Rule Based Systems and Networks:

Deterministic and Fuzzy Approaches

Han Liu
School of Computing
University of Portsmouth
Portsmouth, United Kingdom
han.liu@port.ac.uk

Alexander Gegov
School of Computing
University of Portsmouth
Portsmouth, United Kingdom
alexander.gegov@port.ac.uk

***Abstract*— a rule based system is a special type of expert system which consists of a set of rules. In practice, rule based systems can be built by using expert knowledge or learning from real data. Due to the vast and increasing size of data, the latter approach has become quite popular for building rule based systems. In particular, rule based systems can be built through use of rule learning algorithms, which can be based on statistical heuristics or random basis. This paper focuses on deterministic approaches for classification. This paper also features fuzzy approaches for modelling tasks. In general, this paper is mainly concerned with rule based systems that have a single rule base. However, some of the contents that relate to fuzzy approaches also include some concepts of multiple rule bases.**

Keywords-rule based systems; rule based networks; data mining; machine learning; rule learning; ensemble learning

I. INTRODUCTION

A rule based system is a special type of expert systems, which consists of a set of rules. Rule based systems can be designed by using expert knowledge or through learning from real data. From this point of view, the approaches for the design of rule based systems can be divided into expert based design and data based design. Due to the vast and rapid increasing size of data, the latter approach of design has become increasingly popular.

As introduced in [1], rules can be used for different tasks, e.g. classification, regression and association, and thus these rules are referred to as classification rules, regression rules and association rules respectively. In terms of classification rules, a unfied framework for design of rule based classification systems was developed in [2] and the framework is made up of rule generation, rule simplification and rule represention. In particular, rule generation approaches can be divided into two categories: divide and conquer [3] and separate and conquer [4]. Rule simplification can be achieved through use of pruning algorithms [4], which can be specialized into the following two types: pre-pruning and post-pruning. Rule representation is aimed at managing the computational complexity and interpretability for rule based models [5]. A more detailed explanation of the above framework is presented in Section III.

Rule based systems designed through using machine learning approaches are in general domain independent. For example, as described in [6], inductive learning algorithms are domain independent and can be applied in any practical tasks such as classification and pattern recognition. Some successful applications have been listed in [6], which include lymphography, prognosis of breast cancer recurrence, location of primary tumor and thyroid problem diagnosis in medicine. Mode details can be seen in [7, 8, 9].

This paper is organized as follows: Section II introduces some theoretical preliminaries that are closely related to rule based systems. Section III and IV present respectively a framework for design of single rule based systems and several approaches of ensemble learning for design of ensemble rule based systems in the context of deterministic logic. Section V and VI present respectively multiple rule based systems and networked rule based systems in the context of fuzzy logic. Section VII summaries the contributions of this paper in terms of theoretical and practical importance and suggests the further directions of this research towards achieving advances in rule based systems and networks.

II. THEORETICAL PRELINMINARIES

As mentioned in Section I, some fundamental concepts are closely related to rule based systems and machine learning. In particular, these concepts include *if-then* rules, computational logics, supervised and unsupervised learning.

A. If-Then Rules

As introduced in Section I, a rule based system is made up of a set of rules. It is described in [10] by Ross that a number of different ways have been adopted for knowledge representation in engineering applications of artificial intelligence but one of the most popular ways is to take the form of if-then rules denoted by the expression: IF cause (antecedent) THEN effect (consequent).

The above expression typically provides an inference that if the input (cause, antecedent, condition) is given then the output (effect, consequent, outcome) can be derived [10]. In this paper, each item that makes up a condition (the left hand side of a rule) is referred to as a rule term. It is introduced in [11] that both the left hand side (antecedent) and the right hand side (consequent) of a rule can contain multiple rule terms (inputs/outputs). In this context, an antecedent that contains multiple rule terms linked by 'and' connectives is referred to as a conjunctive antecedent, whereas an antecedent that consists of the rule terms that are linked by 'or' connectives is referred to as a disjunctive antecedent. The above concepts concerning antecedents also apply to the consequents of a rule. Also, it is presented in [11] that rules would be conjunctive, if all of these rules are linked through use of 'and' connectives, or disjunctive, if any of these rules are connected through use of 'or' connectives. A rule may also be inconsistent, which indicates the possible case that a rule may have the same antecedent mapped to a number of different consequents. In this case, the rule would appear to have its antecedent conjunctive and its consequent disjunctive. In addition, rules with the same set of input attributes on their left hand side can make up a rule base. More details about the concept of rule bases will be presented and discussed in Section 5 and can also be seen in [12].

