Classification of Test Pads from Clustered PCB images

HYONTAI SUG
Department of Computer Engineering
Dongseo University
47 Jurye-ro, Sasang-gu, Busan, 47011
REPUBLIC OF KOREA

Abstract: - A robotic flying probe tester can be programmed to move the robotic probe to access all possible locations of test pads in a printed circuit board (PCB), and to record all connection test results like open or short circuits between all possible pairs of test pads in the board. For this purpose, Tan and Kit performed a clustering-based image cluster analysis on the photo image data of printed circuit boards to recover all test pad locations on the board and reported successful results. Their clustered data has been open to the public since 2024. So in this paper, several classification techniques for human comprehension were applied to give the robotic flying probe tester the location of test pads. As the final results of clustering were reviewed and corrected by experts in the original paper, we created machine learning results of classification that are easy for humans to understand, so that it could be easier to review the machine learning results before giving them to the robotic flying probe tester as input. For the classification task, we focused on knowledge discovery methods that can give the coordinates of the grey or test pad to a robot and are readable by humans. Decision trees and rules have the advantage of being relatively easy to understand because the knowledge models are expressed in a single tree structure or a set of rules, so they are widely accepted in the fields where the interpretation of trained knowledge models is important. Three different decision trees and two kinds of rule sets were constructed - J48, Random tree, REP tree (Reduced Error Pruning tree) for the decision trees, and JRIP and PART (PARtial decision Tree) for the rule sets. The accuracy of all four generated knowledge models is 100% except that of the REP tree which is 99.9997%. The size of the generated decision trees was relatively very small compared to the size of the data, 723,552 records, and the generated rule set by JRIP has only two rules. Therefore, we can conclude that the decision trees and the sets of rules for determining the test pads in the PCB have produced very successful results in terms of comprehensibility and accuracy.

Key-Words: - Printed circuit board, test pads, clustering, classification, knowledge model understandability, decision tree, rule sets

Received: April 18, 2024. Revised: October 8, 2024. Accepted: November 11, 2024. Published: December 10, 2024.

1 Introduction

As time goes by, it sometimes happens to have no or little documentation like the circuit boards in the industrial fields of electronics, so it might be necessary to try a data analysis method to generate input data for automatic testing of printed circuit boards with robotic flying probe testers. One example of a test that a robotic flying probe tester can perform is a connection test for the circuit boards. The robotic flying probe tester can be programmed to move the robotic probe to access all possible locations of test pads in a circuit, and to record all connection test results like open or short circuits between all possible pairs of test pads. For this purpose, Tan and Kit performed a clustering-based image cluster analysis on the photo image data of printed circuit boards to recover all test pad locations on the board and reported successful results, [1]. Their method found 128 locations of test pads on a printed circuit board, and they found that 8 of them were not real test pads by visual inspection, where the test pads have grey color. Their clustered data has been open to the public since April 2024. So we need to apply some classification techniques to give a robotic flying probe tester the location of test pads.

Machine learning algorithms for classification can be divided into two categories depending on whether the final result of the machine learning is in a form that is easy for humans to read or not. For example, the trained results of deep learning algorithms are very difficult to understand, [2], while the trained results of decision trees are easy to understand unless the trees are not very large, [3]. So, we want to generate some easy-to-understand machine learning results because of the nature of the

original data. As the final results of clustering were reviewed and corrected by experts in the original paper, we also want to create machine learning results that are easy for humans to understand, so that it is easier for humans to review the machine learning results before giving them to the robotic flying probe tester as input.

From now on section 2 covers related work, section 3 deals with experimental procedure, section 4 covers experimentation, and section 5 presents the conclusion.

2 Related Work

Image processing to detect PCB defects attracts many researchers' attention. For example, Melnyk and Vorovii used artificial neural networks to detect PCB defects, [4], and Cai and Li applied machine vision methods to detect PCB defects, [5]. On the other hand, the target dataset was donated by Tan and Kit in April 2024 and is available at the UCI machine learning repository named printed circuit board processed image, [6]. The dataset was used for test-pad coordinate retrieval of grey pads from PCB images, and the coordinate information can be supplied as input to a robotic flying probe tester. K-means clustering and two-stage clustering approaches were used and achieved a recall of 100% and a precision of 93.25%, [1]. Note that precision = TP/(TP+FP), and recall = TP/(TP+FN), where TP stands for the number of True Positives, FP stands for the number of False Positives, and FN stands for the number of False Negatives. So, we want to do a classification task for the dataset as the next job to do. For classification tasks, there are two main methods of knowledge discovery for classification: those that can be readable by humans and those that are difficult to read but pursue the accuracy of classification only. In this paper, we will focus on knowledge discovery methods that can give the coordinates of the grey pad to a robot and are readable by humans.

