. The vertical height of the rightmost point of each curve shows the total number of ground truth salient regions for which predicted regions meet the 10% overlap threshold requirement. Of the 140 actual ground truth regions, RFW correctly predicted 115, DT 123, and RF 124. The vertical distance of the rightmost point below the top of the chart indicates the number of false negative regions. The horizontal location of the rightmost point of each curve

Table 5: Casing regional results evaluated with 10% and 50% overlap thresholds. (Bold indicates the highest values)

Class	OT %	GT	Preds	TP	FN	FP	Rec.	Prec.	F-m	L-qualities				
										GT ratio	size	size ratio	smm	ts
DT	10	140	319	123	17	196	0.88	0.39	0.54	0.97	**0.98**	0.86	0.90	0.26
RF	10	140	251	**124**	**16**	**127**	**0.89**	**0.49**	**0.63**	**0.98**	0.97	0.93	0.87	0.33
RFW	10	140	257	115	25	142	0.82	0.45	0.58	**0.96**	0.95	0.94	0.94	0.37
						mean:	0.86	0.44	0.58	**0.97**	**0.97**	0.91	0.90	0.32
						sd:	0.04	0.05	0.05	0.01	0.02	0.04	0.04	0.06
DT	50	140	319	104	36	215	0.74	0.33	0.45	**1.00**	0.81	0.91	0.89	0.37
RF	50	140	251	**109**	**31**	**142**	**0.78**	**0.43**	**0.56**	**0.99**	0.79	0.95	0.89	0.43
RFW	50	140	257	100	40	157	0.71	0.39	0.50	**0.99**	0.76	0.97	0.91	0.48
						mean:	0.74	0.38	0.50	**0.99**	0.79	0.94	0.90	0.43
						sd:	0.04	0.05	0.06	0.01	0.03	0.03	0.01	0.06

Class: classifier; OT: overlap threshold; GT: ground truth regions; Preds: predicted regions; TP: true positives; FN: false negatives; FP: false positives; Rec.: Recall; Prec.: Precision; F-m: F-measure; smm: salient margin mean; ts: timestep

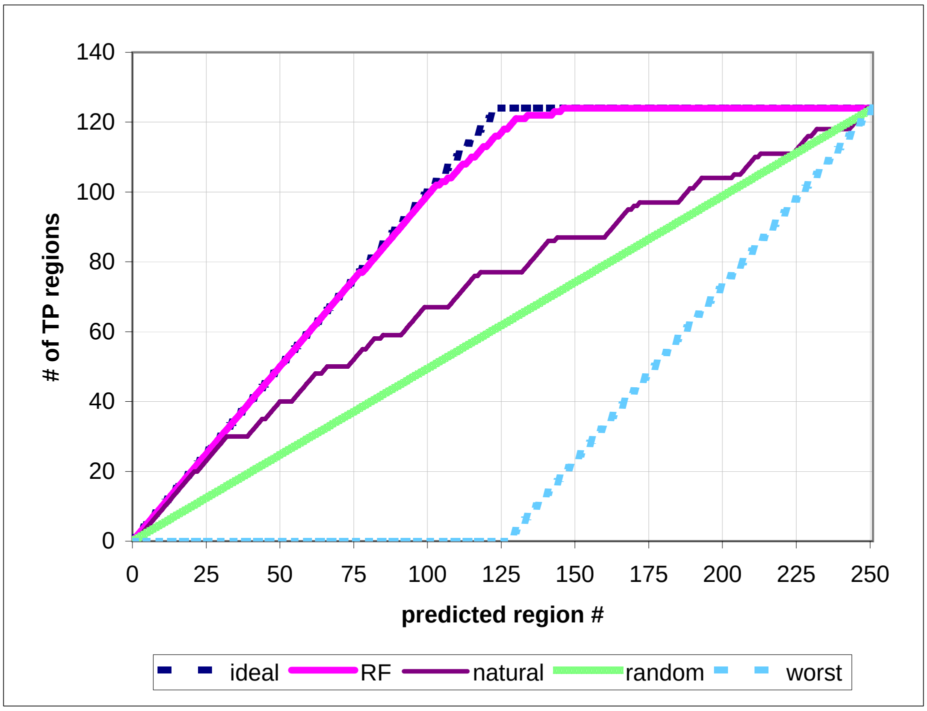


Fig. 6: A visualization of the casing cumulative lift curve for the model trained using random forests unweighted ensembles and evaluated with 10% overlap threshold. The ideal, natural, random, and worst case lift curves are also shown

shows the total number of predicted regions, including true and false positives. RF has 251, RFW has 257, and DT has 319 predicted regions. The high L-qualities for all three classifier/ensemble methods indicates false positives are mostly added at the end, after the user has seen almost all correctly identified salient regions.

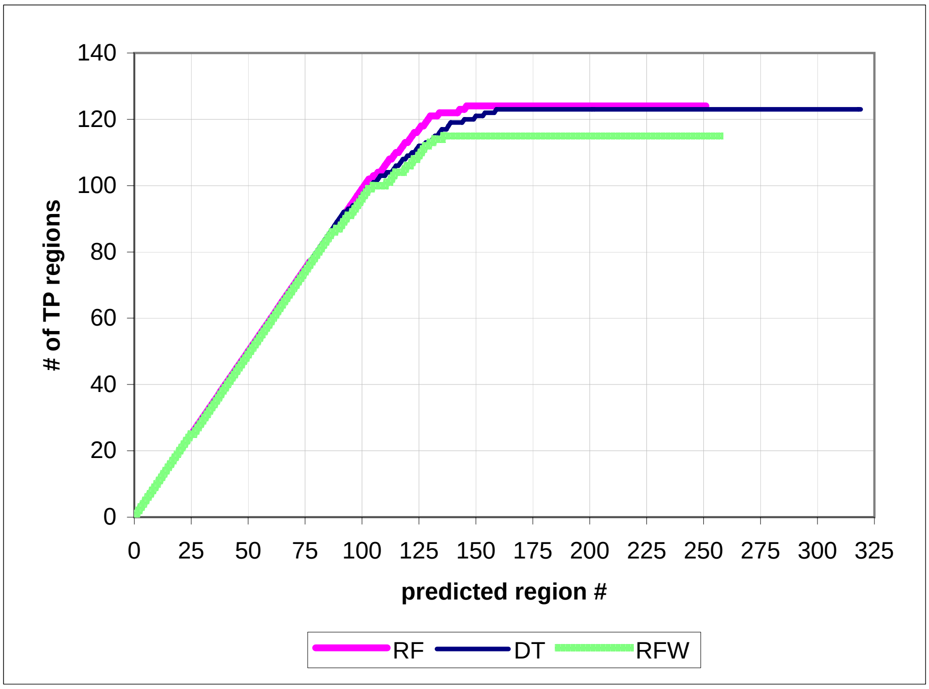


Fig. 7: A visualization of the casing cumulative lift curves for the models trained using single decision tree, and random forests unweighted and weighted ensembles, evaluated with 10% overlap thresholds

Canister Tear Simulation Regional Results Tables 6, 7, and 8 show the canister tear regional results with 10% overlap thresholds using DT, RF, and RFW respectively. The training data from each run was from 14 partitions of a single time step. Most of the lower F-measures involve run 1 (baseline) as either the test run or the training run time step. Run 1 has the fewest salient nodes of any run, and is the only run that has more than one salient region in a single time step (seven time steps each have two smaller salient regions, including the training time step). Only 3 of the 14 partitions of run 1 have at least 50 salient nodes. As discussed in Section 4, predicted regions were ordered by region size from highest to lowest, by the ratio of region size to mean region size, and by the mean of the salient margins of the scaled probabilistic majority votes by ensembles for nodes in each region before smoothing. Regions were also ordered naturally, by timestep (ts), from low to high, then by region number within each timestep from low to high. While some of each table's entries show an L-quality of 1.00, the most notable of these is the 89 false positive regions in Table 6, all after the 29 true positives have been presented. A high L-quality is more significant when there are more false positives along with many true positives.

