Rough Set-based Rule Set Generation to see the Cause of the Occurrence of a Series of Grooves in Gravure Printing

HYONTAI SUG
Division of Computer Engineering
Dongseo University
47 Jurye-ro, Sasang-gu, Busan 47011
KOREA
sht@gdsu.dongseo.ac.kr http://kowon.dongseo.ac.kr/~sht

Abstract: - In the domain of gravure printing the phenomenon of grooves or bands that appear during printing have been hard to solve problems for decades. Such phenomenon causes delays and extra costs for possible maintenance. Therefore, in this paper we want find more accurate knowledge model in rule form to see what could be the main causes by analyzing the related data set. Because rough set theory-based data mining algorithms analyze solely based on data, it is highly possible that they may find more hidden knowledge than other heuristic-based data mining algorithms. But, this good point can be an obstacle when target data sets have many attributes, and the data sets have key-like attributes due to the limitation of the rough set-based algorithms. Because our target data set of gravure printing has such characteristics, we first applied a heuristic-based rule discovery algorithm called RIPPER, after that select attributes only were supplied as an input to rough set theory-based algorithm called MODLEM. Experiments showed very good result in accuracy.

Key-Words: - Gravure printing, data mining, rough set theory, rule learning

1 Introduction

As a means for mass printing in good quality on thin film such as polyester, nylon, etc. in low cost, gravure printing has been the most popular printing method used in flexible-package manufacturing [1]. But, during the printing sometimes grooves or bands appear on the printed surfaces, and if this phenomenon happens, the printer must be halted for maintenance unless the printer has some costly automatic cleaning mechanism [2] so that it causes delays and extra costs. According to [3], the reason for this phenomenon is unknown. Therefore, we want to see what could be the main causes of such grooves or bands by analyzing the related data set in rule form. Good point of data mining in rule form is understandability so that analysis by human is relatively easy. On the other hand, rough set theory has attracted many researchers attention as a tool for data mining in rule form, because the theory analyzes solely based on data so that it may find more hidden knowledge than other heuristic-based data mining algorithms [4]. Rough set theory has been applied to analyze data in mechanics problems also. For example, in [5] rough set is used to reduce attributes after data fusion for the problem of structural damage identification. In [6] rough set theory-based method is applied to extract data mining rules for spinners to choose proper fiber materials for quality and cost. In [7] classification of tunnel wall rock is suggested based on rough set theory and unascertained measurement theory in various evaluation indexes.

Rough set based data mining systems do the job solely based on given data themselves. But, this method can be an obstacle to find better data mining model when target data sets have many attributes, and the data sets have key-like attributes. Key-like attributes have many values and almost all of data instance have different values. On the other hand, because finding optimal rule set for a given data set takes exponential computing [8], there has been effort to find rules effectively based on heuristics and greedy search. C4.5rules that is based on decision tree of C4.5 is one of such kind [9]. On the other hand, RIPPER algorithm has shown to generate better rule sets in accuracy in many data sets than those of C4.5 rules [10]. The performance of RIPPER algorithm is also known to be good for large noisy data sets. So, we may apply heuristics-based algorithms as a pre-processing method for our purpose.

In section 2 we will describe the principle of the algorithms to solve the problem, and in section 3 we will describe the solution. And in section 4 some conclusions will be discussed.

2 Problem and Algorithms

Rough set theory-based rule induction and heuristics–based rule induction methods are our two main methodologies to find better rules of accuracy for the problem of gravure printing. Let's see the principle of related algorithms briefly.

2.1 Rough set theory-based rule induction

An information system S in rough set theory can be described as 4-tuple.

$$S = \langle U, R, V, f\rangle, \text{ where } R = C \cup D.$$

U is a finite nonempty set of objects or data instances, R is a finite nonempty set of attributes, the subsets C and D are condition and decision attribute set respectively. $V = \cup_{a\in R} Va$, where Va is the set of values of attribute a. f: $R\rightarrow V$ is description function. In other words, we can understand that the information system has two dimensional or table form, where the first row of the table has the name of attributes that can be divided into condition and decision attributes. The rows of the table have data instances where each has attributes that describe the property of each data instance. If $C = \{C_1, C_2, \ldots, C_n\}$, we can find rough set-based rules in the form,

$$(C_i = \alpha) \wedge \ldots \wedge (C_j=\beta) \rightarrow (D=\delta),$$

where α, β, δ are respective legal values for the domain of the attributes. Rough set theory-based rule discovery algorithms try to find minimal condition part. If a rule has different decision values, the rule is partially dependent on the condition part. As a rough set theory-based rule discovery system, we'll use MODLEM [11]. MODLEM has a good point of no need of pre-processing to discretize continuous values before applying rough set theory-based rule discovery process. MODLEM can treat continuous values by converting them to nominal values during rule discovery process. The rule discovery process is based on greedy search for condition part where the decision relies on entropy measure [12] or Laplacian accuracy [13]. Laplacian accuracy is calculated by the following equation,

$$(i_c+ 1)/i_{tot} + k,$$

where i_c is the number of data instances in the rule's predicted class, i_{tot} is the total number of data instances covered by the rule, and k is the number of classes in the data set. Larger values are preferred for Laplacian accuracy, while smaller values are preferred for entropy measure.

During the search continuous values of candidate attributes are sorted, then, the values are divided into two intervals by applying one of the measures to find the best condition of the rule. Entropy measure has tendency to more pure rules resulting in smaller number of covering data instances for the rules. MODLEM also has the ability to gather rules and represents as a rule, if the rules have the same conditional attribute in different attribute values having the same decision part.

2.2 RIPPER

RIPPER is abbreviation for Repeated Incremental Pruning to Produce Error Reduction, and the algorithm uses heuristics to find minimal rule set for target data set. The algorithm uses IREP* algorithm to find an initial rule set. IREP* is originated from IREP (Incremental Reduced Error Pruning). IREP algorithm consists of two part; grow_rule and prune_rule. So, IREP first randomly partitions a training data set into a growing set and a pruning set. Then, it starts to grow rules by adding an attribute and corresponding value pair to the condition part of a rule as conjunction of the condition, then it checks if information gain [14] increases. The value of information gain is calculated by the following equation.

$$\text{Gain}(R, R') = t\cdot(\log_2(p'/(p'+n')) - \log_2(p/(p+n))),$$

where R is a rule before adding an attribute and value pair as conjunction, R' is a rule after the adding, p and n are the number of positive and negative data instances covered by R respectively, and p' and n' are the number of positive and negative data instances covered by R' respectively. Finally, t is total number of data instances covered by R and R'. When we classify data instances in two classes, positive and negative, by a rule set, each rule may have positive and, or negative data instances after classification.

