IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 24, No. 5, October, 2019, pp. 553-558
Journal homepage: http://iieta.org/journals/isi

Exception-Tolerant Decision Tree / Rule Based Classifiers

Sayan Sikder[1*], Sanjeev Kumar Metya[2], Rajat Subhra Goswami[1]

[1] Department of Computer Science and Engineering, National Institute of Technology, Arunachal Pradesh 791112, India
[2] Department of Electronics and Communication Engineering, National Institute of Technology, Arunachal Pradesh 791112, India

Corresponding Author Email: sayan.phd@nitap.ac.in

https://doi.org/10.18280/isi.240514

Received: 21 June 2019
Accepted: 15 September 2019

Keywords:
classification, exception tolerant, bagging, boosting, default rule, inefficient rules

ABSTRACT

Many of the existing classifiers cannot deal with exceptions, which are not to be ignored in real life. In this paper, an exception-tolerant methodology is proposed based on each of the following three popular algorithms for multi-class classification problems: C4.5, PRISM and RISE. The performance of each algorithm was improved by converting the outputs into the format of RISE induced rules, applying bagging and boosting techniques, adding exceptions with a default rule, and excluding inefficient rules. The improved versions of C4.5, PRISM and RISE are named as OE^2-C4.5, OE^2-PRISM and OE^2-RISE, respectively. Note that OE^2 stands for Ordering of Efficient Rules and Inclusion of Exceptions. Empirical results show that OE^2-C4.5 and OE^2-RISE significantly outperformed classical C4.5, PRISM and RISE for each dataset. Our methodology provides a reference for improving other weak learners individually or in an ensemble.

1. INTRODUCTION

A plenty of classifiers had already been proposed but none of them deals with exceptions. Theories are formed by observing examples, the phenomenon being called 'Superarticulacy' [1] and there would certainly be instances not abiding by the theories. These had not been dealt with in machine learning till date. The challenge was to differentiate between an exception and a noise. The improvement in the performance of the classifier when exceptions are considered, however, totally depends on the data sets. But in a real life scenario, exceptions are better not to be ignored. The proposed methodology also includes features like n-fold cross-validation, exclusion of inefficient rules, ordering the rules by weighted voting and inclusion of a default rule. This methodology has been applied on three popular classifiers C4.5, PRISM and RISE. These are old algorithms but can perform pretty well with small training sets whereas a few state-of-the-art classifiers like Random Forest [2], Xgboost [3] etc. fail to perform well without extensive training. C4.5 [4, 5] was proposed by J. Ross Quinlan which, from the training set, forms a decision tree. Decision trees come up with amazing outputs but are not easy to understand, work with, or to be manipulated. PRISM was proposed by Jazdia Cendrowska [6] which produces rule sets which are more comprehensible than a tree. The RISE algorithm was proposed by Pedro Domingos [7] which keeps on generalizing instances until a are in rule is obtained. The rule sets formed and their representation the most intelligible form amongst the three algorithms dealt with.

2. BRIEF REVIEW OF THE CLASSIFIERS

The book C4.5 was proposed by Quinlan [4, 5]. It can be seen as a descendant of the ID3 algorithm [8] which induces decision tree/ rules from a given training set. In ID3, Gain criterion was used and it gave descent results, but a strong bias could be seen in favor of the tests with many results. Hence to normalize the biased behavior of ID3, Gain ratio was introduced. Gain ratio helps to identify the attribute which should be considered first as a node in the tree in this algorithm. The attribute contributing the highest Gain ratio is chosen and made a node from which branches propagate to join the attribute which has the next best Gain ratio and so on.

Let 'S' be the training data set with 'n' quantity of outcomes having S_1, S_2, S_3, ….. S_n as subsets. If T be a set of examples, let freq(Ci, T) represent the number of cases from T which belongs to the class C_i and | T | denotes the number of examples in the set T. The probability of selecting one instance from set 'T' and considering C_j to be its class is $\frac{freq(C_j,T)}{|T|}$ resulting in $\log_2\left(\frac{freq(C_j,T)}{|T|}\right)$ of information.

