

# Exception Included, Ordered Rule Induction from the Set of Exemplars (ExIORISE)

Sayan Sikder, Sanjeev Kumar Metya, Rajat Subhra Goswami

**Abstract:** An expert system is one which uses collection of data comprising the knowledge to offer guidance or make inferences. Its work in most cases can be seen as classification which is basically the task of assigning objects to different categories or classes, determined by the properties of those objects. Numerous research works have been and being done to develop efficient knowledge acquisition techniques for expert systems. Some state-of-the-art algorithms are great performers but need extensive learning whereas older rule / decision- tree based algorithms perform pretty well with small data sets. Moreover, co-existence of learners of different levels of expertise and accuracy is believed to be encouraged to achieve a cumulative intelligence just like the human beings have. RISE is one such algorithm that infuses both instance-based learning and rule induction. It proved to be quite efficient handling binary and multi-class classification problems for small data sets in terms of accuracy and cost as well. In this work, features like exclusion of inefficient rules, inclusion of exceptions in the rule set and ordering of the rules using weights beforehand are integrated with the classical RISE algorithm to develop a more efficient classifier system named as ExIORISE. Empirical study shows that ExIORISE outperforms RISE, C4.5 and CN2 significantly.

**Keywords :** Classification, expert system, rule / decision-tree based algorithms, exceptions; inefficient rules, ordering according to weight.

## I. INTRODUCTION

The advent of science and technology and their incorporation in daily life of human beings was not abrupt. The knowledge is inferred after learning from the environment for a prolonged time span. Now the challenge is to make machines do the job for us. The prime goal is now to ensure that these Expert Systems [1,2] should acquire knowledge and form theories previously unknown, and then communicate this knowledge to the people. Thus machine may enhance the global body of scientific and other knowledge, the phenomenon being coined by Donald Michie as "Superarticulacy" [3]. Previously, machine learning was dependent on mono-strategic learning methods. The RISE algorithm, being a multi-strategic learning method [4] has the ability to overcome the limitations of a mono-strategic

learning method. The RISE algorithm was proposed by Pedro Domingos [5,6] which keeps on generalizing instances until a rule is obtained. The rule sets formed and their representation are in the most intelligible form than other popular classification algorithms. It also performs pretty well with a small training set whereas some present day classifiers although being a better per-former needs extensive training [7]. This paper proposes a better accuracy achieving algorithm ExIORISE, based on the classical RISE algorithm and presents the experimental evaluation and analysis of both the algorithms and also other two popular classification algorithms namely C4.5 [8,9,10] and CN2 [11,12] on some real life data sets obtained from the UCI machine learning data repository [13].

## II. BRIEF REVIEW OF RISE

Two different empirical approaches are vastly used in machine learning namely instance-based learning [14] and rule induction [15,16]. Both of them have their own limitations. RISE (Rule Induction from the Set of Exemplars) is a multi-strategy learning method as it combines features of both the aspects to overcome some of the limitations. The algorithm selects an instance from the training set and sees 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 each rule generalization is given by:

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

where  $r$  represents the rule,  $p(r)$  represents the number of instances covered by the rule whose consequences are identical to the rule's,  $S$  is the sample size, and  $\eta \in [0; 1]$  represents the noise tolerance coefficient [5]. The sum of the heuristic values of each rule in a rule set is considered as the heuristic value of the rule set and it is represented by  $H(.)$ .

The reason why  $\eta$  is introduced is that in noisier domain rules should be given chance to cover a greater share of negative instances. However in domains with negligible noise this share could be considered as 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.

Revised Manuscript Received on December 12, 2019.

\* Correspondence Author

**Sayan Sikder\***, Department of Computer Science and Engineering, National Institute of Technology Arunachal Pradesh, Yupia, India. Email: sayan7003@gmail.com

**Sanjeev Kumar Metya**, Department of Electronics and Communication Engineering, National Institute of Technology Arunachal Pradesh, Yupia, India. Email: smetya@gmail.com

**Rajat Subhra Goswami**, Department of Computer Science and Engineering, National Institute of Technology Arunachal Pradesh, Yupia, India. Email: rajat.nitap@gmail.com

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 [17], Unifying Instance-Based and Rule-Based Induction [6], Using partitioning to speed up specific-to-general rule induction [17] and SUNRISE [18] to improve the performance of the RISE Algorithm.