For nominal attribute each nominal value of the attribute is considered to see if adding it increases the information gain. During the process each value θ of continuous attribute Ac is considered whether $Ac \leq \theta$ or $Ac \geq \theta$, and the one that increases information gain is chosen. IREP stops adding a rule when the error rate of the rule is more than 50% with respect to pruning set. After finishing the growing process, IREP starts to prune the rules based on the value v of the following equation for two class data sets.

$$v = (p + (N - n))/(P + N)$$

where P is the total number of positive data instances in the pruning set, and N is the total number of negative data instances in the pruning set. p is the number of positive class training instances that are covered by the rule, and n is the number of negative class training instances that are covered by the rule. So, if reducing a rule generates larger value based on the equation, then pruning is performed.

IREP's stopping criteria in rule generation has tendency not to generate rules having small coverage. In order to avoid such property, IREP is modified to add rules if adding a rule does not increase the description length of the rule set more than given d = 64 bits(default), and this modified version of IREP is called IREP*. For each rule R_j (j = 1, …, n) generated by IREP*, two alternative rules R'_j and R''_j are constructed. By greedily adding an attribute-value pair to R_j makes R'_j, and by empting a pair in the condition part of R_j makes R''_j. Each rule for R_j is considered if changing the rule improves the minimum description length (MDL) [15] with respect to the n rules, then the one with best one among the three will be chosen. The main idea of MDL principle is that we prefer shorter messages during communication to save transmission time. Finally, if there are any positive training data remained, IREP* will be applied again to add rules.

2.3 Property of target data set

The data set used for our experiment is called 'cylinder bands'. The data set can be found in the UCI machine learning repository [16]. It has 40 attributes including one decision attribute which has two distinct values 'band' or 'noband'. The other 39 attributes are condition attributes. Table 1 has the summary of property of the attributes. The total number of data instances is 540. Numeric attributes have continuous values.

Table 1. Property of each attribute

Attribute	Data type	Value range	Number of distinct values
timestamp	numeric	19900330 ~ 19941010	296
cylinder_number	nominal		429
customer	nominal		71
job_number	nominal		262
grain_screened	nominal	yes, no	2
ink_color	nominal	key	1
Proof_on_ctd_ink	nominal	yes, no	2
blade_mfg	nominal	BENTON, UDDEHOLM	2
cylinder_division	nominal	GALLATIN	1
paper_type	nominal	UNCOATED, COATED, SUPER	3
ink_type	nominal	UNCOATED, COATED, SUPER	3
direct_steam	nominal	YES, NO	2
solvent_type	nominal	XYCOL, NAPTHA, LINE	3
type_on_cylinder	nominal	YES, NO	2
press_type	nominal	WoodHoe70, Motter70, Albert70, Motter94	4
press	nominal	802, 813, 815, 816, 821, 824, 827, 828	8
unit_number	nominal	1, 2, 5, 6, 7, 9, 10	7
cylinder_size	nominal	CATALOG, SPIEGEL,TABLOID	3
paper_mil_location	nominal	NorthUS, SouthUS, CANADIAN, SCANDANAVIAN, MidEuropean	5
plating_tank	nominal	1910, 1911	2
proof_cut	numeric	25 ~ 72.5	
viscosity	numeric	35 ~ 72	
caliper	numeric	0.133 ~ 0.533	
ink_temperature	numeric	11.2 ~ 24.5	

humifity	numeric	57 ~ 105	
roughness	numeric	0.056 ~ 1.25	
blade_pressure	numeric	16 ~ 70	
varnish_pct	numeric	0 ~ 35.8	
press_speed	numeric	0 ~ 2600	
ink_pct	numeric	41 ~ 76.9	
solvent_pct	numeric	22 ~ 53.4	
ESA_Voltage	numeric	0 ~ 16	
ESA_Amperage	numeric	0 ~ 6	
wax	numeric	0 ~ 3.1	
hardener	numeric	0 ~ 3	
roller_durometer	numeric	28 ~ 60	
current_density	numeric	30 ~ 45	
anode_space_ratio	numeric	83.33 ~ 117.86	
chrome_content	numeric	90 ~ 100	

3 Problem Solution

As you understand table 1, the data set has many attributes that can be candidates for condition parts of rules. So, we first try to find a rule set for the original data set without performing any pre-processing, then we try to convert timestamp data into 24-hour hour format, and finally we try to use the other algorithm as a method for pre-processing. All experiments will be based on 10-fold cross validation.

3.1 Rule set from original data set

The following rule set consisting of 30 rules was found after applying MODLEM for the original data set. Laplacian measure is used for condition's parameter. The accuracy is 71.48% in 10-fold cross-validation.

Rule 1. (cylinder_number in {F601, R6, G5, I49, AA43, aa58, E83, E84, G465, G55, G634, I331, I347, J60, R14, R32, t26, X13, X292, X817, 1351, aa067, AA1, AA11, AA17, aa28, aa29, AA40, aa45, aa56, aa66, aa7, aa70, aa97, E26, e302, e38, E67, e78, E81, ee304, F108, F219, F227, F242, F26, F329, F372, F416, F590, F620, F672, F77, F98, G145, G3, G38, g433, G459, G462, G48, G496, G519, G604, G625, G628, G648, G657, G67, G71, G84, G95, G98, I010, i25, I301, i305, I309, i311, I317, I324, I343, I349, I365, i366, I383, i400, I46, I820, J29, j44, J6, j70, m272, M362, M372, M374, M45, O4, O5, O8, R2, R25, R30, R38, R5, R7, T117, T176, T195, T204, T233, T244, T245, t30, t31, T312, T313, T351, T365, T8, V376, w368, W395, W785, W90, W92, X12, X126, X185, X196, X203, X216, X243, X253, x255, X281, X295, X304, X352, X356, X36, X388, X53, X66, X67, X7, X720, X746, X768, X779, X787, X791, X795, X798, X802, X821, X828, X838, X89, X95, Y209, Y216, Y255}) => (class = band) (190/190, 83.33%)

Rule 2. (job_number in {36197, 23040, 23055, 25441, 25443, 25466, 25509, 25532, 29442, 34092, 34227, 34905, 35462, 35831, 35880, 37080, 37168, 37199, 37392, 37502, 37575, 37965, 48201}) => (class = band) (33/33, 14.47%)