B. Computational Logic

It is stated in [10] by Ross that logic is a small part of the capability of human reasoning, which can assist people towards making decisions or judgments. A basic type of logic is known as Boolean logic in computer science. As presented in [5], in the context of Boolean logic, each variable is binary, which means that the value of such a variable is 0 (false) or 1 (true). This indicates that reasoning and judgment that are made under certainty would normally lead to deterministic outcomes. From this point of view, Boolean logic can also be known as deterministic logic. However, it is quite usual in reality that people can only make practical operations under uncertainty such as decision making, reasoning and judgment. Due to the above case, the other three types of computational logic, namely probabilistic logic, fuzzy logic and rough logic, have thus become more popular approaches. The differences among the four types of computational logic are comparatively discussed in the perspectives of any related statistical heuristics, set theory and corresponding event type in Table 1.

TABLE.I. TYPE OF COMPUTATIONAL LOGIC [13]

Logic Type	Related Heuristics	Related Set Theory	Related Event type
Deterministic Logic	Binary truth value	Crisp set	Certain event
Probabilistic Logic	Probability	Probabilistic set	Random event
Fuzzy Logic	Fuzzy truth value	Fuzzy set	Fuzzy event
Rough Logic	Possibility	Rough set	Possible event

Deterministic logic deals with certain events. For example, all elements in a crisp set should fully belong to the set without uncertainty, i.e. each of these elements is certainly assigned a full membership to the above set.

Probabilistic logic deals with random events under probabilistic uncertainty. For example, an element may be randomly put into a set with a certain probability. An element must be given a full membership to the set once the element has been put into the above probabilistic set.

Fuzzy logic deals with events under non-probabilistic uncertainty. In this context, each set is known as a fuzzy set, which is due to the fact that each of the elements in such a set may only be given a partial membership to the set, i.e. each of these elements belongs to the fuzzy set to a certain degree.

Rough logic deals with uncertain events which results from incomplete information. In the context of set theory, a rough set is defined as a special type of sets, which restores information on the basis of different subsets of attributes [14], i.e. all instances in a rough set belong to the set subject to specific conditions by means of employing a boundary region of the set [15]. For example, an instance belongs to a rough set subject to two conditions, which have the weight of 0.6 and 0.4 respectively. In this context, if the first condition is met and the second condition is still pending, then the possibility for the instance to belong to the rough set is 0.6.

C. Machine Learning

Machine learning is a branch of artificial intelligence and involves two stages: training and testing [16]. The first stage aims to learn something from known properties by using learning algorithms and the second stage aims to make predictions on unknown properties by using the knowledge learned in the first stage. From this point of view, training and testing are also referred to as learning and prediction respectively. In practice, a machine learning task is aimed at building a model, which is further used to make predictions, through the use of learning algorithms. Therefore, this task is usually referred to as predictive modelling. Machine learning could be divided into two types: supervised learning and unsupervised learning, in terms of the form of learning. Supervised learning means learning with a teacher because all instances from a training set are labelled. The aim of this type of learning is to build a model by learning from labelled data and then to make predictions on other unlabeled instances with regard to the value of a predicted attribute. The predicted value of an attribute could be either discrete or continuous. Therefore, supervised learning could be involved in both classification and regression tasks for categorical prediction and numerical prediction respectively. In contrast, unsupervised learning means learning without a teacher. This is because all instances from a training set are unlabeled. The aim of this type of learning is to discover previously unknown patterns from data sets. It includes association and clustering. The former aims to identify correlations between attributes and the latter aims to group objects on the basis of their similarities to each other.

On the other hand, machine learning algorithms are popularly used in data mining tasks to discover some previously unknown pattern. Therefore, this task is usually referred to as knowledge discovery. From this point of view, data mining tasks also involve classification, regression, association and clustering. Both classification and regression can be used to reflect the correlation between multiple independent variables and a single dependent variable. The difference between classification and regression is that the former typically reflects the correlation in qualitative aspects whereas the latter reflects it in quantitative aspects. Association is used to reflect the correlation between multiple independent variables and multiple dependent variables in both qualitative and quantitative aspects. Clustering can be used to reflect patterns in relation to grouping of objects.

