

# Inductive Learning for Continuous Classes and the Effect of RULES Family

Hebah ElGibreen and Mehmet Sabih Aksoy

**Abstract**—Due to the digital world explosion and its appearance in everyday life, predicting numeric actions became necessary in machine learning. However, due to growth of interest in understanding how problems can be solved, simple prediction algorithms are more helpful than the difficult statistical approaches. Covering algorithm can be used to accomplish difficult problems using simple rules or trees. One family called RULES was found to be very interesting with appealing properties. It is one of the most flexible and simplest families with high learning rate. Nevertheless, even though RULES is actively improving but it is surprisingly neglected, especially with numerical datasets. Thus, the purpose of this paper is to extend the literature and investigate the problems of continuous classes in RULES and other inductive learning families. A theoretical analysis is conducted to show the effect of numerical actions and how it is still an open research area. An empirical evaluation is also provided to prove how RULES family can be used as the base of further improvement. Accordingly, this paper can be used as a reference by researchers to know what research area is still not covered and need further refinement in inductive learning.

**Index Terms**—Continuous classes, covering algorithms, decision tree, inductive learning, RULES family.

## I. INTRODUCTION

In the past few years, digital devices and autonomous machines started to interest both researchers and regular people. Everyone wants to have a self control vacuum cleaner, self activating video recorder, and many other things that require the machine to manage numeric or continuous actions. For that reason, Machine Learning (ML) gained more attention from the societies of artificial intelligent. This domain was developed to create autonomous agents who can make the machine act similarly to humans. ML encountered different learning methods depending on the problems that are needed to be solved. One area of ML, where the agent can learn how to train itself in order to make future prediction, is called Inductive Learning (IL). It has been defined in [1] as “the ability of an agent (like an algorithm) to improve its own performance based on past experience.” In this type of learning the agent is usually provided by previous information as input in order to gain some descriptive knowledge. Hence, it is a supervised learning that works as a data analysis tool and use the knowledge gained through

training to produce general conclusion and identify new objects.

In general, current IL algorithms have been divided into two types: Decision tree (DT) and Covering Algorithm (CA) [2]. Each type has its own purpose, strength, and weakness. DT algorithms discover rules using decision tree. This tree can be used later to represent the rules [3]. CA, however, directly induces rules from the training set based on the concept of separate-and-conquer. DT attracted a lot of attention in the past few years because of the powerful tree structure. However, CA started to become more interesting because of its direct rule representation. Thus, several families have been born under the umbrella of CA.

One important family, that is considered as one of the simplest and most flexible CA families, is called RULE Extraction System (RULES) [4]. RULES family was born in 1995 by Pham and Aksoy [5] to directly induce good and simple rules in simple manner. It discovers inconsistent rules to allow the coverage of some negative examples, in order to handle noisy data, reduce over-fitting problem, and increase the flexibility of rule induction. In addition, RULES does not remove covered examples but, instead, it marks them as covered. This way, repeating the discovery of the same rule is prevented while the accuracy and generality of new rules is preserved. Consequently, RULES can resist fragmentation and small combination problems; i.e. avoid data reduction during the learning process and coverage of small training examples with high error rate.

Nevertheless, it was found that RULES family is surprisingly neglected. Although researchers were interested in CA and conducted different surveys and empirical studies but RULES was always forgotten. Thus, in [6] a theoretical analysis and empirical study were conducted to compare RULES with other conventional IL methods. RULES properties were compared to DT to find that direct induction of rules is more appealing than the use of trees, especially in large and complex data. Moreover, the characteristics of CA families were analyzed and compared to find that RULES can be considered as one of the most flexible one. It was concluded that RULES is an interesting family, and more research should be done to improve its performance. However, the test and comparison was conducted only over discrete datasets while numerical values were neglected. Therefore, the study was extended in [7] to measure the performance of IL with continuous attributes. It was found the CA is still lacking when it comes to continuous attributes but RULES family can be used for further improvement. Nonetheless, in practical and real-life applications, classifiers need to predict numeric actions. Thus, it is important to extend the studies to measure the performance of RULES and

Manuscript received March 30, 2014; revised June 12, 2014.

Hebah ElGibreen is with Information Technology Department, College of Computer and Information Sciences, King Saud University, Saudi Arabia (e-mail: hjibreen@ksu.edu.sa).

Mehmet Sabih Aksoy is with Information System Department, College of Computer and Information Sciences, King Saud University, Saudi Arabia (e-mail: msaksoy@ksu.edu.sa).

other conventional IL methods over data with numerical actions.