Rule 3. (customer in {NTLWILDLIFE}) => (class = band) (1/1, 0.44%)

Rule 4. (press_speed < 1225) => (class = band) (9/9, 3.95%)

Rule 5. (hardener < 0.35) & (timestamp < 19910317.5) => (class = band) (6/6, 2.63%)

Rule 6. (job_number in {35458}) & (type_on_cylinder in {NO}) => (class = band) (2/2, 0.88%)

Rule 7. (cylinder_number in {F374}) & (timestamp < 19900709) => (class = band) (1/1, 0.44%)

Rule 8. (cylinder_number in {F510}) & (timestamp >= 19910308.5) => (class = band) (1/1, 0.44%)

Rule 9. (cylinder_number in {J40}) & (timestamp < 19900806.5) => (class = band) (1/1, 0.44%)

Rule 10. (cylinder_number in {M260}) & (timestamp >= 19900508) => (class = band) (1/1, 0.44%)

Rule 11. (cylinder_number in {M337}) & (timestamp >= 19910506) => (class = band) (2/2, 0.88%)

Rule 12. (cylinder_number in {X199}) & (timestamp >= 19901066) => (class = band) (1/1, 0.44%)

Rule 13. (cylinder_number in {Y270}) & (timestamp >= 19910805) => (class = band) (1/1, 0.44%)

Rule 14. (job_number in {47106}) & (timestamp >= 19901108.5) => (class = band) (3/3, 1.32%)

Rule 15. (cylinder_number in {R15, X823, F126, F159, F257, F337, F615, I353, J42, J582, O14, O21, R23, R9, T178, W30, X116, X118, X138, X197, X220, X251, X264, X271, X273, X291, X346, X35, X390, X405, X60, X766, X818, 3, B181, B6, B7, E303, E510, E68, E69, E72, E74, E77, E85, E86, E90, E91, E92, F067, F103, F121, F133, F135, F146, F169, F170, F19, F192, F236, F237, F25, F261, F264, F267, F294, F308, F310, F321, F331, F38, F383, F450, F466, F482, F492, F569, F571, F629, F633, F685, F72, F76, F85, F89, G089, G404, G412, G416, G467, G572, G60, G640, G78, I303, I320, I325, I337, I346, I354, I358, J15, J33, J34, J38, J43, J45, J62, J68, J9, M103, M200, M208, M257, M309, M332, M39, M4, M402, M410,

M432, M65, M74, M93, O1, O15, O2, O28, O6, O7, R1, R17, R18, R22, R29, R3, R34, R4, R43, S180, S21, T133, T200, T218, T234, T280, T284, T32, T333, T368, T380, T383, T78, W357, W364, W406, W516, W717, X001, X019, X102, X103, X108, X127, X132, X135, X141, X146, X147, X151, X153, X155, X158, X163, X18, X184, X195, X201, X206, X21, X221, X232, X235, X242, X25, X266, X272, X282, X297, X308, X315, X326, X333, X342, X350, X351, X37, X374, X377, X389, X396, X400, X408, X414, X420, X423, X57, X65, X713, X727, X742, X754, X770, X771, X776, X777, X788, X793, X80, X804, X809, X819, X826, X830, X831, X837, X84, X9, X91, X96, Y228}) => (class = noband) (267/267, 85.58%)
Rule 16. (job_number in {47202, 27490, 35752, 35761, 36024, 47105, 23050, 25440, 25447, 34399, 34585, 34685, 35751, 35758, 36066, 37338, 37501, 37505, 37571, 37869, 47104}) => (class = noband) (69/69, 22.12%)
Rule 17. (customer in {ABBEY}) => (class = noband) (4/4, 1.28%)
Rule 18. (cylinder_number in {G81}) & (type_on_cylinder in {YES}) => (class = noband) (2/2, 0.64%)
Rule 19. (cylinder_number in {E310}) & (timestamp < 19910130.5) => (class = noband) (1/1, 0.32%)
Rule 20. (cylinder_number in {F374}) & (timestamp >= 19900709) => (class = noband) (1/1, 0.32%)
Rule 21. (cylinder_number in {G108}) & (timestamp < 19901105.5) => (class = noband) (1/1, 0.32%)
Rule 22. (job_number in {35458}) & (type_on_cylinder in {YES}) => (class = noband) (2/2, 0.64%)
Rule 23. (cylinder_number in {J40}) & (timestamp >= 19900806.5) => (class = noband) (1/1, 0.32%)
Rule 24. (cylinder_number in {M260}) & (timestamp < 19900508) => (class = noband) (1/1, 0.32%)
Rule 25. (cylinder_number in {T281}) & (timestamp < 19901108.5) => (class = noband) (1/1, 0.32%)
Rule 26. (cylinder_number in {X139}) & (timestamp < 19910427) => (class = noband) (1/1, 0.32%)
Rule 27. (cylinder_number in {X160}) & (timestamp < 19910324) => (class = noband) (1/1, 0.32%)
Rule 28. (cylinder_number in {X199}) & (timestamp < 19901066) => (class = noband) (1/1, 0.32%)
Rule 29. (cylinder_number in {Y270}) & (timestamp < 19910805) => (class = noband) (1/1, 0.32%)
Rule 30. (unit_number in {5}) & (timestamp >= 19900421.5) => (class = noband) (12/12, 3.85%)

Table 2 shows the corresponding confusion matrix.

Table 2. Confusion matrix of accuracy of 71.48%

		Number of instances classified as	
		band	no band
Correct class	band	131	97
	no band	57	255

The rules reflect the property of rough set based rule discovery method well. That is, the rules represent the minute details of data instances so that somewhat over-fitted. Especially rule 1 and rule 15 enumerate all the cylinder numbers that have ‘band’ or ‘no band’ respectively so that they just represent the fact well. The other thing to consider is timestamp data. It has the exact time data when the various measurements were performed for certain time. Because MODLEM has tendency to find rules in exact form and timestamp has key-like characteristics, we can see such property in the rule 4 ~ rule 14. In rule 1 and rule 15 we can also see the effect of key-like attribute, cylinder_number. Note that we have data set of 540 instances, and 429 different cylinder numbers exist in the data set.