Table 6: Canister tear decision tree regional results evaluated with 10% overlap threshold

Train	Test	GT	Preds	TP	FN	FP	Rec.	Prec.	F-m	L-qualities			
run	run									size	size	smm	ts
											ratio		
1	1	15	53	8	7	45	0.53	0.15	0.24	0.96	0.57	0.86	-0.01
2	1	17	30	4	13	26	0.24	0.13	0.17	0.71	0.71	0.67	-0.23
3	1	17	36	4	13	32	0.24	0.11	0.15	0.92	0.53	0.53	-0.86
4	1	17	17	5	12	12	0.29	0.29	0.29	0.83	0.17	0.53	-0.27
1	2	10	66	10	0	56	1.00	0.15	0.26	0.94	0.56	0.88	0.06
2	2	9	19	9	0	10	1.00	0.47	0.64	1.00	1.00	0.96	0.51
3	2	10	61	9	1	52	0.90	0.15	0.25	0.97	0.74	0.94	-0.65
4	2	10	18	10	0	8	1.00	0.56	0.71	1.00	0.93	1.00	0.45
1	3	10	75	10	0	65	1.00	0.13	0.24	0.87	0.74	0.86	0.22
2	3	10	20	10	0	10	1.00	0.50	0.67	1.00	0.60	0.80	0.60
3	3	9	45	9	0	36	1.00	0.20	0.33	1.00	0.83	0.96	-0.13
4	3	10	24	10	0	14	1.00	0.42	0.59	1.00	1.00	1.00	0.41
1	4	29	194	25	4	169	0.86	0.13	0.22	0.91	0.20	0.87	0.00
2	4	29	67	29	0	38	1.00	0.43	0.60	1.00	0.98	0.91	0.34
3	4	29	117	28	1	89	0.97	0.24	0.38	1.00	0.81	0.95	-0.39
4	4	28	57	28	0	29	1.00	0.49	0.66	1.00	1.00	0.98	0.39
						mean:	0.81	0.29	0.40	0.94	0.71	0.86	0.03
						sd:	0.30	0.16	0.20	0.08	0.26	0.15	0.43

GT: ground truth regions; Preds: predicted regions; TP: true positives; FN: false negatives; FP: false positives; Rec.: Recall; Prec.: Precision; F-m: F-measure; smm: salient margin mean; ts: timestep

An illustration of the canister tear run 1 cumulative lift curves for the model trained using 250 random forests weighted trees for each of the 14 training partitions of data appears in Figure 8. Predicted regions were ordered by their size (number of nodes) from large to small. Similarly, the cumulative lift curves for the canister tear runs 2, 3, and 4 are shown in Figures 9, 10, and 11. As discussed above, most of the lower F-measures involve run 1 (baseline), and can be distinguished by a lower and/or farther right final point on the lift curve. Those lift curves with higher L-qualities have more diagonally upward steps (true positives) for the initial predictions and more horizontal steps (false positives) for the final predictions.

While precision, recall, and the F-measure are computed on unordered sets of predicted regions, L-quality is computed on ordered or ranked sets of predicted regions. Another ranking quality method is the precision-recall curve, which shows the precision at increasing recall levels. The standard curve is usually smoothed to remove sawtooth patterns by using interpolated precision, which is the highest precision found for any recall level greater than or equal to the given recall level. Eleven-point interpolated

Table 7: Canister tear random forests unweighted regional results evaluated with 10% overlap threshold

Train	Test	GT	Preds	TP	FN	FP	Rec.	Prec.	F-m	L-qualities			
run	run									size	size ratio	smm	ts
1	1	15	18	14	1	4	0.93	0.78	0.85	0.14	0.93	0.96	-0.04
2	1	17	12	7	10	5	0.41	0.58	0.48	0.20	-0.20	-0.14	-0.20
3	1	17	17	5	12	12	0.29	0.29	0.29	0.63	0.40	0.37	0.20
4	1	17	10	7	10	3	0.41	0.70	0.52	0.14	-1.00	-0.52	-0.14
1	2	10	21	10	0	11	1.00	0.48	0.65	0.36	0.47	0.53	0.38
2	2	9	13	9	0	4	1.00	0.69	0.82	1.00	1.00	1.00	0.61
3	2	10	12	10	0	2	1.00	0.83	0.91	1.00	1.00	1.00	0.60
4	2	10	12	10	0	2	1.00	0.83	0.91	1.00	1.00	1.00	-0.10
1	3	10	20	7	3	13	0.70	0.35	0.47	0.58	0.69	0.56	0.52
2	3	10	13	10	0	3	1.00	0.77	0.87	1.00	1.00	1.00	0.60
3	3	9	9	9	0	0	1.00	1.00	1.00	ND	ND	ND	ND
4	3	10	16	10	0	6	1.00	0.62	0.77	1.00	1.00	1.00	0.40
1	4	29	67	26	3	41	0.90	0.39	0.54	0.41	0.25	0.24	0.22
2	4	29	60	29	0	31	1.00	0.48	0.65	0.68	0.54	0.80	0.39
3	4	29	51	29	0	22	1.00	0.57	0.73	1.00	1.00	0.71	0.44
4	4	28	33	28	0	5	1.00	0.85	0.92	1.00	1.00	1.00	0.01
						mean:	0.85	0.64	0.71	0.68	0.61	0.63	0.26
						sd:	0.25	0.20	0.20	0.35	0.57	0.47	0.29

GT: ground truth regions; Preds: predicted regions; TP: true positives; FN: false negatives; FP: false positives; Rec.: Recall; Prec.: Precision; F-m: F-measure; smm: salient margin mean; ts: timestep; ND: not defined

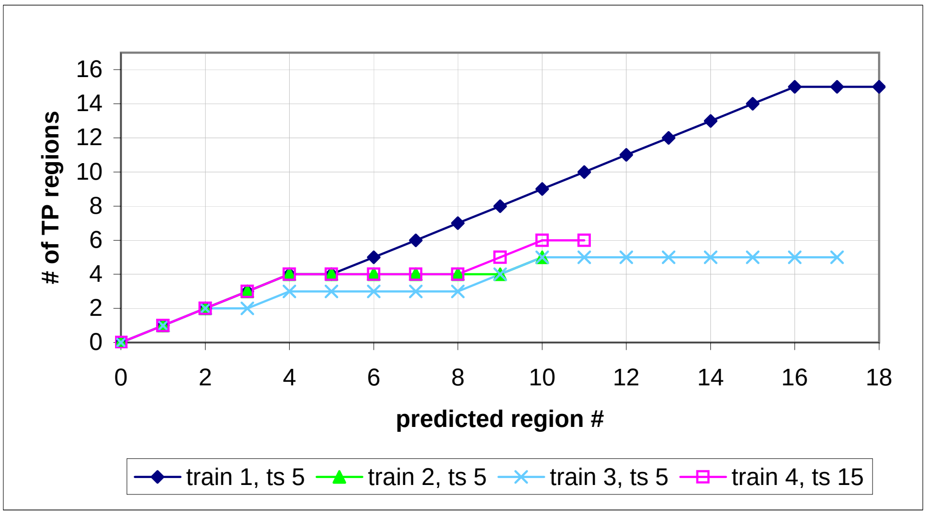


Fig. 8: A visualization of the canister tear cumulative lift curves for the models trained using random forests weighted ensembles and evaluated with 10% overlap threshold on canister tear run 1. In each case the time step (ts) of the training run (train) is specified