Accordingly, the contribution of this paper is to compare RULES with other conventional IL methods and show their effect over datasets with continuous classes in order to emphasize on what is still missing in this domain. Moreover, a practical study will also be conducted to analysis and compare the performance of RULES with other classical DT and CA to show how RULES can be used as the base of further improvement and how it surpass the other families. Based on that, it will be possible to prove the importance of continuous classes' problem in CA, and what part of this problem is still an open research area. Hence, this paper is an extended version of the work conducted in [6], [7] to be used as a reference by researchers to know what area is still not covered in IL with continuous classes.

This paper is organized as follows. First, RULES family and the related work will be discussed to prove the novelty of this paper. After that, continuous classes' problem will be theoretically analyzed. Then, an empirical test will be conducted to test RULES with other DT and CA over continuous classes to show its strength and weakness, and emphasize on the open research questions. Finally, the paper will be concluded and future work will also be presented.

## II. RELATED WORK

In order to understand the purpose of this paper and emphasize on its novelty, it is important to understand the targeted family and discuss its related work. In general, RULES is a CA family that directly induces one rule at a time based on a seed example. It applies specialization to find the best rule. It does not require finding of consistent rules and allows the best rule to cover some negative examples in order to handle noisy data, reduce over-fitting problem, and increase the flexibility of rule induction. Moreover, the examples that are positively covered by the discovered rules are marked only but not removed. This way, repeating the discovery of the same rule is prevented while accuracy and generality of new rules are preserved.

Nevertheless, several versions have been proposed in this family, starting from RULES-1 [5] to RULES-8 [8] and other versions as RULES-TL [9] and RULES-IT [10]. Every version was developed to serve a certain purpose and solve one problem of IL. However, only one version was proposed to deal with continuous classes' problem. This version was called RULES-F [11], and developed to deal with both continuous classes and attributes. In RULES-F, fuzzy set theory was applied to handle continuous classes and it was extended in [12] to RULES-F+, to improve the performance using a new rule space representation scheme. Ultimately, even though each version in RULES family has its own properties, but most of these versions have some common characteristics. In [6], these characteristics were discussed to find that RULES is very appealing when compared with classical CA and DT algorithms. However, the case where datasets contains continuous classes was not covered.

In general, in the field of CA, several surveys and empirical studies have been conducted through the years. In

[13], different CA methods were tested over several pruning techniques in order to show the effect of pre and post pruning techniques. After that, another survey was conducted in [14], to analyze the characteristics of CA. This survey was very interesting and covered a wide range of CA to compare the algorithms based on three biases, namely: language, search, and pruning. However, RULES was neglected, and empirical tests were not conducted.

Moreover, in [15], another study was proposed to explain and test different classification techniques. In this work, different ML techniques have been explained separately and then compared together in an empirical study. Alternatively, in [1], a study was conducted to compare DT algorithms with CA and hybrid IL algorithms. Nevertheless, it only tested DataSqueezer with C5 and CLIP4 empirically without any theoretical comparison. In addition, in [16], an experimental evaluation of different CA was conducted to test its simplicity, but these methods were explained separately, properties and characteristics were not compared, and RULES family was not considered.

Alternatively, in [4] the author focused on RULES family and conducted a survey that explained its versions. Nevertheless, this survey was only concerned with RULES family and theoretically explained version one to five only. Hence, no empirical study was conducted and other families of CA were not considered. On the other hand, in [17], different supervised learning techniques were used to test the performance of its algorithm with different discretization techniques. This work is a discretization techniques survey to identify its taxonomy and analyze its performance. However, RULES family was not included in the study.

Finally, two preceding versions of this paper were conducted in [6] and [7]. The first version tested RULES family with other conventional families of DT and CA. This study showed the importance of RULES and how it can surpass other families in the literature. However, only discrete datasets were considered and the effect of RULES over numerical values was not covered. On the other hand, the second version covered the problem of continuous attributes. It compared RULES with other conventional families of DT and CA in continuous attributes problem. The gaps of such problem were identified and proven in the empirical study. It was also found that RULES family can be used as the base of further improvement. Nevertheless, continuous classes were neglected and this problem is not yet covered in the studies.

Accordingly, it can be noticed from all the studies discussed before that CA and DT conventional algorithms, along with RULES, have not been studied together and the surveys were either focused on the theoretical or the empirical part of the study. Although the preceding studies showed the effect of RULES and other IL families but continuous classes' problem was not considered. Hence, an extended version is needed to consider RULES and compare it with other IL methods in datasets with continuous classes. Thus, this paper will theoretically analyze RULES and other IL methods over numerical actions and discuss its shortcoming and gaps. Then, they will also be empirically tested over datasets with continuous classes to emphasis on the shortcoming and identify the importance of RULES.