3.2 Rule set after transforming timestamp values

Normal UNIX time format uses 10 digits for timestamp data type, but the original data set has 8 digits for the field. But, the 8 digits have enough information, because we are more interested in the time when each measurement was performed in 24 hour time frame. So the timestamp data can be transformed into 24-hour format. For example, because timestamp data 19910108 can be mapped to 08.19.1970 03:35:08, we used 3.58 instead of the timestamp value. In the value 3.58, 3 represents hour and .58 comes from the calculation of 35 minute divided by 60 minute resulting in 0.58. All the numbers are rounded. After transforming the timestamp and dropping cylinder_number attribute, we found more accurate rule set of accuracy of 80%. The following is the found 69 rules.

Rule 1. (job_number in {36197, 34534, 34997, 35816, 38121, 34588, 34996, 35880, 47301, 85813,

25520, 34092, 34453, 34514, 34763, 35522, 36569, 36846, 36858, 37018, 37080, 37191, 37333, 37510, 37938, 39027, 47204, 85814, 23040, 23052, 23055, 23233, 25416, 25427, 25441, 25443, 25445, 25466, 25477, 25502, 25509, 25514, 25515, 25517, 25519, 25532, 25550, 27386, 29442, 34014, 34093, 34156, 34157, 34227, 34250, 34259, 34272, 34293, 34294, 34367, 34402, 34442, 34465, 34483, 34518, 34527, 34533, 34664, 34692, 34749, 34905, 35069, 35335, 35425, 35462, 35502, 35528, 35661, 35831, 35986, 36059, 36423, 36567, 36568, 36648, 36911, 37003, 37046, 37068, 37081, 37157, 37168, 37189, 37199, 37305, 37335, 37392, 37502, 37575, 37964, 37965, 37990, 37998, 38061, 38120, 38218, 39200, 47304, 47401, 47403, 47405, 48201, 71331, 85725, 85741, 85750}) => (class = band) (162/162, 71.05%)
Rule 2. (timestamp >= 3.88) => (class = band) (66/66, 28.95%)
Rule 3. (press_speed < 1225) => (class = band) (9/9, 3.95%)
Rule 4. (job_number in {34551, 47106}) & (varnish_pct < 0.25) => (class = band) (8/8, 3.51%)
Rule 5. (customer in {HANOVRHOUSE, NTLWILDLIFE}) => (class = band) (2/2, 0.88%)
Rule 6. (ink_temperature < 11.6) => (class = band) (1/1, 0.44%)
Rule 7. (humifity >= 104) => (class = band) (1/1, 0.44%)
Rule 8. (job_number in {25513, 35529, 36784, 36805, 47201}) & (unit_number in {7}) => (class = band) (5/5, 2.19%)
Rule 9. (job_number in {36784, 36805}) & (unit_number in {2}) => (class = band) (4/4, 1.75%)
Rule 10. (hardener < 0.35) & (customer in {KMART, MODMAT}) => (class = band) (3/3, 1.32%)
Rule 11. (paper_mill_location in {SouthUS}) & (viscosity >= 39.5) => (class = band) (6/6, 2.63%)
Rule 12. (caliper < 0.18) & (grain_screened in {YES}) => (class = band) (3/3, 1.32%)
Rule 13. (customer in {HANHOUSE}) & (unit_number in {7}) => (class = band) (1/1, 0.44%)
Rule 14. (customer in {HANOVRHOUS}) & (unit_number in {7}) => (class = band) (1/1, 0.44%)
Rule 15. (job_number in {25503}) & (plating_tank in {1911}) => (class = band) (1/1, 0.44%)
Rule 16. (job_number in {34583}) & (unit_number in {2}) => (class = band) (1/1, 0.44%)
Rule 17. (job_number in {35458}) & (type_on_cylinder in {NO}) => (class = band) (2/2, 0.88%)
Rule 18. (job_number in {35811}) & (unit_number in {2}) => (class = band) (1/1, 0.44%)
Rule 19. (job_number in {47201}) & (paper_type in {COATED}) => (class = band) (1/1, 0.44%)
Rule 20. (job_number in {35001}) & (ink_type in {COVER}) => (class = band) (1/1, 0.44%)
Rule 21. (job_number in {35460}) & (type_on_cylinder in {NO}) => (class = band) (1/1, 0.44%)
Rule 22. (job_number in {36165}) & (timestamp >= 1.08) => (class = band) (1/1, 0.44%)
Rule 23. (job_number in {37001}) & (type_on_cylinder in {NO}) => (class = band) (1/1, 0.44%)
Rule 24. (job_number in {37371}) & (viscosity >= 58.5) => (class = band) (1/1, 0.44%)
Rule 25. (blade_pressure >= 59) & (timestamp < 0.94) => (class = band) (1/1, 0.44%)
Rule 26. (job_number in {37352}) & (paper_type in {COATED}) => (class = band) (2/2, 0.88%)
Rule 27. (job_number in {37365}) & (paper_mill_location in {NorthUS}) => (class = band) (2/2, 0.88%)
Rule 28. (job_number in {47203}) & (viscosity >= 55.5) => (class = band) (2/2, 0.88%)
Rule 29. (job_number in {34556}) & (ink_temperature >= 15.54) => (class = band) (1/1, 0.44%)
Rule 30. (job_number in {47103}) & (blade_pressure < 30.89) => (class = band) (4/4, 1.75%)
Rule 31. (humifity >= 99) & (timestamp >= 1.08) => (class = band) (2/2, 0.88%)
Rule 32. (customer in {REI}) & (solvent_type in {XYLOL}) => (class = band) (1/1, 0.44%)
Rule 33. (job_number in {34493}) & (viscosity >= 54.5) => (class = band) (1/1, 0.44%)
Rule 34. (job_number in {47105, 47202, 47104, 27490, 34547, 34585, 35752, 37501, 37822, 37869, 38016, 34545, 34549, 34759, 35751, 36066, 23048, 23050, 25424, 25437, 25450, 25507, 32360, 34066, 34099, 34154, 34175, 34397, 34399, 34522, 34546, 34553, 34554, 34587, 34590, 34685, 34693, 34714, 34752, 34754, 34756, 34781, 34894, 34896, 35334, 35521, 35534, 35683, 35754, 35761, 35794, 35814, 35870, 35871, 35874, 36024, 36053, 36057, 36058, 36166, 36167, 36578, 36649, 36776, 36882, 36894, 36928, 37000, 37177, 37182, 37338, 37354, 37441, 37505, 37572, 37915, 37970, 38013, 38025, 38051, 38059, 88231, 25428, 25433, 25440, 25447, 25451, 25530, 34683, 34694, 35457, 35753, 35758, 35759, 35872, 35982, 36874, 37163, 37169, 37254, 37340, 37351, 37508, 37513, 37571, 37579, 37758, 37874, 38039, 39039, 47169}) => (class = noband) (233/233, 74.68%)