Table 8: Canister tear random forests weighted regional results evaluated with 10% overlap threshold

Train run	Test run	GT	Preds	TP	FN	FP	Rec.	Prec.	F-m	L-qualities size	size ratio	smm	ts
1	1	15	18	15	0	3	1.00	0.83	0.91	0.51	0.96	0.91	0.64
2	1	17	10	5	12	5	0.29	0.50	0.37	0.60	-0.60	-0.92	-0.60
3	1	17	17	5	12	12	0.29	0.29	0.29	0.63	0.40	0.30	0.17
4	1	17	11	6	11	5	0.35	0.55	0.43	0.47	-0.60	-0.33	-0.27
1	2	10	18	10	0	8	1.00	0.56	0.71	0.18	0.40	0.18	0.30
2	2	9	10	9	0	1	1.00	0.90	0.95	1.00	1.00	1.00	0.33
3	2	10	11	10	0	1	1.00	0.91	0.95	1.00	1.00	1.00	0.40
4	2	10	11	10	0	1	1.00	0.91	0.95	1.00	1.00	1.00	0.20
1	3	10	19	9	1	10	0.90	0.47	0.62	0.40	-0.09	0.53	0.29
2	3	10	15	10	0	5	1.00	0.67	0.80	1.00	1.00	1.00	0.56
3	3	9	9	9	0	0	1.00	1.00	1.00	ND	ND	ND	ND
4	3	10	16	10	0	6	1.00	0.62	0.77	1.00	1.00	1.00	0.30
1	4	29	61	24	5	37	0.83	0.39	0.53	0.26	0.23	0.26	0.11
2	4	29	31	29	0	2	1.00	0.94	0.97	1.00	1.00	0.45	-0.17
3	4	29	47	29	0	18	1.00	0.62	0.76	1.00	1.00	1.00	0.54
4	4	28	30	28	0	2	1.00	0.93	0.97	1.00	1.00	0.89	-0.11
						mean:	0.85	0.69	0.75	0.74	0.58	0.55	0.18
						sd:	0.27	0.22	0.23	0.31	0.60	0.58	0.34

GT: ground truth regions; Preds: predicted regions; TP: true positives; FN: false negatives; FP: false positives
Rec.: Recall; Prec.: Precision; F-m: F-measure; smm: salient margin mean; ts: timestep; ND: not defined

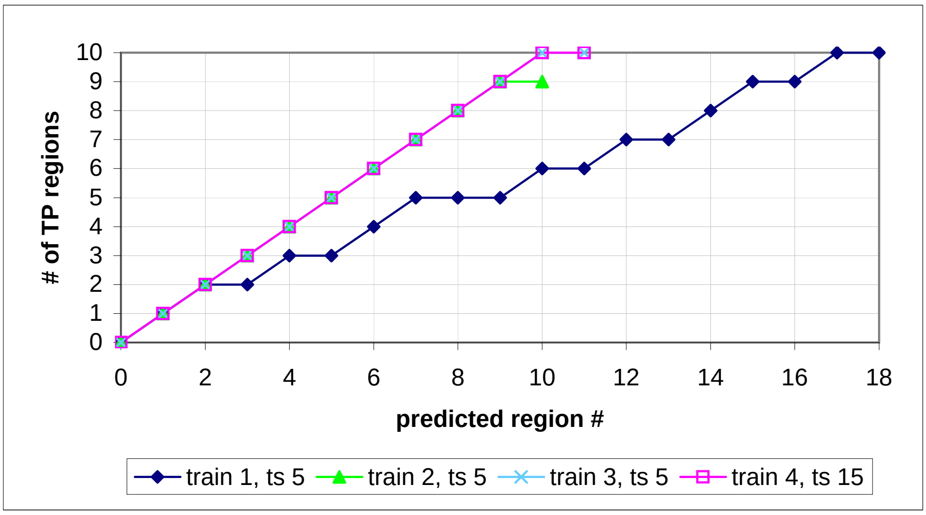


Fig. 9: A visualization of the canister tear cumulative lift curves for the models trained using random forests weighted ensembles and evaluated with 10% overlap threshold on canister tear run 2. In each case the time step (ts) of the training run (train) is specified

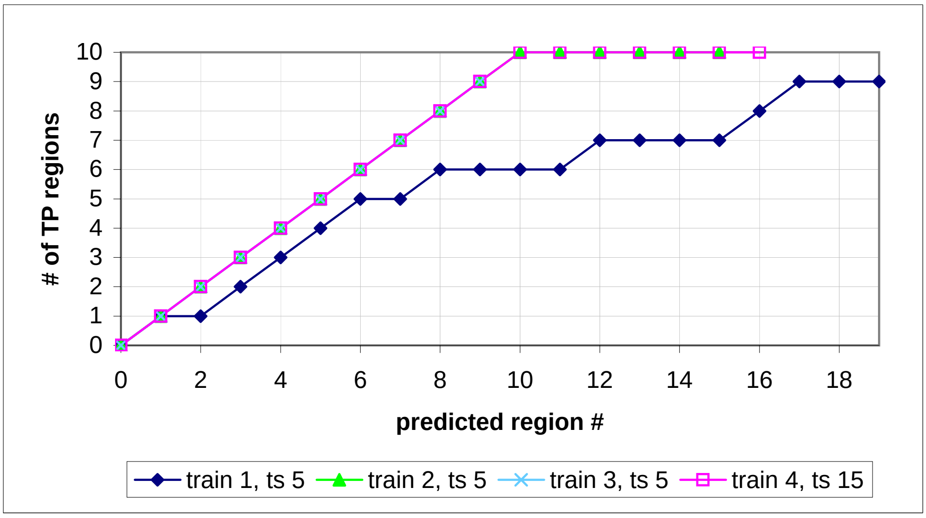


Fig. 10: A visualization of the canister tear cumulative lift curves for the models trained using random forests weighted ensembles and evaluated with 10% overlap threshold on canister tear run 3. In each case the time step (ts) of the training run (train) is specified

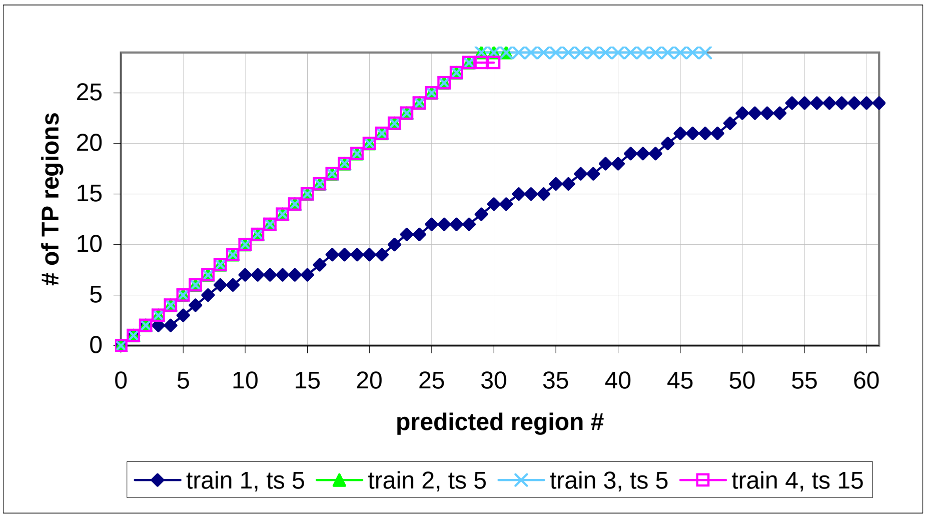


Fig. 11: A visualization of the canister tear cumulative lift curves for the models trained using random forests weighted ensembles and evaluated with 10% overlap threshold on canister tear run 4. In each case the time step (ts) of the training run (train) is specified

precision-recall graphs average the interpolated precision at eleven fixed recall levels from zero to one [3, 31]. Figure 12 shows an eleven-point interpolated precision-recall graph averaged across 16 canister tear train-test combinations using RFW ensembles. Curves for five overlap thresholds (10% to 50%) between predicted and ground truth regions are graphed. As expected, as more salient regions are retrieved from the ranked list (the recall increases), some more false positive regions are also retrieved (the precision decreases). In general, the curves meeting lower overlap threshold requirements show the best performance and are closest to the upper-right corner of the graph, where recall and precision are maximized.