### A. The RISE Algorithm

#### Algorithm 1: RISE algorithm

```

1: Procedure: RISE (D is the training set)
2:   Let RD be ED.
3:   Calculate H(RD).
4:   Repeat
5:   For each rule in RD,
6:   Let R be the immediate version of the rule,
7:   For each one of the R's antecedents F,
8:   Try generalizing F, and
9:   And calculate resulting H(RD').
10:  Select the A whose generalization results in greatest
    H(RD')
11:  If  $H(RD') \geq H(RD)$ ,
12:  Then generalize A,
13:  If R is equivalent to any other rule in RD,
14:  Then remove R from RD.
15:  Substitute H(RD) by max. H(RD'),
16:  Until no increase in H is found.
17:  Return RD.
18: End procedure
    
```

Furry	Age	Size	Class
yes	old	large	lion
no	young	large	Not lion
Yes	young	medium	lion
Yes	old	Small	Not lion
Yes	young	small	Not lion
yes	young	large	lion
no	young	small	Not lion
no	old	medium	Not lion

### B. Solved Example

Table- I: Training Example

Let T be the training example set which is documented as Table 1. Considering for a noiseless case where is 0,  $h(r)$  can be written as  $h(r) = p(r) - S. n(r)$ .

Generalization 1(no generalization of the antecedents done at all):

yes, old, large, lion

Here  $h(r) = 1 - (8 \times 0) = 1$

Generalization 2:

\*, old, large, lion

Here  $h(r) = 1 - (8 \times 0) = 1$

Generalization 3:

yes, \*, large, lion

Here  $h(r) = 2 - (8 \times 0) = 2$

Generalization 4:

yes, old, \*, lion

Here  $h(r) = 1 - (8 \times 1) = -7$

Generalization 5:

\*, \*, large, lion

Here  $h(r) = 1 - (8 \times 1) = -7$

Generalization 6:

\*, old, \*, lion

Here  $h(r) = 1 - (8 \times 2) = -15$

Generalization 7:

yes, \*, \*, lion

Here  $h(r) = 3 - (8 \times 2) = -13$

From the above generalization it can easily be seen that the Generalization 3 gives the greatest value for  $h(r)$ . Hence yes, \*, large, lion is the first rule obtained. Similarly the other rules are obtained by repeating the process until all the instances are covered. For this particular example of the training set, the rules will be as follows:

1. yes, \*, large, lion
2. no, \*, \*, not lion
3. yes, \*, medium, lion
4. \*, \*, small, not lion

## III. PROPOSED ALGORITHM FOR BETTER ACCURACY

In this work a new algorithm Exclusion, Inclusion and Ordering of Rule Induction from the Set of Exemplars (ExIORISE) is proposed to increase the accuracy of classification with respect to the classical RISE. Features like exclusion of inefficient rules, detection and addition of exceptions to the rule set, ordering of rules beforehand using weighted voting are integrated to the classical RISE algorithm. These features, the inclusion of exceptions being the most effective one to increase the overall accuracies, are discussed in the following sub-sections.

### A. Exclusion of inefficient rules

The rules that are generated may often misclassify in more occasions than correctly classifying.

These rules can be eliminated for obtaining better accuracy in classification. 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. This feature, when teamed with ordering by weights does not have a significant effect on the accuracy. This is due to the fact that the probability of an instance being classified by the inefficient rules (if not excluded), would be very less and even zero as these rules score low in weight and hence would have been positioned at the end of the rule set. However exclusion of these rules beforehand in the beginning of the methodology helps to get rid of unnecessary computational costs.

### B. Detection and addition of exceptions to the rule-set

This section introduces an idea of identifying and treating exceptions already present in the training data. These exceptions are isolated from the training data and employed as rules, which ensures that the exceptions in the test set don't fall victim to misclassification.