### III. THEORETICAL STUDY

In rule induction, the training set includes two types of classes: discrete and continuous. The discrete values contain categorical data and have a finite number of values, such as “High, Low, Medium”. The continuous values, however, contain numerical values with an infinite or very large space, such as “1, 2, 3.4, 5.3.” Original algorithms of CA usually assume that the class labels have a discrete and small number of values. Nevertheless, in practice and real-life applications, class labels contain continuous values, and the classifier needs to be based on these numeric labels.

In general, the problem of continuous classes can be defined as the ability to induce new knowledge about numeric conclusions. Hence, regardless of the type of the attributes, the class label is usually numeric. Nevertheless, due to the importance of numeric actions problem, several methods and algorithms were proposed in the literature of IL. However, these methods are still lacking and several gaps needs to be identified. Moreover, in most research, continuous classes’ problem was recognized as regression problem but in recent years researchers started to differently deal with continuous labels without the need for regression. Thus, calling the problem of continuous classes as a regression problem is currently inappropriate and recent researches recognize it as “continuous classes” problem. Yet, whether the problem is called regression or not, several methods were developed to solve it, as follows.

#### A. Regression

In IL, regression has been defined as the problem of developing an approximation function that uses the values of the attributes to predict the class label that has numeric numbers [18]. Basically, the concept of regression has been applied at the beginning, in IL, to deal with continuous classes directly during the rule induction. However, these methods were mostly applied over DT algorithms because a tree structure was required. In general, it can be divided into two types of trees: regression tree and genetic programming. Regression tree methods, such as DecisionStump [19], M5P [20], M5Rules [20], REPTree [21], CARTs [22], CHAID [23], and QUEST [24], breaks down the decision tree into homogeneous regions while the leaf nodes are actually a regression formula rather than a class. On the other hand, Genetic programming methods, such as TARGET [25], GPMCC [26], and E-Motion [27], uses evolutionary algorithm to grow trees with regression model. Nevertheless, although these two types of methods were proposed to handle continuous classes online, but they encountered several problems that made their application over the classification problem very difficult. This is because DT methods usually generate complex trees, so if the regression model is also included then inducing rules from such trees might become NP-complete problem [27].

Moreover, it was stated in [28] that regression models are incomprehensible and very difficult to understand. It is very difficult to predict and discover, and the final result is usually given without explanation. In addition, regression models usually give the exact number which is risky in many domains because the resulted model will be highly affected by noise. Hence, regression models are unable to provide

good point estimates with noisy data or domains that need fault tolerance [29]. Finally, as discussed in [30], regression trees would result in low accuracy, very sensitive to data changes, and require more data than pre-processing discretization. Thus, a new approach has been developed to handle continuous classes without regression, which is Regression via Classification.

#### B. Regression via Classification

Regression via classification (RvC) deals with the regression problem through classification models. This approach was originally used by Weiss & Indurkha [19], [20] in their regression system to conclude that dealing with continuous classes as discrete ones can give excellent results, even better than some regression methods. It is also possible to use such approach in classification systems instead of only regression systems. It includes uncertainty to decrease the specialization level and make the algorithm more general and more resistant to noise. Hence, RvC can take advantage of classification uncertainty to deal with the regression problems. Thus, in [31], three main advantages of RvC have been proposed. First, the use of a classification model produces a clear model represented as rules, which provides better understanding. It produces the class labels as thresholds instead of exact values. Finally, the use of such thresholds increases the uncertainty level and makes the model less specialized and more resistant to faults and noise. Therefore, researchers started to exploit classification in order to solve the regression problem; or what is now called continuous classes’ problem.

Earlier methods of RvC, such as ConjunctiveRule [32] and ZeroR [33], dealt with the problem of continuous classes by applying a certain discretization technique. Consequently, RvC methods can be characterized based on the concept of discretization, including offline and online discretization.

##### 1) Offline discretization

Offline discretization methods deal with the problem of continuous classes by applying two steps [18]. First, the numeric target class is discretized. Second, the output class of the discovered knowledge is transformed back to numeric using a certain technique, such as median. Consequently, methods that belong to this approach apply any discretization technique over the classes before applying the rule induction. It is unsupervised and decides on the intervals without having any input that could direct the discretization. Nevertheless, dealing with continuous classes is not an easy task. Therefore, several attempts have been made to improve RvC approach. In [34] three different discretization methods were integrated with two inductive learning methods. Specifically, Equally Probable Intervals, Equal Width intervals, and K-means clustering were integrated with C4.5 and CN2. The purpose of that paper was to prove that RvC is better than regression approach.

After that, Torgo and Gama [35], proposed a system called (RECLA) to be used as a pre-processing tool that can be integrated with any classification system to deal with continuous classes. Namely, it was tested with C4.5, CN2, and linear discriminant to find that the accuracy of the proposed approach is much better than using regression. Moreover, in [18] and [31], RvC was applied to predict the

number of software defect. The proposed framework was integrated with several classifications and IL methods, including Ripper, PART, and C4.5. In [28], a discretization method called ERD was applied with ensemble learning to handle continuous classes without regression. Furthermore, in [29], [36], RvC was tested with SVM classifier to conclude that RvC approach produce better results than regression approach with less CPU time.