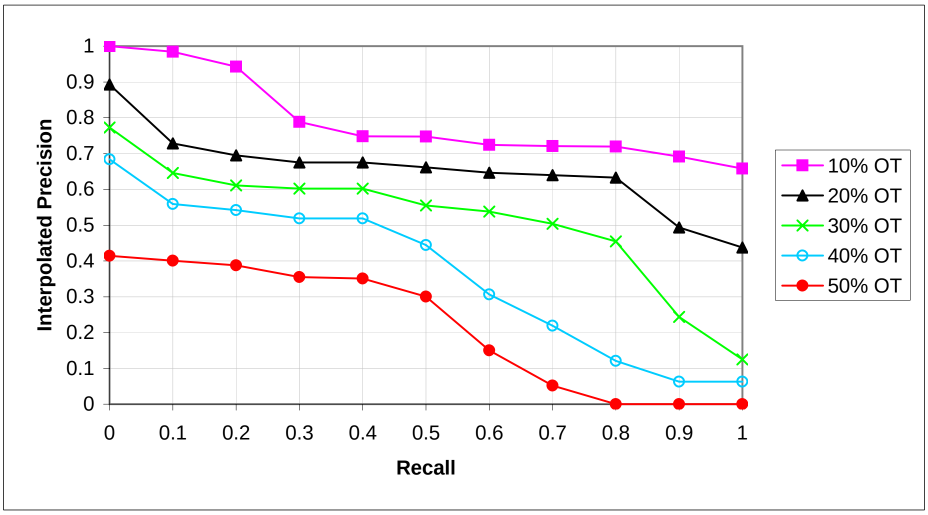


Fig. 12: Eleven-point interpolated precision-recall graph averaged across 16 canister tear train/test combinations using random forests weighted ensembles. Curves are shown for five overlap thesholds (OT) from 10% to 50%

Labeling Noise Results Labeling noise experiments were performed on the casing simulation, since it has perfect ground truth labeling for all bolts (salient regions). The ground truth labels were changed in the training data only, so that 1%, 5%, 10%, 15%, and 20% of the 568 nodes in each bolt were mislabeled as unknown instead of salient. Each bolt has 11 layers of nodes, and noise was thus added to the exterior surface nodes of layer(s) beginning farthest from the bolt head as required. In separate experiments, the labels of the same number of nodes closest to each bolt were changed from unknown to salient at the above five noise levels. In other words, the first set of five noise levels decreased the number of bolt nodes that were labeled correctly, and the second set increased the number of nodes adjacent to each bolt that were labeled incorrectly as bolt nodes.

Table 9: Casing simulation RFW bolt labeling noise results evaluated with 10% overlap threshold.

Noise		GT	Preds	TP	FN	FP	Rec	Prec	F-m	L-qualities			
level	type									GT ratio	size	size ratio	ts
0%	NA	140	257	115	25	142	0.82	0.45	0.58	0.96	0.95	0.94	0.37
1%	b	140	250	118	22	132	0.84	0.47	0.61	0.97	0.95	0.93	0.33
5%	b	140	226	120	20	106	0.86	0.53	0.66	0.92	0.91	0.89	0.37
10%	b	140	195	90	50	105	0.64	0.46	0.54	0.91	0.87	0.87	0.32
15%	b	140	174	60	80	114	0.43	0.34	0.38	0.93	0.93	0.80	0.26
20%	b	140	106	25	115	81	0.18	0.24	0.20	0.86	0.86	0.59	0.19
1%	nb	140	253	104	36	149	0.74	0.41	0.53	0.93	0.94	0.92	0.38
5%	nb	140	230	98	42	132	0.70	0.43	0.53	0.94	0.88	0.95	0.38
10%	nb	140	192	86	54	106	0.61	0.45	0.52	0.95	0.79	0.94	0.38
15%	nb	140	192	78	62	114	0.56	0.41	0.47	0.95	0.85	0.96	0.42
20%	nb	140	160	71	69	89	0.51	0.44	0.47	0.93	0.76	0.97	0.39

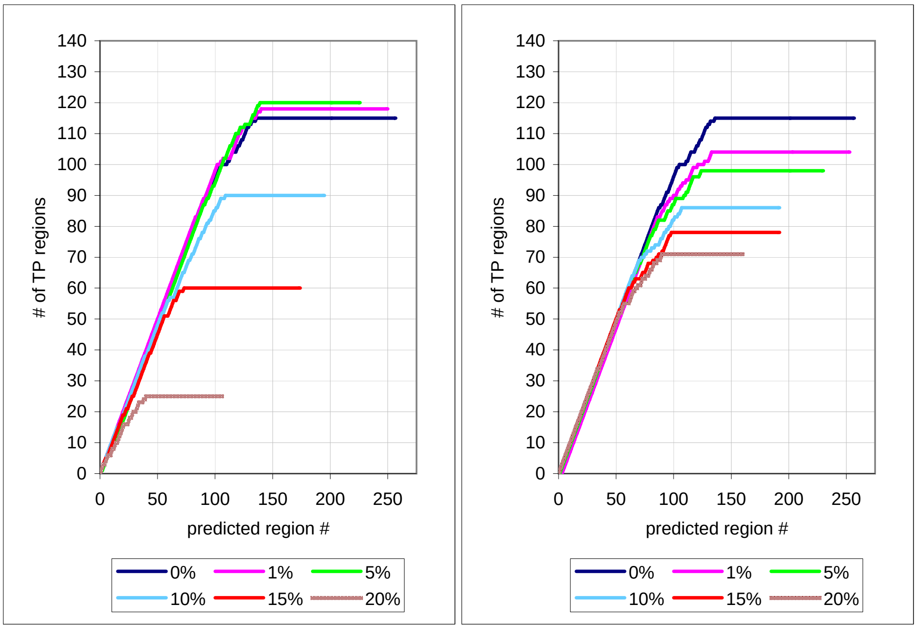
NA: not applicable; b: bolt; nb: non-bolt; GT: ground truth regions; Preds: predicted regions

TP: true positives; FN: false negatives; FP: false positives;

Rec.: Recall; Prec.: Precision; F-m: F-measure; ts: timestep

Random forests unweighted (RF), weighted (RFW), and a single decision tree (DT) were separately trained on each of the 12 partitions of the training data. For reference, a single decision tree was also trained on all of the combined training data (SDT), although this method would not be feasible for much larger datasets. An overlap threshold of 10% was used. The results of the labeling noise experiments for RFW are shown in Table 9 and Figure 13. The results for RF, DT, and SDT are shown in the Appendix—Tables 15, 16, and 17, and Figures 14, 15, and 16. All figures show cumulative lift curves that were produced with the ground truth (GT) ratio method of ordering predicted regions, as discussed in Section 4. For all noise level cases, the L-qualities remained consistently high. Methods that produce high L-qualities compensate for low precision and make higher recall the key measurement, since most true positive regions (bolts) are detected before false positive regions. While the recall for each method steadily decreased as the non-bolt noise level increased, there were exceptions as the bolt noise level increased. Both SDT and RFW showed a small gain in recall for some increase(s) in bolt noise levels. DT showed a larger decrease in recall (though still high overall) at bolt noise levels of 1% and 10% than at the other bolt noise levels. This was likely due to a combination of decision tree instability and partitioning of the data.