III. SIGNLE RULE BASED SYSTEMS

Single rule based systems generally means that each of such systems is made from a single rule based model. As mentioned in Section I, a single rule based system can be designed through adopting a unified framework that consists of rule generation, rule simplification and rule representation. This section illustrates this framework in detail.

A. Rule Generation

As mentioned in Section I, rules can be generated following two main approaches: divide and conquer and separate and conquer. The former aims to generate a set of rules in the form of a decision tree, such as ID3 [17] and C4.5 [3], whereas the latter aims to generate a set of if-then rules directly from training instances, such as Prism [18] and IEBRG [19].

The divide and conquer approach is also known as Top-Down Induction of Decision Trees (TDIDT). This is because of the fact that rules generated through use of this approach are represented in the form of decision trees and that the induction procedure is from general to specific like the top-down approach [20] in the context of software engineering. The basic procedure of the TDIDT is illustrated in Fig.1.

```
Input: A set of training instances, attribute A_i, where i is the index of the attribute A, value
V_j, where j is the index of the value V
Output: A decision tree.
  if the stopping criterion is satisfied then
     create a leaf that corresponds to all remaining training instances
  else
       choose the best (according to some heuristics) attribute A_i
       label the current node with A_i
       for each value V_j of the attribute A_i do
        label an outgoing edge with value V_j
        recursively build a subtree by using a corresponding subset of training   instances
       end for
  end if
```

Figure 1 Decision tree learning algorithm [21]

The separate and conquer approach is also known as the covering approach. This is because of the fact that this approach typically involves generating if-then rules sequentially. In particular, the aim of this approach is to generate a rule which covers the instances that belong to the same class and then iteratively start the generation of the next rule through learning from the rest of the training instances that should not have been covered by the rules generated previously. In other words, all the above instances that are covered by the previously generated rules need to have been deleted from the current training subset. The basic procedure of the separate and conquer approach is illustrated in Fig. 2.

B. Rule Simplification

As mentioned in Section I, rule simplification can be achieved through using pruning algorithms, which can generally be divided into two categories: pre-pruning and post-pruning. In practice, simplification of rules can lead to reduction of overfitting and complexity of rule based models. Pre-pruning generally means to take pruning actions when rules are being generated. In contrast, post-pruning generally means that pruning actions are not taken until the rule generation has been completed.

```
Input: A set of training instances
Output: An ordered set of rules
  while training set is not empty do
    generate a single rule from the training set
    delete all instances covered by this rule
    if the generated rule is not good then
      generate the majority rule and empty the training set
    end if
  end while
```

Figure 2 Rule covering approach [21]

On the other hand, pruning strategies are varied between different rule generation approaches. In fact, rules generated following the divide and conquer approach automatically fit in a tree structure whereas rules generated following the separate and conquer approach are automatically represented in the form of if-then rules. Therefore, the strategies for pruning decision trees and if-then rules are different.

With regard to decision tree pruning, pre-pruning aims to stop a branch of a tree being generated further whereas post-pruning aims to simplify a tree by replacing a particular sub-tree with a leaf node following the completion of the tree generation. In addition, post-pruning of a decision tree can also be done through converting the tree into a set of if-then rules and then simplifying each of the rules, following the completion of the tree generation.

With regard to pruning of if-then rules, pre-pruning aims to stop a rule being specialized on its left hand side whereas post-pruning aims to simplify each single rule after its generation has been completed. However, the main difference to post-pruning of decision trees is that each rule is simplified immediately after its generation has been completed and before the start of the generation of the next rule.

More specific pruning algorithms towards simplification of rules can be seen at [2, 13, 22].

C. Rule Representation

As mentioned in Section I, rule representation aims to manage the computational complexity and interpretability for rule based models. Existing rule representation techniques include decision tree and linear list. The former usually represents rules generated through the divide and conquer approach in an automatic manner whereas the latter usually represents rules generated through the separate and conquer approach in an automatic manner [5]. In addition, a novel rule representation technique, which is referred to as rule based network, has recently been developed in [5].

In a decision tree, the root or each internal node represents an input attribute and each branch from the current node to one of its child nodes represents a condition judgment. In addition, each leaf node represents a class label. An example of decision trees is illustrated in Fig.3.