In addition, when it comes to IL, different techniques have also been developed to handle continuous classes offline. These algorithms were basically developed for DT methods. Nevertheless, one attempt to apply offline discretization over CAs using genetic algorithms have been presented in [37]. However, their method needed to sort and test the variance of every attribute, which increased the computation and time overhead. In addition, the result accuracy is highly dependent on the strength of the fitness function which is decided by the user.

Consequently although offline discretization reduces the time of rule induction and can be better than regression approach, but it can seriously affect the rules' quality resulting from the CA [38]. In specific, there is a great tradeoff between the number of intervals and the consistency of the rule. Such that, choosing a small number of split points would indicate that the interval size will be very large. Hence, the resultant rules might not be consistent. Moreover, when the number of split points is large, it will indicate that the interval size is small, which might also affect the number of rules and overspecialize it. As a result, online discretization was also introduced in continuous classes' problem.

#### 2) *Online discretization*

Online discretization was basically designed for DT algorithms, where it must re-discretize all continuous attributes at every node. Thus, it wastes a lot of time and increases the computational complexity. In [30], a dynamic discretization algorithm, called CLC, was developed to discretize the continuous classes online. During the process of tree construction, the algorithm discretized based on the data associated with each node. It was found that this algorithm has better performance than pre-processing discretization but the problem of fixed and none-overlapping intervals still exists. Moreover, it cannot handle any noise that might occur in the data where noisy instances must be removed from the training set before processing it.

Alternatively, when it comes to rule learners continuous classes are usually handled using fuzzy set theory [39], as in WM [40] and FRSBM [41]. Similarly, in RULES-F, fuzzy set theory was also applied to handle continuous classes and reduce the overlapping between the rules. The algorithm used the membership function to convert the continuous class into a fuzzy range. After discovering the best rule, a post-processing technique is applied to transform all the attributes into the membership degree and create fuzzy rule set instead of intervals. Hence, it reduces the coverage space and avoids the possibility of overlapping. However, it was found that RULES-F is highly dependent on the number of member function used, which is based on the intervals given by the user. Hence, the good performance cannot always be guaranteed, and the process is not fully automated. As a

result, another attempt was made in [42] by developing FR3 algorithm. It was developed to handle continuous classes using fuzzy theory. Nevertheless, it was noted that the performance improvement was at the cost of model complexity and the algorithm is still not generalized, and it can only be applied to a specific number of domains.

#### C. *Discussion*

From the work conducted to solve continuous classes' problem and discussed before, it can be concluded that both RULES and other method in CA and DT are still lacking. In general, each approach has some common deficiencies when solving the problem of continuous classes. These deficiencies can be summarized as in the following points.

- 1) Regression approach faced several problems and became difficult to apply in classification. It generates complex trees that can make the induction process NP-complete problem. Its models are incomprehensible and very difficult to understand. It is very difficult to predict and discover, and the final result is usually given without explanation. In addition, regression models are unable to provide good point estimates with noise or domains that need fault tolerance. Regression trees result in low accuracy, very sensitive to data changes, and require more data than offline discretization.
- 2) RvC improve regression and solve its problems, but it still has its own deficiencies.
- 3) In RvC with offline discretization, future cases are not considered, and intervals are fixed in advance. Thus, it can cause major problems on the long run, because values of unseen data do not remain in the same distribution as the training set. In addition, the update of discretized values can cause another problem with incremental learning. It will be difficult to update the interval of older rules, and it is highly anticipated that it would reduce the accuracy of the algorithm.
- 4) In RvC with offline genetic programming, Potgieter and Engelbrecht [26] discussed its difficulties to conclude that these methods are highly dependent on the fitness function quality and stopping condition. It needs to tradeoff between the search cost and finding of good solutions, and usually used for optimization problem.
- 5) In RvC with online discretization, the performance of discretization is improved comparing to offline. However, it still has the problems of fixed intervals, and causes an increase in computation and time complexity.
- 6) In RvC with online fuzzy discretization, the performance of online discretization was improved but the complexity is also increased. Moreover, the member functions need to be defined by the user, which cannot guarantee the performance of the method.

Consequently, even though RULES family showed better properties and characteristics than the conventional families of IL, but it is still lacking with numerical actions. Similarly to the other families, RULES needs further improvement to cover the gaps indicated previously and this type of data is still an open research area. Nevertheless, in order to show how RULES family can be used as the base of further improvement and how it can surpass the other families, empirical evaluations are conducted in the next section.