The decision tree methods each proved more robust to noise inside the bolts than the random forests methods for 15% and 20% noise levels. Conversely, both decision tree methods were less robust to 20% noise added to nodes outside the bolts. For the recall at the bolt and non-bolt 10% noise levels, SDT averaged a recall of 0.89, followed by RFW at 0.63, RF at 0.62, and DT at 0.55. Excluding SDT, which is not viable for larger datasets, RFW performed best overall for both types of noise. SDT and DT predicted more salient regions (true and false positives combined) than RF and RFW, and the number of predicted regions for all methods tended to decrease at the higher noise levels. In general, our methods are best used for noise levels less than about 10%.



(a) Bolt noise. Top to bottom: 5, 1, 0, 10, 15, 20

(b) Non-bolt noise. Top to bottom: 0 to 20

Fig. 13: Casing simulation bolt and non-bolt noise cumulative lift curves using RFW

Voting Method Results A simple majority vote (mv) and a probabilistic majority vote (pmv) without scaling each produce inferior results when compared to a scaled probabilistic majority vote (spmv) for the experiments described in this paper. If only partitions that have examples of both classes (unknown and salient) are used for training, the probabilistic component of spmv is the same for both classes and has no effect on each classifier or ensemble vote. However, both mv and a scaled majority vote (smv) using only two-class partitions could be used to examine alternative voting methods. Table 10 shows the average results of such tests on the 16 canister tear train-test combinations. The smv method almost always yields higher recall, precision, and F-measure than the mv method. The final three lines of Table 11 show the corresponding spmv results, which are better than the mv and smv results for recall, precision, and F-measure in Table 10 for RF and RFW. The difference lies in the adjustment to the probabilistic component of the scaled probabilistic majority vote, which assigns higher relative weights to salient votes in these cases. Of course this adjustment could be made without building one-class partition classifiers, since they always predict the class as unknown.

Statistical Significance Results Ensembles often result in a higher accuracy classifier than a single classifier. Many times an ensemble is trained on a subset of the data (e.g. bagging). The data may be implicitly weighted as in bagging, features left out to create random subspaces, etc. Other work has shown that you can get better accuracy from an ensemble of classifiers built on subsets of the data [10, 16]. The disjoint subsets of data

Table 10: Canister tear average regional results using only two-class partitions and evaluated with 10% overlap threshold. (Bold indicates the highest values)

Classifier	Voting method	Recall	Precision	F-measure	L-qualities		
					size	size ratio	timestep
DT	mv	0.61	0.31	0.40	**0.85**	0.42	0.32
DT	smv	0.80	0.34	0.47	**0.84**	0.71	0.18
RF	mv	0.60	0.47	0.43	**0.76**	0.58	0.51
RF	smv	**0.84**	**0.50**	**0.61**	0.65	**0.71**	0.40
RFW	mv	0.63	0.47	0.50	**0.70**	0.50	0.48
RFW	smv	0.73	0.45	0.54	**0.66**	**0.66**	0.50

mv: majority vote; smv: scaled majority vote

here will result in classifiers that make different errors, which can (and often does) lead to better accuracy as has been seen with smaller datasets previously.

The average canister and casing regional results for a 10% overlap threshold are shown in Tables 11 and 12. For a baseline comparison, each table includes results for a single decision tree (SDT) built on the unpartitioned training data, which includes all available labeled training data. To illustrate, instead of training one DT for the data in each of the 14 tear training partitions, only one decision tree was trained on all of the combined training data of the 14 partitions. For the combined casing experiments and canister tear experiments we applied the Friedman test, an average algorithm rank method, and the Holm step-down procedure, both described in [12], to show that the precision and F-measure from DT are significantly worse than either RF or RFW with a 99% confidence level. Each of the four canister tear simulation runs was considered as a separate dataset, since each has a unique % of salient nodes in the training time step and in all time steps, and either unique material properties, or a different number of time steps or nodes per time step. The casing experiments and all 16 tear train-test combinations were used in the evaluations. We also found that the natural time step ordering is significantly worse than all of the predicted region ordering methods with a 99% confidence level. For the combined casing and canister tear simulations, the precision and F-measure from DT using unpartitioned training data are significantly worse than either RF or RFW with a 99% confidence level. This demonstrates that partitioning obstacles to data mining can be more than overcome with the diversity of random forests.

6 Summary and Discussion

Large simulations (terabyte to petabyte scale) must be partitioned across multiple processors in order to obtain results in a reasonable amount of time. The method of breaking data into pieces may cause highly skewed class distributions, as it violates the usual assumption of independent and identically distributed datasets. In this paper, we showed how such data may nonetheless be effectively used for data mining. We showed that

Table 11: Canister tear average regional results evaluated with 10% overlap threshold. (Bold indicates the highest values)

Classifier	Training data	Recall	Precision	F-measure	L-qualities		
	partitioned?				size	size ratio	timestep
SDT	no	0.77	0.24	0.35	**0.87**	0.72	0.05
DT	yes	0.81	0.29	0.40	**0.94**	0.71	0.03
RF	yes	**0.85**	0.64	0.71	**0.68**	0.61	0.26
RFW	yes	**0.85**	**0.69**	**0.75**	**0.74**	0.58	0.18

Table 12: Casing regional results evaluated with 10% overlap threshold. (Bold indicates the highest values)

Classifier	Training data	Recall	Precision	F-measure	L-qualities			
	partitioned?				GT	size	size	timestep
					ratio		ratio	
SDT	no	**0.95**	0.41	0.57	0.97	**1.00**	0.84	0.28
DT	yes	0.88	0.39	0.54	0.97	**0.98**	0.86	0.26
RF	yes	0.89	**0.49**	**0.63**	**0.98**	0.97	0.93	0.33
RFW	yes	0.82	0.45	0.58	**0.96**	0.95	0.94	0.37

GT: ground truth regions

results from the distributed training data are as good or better than one can obtain with a single decision tree trained on all the labeled training data. Our approach uses fast ensemble learning algorithms, scaled probabilistic majority voting, and ordering of predicted regions of saliency.

The results show that a simulation experiment that yields only somewhat above average regional F-measures can provide efficient visual analysis of those results by effective ordering of the predicted regions. The vast majority of false positives were ordered last, after the user has already seen most of the true positive salient regions. The canister tear results often showed higher F-measures than the casing results in spite of the relatively fewer examples used for training ensembles. Again, the quality of ordering predicted regions is typically reflected in high L-quality measures.

The results indicate that simulation developers and users would be accurately directed to regions of interest with only occasional misdirection. This has the potential for saving significant time during debugging and use by allowing for a much improved focus of attention on areas of interest without highly time-consuming search.

Acknowledgements. This research was partially supported by the Department of Energy through the ASCI CSEE Data Discovery Program, Contract number: DE-AC04-76DO00789 and the National Science Foundation under grant EIA-0130768.