Rule sets and decision trees as knowledge models have the advantage of being relatively easy to understand because the knowledge models are expressed in a single tree structure or a set of rules, so they are widely accepted in the field where interpretation of knowledge models is important, [7].

J48 generates a decision tree and is a C4.5 program written in Java. C4.5 was developed by J.R. Quinlan in 1993, [8]. C4.5 uses a greedy search algorithm to generate a decision tree. To determine the root node of each subtree when generating a decision tree, the classification suitability of each possible sub-node is calculated by an entropy-based calculation formula, where the entropy of an attribute A is,

$$E(A) = -\sum_{k=0}^{n} p_k log_2 p_k \quad (1)$$

Each p_k is the probability of each class of the instances. Note that E(A) becomes smaller if we have a purer class distribution. The attribute with the best value among them becomes the root node. The pruning method in C4.5 first creates all the branches to the end of the decision tree as far as possible using the training dataset, and if the sum of estimated errors in terminal nodes of a subtree is greater than the estimated error of the root node of the subtree, the subtree is replaced by the root node of the subtree embracing all the instances in the terminal nodes of the subtree. In estimating error rates a normal distribution is assumed. The treatment for numerical attributes is simple; To select whether a numerical attribute can be the root node of the subtree compared to other attributes, the data of the numerical attribute is sorted first, then separated into two groups centered on the median value, and then the entropy-based value is calculated.

Random tree builds a CART [9] like decision tree that considers K randomly chosen attributes at each node, and performs no pruning. The default K value is calculated by INT($\log_2$(the number of attributes) + 1). The difference between CART and a random tree is that CART uses a greedy search algorithm using a purity-based calculation method called the GINI index to determine the root node of each subtree among all the candidate attributes, where the GINI index for attribute A is calculated by,

$$G(A) = 1 - \sum_{k=0}^{n} p_k^2 \quad (2)$$

Each p_k is the probability of each class of the instances. Note that G(A) becomes smaller if we have a purer class distribution. A random tree limits the number of candidate attributes to K.

REP tree (Reduced Error Pruning tree) generates a decision tree using a similar algorithm to C4.5, but pruning is slightly different, [10]. The pruning is done for every non-leaf subtree by replacing a subtree with the best possible leaf labeled by the majority class of the instances on the condition that the new tree would give an equal or fewer number of errors over the test set.

JRIP implements a propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), [11]. RIPPER algorithm uses a greedy approach and has a growing and pruning phase for rules. The growing phase grows a rule by

greedily adding conditions to the rule and the pruning phase gradually prunes each rule and allows the pruning of any final sequences of the conditions. After finding initial rule sets, it applies random samples to fine-tune the found rules by applying the growing and pruning phases again. Finally, any rules that would increase the description length of the whole ruleset are deleted.

PART (PARtial decision Tree) rule learner algorithm uses a separate-and-conquer approach, [12]. It builds a partial C4.5 decision tree in each iteration, and it chooses the best branch and makes it into a rule. In other words, once a partial tree has been built, a single rule is extracted by choosing a leaf that covers the greatest number of instances. This choice leads to finding the most general rule in the partial tree.

3 Experimental Procedure

The dataset in the 'Printed Circuit Board Processed Image' dataset in the UCI machine learning repository, [6], will be used for experiments. To generate understandable knowledge models for the dataset, three kinds of decision trees, J48, random tree, and REP tree, will be built in turn. Moreover, two kinds of rule-generating knowledge models, JRIP and PART, will be built in turn. For the experiment, an open-source tool called Weka will be used, [13]. Weka has all the above-mentioned machine learning algorithms implemented. For thorough testing 10-fold CV(cross-validation) will be applied. The 10-fold cross-validation randomly splits the data into 10 equal-sized groups, with 9 groups for training and 1 group for testing, alternating between 10 times. Note that by experimenting with 10-fold cross-validation, all the data used to build a machine learning model can be tested.