Rule 35. (customer in {ABBEY, SERVMERCH}) => (class = noband) (6/6, 1.92%)
Rule 36. (job_number in {37355, 36644}) & (press_type in {Motter94, Albert70}) => (class = noband) (9/9, 2.88%)
Rule 37. (job_number in {34556, 34716}) & (humifity >= 70.5) => (class = noband) (6/6, 1.92%)
Rule 38. (unit_number in {5}) & (timestamp >= 0.91) => (class = noband) (12/12, 3.85%)
Rule 39. (proof_cut >= 68.75) => (class = noband) (3/3, 0.96%)
Rule 40. (job_number in {34493, 37352, 37365, 35001, 35460, 36165, 37371}) & (ESA_Voltage >= 0.25) => (class = noband) (5/5, 1.6%)
Rule 41. (job_number in {34493, 37352, 37365}) & (press in {824, 828}) => (class = noband) (4/4, 1.28%)
Rule 42. (job_number in {37352}) & (paper_type in {UNCOATED}) => (class = noband) (4/4, 1.28%)
Rule 43. (job_number in {47203}) & (caliper < 0.27) => (class = noband) (5/5, 1.6%)
Rule 44. (customer in {NATLWILDLIFE}) & (plating_tank in {1911}) => (class = noband) (2/2, 0.64%)
Rule 45. (viscosity >= 69.5) & (timestamp >= 1.04) => (class = noband) (2/2, 0.64%)
Rule 46. (customer in {HANHOUSE}) & (unit_number in {2}) => (class = noband) (1/1, 0.32%)
Rule 47. (customer in {HANOVRHOUS}) & (unit_number in {2}) => (class = noband) (1/1, 0.32%)
Rule 48. (job_number in {25452}) & (unit_number in {2}) => (class = noband) (1/1, 0.32%)
Rule 49. (job_number in {25503}) & (plating_tank in {1910}) => (class = noband) (1/1, 0.32%)
Rule 50. (job_number in {34493}) & (humifity >= 75.5) => (class = noband) (2/2, 0.64%)
Rule 51. (job_number in {34583}) & (unit_number in {9}) => (class = noband) (1/1, 0.32%)
Rule 52. (job_number in {35001}) & (ink_type in {UNCOATED}) => (class = noband) (2/2, 0.64%)
Rule 53. (job_number in {35458}) & (type_on_cylinder in {YES}) => (class = noband) (2/2, 0.64%)
Rule 54. (job_number in {35811}) & (unit_number in {9}) => (class = noband) (1/1, 0.32%)
Rule 55. (job_number in {36054}) & (timestamp < 0.94) => (class = noband) (1/1, 0.32%)
Rule 56. (job_number in {36165}) & (timestamp < 1.08) => (class = noband) (2/2, 0.64%)
Rule 57. (job_number in {37001}) & (plating_tank in {1910}) => (class = noband) (2/2, 0.64%)
Rule 58. (job_number in {37371}) & (unit_number in {9}) => (class = noband) (1/1, 0.32%)
Rule 59. (job_number in {47103}) & (blade_pressure >= 30.89) & (caliper >= 0.21) => (class = noband) (5/5, 1.6%)
Rule 60. (anode_space_ratio >= 117.78) & (timestamp < 2.35) => (class = noband) (3/3, 0.96%)
Rule 61. (customer in {LURIAS}) & (unit_number in {9}) => (class = noband) (1/1, 0.32%)
Rule 62. (job_number in {25513}) & (unit_number in {2}) => (class = noband) (1/1, 0.32%)
Rule 63. (job_number in {35529}) & (unit_number in {2}) => (class = noband) (1/1, 0.32%)
Rule 64. (job_number in {36784}) & (timestamp >= 0.97) => (class = noband) (1/1, 0.32%)
Rule 65. (job_number in {38064}) & (timestamp < 3.59) => (class = noband) (2/2, 0.64%)
Rule 66. (job_number in {47201}) & (press_type in {WoodHoe70}) => (class = noband) (1/1, 0.32%)
Rule 67. (job_number in {37386}) & (viscosity < 39.5) => (class = noband) (2/2, 0.64%)
Rule 68. (job_number in {34551}) & (unit_number in {9}) => (class = noband) (1/1, 0.32%)
Rule 69. (job_number in {47106}) & (paper_mill_location in {CANADIAN}) => (class = noband) (1/1, 0.32%)

Table 3 shows the corresponding confusion matrix.

Table 3. Confusion matrix of accuracy of 80%

		Number of instances classified as	
		band	no band
Correct class	band	149	79
	no band	29	283

The accuracy of above model is the best accuracy among other data mining models in rule form. For example, the accuracy of C4.5 [17] is 69.07% and RIPPER [10] is 75.37%. If we inspect the found rule set carefully, we see the complexity of found rules, and we still have another key-like attribute, job_number. On the contrary, RIPPER and C4.5 generate simpler rule set with less accuracy.

3.3 Rule set after transforming timestamp and selecting attributes

The following is the rule set found by RIPPER.

Rule 1. (timestamp >= 3.82) => class=band (77.0/3.0)

Rule 2. (press = 815) & (press_speed <= 1600) and (roller_durometer >= 30) => class=band (22.0/0.0)
Rule 3. (unit_number = 7) & (humifity >= 71) => class=band (41.0/13.0)
Rule 4. (viscosity >= 62) & (grain_screened = YES) => class=band (10.0/0.0)
Rule 5. (press_type = WoodHoe70) & (humifity <= 72) => class=band (10.0/1.0)
Rule 6. (press = 815) & (timestamp >= 1.03) and (varnish_pct <= 1) => class=band (17.0/4.0)
=> class=noband (363.0/72.0)

As we understand the rule set by RIPPER, only ten attributes are used in the condition part. Therefore, we used the ten conditional attributes by RIPPER only for MODLEM at this time, and the following 113 rules were found with accuracy of 82.78%.