References

1. ASC, National Nuclear Security Administration in collaboration with Sandia, Lawrence Livermore, and Los Alamos National Laboratories, http://www.sandia.gov/nnsa/asc/ (accessed on 29 Nov 2008)
2. Aggarwal, C.C., Han, J., Wang, J., Yu, P.S.: On demand classification of data streams. In: KDD '04: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 503–508. ACM Press, New York, NY, USA (2004)
3. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. ACM Press (1999)
4. Banfield, R.E., Hall, L.O., Bowyer, K.W., Kegelmeyer, W.P.: Ensembles of classifiers from spatially disjoint data. In: Multiple Classifier Systems, Sixth International Workshop. Lecture Notes in Computer Science, vol. 3541, pp. 196–205. Springer, Seaside, CA, USA (2005)
5. Banfield, R.E., Hall, L.O., Bowyer, K.W., Kegelmeyer, W.P.: A comparison of decision tree ensemble creation techniques. IEEE Transactions on Pattern Analysis and Machine Intelligence. 29(1):173–180 (2007)
6. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
7. Brinker, K.: Active learning of label ranking functions. In: Proceedings of the 21th International Conference on Machine Learning. pp. 129–136 (2004)
8. Chawla, N.V., Hall, L.O., Bowyer, K.W., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16, 321–357 (2002)
9. Chawla, N.V., Moore, T.E., Hall, L.O., Bowyer, K.W., Kegelmeyer, W.P., Springer, C.: Distributed learning with bagging-like performance. Pattern Recognition Letters 24(1-3), 455–471 (2003)
10. Chawla, N.V., Hall, L.O., Bowyer, K.W., Kegelmeyer, W.P.: Learning ensembles from bites: A scalable and accurate approach. J. Mach. Learn. Res. 5, 421–451 (2004)
11. Cohen, W.W., Schapire, R.E., Singer, Y.: Learning to order things. J ARTIF INTELL RES 10, 243–270 (1999)

12. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)
13. Domingos, P.: Metacost: a general method for making classifiers cost-sensitive. In: KDD '99: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 155–164. ACM Press, New York, NY, USA (1999)
14. Domingos, P., Hulten, G.: Mining high-speed data streams. In: KDD '00: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 71–80. ACM Press, New York, NY, USA (2000)
15. Erdem, Z., Polikar, R., Gurgen, F., Yumusak, N.: Ensemble of SVMs for incremental learning. In: Multiple Classifier Systems, 6th International Workshop. Lecture Notes in Computer Science, vol. 3541, pp. 246–256. Springer, Seaside, CA, USA (2005)
16. Eschrich, S., Hall, L.O.: Learning from soft partitions of data: reducing the variance. In: Fuzzy Systems, 2003. FUZZ '03. The 12th IEEE International Conference on. vol. 1, pp. 666–671 (2003)
17. Fan, W.: Systematic data selection to mine concept-drifting data streams. In: KDD '04: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 128–137. ACM Press, New York, NY, USA (2004)
18. Fan, W., Wang, H., Yu, P.S., Stolfo, S.J.: A fully distributed framework for cost-sensitive data mining. In: Proceedings 22nd International Conference on Distributed Computing Systems. pp. 445–446 (2002)
19. Gionis, A., Mannila, H., Puolamäki, K., Ukkonen, A.: Algorithms for discovering bucket orders from data. Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining pp. 561–566 (2006)
20. Hall, L.O., Bhadoria, D., Bowyer, K.W.: Learning a model from spatially disjoint data. In: 2004 IEEE International Conference on Systems, Man, and Cybernetics, Vol. 2. pp. 1447–1451 (October 2004)
21. Henderson, A.: The ParaView Guide. Kitware, Inc., United States (2004)
22. Hullermeier, E., Furnkranz, J.: Learning label preferences: Ranking error versus position error. Proceedings IDA05, 6th International Symposium on Intelligent Data Analysis. pp. 180–191 (2005)
23. Koegler, W.S., Kegelmeyer, W.P.: FCLib: a library for building data analysis and data discovery tools. Advances in Intelligent Data Analysis VI IDA 2005, 192–203 (2005)
24. Kong, R., Zhang, B.: A fast incremental learning algorithm for support vector machine. Control and Decision 20(10), 1129–1136 (2005)
25. Korecki, J.N., Banfield, R.E., Hall, L.O., Bowyer, K.W., Kegelmeyer, W.P.: Semi-supervised learning on large complex simulations. In: Proceedings of the 19th Conference of the International Association for Pattern Recognition (2008)
26. Kotsiantis, S., Kanellopoulos, D., Pintelas, P.: Handling imbalanced datasets: a review. GESTS International Transactions on Computer Science and Engineering 30(1), 25–36 (2006)
27. Kusnezov, D.F.: Advanced Simulation & Computing: the next ten years. Tech. rep., Sandia National Labs, Albuquerque, NM 87185 (2004)
28. Lazarevic, A., Obradovic, Z.: Boosting algorithms for parallel and distributed learning. Distributed and Parallel Databases Journal 11(2), 203–229 (2002)
29. Ling, C.X., Li, C.: Data mining for direct marketing: Problems and solutions. In: Knowledge Discovery and Data Mining. pp. 73–79 (1998)
30. Maloof, M.A., Michalski, R.S.: Incremental learning with partial instance memory. Artificial Intelligence 154(1-2), 95–126 (2004)
31. Manning, C., Raghavan, P., Schutze, H.: Introduction to Information Retrieval. Cambridge University Press (2008)

32. Otsu, N.: A threshold selection method from gray level histograms. IEEE Trans. Systems, Man and Cybernetics 9, 62–66 (1979)
33. Piatetsky-Shapiro, G., Steingold, S.: Measuring lift quality in database marketing. SIGKDD Explor. Newsl. 2(2), 76–80 (2000)
34. Schoof, L.A., Yarberry, V.R.: EXODUS II: a finite element data model, Technical Report # SAND92–2137. Tech. rep., Sandia National Labs, Albuquerque, NM 87185 (1998)
35. Shipp, C.A., Kuncheva, L.I.: Relationships between combination methods and measures of diversity in combining classifiers. Information Fusion 3(2), 135–148 (2002)
36. Shoemaker, L., Banfield, R.E., Hall, L.O., Bowyer, K.W., Kegelmeyer, W.P.: Learning to predict salient regions from disjoint and skewed training sets. In: 18th IEEE Conference on Tools with Artificial Intelligence (ICTAI 2006), Arlington, Virginia, USA. pp. 116–123 (2006)
37. Shoemaker, L., Banfield, R.E., Hall, L.O., Bowyer, K.W., Kegelmeyer, W.P.: Detecting and ordering salient regions for efficient browsing. In: Proceedings of the 19th Conference of the International Association for Pattern Recognition (2008)
38. Shoemaker, L., Banfield, R.E., Hall, L.O., Bowyer, K.W., Kegelmeyer, W.P.: Using classifier ensembles to label spatially disjoint data. Inf. Fusion 9(1), 120–133 (2008)
39. Wang, F., Ma, S., Yang, L., Li, T.: Recommendation on Item Graphs. Proceedings of the Sixth International Conference on Data Mining pp. 1119–1123 (2006)
40. Webb, G.I., Boughton, J.R., Wang, Z.: Not so naive Bayes: aggregating one-dependence estimators. Machine Learning 58(1), 5–24 (2005)
41. Weiss, G.: Mining with rarity: a unifying framework. SIGKDD Explorations 6(1), 7–19 (2004)
42. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, second edition. Morgan Kaufmann, San Francisco (2005)