## IV. EMPIRICAL STUDY

In the previous section, the significant of continuous classes' problem was verified and its gaps were emphasized. However, it is also important to know what RULES family can add in comparison to the other classical families of IL. Thus, this section shows the result of comparing RULES family with other IL algorithms in continuous classes' problem. The experiments were conducted on a PC with Intel®Core™ i7 CPU, 2.67 GHz processes, and 6GB RAM and KEEL tool [43], [44] was used to build the experiments. Moreover, several dataset with different properties were gathered from KEEL repository [44], as illustrated in TABLE I, in order to measure the performance of the algorithms over different situations. The result of the algorithms is also validated using 10-cross-fold validation [45].

TABLE I: DATASETS PROPERTIES

Dataset	#Examples	#Attributes	Class Type
Abalone	4177	8	Integer [1,29]
AutoMPG6	392	5	Real [9.0,46.6]
AutoMPG8	392	5	Real [9.0,46.6]
Baseball	337	16	Integer [109,6100]
Dee	365	6	Real [0.766,5.119]
Diabetes	43	2	Real [3.0,6.6]
ele-1	495	2	Real [80, 7675]
ele-2	1056	4	Real [64.5,8546]
ForestFires	517	12	Real [0.0, 1090.8]
Friedman	1200	5	Real [0.66, 28.59]
Laser	993	4	Real [0.0, 255.0]
MachineCPU	209	6	Integer [6, 1150]
Treasury	1049	16	Real [3.02, 20.76]

The only version of RULES family that deal with continuous classes, two DT, and three rule learner algorithms are included in the study. RULES-F+ is used as one of RULES family versions because it is currently the only version that was developed to deal with continuous classes. M5 and CART are used as DT methods because they are usually used as a benchmark. Finally, in rule learner, M5Rules, WM, and FRSBM algorithms are applied since they are well-known in the literature of regression. Note that the continuous attributes are discretized in this experiment to focus on solving the continuous classes' problem. The method used for discretization is proportional k-interval discretization (PKID) [46]. In addition, two measures were recorded to determine the performance of the tested

algorithms and compare them together, as follows.

- **Root mean square error (RMSE) [47]:** It is used to measure the difference between the actual and the predicted value in order to show the quality of the prediction model. It is one of the mostly used measures to test the result of numeric prediction.
- **Execution Time:** It records the time interval between generating the result of the first and last fold of the dataset in seconds.

Note that it was not possible to record the learning rate because most of the algorithms do not provide their rule set. Instead, they only produce the model and make the prediction without showing how. Thus, it was not possible to compare the algorithms learning rate or complexity. Moreover, it was found that FRSBM could not handle datasets with large number of examples, as in Abalone dataset. It took weeks to execute the algorithm without finishing. Hence, FRSBM is not scalable over large datasets, so it will not be considered in that case.

Starting from the RMSE, as shown in Table II, it can be noticed in all datasets that RULES-F+ is never the best. Even though, on average, its results in better than WM but it was never the best at any of the investigated datasets. Nevertheless, what is important is to note that rule learning methods are better than DT methods in most cases. At least one rule learning method is equal or better than the best DT.

Specifically, WM gives the best RMSE in Friedman dataset, which is also similar to the other rule learning methods. Moreover, it has the best RMSE in Dee, Ele-1, and Diabetes datasets. However, this algorithm is not scalable over Treasury dataset because it has a large number of examples and attributes, in comparison to the other datasets. Additionally, when it comes to M5Rules, this algorithm has the best RMSE in lots of datasets; as in Ele-2, Abalone, Baseball, Treasury, ForestFire, and Machine CPU datasets. Hence, it is scalable over large datasets. Alternatively, when it comes to DT methods, CART performance is not one of the best but M5 gives higher accuracy in AutoMPG6, AutoMPG8, and Laser datasets. Nevertheless, M5Rules was not far away from the resulted RMSE in these three datasets. Thus, it can be concluded that rule based algorithms that deals with continuous classes can result in better or equal error rate compared to DT algorithms.

TABLE II: ROOT MEAN SQUARED ERROR (RMSE) WITH 10-CROSS-VALIDATION

Datasets	WM	FRSBM	M5Rules	M5	CART	RULES-F+
Abalone	2.56	--	2.12	2.15	2.97	2.81
AutoMPG6	3.52	3.45	2.91	2.83	3.67	3.81
AutoMPG8	3.61	3.32	2.94	2.86	4.03	3.56
Baseball	1063.98	709.19	694.74	704.77	961.01	847.12
Dee	0.47	0.41	0.42	0.42	0.55	0.49
Diabetes	0.75	0.54	0.57	0.60	0.75	0.65
ele-1	705.44	638.18	678.02	678.02	782.38	788.47
ele-2	336.99	164.06	85.97	86.13	981.78	413.04
ForestFires	242.16	46.47	45.64	56.99	47.14	70.49
Friedman	2.21	2.69	2.38	2.34	2.95	2.78
Laser	16.77	22.69	8.77	7.80	12.49	20.78
MachineCPU	79.44	61.72	60.09	60.09	105.49	78.34
Treasury	1.00	0.24	0.24	0.25	0.27	1.17
<b>Average</b>	<b>189.15</b>	<b>--</b>	<b>121.91</b>	<b>123.48</b>	<b>223.50</b>	<b>171.81</b>