As the training data induces the resultant rule set there would be certain instances in the training data itself, which may differ from a rule in the rule set. It means the obtained class using the rule which covers the instance contradicts with the actual class resulting to a misclassification.

So that certain instance of the example set could either be a noise or be an exception. If it contradicts with any other instance or more, then it is being seen as a noise or else as an exception.

**Table-II: Training Example**

A	B	C	D	Class
4	2	3	3	2
4	3	2	2	1
4	3	4	2	2
4	3	4	2	2
4	3	4	2	1
3	3	2	2	2

**Table- III: Test samples covered by rules**

A	B	C	D	Class	hit/miss	Classification status
4	2	3	3	2	miss	Nil
4	3	2	2	1	hit	wrong
4	3	4	2	2	hit	right
4	3	4	2	2	hit	right
4	3	4	2	1	hit	wrong
3	3	2	2	2	hit	right

**Table- IV: Misclassified Instances**

A	B	C	D	Class
4	3	2	2	1
4	3	4	2	1

An example is shown here for further clarity. Let the training

data be as follows as shown in Table-II. Let the rule be: \*, 3, \*, 2, 2 and let the test set be identical to the training set. In trial of classification, the result obtained is documented in Table-III. The instances which are hit by the rule but misclassified are documented in Table-IV. Between these two instances, the later one is conflicting with the fourth instance of the example set. Henceforth, it is considered as a noise and the first one is considered as an exception.

### C. Ordering of Rules Using Weights

We have already eliminated those rules which did not satisfy the threshold. And the threshold is 50%, so there will be rules in the rule set with accuracies varying from 50% to 100%, and some of these rules can predict wrong classes for some instances. To minimize the error, weighted voting [19] is used in RISE whenever an instance in the test set is covered by more than one rule. In our case we have used the following metric which is as follows:

$$\frac{\text{Correct classifications} - \text{misclassifications}}{\text{hits}} \quad (2)$$

This metric is calculated for all the rules (not the exceptions), and the rules are ordered according to the values of metric from highest to lowest. Let 2, 1, 1, 2 be an instance. Table-V contains such two rules, which predicts different classes for this particular instance. To avoid a misclassification here and to increase overall accuracy this measure helps. It may not completely solve the problem but the overall accuracy will be trivially increased.

**Table-V. Rules differing in prediction**

A	B	C	D	Class
*	*	1	2	1
*	1	1	*	2

### D. General Methodology

The process initiates with a data set as the input. First of all n fold cross validation [20, 21] is performed where n = 10, hence as results are obtained 10 sample data sets and 10 respective sample test sets. Then 10 separate rule sets from 10 sample data sets are formed by deployment of each of the four algorithms C4.5, CN2, RISE and ExIORISE respectively. By using the respective rule sets, classes are predicted for each instance in all the 10 sample test sets. Then the accuracy of classification and the standard deviation are calculated for each of the 10 cases. The accuracy of classification is the ratio of the number of correct classification to the total number of instances in the test set.

This whole procedure is repeated for 5 times. Hence for each data set classified by any one of the four algorithms, there will be 50 (10×5) different accuracies and standard deviations. The average of the accuracies and the respective standard deviations are considered in this paper.