Appendix

Table 13: Feature ranges for the canister tear data in runs 1, 2, 3, and 4

Feature	Run 1		Run 2		Run 3		Run 4	
	min	max	min	max	min	max	min	max
DISPLX	-262.1	374.4	-186.5	472.4	-223.3	231.2	-134.8	190.2
DISPLY	-30.68	507.3	-30.52	338.8	-30.16	435.4	-30.66	235.3
DISPLZ	-486.9	206.9	-416.4	215.6	-212.4	214.6	-182.3	142.9
VELX	-144,943	262,027	-170,385	164,370	-122,159	133,943	-111,141	151,133
VELY	-111,516	234,437	-129,884	212,983	-133,301	312,411	-161,039	227,234
VELZ	-171,581	102,341	-214,932	122,208	-117,727	118,168	-96,732	119,858
ACCLX	-5.74E+11	3.71E+11	-5.36E+11	6.65E+11	-5.67E+11	6.50E+11	-2.12E+11	2.99E+11
ACCLY	-5.59E+11	3.49E+11	-6.67E+11	3.31E+11	-3.84E+11	1.88E+11	-2.50E+11	2.35E+11
ACCLZ	-3.55E+11	6.54E+11	-3.80E+11	3.87E+11	-2.82E+11	1.80E+11	-1.51E+11	1.33E+11
ELEMDEATH	0	4	0	4	0	4	0	4
ELEMVAR7	0	324.8	0	288.4	0	326.0	0	324.8
ELEMVAR8	0	0.218	0	0.226	0	0.229	0	0.217
ELEMVAR9	0	47,968	0	22,256	0	26,819	0	24,105
ELEMVAR10	0	538.9	0	597.4	0	680.4	0	557.0
ELEMVAR11	0	9.18E+06	0	5.93E+06	0	7.05E+06	0	5.06E+06
ELEMVAR12	0	1.04	0	1.04	0	1.01	0	1.00
ELEMVAR18	0	1.56	0	1.63	0	2.19	0	1.20
ELEMVAR19	0	64,043	0	30,995	0	54,581	0	29,743
ELEMVAR20	-1640	1021	-2202	1165	-1929	1038	-1974	1025
PLASTICSTRAIN	0	1.56	0	1.63	0	2.19	0	1.21
SIGMAXX	-1913	1021	-2202	1173	-1929	1051	-1974	1025
SIGMAXY	-568.7	520.5	-625.4	593.8	-540.1	527.8	-541.0	521.4
SIGMAYY	-2513	1222	-2871	1238	-2366	1084	-2327	976.7
SIGMAYZ	-566.1	515.8	-640.6	581.2	-595.4	527.7	-564.5	518.8
SIGMAZX	-574.6	515.2	-660.7	620.9	-585.9	532.5	-582.6	514.8
SIGMAZZ	-1819	1025	-2387	1300	-2268	1134	-1734	1046

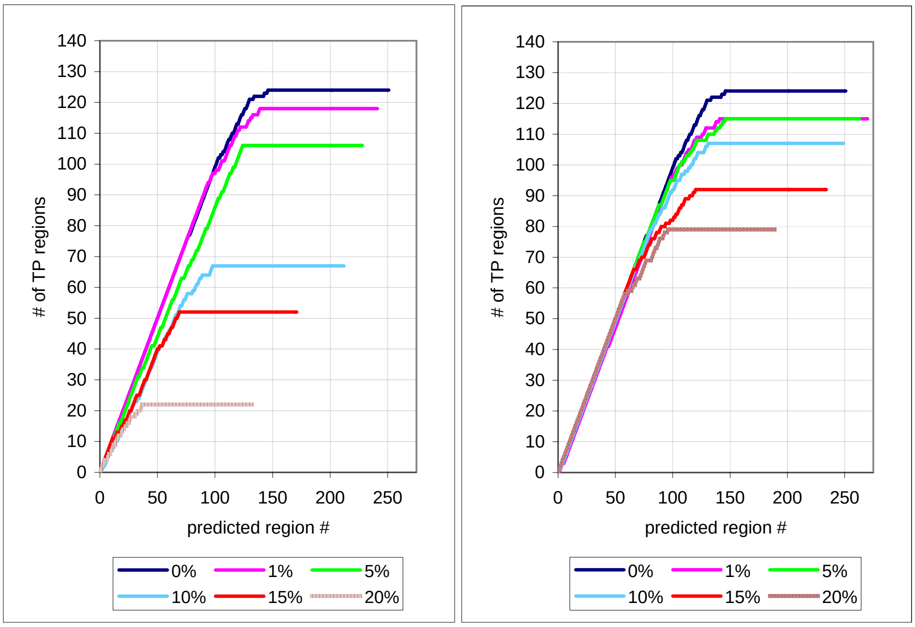
Table 14: Feature ranges for the casing simulation

Feature	Minimum	Maximum	Feature	Minimum	Maximum
DISPLX	-2.62	5.00	F-EXT-X	-1550	877.2
DISPLY	-0.24	0.23	F-EXT-Y	-354.1	345.8
DISPLZ	-10.34	0.55	F-EXT-Z	-2561	2329
VELX	-4306	7437	F-INT-X	-1550	877.0
VELY	-2108	5943	F-INT-Y	-470.0	473.1
VELZ	-11,518	3922	F-INT-Z	-4920	2354
ACCELX	-1.30E+09	8.79E+09	REACT-X	-558.3	596.4
ACCELY	-1.47E+09	1.46E+09	REACT-Y	-354.1	345.8
ACCELZ	-2.23E+09	3.29E+09	REACT-Z	-165.4	2328
F-CONTACT-X	-463.9	392.4			
F-CONTACT-Y	-469.1	478.6			
F-CONTACT-Z	-4917	2354			

Table 15: Casing simulation RF bolt labeling noise results evaluated with 10% overlap threshold.

Noise		GT	Preds	TP	FN	FP	Rec	Prec	F-m	L-qualities			
level	type									GT ratio	size	size ratio	ts
0%	NA	140	251	124	16	127	0.89	0.49	0.63	0.98	0.97	0.93	0.33
1%	b	140	241	118	22	123	0.84	0.49	0.62	0.97	0.98	0.92	0.32
5%	b	140	228	106	34	122	0.76	0.46	0.58	0.87	0.86	0.83	0.35
10%	b	140	212	67	73	145	0.48	0.32	0.38	0.84	0.83	0.78	0.17
15%	b	140	171	52	88	119	0.37	0.30	0.33	0.87	0.86	0.75	0.20
20%	b	140	132	22	118	110	0.16	0.17	0.16	0.90	0.90	0.47	0.18
1%	nb	140	270	115	25	155	0.82	0.43	0.56	0.95	0.96	0.93	0.31
5%	nb	140	263	115	25	148	0.82	0.44	0.57	0.96	0.94	0.90	0.27
10%	nb	140	249	107	33	142	0.76	0.43	0.55	0.95	0.95	0.94	0.25
15%	nb	140	234	92	48	142	0.66	0.39	0.49	0.95	0.89	0.96	0.31
20%	nb	140	189	79	61	110	0.56	0.42	0.48	0.95	0.85	0.97	0.34

NA: not applicable; b: bolt; nb: non-bolt; GT: ground truth regions; Preds: predicted regions
TP: true positives; FN: false negatives; FP: false positives;
Rec.: Recall; Prec.: Precision; F-m: F-measure; ts: timestep