In addition to the RMSE, execution time is also important. Thus, the time spent by every algorithm was also recorded in seconds, as illustrated in

Fig. 1 From the graph it can be noticed that RULES-F+, WM, M5, and M5Rules are not highly affected by the datasets properties. In all datasets they only consumed seconds to complete the job. Moreover, it was found that although RULES-F+ did not give the best RMSE when compared to the other algorithms but it actually is the fastest one. Its execution time is the least and its average speed is the highest.

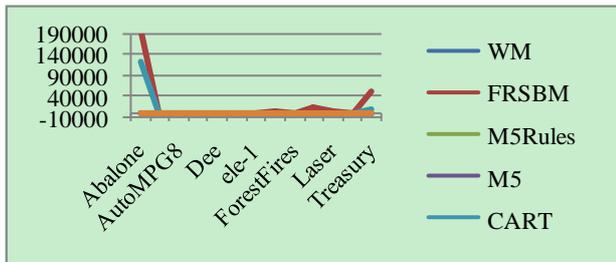


Fig. 1. Average time with continuous classes in seconds.

Nevertheless, when it comes to CART and FRSBM it can be noticed that the dataset properties highly affect their performance. In specific, CART encountered an increase in Abalone and Treasury datasets, where they include large number of examples and attributes. Alternatively, FRSBM speed is also affected by datasets with large number of examples, as in Abalone, Ele-2, Friedman, Laser, and Treasury datasets. The more number of examples in the dataset the longer the algorithm would take.

Consequently, from all above, it can be said that RULES family is one of the most flexible families in CA. It was previously proven that this family is very promising to be used with discrete datasets. However, when it comes to numerical actions the performance is different. It was found that CA can be a better option in Information System domain but from the empirical study conducted in this paper it was found that this domain needs further improvement. In specific, when it comes to datasets with continuous classes the following points can be concluded about the methods developed to manage such problem.

- 1) The error rate of rule learning algorithms are better than or equal to DT algorithms and their speed are similar.
- 2) In the execution time, it was not possible to find a general conclusion because the speed was different from algorithm to another depending on the approach used.
- 3) The fuzzy technique used in RULES-F+ resulted in the best execution time but affected its accuracy.
- 4) RULES family needs further improvement to cover the shortcoming resulting from the use of fuzzy theory and to reduce the tradeoff between speed and accuracy.

From all above, it can be concluded that the empirical result emphasized on the conclusion discussed in the theoretical analysis. Thus, IL needs further improvement with continuous classes' problem. Moreover, because of RULES family properties, discovered in the preceding paper, it would be a good idea to make such improvement using this family. A new non-discretization technique can be integrated into RULES to solve the problem of continuous classes

without the need for discretization or regression.

## V. CONCLUSION

In CA, different families have been developed and several surveys were also conducted. However, one family called RULES was found to be neglected. Although two preceding studies tried to find the importance of RULES family and how it can be used to improve IL but these studies did not consider the problem of continuous classes. Hence, in addition to the neglects, the effect of numerical actions over this family was also missing. Hence, the contribution of this paper was to extend the preceding versions to show the effect of RULES over datasets with continuous classes to be used as a reference by recent researchers in IL. From the theoretical analysis, it was found that even though RULES family classical families of IL but it is lacking when applied over datasets with continuous classes. Moreover, as a result of the empirical evaluation it was concluded that current algorithms needs to tradeoff between the accuracy and speed. A non-discretization technique was not developed in this kind of problem so the gaps encountered because of discretization are not yet eliminated. Consequently, continuous classes' problem is still an open research area in CA and non-discretization technique can be integrated with RULES family for further improvement.

## ACKNOWLEDGEMENT

This research project is supported by a grant from King AbdulAzaiz City for Science & Technology. We also thank Dr. Samuel Bigot for providing us with RULES-F+.