Therefore,

$$info(T)=\left[-\sum_{j=1}^{k}\frac{freq(C_j,T)}{|T|}\times\log_2\frac{freq(C_j,T)}{|T|}\right]bits \quad (1)$$

When applied to the training instances S, info(S) measures the mean of the total amount of information required for predicting the class of an instance in S. It can also be seen as the Entropy of the set T. Suppose 'S' has been partitioned according to the 'n' outcomes of a test case 'X'. The weighted sum over the subsets gives the expected information

requirement as:

$$info\_X(S) = \sum_{i=1}^{n} \frac{|S_i|}{S} \times info(S_i) \quad (2)$$

Therefore, the gain quantity is given by

$$gain(X) = info(S) - info\_X(S) \quad (3)$$

This Gain criteria was used in ID3, but C4.5 uses gain ratio which is:

$$gain\_ratio(X) = gain(X) / split\_info(X) \quad (4)$$

Here a new term was introduced known as split_info which is given by:

$$split\_info(X) = -\sum_{i=1}^{n} \frac{|S_i|}{|S|} \times \log_2 \frac{|S_I|}{|S|} \quad (5)$$

The attribute for which the Gain Ratio is maximum, is selected as the node from which the tree propagates further. Then similar calculation is performed with the left over attributes and the one with the highest Gain Ratio is selected as the next node. The above mentioned process is followed until and unless all the leaves are individual classes themselves. The basic C4.5 algorithm is given below:

Algorithm 1: The C4.5 algorithm

1: Tree = {}
2: if E is pure OR other stopping criteria met then
3: terminate
4: end if
5: for all features f ∈ E do
6: calculate information-theoretic criteria if we split on a
7: end for
8: f_{best} ← Best attribute according to above computed criteria
9: Tree = Create a decision node that tests f_{best} in the root
10: E_v= Induced sub-datasets from E based on f_{best}
11: for all E_v do
12: $Tree_v$ = C4.5(E_v)
13: Attach $Tree_v$ to the corresponding branch of Tree
14: end for
15: return Tree

C4.5 is hugely applied for data classification in various domains and a few improvements had been proposed in Efficient C4.5 [9] and C5.0.

PRISM was proposed by Cendrowska [6] and like C4.5, it also evolved ID3. ID3 produced decision trees and those were not very comprehensible. In PRISM a branch could be considered as an attribute-value pair. The relevant relation between an attribute-value pair and the specific classification is considered in PRISM. The attribute-value pair which contains maximum information is chosen as one term of one rule for a particular classification. Now the subset of the training set which has the attribute-value pair is considered and again another attribute-value pair which contains the maximum information is searched and it goes on until the subset contains only one class. A conjunct of all the attribute-value pairs chosen resulting to one class becomes a rule. If the training data constitutes examples resulting in more than one class, then for each classification, δ_n. The basic PRISM algorithm is as follows:

Algorithm 2: The PRISM algorithm

1: for all attribute-value pair α_x do
2: compute the probability of occurrence, $p(\delta_n | \alpha_x)$, of the classification α_x
3: end for
Ensure: select the α_x for which $p(\delta_n | \alpha_x)$ is highest and create a subset of the training data including all the instances which contain the selected α_x
4: repeat Steps 1 and 2 for this subset
5: until the subset contains only instances of the certain class
Ensure: all the instances, covered by such rule from the training set, are removed
6: repeat Steps 1-4
7: until all instances of class δ_n are removed

After the induction of a rule, the training set is considered without those instances which are completely covered by the rule and the same procedure is repeated to induce a collection of rules concluding to the same class. After the induction of all the rules resulting in one specific class, the training data is reinstated to its inceptive state and the algorithm is employed again. Because the classifications are considered individually, the order of their representation is insignificant. When no other instance is left uncovered, the algorithm terminates.