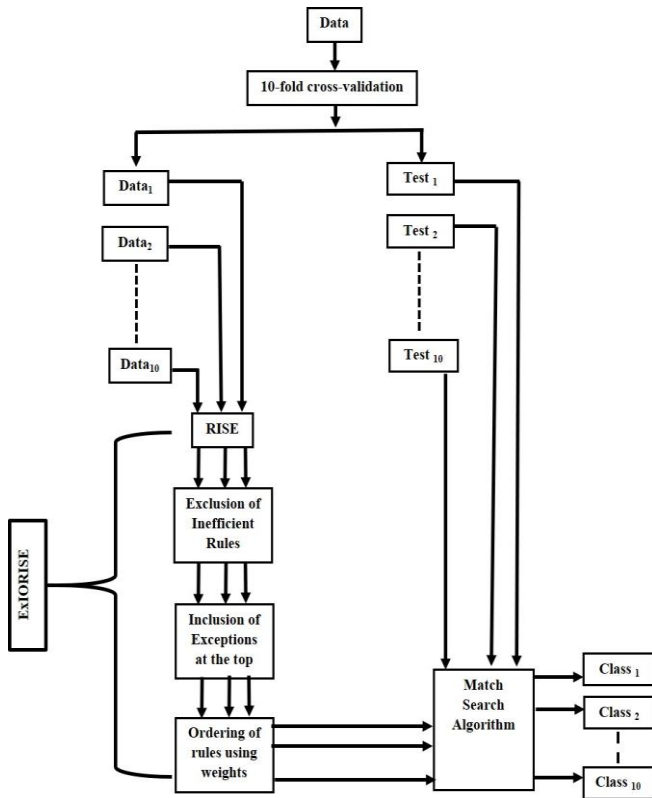


Fig. 1 Skeletal architecture of the methodology

#### E. The ExIORISE Algorithm

Algorithm 2: ExIORISE algorithm

- 1: procedure ExIORISE(datafile; namefile)
- 2:   while  $a \leq 5$  do
- 3:     10 fold cross-validation to the data file
- 4:     procedure RISE(cross-validated data files)
- 5:     end procedure
- 6:   Read each rule, compare it with every instance for the corresponding data set and when they match, add 1 to the counter<sub>1</sub>, initial value being 0
- 7:   Remove the last elements of each lines of the 100 data sets and their corresponding rule sets
- 8:   Store the last elements of both the files in a temporary file
- 9:   Read each rule, compare it with every instance for the corresponding data set and when they match, add 1 to the counter<sub>2</sub>, initial value being 0
- 10:   Check the corresponding last elements and when they do not match, store the instance in misclassified
- 11:   Calculate the differences between the values of both the counters and store in order in difference<sub>1</sub>
- 12:   Calculate the differences between the values of the counter<sub>2</sub> and the difference<sub>1</sub> and store in order in difference<sub>2</sub>
- 13:   For all the negative values of difference<sub>2</sub> remove the corresponding rule

- 14:   Reorder the rules in descending order the corresponding values in counter<sub>1</sub>
- 15:   Compare the instances minus the last elements in misclassified to the instances of the data sets minus the last elements
- 16:   If there is a perfect match, check whether the corresponding last elements match
- 17:   If they do match, store the instances in exception and append in the top of the rule sets
- 18:   Classify the test sets using corresponding rule sets and calculate accuracy and standard deviation in each case
- 19:   end while
- 20:   Calculate mean of accuracy, standard deviation
- 21: End procedure

#### IV. DATA SETS

In this work, 20 real life data sets are used to understand whether the proposed algorithm helps to increase the classification accuracy or not. The data sets are taken from different domains such as medical, biological, marketing etc and all are obtained from UCI machine learning data repository. All the data sets vary in size. The one having minimum attributes (Soybean) contains 47 examples and the one having maximum attributes (Thyroid disease) contains 3163 examples. They are also having varying number of attributes with the minimum one (Iris) having 4 attributes and maximum one (Audiology) having 69 attributes. Out of these 20 data sets, as many as 11 data sets (Annealing, Audiology, Breast cancer, Cleveland, Credit, Heart disease, Hepatitis, Horse colic, Thyroid disease, Voting, Wisconsin) contain missing valued attribute(s). In Table 6 information about the data sets used, is shown. “Exs.”, “Att.”, “Num.”, “Class”, “Missing” columns in the table stores the number of examples, the number of attributes, the number of numerical attributes, the number of classes and the information of whether it contains any missing valued attributes or not respectively for each data set.