## REFERENCE

- [1] K. J. Cios, R. W. Swiniarski, W. Pedrycz, L. A. Kurgan, K. Cios, R. Swiniarski, and L. Kurgan, "Supervised learning: Decision trees, Rule algorithms, and their hybrids," *Data Mining*, Springer US, 2007, pp. 381-417.
- [2] D. Pham and A. Afify, "Machine-learning techniques and their applications in manufacturing," in *Proc. the Institution of Mechanical Engineers Part B Journal of Engineering Manufacture*, vol. 219, pp. 395-412, 2005.
- [3] F. Stahl and M. Bramer, "Computationally efficient induction of classification rules with the PMCRI and J-PMCRI frameworks," *Knowledge-Based Systems*, vol. 35, pp. 49-63, 2012.
- [4] M. S. Aksoy, "A review of rules family of algorithms," *Mathematical and Computational Applications*, vol. 13, pp. 51-60, 2008.
- [5] D. T. Pham and M. S. Aksoy, "RULES: A simple rule extraction system," *Expert Systems with Applications*, vol. 8, pp. 59-65, 1995.
- [6] H. ElGibreen and M. S. Aksoy, "RULES Family: Where does it stand in Inductive Learning?" in *Proc. 8th International Conference on Computer Engineering and Applications*, Tenerife, Spain, 2014.
- [7] H. ElGibreen and M. S. Aksoy, "Continuous features in inductive learning and the effect of RULES family," *International Journal of Computers*, 2014.
- [8] D. Pham, "A novel rule induction algorithm with improved handling of continuous valued attributes," Doctor of Philosophy, School of Engineering, Cardiff University, Cardiff, 2012.
- [9] H. ElGibreen and M. S. Aksoy, "Multi model transfer learning with RULES family," in *Machine Learning and Data Mining in Pattern Recognition*. P. Perner, Ed., Springer Berlin Heidelberg, 2013, vol. 7988, pp. 42-56.
- [10] H. ElGibreen and M. S. Aksoy, "RULES-IT: Incremental transfer learning with RULES family," *Frontiers of Computer Science*, 2014.
- [11] D. T. Pham, S. Bigot, and S. S. Dimov, "RULES-F: A fuzzy inductive learning algorithm," in *Proc. the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 220, pp. 1433-1447, 2006.