Instance-based learning [10] and Rule induction [11, 12] are two different empirical approaches vastly used in machine learning having their own constraints. RISE (Rule Induction from the Set of Exemplars) amalgamates features of both the approaches to overcome some of the constraints and hence called a multi-strategy learning method [7]. The algorithm chooses an instance from the training data and assumes it as a rule and tries to generalize it. It is done by generalizing each antecedent by assigning it the value "*" which means "any" and is same as dropping the antecedent. For each such generalization the algorithm finds out the heuristic value h(.). The Heuristic value of a generalized individual example is given by:

$$h(r) = p(r) - (1-\eta).S.n(r) \quad (6)$$

where, r represents the rule, $p(r)$ stands for the number of instances covered by the rule whose consequences are identical to the rule's, S represents the sample size, and $\eta \in [0;1]$ is the noise coefficient of tolerance [7]. The sum of the heuristic values of all the rules in the rule set is considered as the heuristic value of the rule set and it is represented by H(.). The reason why η is introduced is that in noisier domains, rules are meant to cover a higher proportion of negative instances. Whereas in domains with negligible noise this proportion is negligible too, hence η is considered to be 0. The generalization with the highest h(.) is made a rule. Similarly, the next instance is selected and checked whether it is covered by the rule already formed. If it is not covered, the generalization process will start again. The one with the highest. h(:) will be made a rule similarly. These steps will be carried out repeatedly until all the instances are covered.

A test example is classified by RISE by matching it with the rules in the rule set. Whenever more than one match is found,

the conflict is resolved by weighted voting. Test examples with no matches at all are classified by the default rule. The default rule is nothing but all the antecedents generalized with the class which is found most in the training data set. Further modifications to the classical RISE (also known as the RISE system 1.0) had been made and reported such as The RISE 2.0 System, Unifying Instance-Based and Rule-Based Induction [13], Using partitioning to speed up specific-to-general rule induction [14] and SUNRISE [15] for performance enhancement of the RISE Algorithm.

Algorithm 3: The RISE algorithm

```
1: procedure RISE (T be the training set)
2:   Let R be T (R be rule set).
3:   Calculate H(R).
4:   Repeat
5:     For each rule in R,
6:       Let r be the current version of the rule,
7:       For each one of the r's antecedents A,
8:       Try generalising A, and
9:       And calculate resulting H(R').
10:      Select the A whose generalisation results in
         maximum value for H(R')
11:      If H(R') ≥ H(R),
12:      Then generalise A,
13:      If r is identical to another rule in R,
14:      Then remove r from R.
15:      Replace H(R) by max. H(R'),
16:  Until no increase in H is obtained.
17:  Return R.
18: end procedure
```

3. PROPOSED METHODOLOGY

The methodology involves features like re-sampling, 100-fold cross-validation, inclusion of the exceptions at the beginning of the rule set, exclusion of inefficient rules, ordering of rules beforehand using weighted voting and inclusion of a default rule at the end of the rule set. These features are discussed as follows.

3.1 Re-sampling and cross-validation

The process initiates with a data set as the input. Data is re-sampled and n-fold cross-validation [16, 17] is performed where n = 100, hence resulting in 100 sample datasets and 100 respective sample test sets. 100 separate rule sets from 100 sample datasets are formed by deployment of each of the 3 algorithms viz. C4.5, PRISM and RISE respectively. Then the rest of the methodology is performed. By using the respective rule sets, classes are predicted for each instance in all the 100 sample test sets.

The accuracy of classification (the ratio of the number of correct classifications to the total number of instances in the test set) and the standard deviation are calculated for each of the 100 cases. The whole procedure is repeated for 5 times. Hence for each data set classified by any one of the three algorithms, there will be 500 (100*5) different accuracies and standard deviations. The average of the accuracies and the respective standard deviations are considered in this paper.

3.2 Exclusion of inefficient rules

Instead of correctly classifying, the rules generated may misclassify more often. The threshold is taken 50 % for an individual rule. At first rule sets are formed. Then the number of correct classifications and misclassifications for a particular rule is calculated. If the later exceeds the former for a particular rule, that rule is eliminated from the rule set. Exclusion of these rules in the primary stage of the classification procedure helps to avoid unnecessary computational efforts.