Rule 1. (timestamp >= 3.88) => (class = band) (66/66, 28.95%)
Rule 2. (press_speed < 1225) => (class = band) (9/9, 3.95%)
Rule 3. (timestamp >= 3.81) & (press_type in {WoodHoe70, Motter94}) => (class = band) (56/56, 24.56%)
Rule 4. (varnish_pct >= 35.15) => (class = band) (1/1, 0.44%)
Rule 5. (viscosity >= 62.5) & (grain_screened in {YES}) => (class = band) (21/21, 9.21%)
Rule 6. (press in {815}) & (unit_number in {1}) => (class = band) (12/12, 5.26%)
Rule 7. (humifity >= 99) & (varnish_pct < 0.25) => (class = band) (4/4, 1.75%)
Rule 8. (press in {815}) & (press_speed < 1610) & (roller_durometer >= 29) => (class = band) (26/26, 11.4%)
Rule 9. (unit_number in {7}) & (viscosity >= 61.5) => (class = band) (8/8, 3.51%)
Rule 10. (unit_number in {7}) & (timestamp < 0.94) => (class = band) (3/3, 1.32%)
Rule 11. (unit_number in {7}) & (humifity < 61) => (class = band) (1/1, 0.44%)
Rule 12. (unit_number in {7}) & (timestamp >= 3.63) & (varnish_pct >= 14.35) => (class = band) (5/5, 2.19%)
Rule 13. (timestamp < 0.91) & (grain_screened in {YES}) => (class = band) (5/5, 2.19%)
Rule 14. (press in {815}) & (humifity < 74.5)& (timestamp < 1.1) => (class = band) (7/7, 3.07%)
Rule 15. (unit_number in {7}) & (press_type in {Motter94}) & (humifity >= 72.5) & (timestamp in [1.01, 3.71]) => (class = band) (13/13, 5.7%)
Rule 16. (unit_number in {6}) & (roller_durometer >= 35.31) => (class = band) (1/1, 0.44%)
Rule 17. (press in {816}) & (humifity < 72.5) => (class = band) (11/11, 4.82%)
Rule 18. (press in {815}) & (timestamp in [3.63, 3.67]) => (class = band) (3/3, 1.32%)
Rule 19. (press in {815}) & (press_speed >= 2165) & (timestamp >= 0.91) => (class = band) (4/4, 1.75%)
Rule 20. (press in {816}) & (timestamp < 0.94) => (class = band) (4/4, 1.75%)
Rule 21. (timestamp >= 3.76) & (press in {821}) => (class = band) (17/17, 7.46%)
Rule 22. (press in {816}) & (press_speed < 1458.5) => (class = band) (5/5, 2.19%)
Rule 23. (press in {815}) & (press_speed < 1655) & (humifity >= 88.5) => (class = band) (10/10, 4.39%)
Rule 24. (timestamp >= 3.77) & (humifity < 70.5) => (class = band) (5/5, 2.19%)
Rule 25. (press in {816}) & (humifity < 75.5) & (timestamp < 1.01) => (class = band) (6/6, 2.63%)
Rule 26. (press in {827}) & (grain_screened in {YES}) & (varnish_pct < 8.85) => (class = band) (9/9, 3.95%)
Rule 27. (press in {816}) & (humifity < 75.5) & (viscosity < 43.5) & (grain_screened in {NO}) => (class = band) (6/6, 2.63%)
Rule 28. (timestamp >= 3.77) & (press_speed >= 2010) => (class = band) (12/12, 5.26%)
Rule 29. (viscosity >= 67.5) & (timestamp < 3.71) => (class = band) (4/4, 1.75%)
Rule 30. (press in {815}) & (varnish_pct >= 15.7) & (timestamp < 1.06) => (class = band) (3/3, 1.32%)
Rule 31. (press in {815}) & (roller_durometer >= 38.25) & (humifity >= 82.5) => (class = band) (9/9, 3.95%)
Rule 32. (humifity >= 91.5) & (viscosity in [43.5, 47.5]) => (class = band) (3/3, 1.32%)
Rule 33. (press in {821}) & (press_speed < 1710) & (viscosity >= 44.5) => (class = band) (10/10, 4.39%)
Rule 34. (press in {828}) & (humifity < 70.5) => (class = band) (1/1, 0.44%)
Rule 35. (viscosity < 42.5) & (press in {824}) & (varnish_pct < 0.25) => (class = band) (3/3, 1.32%)
Rule 36. (press in {816}) & (humifity < 78.5) & (timestamp >= 3.67) => (class = band) (17/17, 7.46%)
Rule 37. (timestamp >= 3.77) & (humifity < 75.5) & (grain_screened in {NO}) => (class = band) (7/7, 3.07%)
Rule 38. (unit_number in {7}) & (varnish_pct < 0.25) & (roller_durometer >= 38.25) => (class = band) (8/8, 3.51%)