4 Experimentation

The dataset in the 'Printed Circuit Board Processed Image' dataset in the UCI machine learning repository, [6], is used for experiments. The goal of this experiment is to find the various readable classification models based on 10-fold cross-validation.

4.1 Printed Circuit Board Processed Image Dataset

The dataset comes from an electronic circuit board image of 71040 pixels and has 120 locations of test pads. The test pads are distinguished by grey color. There are five attributes and one class attribute. The dataset has 723552 records and each record represents one pixel. The five attributes consist of numerical attributes as in Table 1. Attributes X and Y represent the coordinates of a pixel. R, G, and B mean the primary three colors, Red, Green, and Blue. The attribute Grey represents whether the pixel has a grey color or not.

Table 1. The Property of attributes of the data set

Attribute name	Value Range	D. values	Mean	SD
X	0 ~ 965	966	482.819	273.251
Y	0 ~ 778	779	401.553	219.576
R	0.027 ~ 0.969	240	0.311	0.184
G	0.235 ~ 0.976	190	0.566	0.134
B	0.173 ~ 0.976	206	0.429	0.159
Grey	2 class values (0, 1)			

In the first row of Table 1, 'D. values' means distinct values, and 'SD' means Standard Deviation. Only 1.645% have class values of 1 in the attribute Grey. Because X and Y represent the coordinates of pixels in the PCB, the mean and SD are meaningless.

4.1.1 Decision trees

Three kinds of decision trees were generated, J48, random decision tree, and REP tree.

Table 2 shows the result of the decision tree of J48 for the data set. The tree was trained and tested with 10-fold cross-validation with a default pruning parameter of confidence of 25%.

Table 2. The accuracy and confusion matrix of the J48 decision tree for the dataset

Accuracy in 10-fold CV (%)	Confusion matrix		No. of Misclassified
100	7111649	0	0
	0	11903	

Note that even though we have a very skewed class distribution, we found a knowledge model of very good accuracy because we have a very large dataset. The following shows the generated decision tree of J48.

```
R <= 0.509804: 0 (593429.0)
R > 0.509804
|  G <= 0.662745
|  |  B <= 0.662745
|  |  |  B <= 0.541176
|  |  |  |  B <= 0.509804: 0 (5.0)
|  |  |  |  B > 0.509804: 1 (29.0)
```

| | | B > 0.541176: 1 (11874.0)
| | B > 0.662745: 0 (5690.0)
| G > 0.662745: 0 (112525.0)

Both precision and recall are 100%.

Table 3 shows the result of the random tree for the data set. Because INT(log2(5) + 1) = 3, so three attributes are considered to choose the root attribute of each subtree. The tree was trained and tested with 10-fold cross-validation with default parameter settings.

Table 3. The accuracy and confusion matrix of the random tree for the dataset

Accuracy in 10-fold CV (%)	Confusion matrix		No. of Misclassified
100	7111649	0	0
	0	11903	

The following shows the generated decision tree of the random tree.

B < 0.55
| R < 0.51 : 0 (566946/0)
| R >= 0.51
| | X < 68.5
| | | B < 0.51 : 0 (5/0)
| | | B >= 0.51 : 1 (19/0)
| | X >= 68.5 : 1 (60/0)
B >= 0.55
| B < 0.66
| | G < 0.66
| | | B < 0.6
| | | | G < 0.54
| | | | | R < 0.51 : 0 (4053/0)
| | | | | R >= 0.51 : 1 (88/0)
| | | | G >= 0.54
| | | | | B < 0.57
| | | | | | R < 0.51 : 0 (2494/0)
| | | | | | R >= 0.51 : 1 (408/0)
| | | | | B >= 0.57
| | | | | | X < 754
| | | | | | | G < 0.55
| | | | | | | | R < 0.51 : 0 (357/0)
| | | | | | | | R >= 0.51 : 1 (120/0)
| | | | | | | G >= 0.55
| | | | | | | | R < 0.51 : 0 (1132/0)
| | | | | | | | R >= 0.51 : 1 (1627/0)
| | | | | | X >= 754
| | | | | | | R < 0.51 : 0 (764/0)
| | | | | | | R >= 0.51 : 1 (302/0)
| | | B >= 0.6
| | | | R < 0.51 : 0 (1317/0)
| | | | R >= 0.51 : 1 (9279/0)
| | G >= 0.66 : 0 (45380/0)
| B >= 0.66 : 0 (89201/0)

Both precision and recall are 100%.