3.3 Detection and addition of exceptions to the rule set

In this section a concept of exceptions in the training sets is introduced. These exceptions are extracted from the training set and used as rules so that the exceptions in the test set are not misclassified.

3.3.1 Exceptions

Rule generation by the inducers/ classifiers is done depending upon the training set. There may be instances in the training set itself, which may contradict a rule in the rule set. That means the class obtained by the rule which covers the instance does not match with the actual class. This can be seen as a misclassification. So that particular instance in the training set is either a noise or an exception. Now if it conflicts with some other instances, then it is considered as a noise or else considered as an exception.

3.3.2 Example

For further clarity, an example is shown below. Let the training set be as follows as shown in Table 1. Let the rule be: *, 2, *, 1, 1 and let the test set be identical to the training set. Now, trying to classify, the following is obtained which is shown in Table 2. The instances which are hit by the rule but not correctly classified are shown in Table 3. Among these two instances, the second one is contradicting with the fourth instance of the training set. Henceforth, it is considered as a noise. The first one is considered as an exception.

Table 1. Training example

A	B	C	D	Class
3	1	2	2	1
3	2	1	1	2
3	2	3	1	1
3	2	3	1	1
3	2	3	1	2
2	2	1	1	1

Table 2. Test example covered by rules

A	B	C	D	Class	hit/miss	Classification status
3	1	2	2	1	miss	nil
3	2	1	1	2	hit	wrong
3	2	3	1	1	hit	right
3	2	3	1	1	hit	right
3	2	3	1	2	hit	wrong
2	2	1	1	1	hit	right

Table 3. Misclassified instances

A	B	C	D	Class
3	2	1	1	2
3	2	3	1	2

3.4 Ordering with respect to specificity

In this section we are proposing a cost efficient yet well performing (in terms of accuracy) combination method. Here after generating a pool of rules as the end result of the actions documented in former sections, we order them according to their specificity. The less the number of generalized attributes in a rule is, the more specific the rule becomes. Hence the exceptions remain in the first cluster, the second cluster comprises of rules with only one of the attributes generalized and so on.

When the system is fetched with an unlabeled test instance it first tries to find a match in the cluster of exceptions. If a match is found (in the cluster comprised of exceptions, if a match is found it will be the only one) the class of the rule is given as the class of the test instance and the search process terminates. If no matches are found in the first cluster the system tries to find a match in the second cluster and so on. It helps to avoid conflicts between rules of different specificity and due to the fact that specific rules are given priority, the chances of a test case being misclassified are reduced. Still there remain chances of conflict between rules of same specificity. Such cases are taken care of by the use of weighted voting.

3.5 Inclusion of a default rule

Instances which remain uncovered by the rules are not assigned any classes, contribute to the misclassification count. This reduces the accuracy of prediction. To address this issue many known and popular classifiers had already included a default rule for better results. In this work we have also successfully reduced the number of misclassifications by adding a default rule. The default rule assigns the most occurrent class to the test sample irrespective of the values of the attributes. For example, if the most occurrent class in a data set be 1, and the number of attributes be 4, then the default rule would look like *, *, *, *, 1.

4. DATASETS

In this paper, 20 real life data sets, obtained from UCI data repository for machine learning are used to understand whether the proposed methodology helps popular algorithms like C4.5, PRISM and RISE to increase the accuracy in class prediction or not. Different domains (medical, biological, marketing etc.) have contributed to the data sets. All the data sets vary in size. The one having minimum attributes (Zoology) contains 101 examples and the one having maximum attributes (Hypothyroid) contains 3162 examples. The number of attributes in each data set also reflects variety, the Iris data set having the minimum (4 attributes) and the Annealing data set having the maximum (38 attributes). Amongst these 20 data sets, as many as 11 data sets (Annealing, Breast cancer, Cleveland, etc.) contain attribute(s) with missing values. In Table 4 information about the data sets used, is shown. “Exs.”, “Att.”, “Class”, “Missing” columns in the table stores the number of training instances, the number of features, the number of classes and the information of whether it contains any missing valued attributes or not respectively for each data set.