Table- VI: Data-sets

Data set	Exs.	Att.	Num	Class	Missing
Annealing	798	38	9	5	Yes
Audiology	200	69	0	24	Yes
Breast Cancer	286	9	4	2	Yes
Cleveland	272	13	6	5	Yes
Glass	690	15	6	2	Yes
Heart disease	214	9	9	6	No
Hepatitis	303	13	6	2	Yes
Horse colic	155	19	6	2	Yes
Iris	300	22	7	2	Yes
Liver disorder	150	4	4	3	No
Lymphography	345	6	6	2	No
Pima diabetes	148	18	3	4	No
Soybean	768	8	8	2	No



Tae	47	35	0	4	No
Thyroid disease	151	5	1	3	No
Voting	3163	25	7	2	Yes
Wisconsin	435	16	0	2	Yes
Zoology	178	13	13	3	No
Wisconsin	699	9	1	2	Yes
Zoology	101	16	1	7	No

## V. RESULT AND DISCUSSION

RISE outperformed C4.5 and CN2 for two third of the data sets in UCI machine learning data repository. Table-VII constitutes the average accuracies and the standard deviations of C4.5, CN2, RISE and ExIORITY respectively for 20 real life data sets obtained from the repository. For all the 20 data sets (except for Iris and Soybean, where accuracy for both the algorithms are equal), ExIORITY performs with better accuracy than classical RISE, the average increase in accuracy being 7.19% rounded off to 2 decimal places. C4.5 and CN2 algorithms perform better than RISE in 13 of the 20 considered data sets. For 6 out of these 13 data sets ExIORITY comes up with accuracies which are very close to that obtained by the winner algorithm. And for the other 7 (Breast cancer, Hepatitis, Horse colic, Liver disorder, Pima diabetes, Tae, Wisconsin) out of the 13 data sets, ExIORITY even yields the best accuracies amongst all the algorithms used in this work.

**Table- VII: Comparative analysis of Algorithms**

Data Set	C4.5	CN2	RISE	ExIORITY
Annealing	93.6 ± 1.6	82.1 ± 6.6	97.4 ± 1.6	99.0 ± 0.8
Audiology	70.9 ± 6.0	69.4 ± 6.6	76.9 ± 5.3	78.2 ± 4.58
Breast cancer	68.6 ± 5.4	68.2 ± 5.7	67.7 ± 4.7	73.4 ± 7.6
Cleveland	50.4 ± 5.7	57.0 ± 7.0	41.6 ± 8.4	56.3 ± 11.4
Credit	84.6 ± 2.4	82.2 ± 2.5	78.6 ± 2.7	82.0 ± 2.3
Glass	64.4 ± 7.7	64.2 ± 6.0	67.3 ± 9.2	76.2 ± 7.4
Heart disease	76.2 ± 4.3	78.5 ± 3.6	60.4 ± 12.2	75.6 ± 4.7
Hepatitis	78.4 ± 5.5	80.6 ± 4.6	73.5 ± 9.6	83.9 ± 6.4
Horse colic	81.0 ± 3.8	83.0 ± 3.6	82.0 ± 3.0	83.7 ± 4.7
Iris	93.8 ± 2.9	93.0 ± 3.0	98.0 ± 3.1	98.0 ± 3.1
Liver Disorder	64.1 ± 4.3	65.2 ± 4.2	62.3 ± 4.8	73.1 ± 8.4
Lymphography	75.0 ± 4.6	77.6 ± 5.5	78.6 ± 6.3	79.9 ± 6.8
Pima diabetes	73.7 ± 3.0	73.7 ± 2.4	60.9 ± 6.1	75.7 ± 4.7
Soybean	96.0 ± 6.5	97.1 ± 5.4	100.0 ± 0.0	100.0 ± 0.0
Tae	51.7 ±	57.6 ±	53.6 ±	63.3 ±

	9.8	14.7	15.2	9.7
Thyroid disease	98.7 ± 0.7	98.0 ± 0.7	92.5 ± 5.3	96.3 ± 2.8
Voting	95.4 ± 1.6	95.3 ± 1.5	95.2 ± 1.5	95.3 ± 1.6
Wine	91.3 ± 4.9	91.5 ± 4.6	93.8 ± 4.8	97.8 ± 2.8
Wisconsin	94.2 ± 2.1	95.3 ± 2.6	92.3 ± 2.9	95.6 ± 2.2
Zoology	90.8 ± 5.4	91.9 ± 5.2	83.1 ± 9.1	86.1 ± 10.2