Table 4 shows the result of the REP tree for the data set. The tree was trained and tested with 10-fold cross-validation with default parameter settings.

Table 4. The accuracy and confusion matrix of the REP tree for the dataset

Accuracy in 10-fold CV (%)	Confusion matrix		No. of Misclassified
99.9997	7111647	2	2
	0	11903	

The following shows the generated decision tree of the REP tree.

R < 0.51 : 0 (395545/0) [197884/0]
R >= 0.51
| G < 0.66
| | B < 0.66 : 1 (7941/5) [3967/0]
| | B >= 0.66 : 0 (3819/0) [1871/0]
| G >= 0.66 : 0 (75063/0) [37462/0]

Both precision and recall are 100%.

4.1.2 Rule-based knowledge models

Two kinds of rule-based knowledge models were generated, JRIP and PART.

Table 5 shows the result of JRIP for the data set.

Table 5. The accuracy and confusion matrix of JRIP for the dataset

Accuracy in 10-fold CV (%)	Confusion matrix		No. of Misclassified
100	7111649	0	0
	0	11903	

The following shows the generated two rules of JRIP.

(R >= 0.513725) and (G <= 0.662745) and (B <= 0.662745) and (B >= 0.513725) => Grey=1 (11903.0/0.0)
=> Grey=0 (711649.0/0.0)

Both precision and recall are 100%.

Table 6 shows the result of PART for the data set.

Table 6. The accuracy and confusion matrix of PART for the dataset

Accuracy in 10-fold CV (%)	Confusion matrix	No. of Misclassified

100	7111649	0	0
	0	11903	

The following shows the generated rules of PART.

R <= 0.509804: 0 (593429.0)

G > 0.662745: 0 (112525.0)

B <= 0.662745 AND
B > 0.541176: 1 (11874.0)

B > 0.603922: 0 (5690.0)

B > 0.509804: 1 (29.0)

: 0 (5.0)

The rules should be applied in the manner of if ~ then ~ else if ~ statement. Both precision and recall are 100%.

From the experiment, we can see that rules from JRIP are the simplest knowledge model among the generated trees and rules. If the rules of JRIP are converted to code, they will look like;

```
IF (R >= 0.513725) AND (G <= 0.662745)
   AND (B <= 0.662745) AND (B >= 0.513725)
   THEN Grey :=1
 ELSE Grey := 0;
```

The above code can be applied to the original dataset to retrieve the X and Y coordinates of the location of the grey color in the PCB, and the retrieved X and Y coordinates can be used to locate the robotic flying probe testers.

5 Conclusion

It is known that it would be necessary to produce information about electrical circuits using reverse engineering technology because information such as circuit drawings used by electrical circuit production factories may be lost over time. So it might be necessary to try a data analysis method to generate input data for automatic testing of printed circuit boards with robotic flying probe testers. One example of a test that a robotic flying probe tester can perform is a connection test for the printed circuit boards. The robotic flying probe tester can be programmed to move the robotic probe to access all possible locations of test pads in a circuit, and to record all connection test results like open or short circuits between all possible pairs of test pads. For this purpose, Tan and Kit performed a clustering-based image cluster analysis on the photo image data of a printed circuit board to recover all test pad locations on the board and reported successful results. Their clustered data have been open to the public since April 2024. So we need to apply some classification techniques to give a robotic flying probe tester the location of test pads. As the final results of clustering were reviewed and corrected by experts in the original paper, we also wanted to create machine learning results that are easy for humans to understand, so that it could be easier for humans to review the machine learning results of classification before giving them to the robotic flying probe tester as input.