Table 4. Data sets used in experimental evaluation

Data set	Exs.	Att.	Class	Missing
Annealing	798	38	5	Yes
Breast cancer	286	9	2	Yes
Cleveland	272	13	5	Yes
Credit	690	15	2	N/A
Echocardio	132	12	3	Yes
Ecoli	336	8	8	No
German	201	25	7	Yes
Glass	214	9	7	No
Hdva	200	13	5	Yes
Horse colic	300	22	2	Yes
Hypothyroid	3162	25	2	N/A
Iris	150	4	3	No
Liver disorder	345	6	2	No
Lymphography	148	18	4	No
Pima diabetes	768	8	2	Yes
Primary tumor	339	17	20	Yes
Tae	151	5	3	No
Voting	435	16	2	Yes
Wisconsin	699	9	2	Yes
Zoology	101	16	7	No

5. RESULTS AND DISCUSSION

OE^2-C4.5, OE^2-PRISM and OE^2-RISE all yielded better accuracy in class prediction than their respective conventional versions for most of the data sets considered for empirical evaluation. Table 5 constitutes the average accuracy and the standard deviations obtained by OE^2-C4.5, OE^2-PRISM, OE^2-RISE and the best of C4.5, PRISM and RISE using conventional methodology for 20 real life data sets, taken from the UCI data repository for machine learning. For 1 data set (Credit), OE^2-C4.5, OE^2-PRISM and OE^2-RISE all fail to perform better than the best of the conventional versions of the algorithms and for 1 data set (Hypothyroid), accuracy obtained by the best of OE^2-C4.5, OE^2-PRISM and OE^2-RISE happens to be equal to that obtained by the best of the conventional versions of the algorithms. Out of 20 data sets, for 13 data sets all the three proposed algorithms have outperformed the best of conventional versions of the algorithms. However, it still cannot be concluded that the algorithms using proposed methodology has significantly outperformed the best of algorithms using conventional one. Henceforth, Friedman test [18, 19] is performed. For all the training sets the methodologies are ranked individually, the one with the highest accuracy being ranked 1, the next highest being ranked 2 and so on, and for a tie. For all the algorithms, the average of ranks is calculated which are 3.425 for the best of conventional algorithms, 1.8 for OE^2-C4.5, 2.55 for OE^2-PRISM and 2.225 for OE^2-RISE.

The Friedman statistic is then calculated and it is given by [19]:

$$X^2F = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right] \quad (8)$$

where, the number of data sets is represented by N, the number of methods / algorithms is represented by k and $\sum_j R_j^2$ is the sum of the squares of the average ranks of the individual methods / algorithms. The value of The Friedman statistic is found to be 17.085 in this particular scenario.

A more useful statistic could be derived from the Friedman statistic which is given by [20]:

$$F_F = \frac{(N-1)}{N(k-1)-X_F^2} \quad (9)$$

It is distributed according to the F-distribution with (k - 1) and (k - 1)(N - 1)degrees of freedom. The value obtained, in this case is 7.56 rounded off to two decimal places. As the number of methods is 4 and the number of data sets is 20 we can say F_F is distributed according to the F distribution with (4 - 1) = 3 and (4 - 1)(20 - 1) =57 degrees of freedom. Hence the critical value of F(3, 57) for α = 0.05 is 2.77 rounded off to two decimal places which is less than F_F , so we reject the null-hypothesis of the algorithms performing with no significant differences.