- [12] S. Bigot, "A new rule space representation scheme for rule induction in classification and control applications," in *Proc. the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 2011.
- [13] J. Fürnkranz, "Pruning Algorithms for Rule Learning," *Machine Learning*, vol. 27, pp. 139-172, 1997.
- [14] J. Fürnkranz, "Separate-and-conquer rule learning," *Artificial Intelligence Review*, vol. 13, pp. 3-54, 1999.
- [15] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," *Informatica*, vol. 31, pp. 249-268, 2007.
- [16] U. Ruckert and L. Deraedt, "An experimental evaluation of simplicity in rule learning," *Artificial Intelligence*, vol. 172, pp. 19-28, 2008.
- [17] S. Garcia, J. Luengo, J. A. Saez, V. Lopez, and F. Herrera, "A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning," *IEEE Transactions on Knowledge and Data Engineering*, 2012.
- [18] S. Bibi, G. Tsoumakas, I. Stamelos, and I. P. Vlahavas, "Software defect prediction using regression via classification," *AICCSA*, 2006, pp. 330-336.
- [19] I. H. Witten, E. Frank, L. E. Trigg, M. A. Hall, G. Holmes, and S. J. Cunningham, "Weka: Practical machine learning tools and techniques with Java implementations," 1999.
- [20] G. Holmes, M. Hall, and E. Prank, *Generating rule sets from model trees*, Sydney, Australia: Springer-Verlag, 1999.
- [21] Y. Zhao and Y. Zhang, "Comparison of decision tree methods for finding active objects," *Advances in Space Research*, vol. 41, pp. 1955-1959, 2008.
- [22] D. Steinberg, "CART: Classification and regression trees," in *The Top Ten Algorithms in Data Mining*, X. Wu and V. Kumar, Eds., Taylor & Francis Group, LLC, 2010, pp. 179-201.
- [23] G. V. Kass, "An exploratory technique for investigating large quantities of categorical data," *Applied Statistics by Journal of the Royal Statistical Society*, vol. 29, pp. 119-127, 1980.
- [24] W.-y. Loh and Y.-s. Shih, "Split selection methods for classification trees," *Statistica Sinica*, vol. 7, pp. 815-840, 1997.
- [25] J. B. Gray and G. Fan, "Classification tree analysis using TARGET," *Computational Statistics & Data Analysis*, vol. 52, pp. 1362-1372, 2008.
- [26] G. Potgieter and A. P. Engelbrecht, "Evolving model trees for mining data sets with continuous-valued classes," *Expert Systems with Applications*, vol. 35, pp. 1513-1532, 2008.
- [27] R. C. Barros, D. D. Ruiz, and M. P. Basgalupp, "Evolutionary model trees for handling continuous classes in machine learning," *Information Sciences*, vol. 181, pp. 954-971, 2011.
- [28] S. M. Halawani, I. A. Albidewi, and A. Ahmad, "A novel ensemble method for regression via classification problems," *Journal of Computer Science*, vol. 7, p. 387, 2011.
- [29] R. Salman and V. Kecman, "Regression as classification," in *Proc. IEEE*, 2012, pp. 1-6.
- [30] H. Hsiao-Wei, C. Yen-Liang, and T. Kwei, "A dynamic discretization approach for constructing decision trees with a continuous label," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, pp. 1505-1514, 2009.
- [31] S. Bibi, G. Tsoumakas, I. Stamelos, and I. Vlahavas, "Regression via Classification applied on software defect estimation," *Expert Systems with Applications*, vol. 34, pp. 2091-2101, 2008.
- [32] M. F. bin Othman and T. M. S. Yau, "Comparison of different classification techniques using WEKA for breast cancer," in *Proc. 3rd Kuala Lumpur International Conference on Biomedical Engineering 2006, 2007*, pp. 520-523.
- [33] S. Inamdhar, S. Narangale, and G. Shinde, "Preprocessor agent approach to knowledge discovery using Zero-R algorithm," *International Journal*, vol. 2, 2011.
- [34] L. Torgo and J. Gama, "Regression by classification," in *Advances in Artificial Intelligence*, D. Borges and C. A. Kaestner, Eds., Springer Berlin Heidelberg, vol. 1159, 1996, pp. 51-60.
- [35] L. Torgo and J. Gama, "Regression using classification algorithms," *Intelligent Data Analysis*, vol. 1, pp. 275-292, 1997.
- [36] R. Salman, "Contributions to K-means clustering and regression Via classification algorithms," Doctor of Philosophy, Computer Science Virginia Commonwealth University, Richmond, Virginia, 2012.
- [37] C. Chiu and N. S. Chiu, "An adapted covering algorithm approach for modeling airplanes landing gravities," *Expert Systems with Applications*, vol. 26, pp. 443-450, 2004.
- [38] D. Pham, S. Bigot, and S. Dimov, "RULES-5: a rule induction algorithm for classification problems involving continuous attributes," *Institution of Mechanical Engineers*, 2003, pp. 1273-1286.
- [39] L. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338-353, 1965.
- [40] L.-X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, pp. 1414-1427, 1992.
- [41] L. Sánchez, "A random sets-based method for identifying fuzzy models," *Fuzzy Sets and Systems*, vol. 98, pp. 343-354, 1998.
- [42] J. C. Huhn and E. Hullermeier, "FR3: a fuzzy rule learner for inducing reliable classifiers," *Trans. Fuz Sys.*, vol. 17, pp. 138-149, 2009.
- [43] J. Alcalá-Fdez, L. Sánchez, S. Garc á, M. J. d. Jesus, S. Ventura, J. M. Garrell, J. Otero, C. Romero, J. Bacardit, V. M. Rivas, J. C. Fern ández, and F. Herrera, "KEEL: A software tool to assess evolutionary algorithms to data mining problems," *Soft Computing*, vol. 13, pp. 307-318, 2009.
- [44] J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. Garc á, L. Sánchez, and F. Herrera, "KEEL Data-Mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17, pp. 255-287, 2011.
- [45] B. Efron and R. Tibshirani, *An Introduction to the Bootstrap*, USA: Chapman & Hall, 1993.
- [46] Y. Yang and G. Webb, "Proportional k-Interval discretization for Naive-Bayes classifiers," in *Machine Learning: ECML 2001*, L. Raedt and P. Flach, Eds., Springer Berlin Heidelberg, vol. 2167, 2001, pp. 564-575.
- [47] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining Practical Machine Learning Tools and Techniques*, Third ed., Morgan Kaufmann, 2011.

**Hebah ElGibreen** is currently working on her PhD in the field of ML. In 2009, she received her MS in information system at Kind Saud University. In 2007, she worked as a TA at NXT LEGO robot program for gifted girls, sponsored by "King Abdul-Aziz & His Companions Foundation for the Gifted". She also won the 2nd place of educating by computers section in the National Competition in Computer Skills, sponsored by College of Telecom and Information. She worked for two years at Kind Saud University as a TA and currently as a lecturer.



**Mehmet Sabih Aksoy** completed his PhD in the field of AI in Cardiff University, U.K in 1994. He did his master study in Yildiz University, Istanbul in 1985. He graduated from Istanbul Technical University in 1982. He worked in Istanbul Technical University, Fatih University, Sakarya University and Istanbul University in Turkey from 1984 to 2002. His area of interest includes ML, inductive learning, expert systems, artificial neural networks and data mining. He joined King Saud University in 2002. He is currently working in the same university as a full professor.