However it still cannot be said that ExIORITY has significantly outperformed the classical algorithms. That is why Friedman test [22, 23] is performed. ranked for each data set individually, the one with the highest accuracy being ranked 1, the one with the second highest accuracy being ranked 2 and so on, and in case of ties (as in Credit, Glass, Iris, Pima diabetes, Soybean, Voting). The average ranks for each algorithm are then calculated which are 2.675 for C4.5, 2.625 for CN2, 3.15 for RISE and 1.55 for ExIORITY. Using these values the Friedman statistic is calculated which is given by [22]:

$$X^2F = \frac{12N}{k(k+1)} \left[ \sum_j R_j^2 - \frac{k(k+1)^2}{4} \right] \quad (3)$$

where the quantity of data sets is represented by N, the number of methods / algorithms is represented by k and is the sum of the squares of the mean ranks of the individual methods / algorithms. The value of The Friedman statistic is found to be 16.455 in this particular scenario. A more useful statistic could be derived from the Friedman statistic which is given by [22]:

$$F_F = \frac{(N-1)}{N(k-1) - X_F^2} \quad (4)$$

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 7.18 rounded off to two decimal places. As the quantity of algorithms is 4 and the quantity of data sets is 24 we can say  $F_F$  is distributed according to the F distribution with 41 = 3 and (4 - 1)(20 - 1) = 57 degrees of freedom. Hence the critical value of F(3, 57) for  $\alpha = 0.05$  is 2.77 rounded off to two decimal places which is less than  $F_F$ , so the null-hypothesis of the algorithms performing with no significant differences is rejected. As no algorithm is singled out, the Nemenyi test [24] is done for pair wise comparisons. The critical value is 2.569 for 4 algorithms considering  $\alpha = 0.05$  and the corresponding CD (critical difference) is found to be 1.05 rounded off to two decimal places using [22,24]:

$$CD = q_{\alpha} \sqrt{\frac{k(k+1)}{6N}} \quad (5)$$

Since the differences between the mean ranks of ExIORITY and the other three algorithms (ExIORITY - C4.5 = 1.125, ExIORITY - CN2 = 1.075, ExIORITY - RISE = 1.6) are all more than the obtained CD, we can affirm that ExIORITY is significantly performing better than those.

## VI. CONCLUSION

In this paper, a new algorithm ExIORISE is proposed to classify (predict classes from the attribute values) with better accuracy. It is observed that ExIORISE yields better accuracy than classical RISE for all the 20 data sets except for Iris and Soybean data sets where accuracy for both the algorithms are exactly the same. The average increases in accuracy of ExIORISE over RISE and of the same over the best of C4.5, CN2 and RISE are found to be 7.19% and 2.87% respectively, rounded off to 2 decimal places. In some data sets taken for experimentation in this work, the accuracies given by C4.5 and CN2 algorithms are better than that obtained by the RISE algorithm. In case of those data sets ExIORISE yields accuracies which are very close to that obtained by the maximum accuracy yielding algorithm and for some of such data sets ExIORISE even yields the best accuracies amongst all the algorithms used in this work. Thus it is clearly visible that ExIORISE can classify with more accuracy than the classical ones in most cases.