Table 5. Data sets used in experimental evaluation

Data Set	Best of conventional acc. ± s.d	Algorithm	OE²C4.5 acc ± s.d	OE²PRISM acc ± s.d	OE²RISE acc ± s.d
Annealing	97.40 ± 1.61	RISE	99.27 ± 0.87	98.85 ± 1.25	98.92 ± 1.00
Breast cancer	72.71 ± 7.29	RISE	77.17 ± 7.38	75.33 ± 7.86	77.51 ± 6.61
Cleveland	50.04 ± 5.70	C4.5	58.31 ± 8.64	59.05 ± 7.81	57.75 ± 8.96
Credit	86.09 ± 4.31	RISE	86.00 ± 3.10	55.51 ± 5.42	86.00 ± 3.10
Echocardio	65.99 ± 11.69	RISE	67.02 ± 12.42	65.02 ± 10.54	70.29 ± 11.54
Ecoli	81.26 ± 4.71	RISE	81.18 ± 6.64	42.56 ± 8.19	83.32 ± 6.46
German	71.92 ± 1.44	C4.5	75.38 ± 4.20	75.08 ± 4.40	72.32 ± 4.11
Glass	72.86 ± 7.36	RISE	78.99 ± 9.16	78.23 ± 7.91	81.57 ± 7.28
Hdva	31.50 ± 13.24	PRISM	35.10 ± 9.35	36.30 ± 8.23	36.30 ± 9.84
Horse colic	82.00 ± 3.01	RISE	86.13 ± 6.34	84.53 ± 7.27	84.20 ± 6.56
Hypothyroid	99.20 ± 0.39	C4.5	99.20 ± 0.44	95.26 ± 1.19	98.92 ± 0.43
Iris	96.65 ± 3.35	C4.5	97.73 ± 4.13	97.60 ± 4.57	98.67 ± 3.77
Liver disorder	67.36 ± 7.22	RISE	75.38 ± 6.42	75.38 ± 6.28	74.97 ± 6.59
Lymphography	81.44 ± 9.30	C4.5	84.18 ± 8.96	82.96 ± 9.40	81.77 ± 9.77
Pima diabetes	74.47 ± 6.02	C4.5	76.98 ± 4.53	76.65 ± 4.74	76.36 ± 5.44
Primary tumor	41.42 ± 7.25	C4.5	50.62 ± 8.60	48.55 ± 7.74	48.97 ± 8.28
Tae	53.63 ± 15.2	RISE	64.07 ± 11.54	22.37 ± 8.01	66.88 ± 10.65
Voting	95.39 ± 3.09	C4.5	96.00 ± 3.33	96.23 ± 3.16	95.09 ± 3.45
Wisconsin	94.20 ± 2.10	C4.5	96.91 ± 2.17	97.08 ± 1.74	97.08 ± 2.12
Zoology	90.80 ± 5.11	C4.5	95.84 ± 6.32	92.11 ± 9.54	88.44 ± 10.47

The Nemenyi test [19] is done then for pairwise comparisons. The critical valueis 2.569 for comparing 4 methods considering α = 0.05 and the corresponding CD (critical difference) is found to be 1.05 rounded off to two decimal places using [19, 20].

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \quad (10)$$

The difference between the average rank of OE²-C4.5 and the best of the conventional algorithms is found to be 1.625 which is greater than the obtained CD so we can affirm that OE²-C4.5 is significantly outperforming the conventional versions of all the 3 algorithms. Similarly, we can conclude the same for OE²-RISE as the difference is 1.2 in this case. Although OE²-PRISM has outperformed its conventional version it can be inferred that it fails to perform better than the best of all conventional versions significantly as the difference here is 0.875.

6. CONCLUSION

In this work, a methodology is proposed to classify (predict classes from the attribute values) with better accuracy using C4.5, PRISM and RISE algorithms. It is observed that it yields better accuracy than the classical versions of C4.5, PRISM and RISE for all the 19 data sets amongst 20, except for Credit data set. And for Hypothyroid data set the accuracy for the best of both the methodologies are equal. Performing Friedman test and Nemenyi test it is clearly visible that OE²-C4.5 and OE²-RISE performs better than the conventional versions of C4.5, PRISM and RISE significantly. OE²-PRISM performs better than the classical PRISM, but fails to outperform the conventional versions of all the three algorithms significantly. Conventional PRISM, for the data sets considered, actually failed to outperform conventional C4.5 and RISE. Hence the performance of OE²-PRISM was not very impressive. However, the proposed methodology helped the individual classifiers to perform better and would be useful for similar weak learners individually or in an ensemble.