For the classification task, we focused on knowledge discovery methods that can give the coordinates of the grey pad or test pad to a robot and be readable by humans. Decision trees and rules as knowledge models have the advantage of being relatively easy to understand because the knowledge models are expressed in a single tree structure or a set of rules, so they are widely accepted in the fields where the interpretation of data and discovered knowledge models are important. Each machine learning algorithm is based on different ideas, and it creates its unique knowledge model according to the data, so it is necessary to apply various algorithms, therefore, as a result of applying a total of 5 algorithms, it can be said that the contribution of this paper is that it is possible to find even a very simple knowledge model despite the relatively large number of data sizes of about 700,000.

We built the knowledge models of three different decision trees and two kinds of rule sets - J48, Random tree, REP tree for the decision trees, and JRIP and PART for the rule sets. The accuracy of all four generated knowledge models is 100% except that of the REP tree which is 99.9997%. The size of the generated decision tree was relatively very small compared to the size of the data, 723552 records, and the generated rule set by JRIP has only two rules. Therefore, it can be assessed that the decision trees and the sets of rules for determining the test pads in the PCB boards have produced very successful results in terms of comprehensibility and accuracy. Currently, the clustering is done by two different clustering methods and classification has been done with different tools, so future research could be making a software tool to integrate the two

processes, clustering and classification as a seamless process for convenience.

References:

[1] S.C. Tan, S.T.W. Kit, Fast retrievals of test-pad coordinates from photo images of printed circuit boards, *2016 International Conference on Advanced Mechatronic Systems*, Melbourne, Australia, Nov. 30 – Dec. 3, 2016, pp. 464-467.

[2] S. Dong, P. Wang, K. Abbas, A survey on deep learning and its applications, *Computer Science Research*, Vol. 40, 2021, 100379, ISSN 1574-0137, https://doi.org/10.1016/j.cosrev.2021.100379.

[3] A. Sivaprasad, E. Reiter, N. Tintarev, N. Oren, Evaluation of Human-Understandability of Global Model Explanations Using Decision Tree, *Artificial Intelligence. ECAI 2023 International Workshops*, Part I, Kraków, Poland, Sep. 30 - Oct. 4, 2023, pp. 43-65, https://doi.org/10.1007/978-3-031-50396-2_3.

[4] R. Melnyk, V. Vorobii, PCB Image Defects Detection by Artificial Neural Networks and Resistance Analysis, *WSEAS Transactions on Circuits and Systems*, Vol. 23, 2024, pp. 70-83.

[5] L. Cai, J. Li, PCB defect detection system based on image processing, *Journal of Physics: Conference Series*, 2383(2022) 012077. doi:10.1088/1742-6596/2383/1/012077

[6] S. Tan, S. Tan, Printed Circuit Board Image [Dataset], *UCI Machine Learning Repository*, 2016. https://doi.org/10.24432/C5DK8H. (Accessed Date: Sep. 18, 2024).

[7] V.F. Souza, F. Cicalese, E.S. Laber, M. Molinaro, Decision Trees with Short Explainable Rules, *Advances in Neural Information Processing Systems*, Vol. 35, Nov. 28 – Dec. 9, New Orleans, USA, 2022, pp. 12365-12379.

[8] J.R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1992.

[9] L. Breiman, J. Friedman, R.A. Olshen, C.J. Stone, *Classification and Regression Trees*, Chapman and Hall/CRC, 1984, https://doi.org/10.1201/9781315139470.

[10] T. Elomaa, M. Kaariainen, An Analysis of Reduced Error Pruning, *Journal of Artificial Intelligence Research*, Vol. 15, pp. 163-187, 2011.

[11] W.W. Cohen, Fast Effective Rule Induction, *Proceedings of the Twelfth International Conference on Machine Learn*ing, Tahoe City, California, July 9–12, 1995, pp. 115-123. https://doi.org/10.1016/B978-1-55860-377-6.50023-2.

[12] E. Frank, I.H. Witten, Generating Accurate Rule Sets Without Global Optimization, *Fifteenth International Conference on Machine Learning*, pp. 144-151, 1998.

[13] E. Frank, M.A. Hall, I.H. Witten, *Data Mining: Practical Machine Learning Tools and Techniques,* Morgan Kaufmann, Fourth Edition, 2016.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The sole author contributed to the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

This work was supported by personal expenses.

Conflict of Interest

The author has no conflicts of interest to declare that are relevant to the content of this article.