Rule 39. (press in {827}) & (humifity < 74.5) & (viscosity < 42.5) & (timestamp >= 2.35) => (class = band) (5/5, 2.19%)
Rule 40. (press in {828}) & (grain_screened in {YES}) & (timestamp >= 1.08) => (class = band) (5/5, 2.19%)
Rule 41. (press in {816}) & (viscosity < 42.5) & (timestamp < 1.08) => (class = band) (1/1, 0.44%)
Rule 42. (timestamp < 0.94) & (unit_number in {1}) & (viscosity < 48.5) => (class = band) (1/1, 0.44%)
Rule 43. (viscosity >= 62.5) & (press in {824}) & (timestamp in [3.71, 3.74]) => (class = band) (2/2, 0.88%)
Rule 44. (press in {816}) & (viscosity in [41.5, 42.5]) & (unit_number in {2}) => (class = band) (2/2, 0.88%)
Rule 45. (press in {828}) & (roller_durometer < 31) & (humifity < 78.5) & (timestamp >= 2.35) => (class = band) (2/2, 0.88%)
Rule 46. (press in {815}) & (timestamp >= 3.63) & (viscosity < 52.5) => (class = band) (8/8, 3.51%)
Rule 47. (press in {821}) & (timestamp >= 3.63) & (humifity < 70.5) => (class = band) (1/1, 0.44%)
Rule 48. (timestamp in [3.71, 3.73]) & (viscosity in [54.5, 62.5]) => (class = band) (3/3, 1.32%)
Rule 49. (press_speed < 1410) & (unit_number in {9}) & (timestamp in [0.99, 3.59]) => (class = band) (3/3, 1.32%)
Rule 50. (viscosity in [42.5, 43.5]) & (timestamp < 1.08) => (class = band) (4/4, 1.75%)
Rule 51. (press in {821}) & (press_speed < 1610) & (timestamp < 1.1) => (class = band) (3/3, 1.32%)
Rule 52. (press in {816}) & (timestamp in [3.66, 3.67]) & (viscosity >= 54.5) => (class = band) (1/1, 0.44%)
Rule 53. (press in {827}) & (timestamp < 0.97) & (humifity < 73.5) => (class = band) (1/1, 0.44%)
Rule 54. (press_type in {Albert70}) & (press_speed >= 1730) & (varnish_pct < 0.25) => (class = band) (2/2, 0.88%)
Rule 55. (press in {816}) & (timestamp in [1.04, 1.06]) => (class = band) (3/3, 1.32%)
Rule 56. (unit_number in {5}) & (timestamp >= 0.91) => (class = noband) (12/12, 3.85%)
Rule 57. (press_speed >= 2413) & (timestamp < 3.77) => (class = noband) (12/12, 3.85%)
Rule 58. (roller_durometer < 29) & (humifity >= 61) => (class = noband) (10/10, 3.21%)
Rule 59. (humifity < 69.5) & (grain_screened in {NO}) => (class = noband) (19/19, 6.09%)
Rule 60. (press_speed >= 2210) & (varnish_pct >= 5.21) => (class = noband) (20/20, 6.41%)
Rule 61. (varnish_pct >= 18.95) & (unit_number in {2}) => (class = noband) (14/14, 4.49%)
Rule 62. (press_speed >= 2210) & (viscosity < 45.5) & (humifity >= 70.5) => (class = noband) (19/19, 6.09%)
Rule 63. (timestamp < 1.04) & (press in {827}) & (viscosity < 54.5) => (class = noband) (10/10, 3.21%)
Rule 64. (press_type in {Motter70}) & (grain_screened in {NO}) => (class = noband) (12/12, 3.85%)
Rule 65. (press in {824}) & (timestamp < 3.71) & (viscosity >= 42.5) => (class = noband) (39/39, 12.5%)
Rule 66. (press in {802}) & (varnish_pct >= 11.75) => (class = noband) (14/14, 4.49%)
Rule 67. (timestamp in [0.97, 1.04]) & (press_type in {Motter70, Albert70, Motter94}) => (class = noband) (59/59, 18.91%)
Rule 68. (press in {828}) & (viscosity >= 60.5) => (class = noband) (6/6, 1.92%)
Rule 69. (unit_number in {9}) & (viscosity < 39.5) => (class = noband) (3/3, 0.96%)
Rule 70. (varnish_pct >= 5.45) & (press_speed >= 2112.5) => (class = noband) (24/24, 7.69%)
Rule 71. (press in {802}) & (grain_screened in {NO}) & (timestamp >= 0.96) => (class = noband) (9/9, 2.88%)
Rule 72. (varnish_pct >= 23.75) & (viscosity >= 49.5) => (class = noband) (2/2, 0.64%)
Rule 73. (varnish_pct in [10.1, 10.35]) => (class = noband) (4/4, 1.28%)
Rule 74. (unit_number in {9}) & (humifity >= 87.5) & (roller_durometer < 33.5) => (class = noband) (13/13, 4.17%)
Rule 75. (humifity in [68.5, 69.5]) => (class = noband) (4/4, 1.28%)
Rule 76. (varnish_pct >= 10.45) & (viscosity < 45.5) & (unit_number in {9, 1}) => (class = noband) (11/11, 3.53%)
Rule 77. (press_speed >= 2301.5) & (viscosity >= 60.5) => (class = noband) (7/7, 2.24%)
Rule 78. (timestamp < 0.96) & (roller_durometer < 34.5) => (class = noband) (31/31, 9.94%)
Rule 79. (press in {802}) & (press_speed >= 1817.5) => (class = noband) (1/1, 0.32%)
Rule 80. (press in {827}) & (timestamp in [0.97, 1.1]) => (class = noband) (10/10, 3.21%)
Rule 81. (press in {802}) & (press_speed >= 1666.5) & (roller_durometer < 38.25) => (class = noband) (4/4, 1.28%)
Rule 82. (varnish_pct in [5.45, 5.8]) & (press in {816, 824, 827}) => (class = noband) (15/15, 4.81%)
Rule 83. (varnish_pct >= 14.35) & (unit_number in {2}) & (viscosity < 47.5) => (class = noband) (12/12, 3.85%)