ACKNOWLEDGEMENT

Authors would like to thank the Ministry of Electronics & Information Technology, Government of India for providing fund under Visvesvaraya PhD scheme for Electronics & IT to carry out the research work.

REFERENCES

[1] Michie, D. (1986). Technology lecture: The super articulacy phenomenon in the context of software manufacture. In Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 405(1829): 185-212. https://doi.org/10.1098/rspa.1986.0049

[2] Liaw, A., Wiener, M. (2002). Classification and

regression by random forest. R news, 2(3): 18-22, 2002.
[3] Chen, T., Guestrin, C. (2016). Xgboost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785-794. http://dx.doi.org/10.1145/2939672.2939785
[4] Quinlan, J.R. (1993). C4.5: Programs for machine learning. Machine Learning, 16(3): 235-240. https://doi.org/10.1007/BF00993309
[5] Quinlan, J.R. (2014). C4. 5: Programs for Machine Learning. Elsevier.
[6] Cendrowska, J. (1987). Prism: An algorithm for inducing modular rules. International Journal of Man-Machine Studies, 27(4): 349-370. https://doi.org/10.1016/S0020-7373(87)80003-2
[7] Domingos, P. (1994). The RISE system: Conquering without separating. In Tools with Artificial Intelligence Proceedings Sixth International Conference on. IEEE, pp. 704-707. https://doi.org/10.1109/TAI.1994.346421
[8] Quinlan, J.R. (1986). Induction of decision trees. Machine Learning, 1(1): 81-106. https://doi.org/10.1007/BF00116251
[9] Ruggieri, S. (2002). Efficient c4.5 [classification algorithm]. IEEE Transactions on Knowledge and Data Engineering, 14(2): 438-444. https://doi.org/10.1109/69.991727
[10] Aha, D.W., Kibler, D., Albert, M.K. (1991). Instance-based learning algorithms. Machine Learning, 6(1): 37-66. https://doi.org/10.1007/BF00153759
[11] Michie, D. (1983). Inductive rule generation in the context of the fifth generation. In Proceedings of the Second International Machine Learning Workshop, pp. 66-70.
[12] Langley, P., Simon, H.A. (1997). Applications of machine learning and rule induction. Communications of the ACM, 38(11): 54-64. https://doi.org/10.1145/219717.219768
[13] Domingos, P. (1996). Unifying instance-based and rule-based induction. Machine Learning, 24(2): 141-168. https://doi.org/10.1007/BF00058656
[14] Peda Gopi, A., Vejendla, L.N. (2019). Certified node frequency in social network using parallel diffusion methods. Ingénierie des Systèmes d' Information, 24(1): 113-117. https://doi.org/10.18280/isi.240117
[15] De Pina, A.C., Zaverucha, G. (2004). SUNRISE: improving the performance of the RISE algorithm. European Conference on Principles of Data Mining and Knowledge Discovery, pp. 132. https://doi.org/10.1007/978-3-540-30116-5_52
[16] Vejendla, L.N., Peda Gopi, A. (2019). Avoiding interoperability and delay in healthcare monitoring system using block chain technology. Revue d'Intelligence Artificielle, 33(1): 45-48.
[17] Rodriguez, J.D., Perez, A., Lozano, J.A. (2010). Sensitivity analysis of k-fold cross validation in prediction error estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(3): 569-575. https://doi.org/10.1109/TPAMI.2009.187
[18] Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research, 7: 1-30.
[19] Pohlert, T. (2004). The pairwise multiple comparison of mean ranks package (PMCMR). R Package, pp. 2004-2006.
[20] Zimmerman, D.W., Zumbo, B.D. (1993). Relative power of the Wilcoxon test, the Friedman test, and repeated-measures ANOVA on ranks. The Journal of Experimental Education, 62(1): 75-86. https://doi.org/10.1080/00220973.1993.9943832