## ACKNOWLEDGMENT

Authors whole-heartedly 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, "Expert systems in the micro-electronic age", Edinburgh University Press Edinburgh, 1979.
2. Giarratano JC, Riley G, "Expert systems", PWS Publishing Co., 1998.
3. Michie D, "Technology lecture. the superarticulacy phenomenon in the context of software manufacture" In: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, The Royal Society*, vol 405, pp. 185-212, 1986.
4. Michalski RS, Michalski RS, Tecuci G, Carbonell JG, Mitchell TM "Machine learning: A multistrategy approach", Morgan Kaufmann, vol 4, 1994.
5. Domingos P, "The RISE system: Conquering without separating .In: Tools with Artificial Intelligence", *Proceedings Sixth International Conference on Tools with Artificial Intelligence*, IEEE, pp. 704-707, 1994.
6. Domingos P, "Rule induction and instance-based learning: A unified approach", In: *Proc. IJCAI-95*, pp. 1226-1232, 1995.
7. Chen T, Guestrin C (2016), "Xgboost: A scalable tree boosting system", In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, ACM, pp. 785-794, 2016.
8. Quinlan JR, "Generating production rules from decision trees" In: *ijcai*, vol 87, pp. 304-307, 1987.
9. Quinlan JR "C4. 5: programs for machine learning", vol. 1, Morgan kaufmann, 1993.
10. Quinlan JR "C4. 5: programs for machine learning", Elsevier, 2014.
11. Clark P, Niblett T, "The CN2 induction algorithm. Machine learning", pp. 261-283, 1989.
12. Clark P, Boswell R, "Rule induction with CN2: Some recent improvements", In: *European Working Session on Learning*, Springer, pp. 151-163, 1991.
13. Asuncion A, Newman D, "UCI machine learning repository", 2007.
14. Aha DW, Kibler D, Albert MK, "Instance-based learning algorithms", *Machine learning*, pp.37-66, 1991.
15. Michie D, "Inductive rule generation in the context of the fifth generation", In: *Proceedings of the Second International Machine Learning Workshop*, 1983.
16. Langley P, Simon HA, "Applications of machine learning and rule induction", *Communications of the ACM*, pp. 54-64, 1995.
17. Domingos P, "Using partitioning to speed up specific-to-general rule induction", In: *Proceedings of the AAAI-96 Workshop on Integrating Multiple Learned Models*, Citeseer, pp. 29-34, 1996.
18. De Pina AC, Zaverucha G, "SUNRISE: Improving the Performance of the RISE Algorithm", *Advances in Inductive Rule Learning* p 132, 2004.
19. Dietterich TG, "Ensemble methods in machine learning", *Multiple classifier systems*, pp. 1-15, 2000.
20. Rodriguez JD, Perez A, Lozano JA, Sensitivity analysis of k-fold cross validation in prediction error estimation, *IEEE transactions on pattern analysis and machine intelligence*, pp. 569-575, 2010.
21. Kohavi R, "A study of cross-validation and bootstrap for accuracy estimation and model selection", In: *Ijcai*, Stanford, CA, vol. 14, pp. 1137-1145, 1995.
22. Demsar J, "Statistical comparisons of classifiers over multiple datasets", *Journal of Machine learning research* 7(Jan):1-30, 2006.
23. Zimmerman DW, Zumbo BD, "Relative power of the Wilcoxon test, the Friedman test, and repeated-measures ANOVA on ranks", *The Journal of Experimental Education*, pp.75-86, 1993.
24. Pohlert T, "The pairwise multiple comparison of mean ranks package (PMCMR)", *R package* pp. 2004-2006, 2014.



**Sayan Sikder** received B.Tech degree in Computer Science and Engineering from Haldia Institute of Technology (HIT) in 2012 and M.Tech degree in Computer Science and Technology from I.I.E.S.T Shibpur in 2015. He is currently pursuing Ph.D degree in Computer Science and Engineering at National Institute of Technology, Arunachal Pradesh. His areas of interest include Machine Learning, Pattern recognition and Artificial intelligence.



**Dr. Sanjeev Kumar Metya** is serving the Department of Electronics and Communication Engineering at National Institute of Technology Arunachal Pradesh, India. He obtained his B.Tech degree in Electronics and Communication Engineering from Haldia Institute of Technology (HIT) in 2005, M.Tech degree in Electronics and Communication Engineering from KIIT-University in 2007 and PhD from Malaviya National Institute of Technology (MNIT) Jaipur. His area of interest includes Optical Communication, Photonic Crystals, Nonlinear Optoelectronics, All-optical systems numerical modeling and Machine Learning.



**Dr. Rajat Subhra Goswami** is serving the Department of Computer Science and Engineering at National Institute of Technology Arunachal Pradesh, India. He obtained his B.Tech degree in Information Technology from West Bengal University of Technology (WBUT) in 2005, M.Tech degree in Multimedia Development from Jadavpur University, West Bengal in 2009 and PhD from National Institute of Technology Arunachal Pradesh, India in 2015. His area of interest includes Information Security, Cryptography, Image Processing, Big Data, Network Traffic Classification and Machine Learning.