Rule 84. (roller_durometer >= 42.5) & (unit_number in {7}) => (class = noband) (1/1, 0.32%)
Rule 85. (viscosity >= 68.5) & (humifity >= 82.5) => (class = noband) (2/2, 0.64%)
Rule 86. (press_speed in [2184.5, 2194.5]) => (class = noband) (3/3, 0.96%)
Rule 87. (press_type in {Motter70}) & (varnish_pct >= 20.5) => (class = noband) (1/1, 0.32%)
Rule 88. (press in {802}) & (unit_number in {2}) & (timestamp >= 3.76) => (class = noband) (4/4, 1.28%)
Rule 89. (viscosity < 43.5) & (press_type in {Motter70}) & (timestamp >= 0.91) => (class = noband) (7/7, 2.24%)
Rule 90. (viscosity < 43.5) & (press_speed >= 2184.5) & (timestamp < 1.1) => (class = noband) (11/11, 3.53%)
Rule 91. (viscosity < 43.5) & (varnish_pct in [5.21, 6.55]) => (class = noband) (5/5, 1.6%)
Rule 92. (roller_durometer >= 42.5) & (varnish_pct < 0.25) & (viscosity < 54.5) => (class = noband) (1/1, 0.32%)
Rule 93. (viscosity < 42.5) & (press in {821}) & (unit_number in {2}) => (class = noband) (6/6, 1.92%)
Rule 94. (press in {816}) & (timestamp in [2.35, 3.66]) & (varnish_pct < 8.7) => (class = noband) (8/8, 2.56%)
Rule 95. (varnish_pct in [18.7, 18.95]) & (viscosity >= 48.5) => (class = noband) (1/1, 0.32%)
Rule 96. (press in {802}) & (press_speed >= 1710) & (viscosity < 43.5) => (class = noband) (2/2, 0.64%)
Rule 97. (humifity >= 95.5) & (varnish_pct >= 7.1) & (timestamp >= 0.91) => (class = noband) (4/4, 1.28%)
Rule 98. (press in {827}) & (viscosity >= 58.5) & (timestamp < 3.67) => (class = noband) (2/2, 0.64%)
Rule 99. (press in {828}) & (press_speed < 2010) & (timestamp < 3.66) => (class = noband) (7/7, 2.24%)
Rule 100. (press in {802}) & (press_speed in [1388.5, 1432.5]) => (class = noband) (12/12, 3.85%)
Rule 101. (press in {824}) & (viscosity >= 58.5) & (timestamp >= 3.74) => (class = noband) (5/5, 1.6%)
Rule 102. (timestamp in [0.97, 1.01]) & (press in {815}) => (class = noband) (3/3, 0.96%)
Rule 103. (press in {827}) & (varnish_pct >= 8.85) & (timestamp >= 3.63) => (class = noband) (6/6, 1.92%)
Rule 104. (viscosity < 42.5) & (press in {821}) & (humifity < 70.5) => (class = noband) (4/4, 1.28%)
Rule 105. (press in {816}) & (humifity >= 78.5) & (timestamp in [0.99, 2.35]) => (class = noband) (17/17, 5.45%)
Rule 106. (varnish_pct in [6.33, 6.65]) & (timestamp < 3.59) => (class = noband) (3/3, 0.96%)
Rule 107. (varnish_pct in [7.45, 7.95]) => (class = noband) (9/9, 2.88%)
Rule 108. (varnish_pct in [11.15, 11.6]) => (class = noband) (4/4, 1.28%)
Rule 109. (press in {802}) & (humifity in [72.5, 74.5]) => (class = noband) (2/2, 0.64%)
Rule 110. (viscosity in [40.5, 41.5]) & (timestamp in [0.89, 2.35]) => (class = noband) (11/11, 3.53%)
Rule 111. (timestamp < 0.99) & (roller_durometer < 32.5) => (class = noband) (35/35, 11.22%)
Rule 112. (viscosity in [57.5, 58.5]) & (press in {816, 821}) => (class = noband) (5/5, 1.6%)
Rule 113. (viscosity in [41.5, 42.5]) & (press in {821}) & (timestamp >= 1.1) => (class = noband) (4/4, 1.28%)

Table 4 shows the corresponding confusion matrix.

Table 4. Confusion matrix of accuracy of 82.78%

		Number of instances classified as	
		band	no band
Correct class	band	161	67
	no band	26	286

The result above is very accurate yet found in rule form for the data set. Moreover, the number of rules is not much increased, because the rules of key-like attributes having disjunction of attributes are actually distinct rules as we found in section 3.1 and 3.2.

4 Conclusion

Gravure printing has been used for almost one hundred years as a means of mass printing with good quality in low cost. But, during the printing sometimes grooves appear on the printed surfaces, and if this phenomenon happens, the printer must be halted for maintenance so that it causes delays and extra costs. Therefore, we may want to see what could be the main causes of such grooves by analyzing the related data set.

Because rough set theory-based data mining algorithms analyzes solely based on data, it is highly possible that they may find more hidden knowledge in higher accuracy than other heuristic-based data mining algorithms. But, this good point can be an obstacle to find better data mining models when target data set have many attributes, and the data set has key-like attributes due to the limitation of rough set theory-based algorithms. But, because our target data set of gravure printing has such properties, we first applied a heuristic-based rule discovery machine learning algorithm called RIPPER, after that select attributes only were supplied as an input to rough set theory-based machine learning algorithm called MODLEM. Experiments showed very good result in accuracy.

References:

[1] R. Kitsomboonloha, *Modeling and Applications of Highly-Scaled Gravure*, Technical Report No. UCB/EECS-2015-213, PhD Thesis, UC Berkeley, 2015.

[2] M.T. Pudas, J. Hagberg, S. Leppävuori, The self-cleaning gravure(SCG), A solution for gravure groove blocking and a novel printing method, *Journal of Imaging Science and Technology*, Vol.48, No.4, 2004, pp. 375-381.

[3] W.W. Cohen, Fast Effective Rule Induction, *IEEE Expert*, Vol. 9, Issue 1, 1994, pp. 60-66.

[4] Q. Zhang, Q. Xie, G. Wang, A survey on rough set theory and its applications, *CAAI Transactions on Intelligence Technology*, Vol. 1, Issue 4, 2016, pp.323-33.

[5] S. Jiang, J. Yao, Structural damage identification method based on rough set and data fusion, *Engineering Mechanics*, Vol. 26, No. 4, 2009, pp. 207-213.

[6] Q. Xiang, Z. lv, J. Yang, X. Yin, Mining rule of quality for spinning process with rough set theory, *Applied Mechanics and Materials*, Vol. 80-81, 2011, pp. 80-81.

[7] C. Huang, S. Tian, Q. Li, J. Huang, Evaluation of rock quality of tunnel wall rock based on rough set theory and unascertained measurement theory, *Mathematical Problems in Engineering*, Vol. 2018, Article ID 3571028, 2018.

[8] P. Domingos, A few useful things to know about machine learning, *Communications of the ACM*, Vol. 55, No. 10, 2012, pp. 78-87.

[9] B. Hssina, A. Merbouha, H. Ezzikouri, M. Erritali, A comparative study of decision tree ID3 and C4.5, *International Journal of Advanced Computer Science and Applications, Special Issue on Advances in Vehicular Ad Hoc Networking and Applications 2014*, Vol. 4, No. 4, 2014.

[10] W.W. Cohen, Fast effective rule induction, In *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 115-123, 1995.

[11] J.W. Grzymala-Busse, J. Stefanowski, Three discretization methods for rule induction, International Journal of Intelligent Systems, Vol. 16, 2001, pp. 29-38.

[12] C.E. Shannon, A mathematical theory of communication, *Bell System Technical Journal*, Vol. 27, No. 3, pp. 379-423, 1948.

[13] P. Clark, R. Boswell, Rule induction withCN2: some recent improvements, In *Proceedings of the 5th Euro Working Session on Learning*, pp. 151-163, 1991.

[14] J.R. Quinlan, Learning logical definitions from relations, *Machine Learning*, Vol. 5, pp. 239-266, 1990.

[15] T. Roos, Minimum description length principle, *Encyclopedia of Machine Learning and Data Mining*, C. Sammut, G.I. Webb eds., Springer Science+Business Media New York, 2016.

[16] D. Dua, C. Graff, *UCI Machine Learning Repository* [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2019.

[17] J.R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufman Publishers, Inc., 1993.