(a) Bolt noise. Top to bottom: 0 to 20

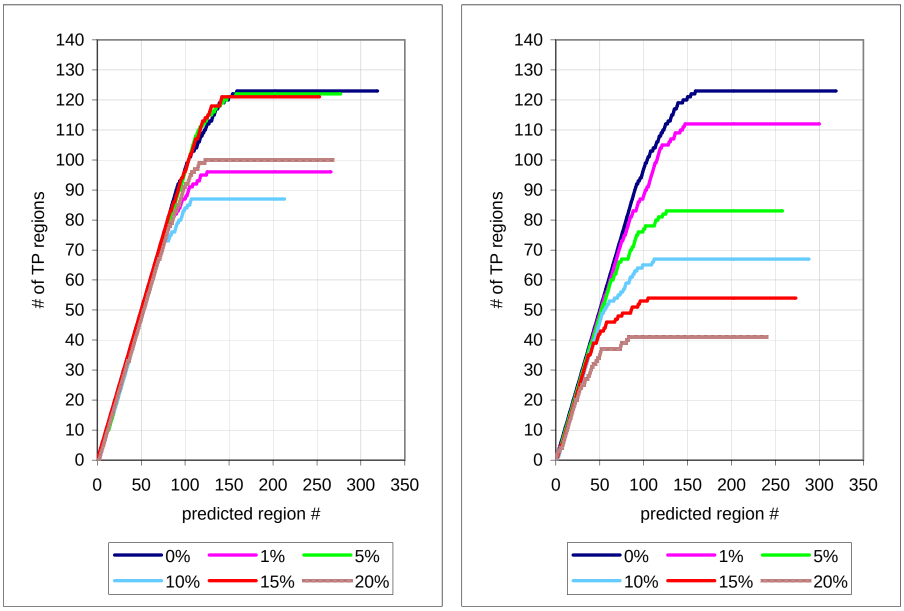
(b) Non-bolt noise. Top to bottom: 0 to 20

Fig. 14: Casing simulation bolt and non-bolt noise cumulative lift curves using RF

Table 16: Casing simulation DT bolt labeling noise results evaluated with 10% overlap threshold.

Noise		GT	Preds	TP	FN	FP	Rec	Prec	F-m	L-qualities			
level	type									GT ratio	size	size ratio	ts
0%	NA	140	319	123	17	196	0.88	0.39	0.54	0.97	0.98	0.86	0.26
1%	b	140	264	96	44	168	0.69	0.36	0.48	0.98	0.90	0.98	0.37
5%	b	140	277	122	18	155	0.87	0.44	0.59	0.94	0.94	0.93	0.22
10%	b	140	213	87	53	126	0.62	0.41	0.49	0.93	0.91	0.97	0.44
15%	b	140	253	121	19	132	0.86	0.48	0.62	0.97	0.92	0.95	0.18
20%	b	140	266	100	40	166	0.71	0.38	0.49	0.97	0.92	0.96	0.30
1%	nb	140	300	112	28	188	0.80	0.37	0.51	0.94	0.99	0.85	0.25
5%	nb	140	258	83	57	175	0.59	0.32	0.42	0.93	0.91	0.95	0.36
10%	nb	140	288	67	73	221	0.48	0.23	0.31	0.93	0.91	0.95	0.40
15%	nb	140	273	54	86	219	0.39	0.20	0.26	0.93	0.89	0.96	0.35
20%	nb	140	240	41	99	199	0.29	0.17	0.22	0.91	0.87	0.96	0.64

NA: not applicable; b: bolt; nb: non-bolt; GT: ground truth regions; Preds: predicted regions
TP: true positives; FN: false negatives; FP: false positives;
Rec.: Recall; Prec.: Precision; F-m: F-measure; ts: timestep



(a) Bolt noise. Top to bottom: 0, 5, 15, 20, 1, 10

(b) Non-bolt noise. Top to bottom: 0 to 20

Fig. 15: Casing simulation bolt and non-bolt noise cumulative lift curves using DT

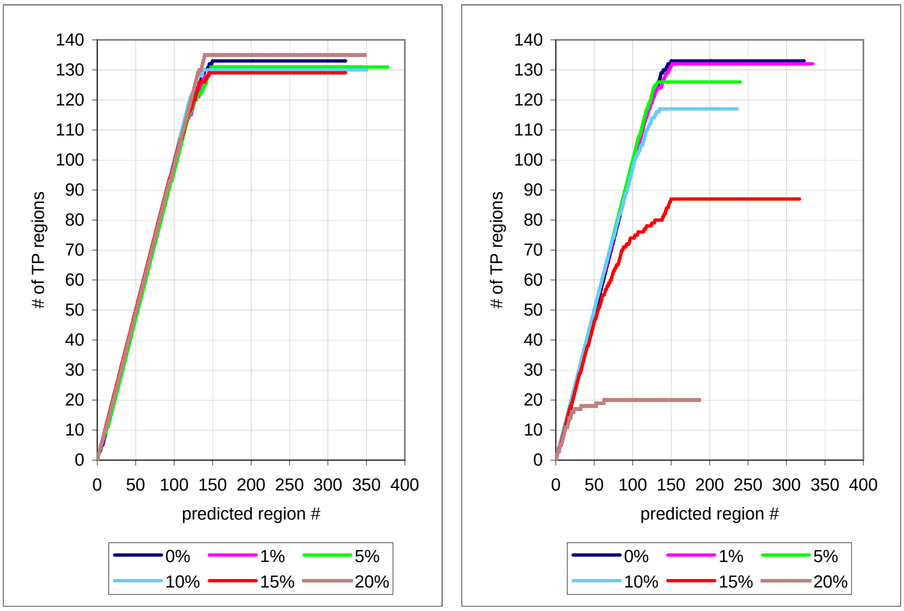
Table 17: Casing simulation SDT bolt labeling noise results evaluated with 10% overlap threshold.

Noise		GT	Preds	TP	FN	FP	Rec	Prec	F-m	L-qualities			
level	type									GT ratio	size	size ratio	ts
0%	NA	140	323	133	7	190	0.95	0.41	0.57	0.97	1.00	0.84	0.28
1%	b	140	345	131	9	214	0.94	0.38	0.54	0.97	0.99	0.83	0.29
5%	b	140	378	131	9	247	0.94	0.35	0.51	0.97	1.00	0.83	0.18
10%	b	140	351	130	10	221	0.93	0.37	0.53	1.00	1.00	0.89	0.20
15%	b	140	323	129	11	194	0.92	0.40	0.56	0.99	0.99	0.86	0.25
20%	b	140	348	135	5	213	0.96	0.39	0.55	0.99	1.00	0.85	0.23
1%	nb	140	334	132	8	202	0.94	0.40	0.56	0.98	0.99	0.82	0.27
5%	nb	140	240	126	14	114	0.90	0.53	0.66	0.99	1.00	0.97	0.20
10%	nb	140	236	117	23	119	0.84	0.50	0.62	0.97	0.99	0.93	-0.02
15%	nb	140	317	87	53	230	0.62	0.27	0.38	0.89	0.96	0.80	0.27
20%	nb	140	186	20	120	166	0.14	0.11	0.12	0.92	0.82	0.97	-0.33

NA: not applicable; b: bolt; nb: non-bolt; GT: ground truth regions; Preds: predicted regions

TP: true positives; FN: false negatives; FP: false positives;

Rec.: Recall; Prec.: Precision; F-m: F-measure; ts: timestep



(a) Bolt noise. Top to bottom: 20, 0 to 15

(b) Non-bolt noise. Top to bottom: 0 to 20

Fig. 16: Casing simulation bolt and non-bolt noise cumulative lift curves using SDT