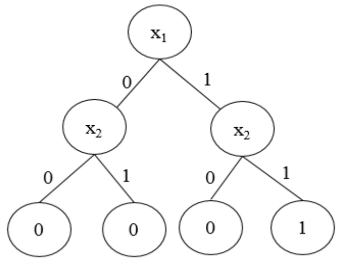

Figure 3 Decision Tree [13]

The decision tree illustrated in Fig.3 can be simply converted into a set of rules as follows:

Rule 1: if x_1=0 and x_2=0 then y=0;

Rule 2: if x_1=0 and x_2=1 then y=0;

Rule 3: if x_1=1 and x_2=0 then y=0;

Rule 4: if x_1=1 and x_2=1 then y=1;

The above rule set can also be represented in a rule based network topology as illustrated in Fig.4. In this network topology, each of the nodes (e.g. x_1 and x_2) in the input layer represents an input attribute. Each node in the conjunction layer represents a rule antecedent, and the corresponding consequent of the same rule is a class label which is represented as a node in the output layer. In addition, each of the connections between the nodes in the input and conjunction layers represents the condition judgement, and each of the connections between the nodes in the last two layers (conjunction and output) represents the mapping between a rule antecedent and a rule consequent.

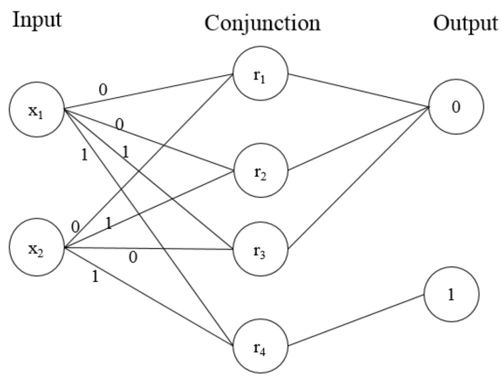

Figure 4 Rule Based Network Topology [13]

IV. ENSEMBLE RULE BASED SYSTEMS

An ensemble rule based system consists of a number of single rule based systems by means of a system of systems. In this context, each single rule based systems can be seen as a subsystem of the ensemble rule based system. Ensemble rule based systems can be designed through adopting ensemble learning approaches.

Ensemble learning is usually adopted to improve the overall accuracy in prediction. As mentioned in [21], ensemble learning can be done in parallel or sequentially. In the former way, there is no collaboration involved in training stage and only the predictions by different models are combined towards making a final prediction. In the latter way, the first algorithm learns a model that is further corrected by all the subsequent algorithms.

A. Bagging

The term Bagging stands for bootstrap aggregating. It is a popular method developed by Breiman [23] and follows the parallel ensemble learning approach. Bagging involves sampling of data with replacement. In particular, the Bagging method manages to take a sample with the size *n*, where *n* is the size of the training set, and to have instances from the training set randomly selected into the sample set. This indicates that some instances in the training set may appear more than once in the sample set and some other instances may never appear in the sample set. On average, a sample is expected to contain 63.2% of the training instances [21, 25]. In the training stage, the classifiers, each of which results from a particular sample set mentioned above, are parallel to each other. In the testing stage, their independent predictions are combined towards predicting the final classification through equal voting. The detailed procedure of Bagging is illustrated in Fig.5. As concluded in the literature [21, 25], Bagging is robust and does not lead to overfitting due to the increase of the number of generated models. Therefore, it is useful especially for those non-stable learning methods with high variance.

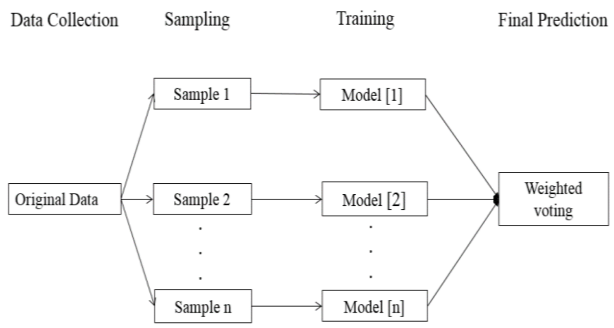

Figure 5 Bagging Method [22]

B. Random Forests

Random forests is another popular method [24] that can be seen as a special case of bagging. In particular, the base classifier must be a decision tree, which is generated on each sample of training data. In addition, the attribute selection at each node of a decision tree is random to some extent. Otherwise, this ensemble learning method only belongs to Bagging. In this sense, at each node, a subset of attributes is randomly selected from the training set and the attribute which can provide the best split for the node is finally chosen [26]. In the training stage, the employed algorithm of decision tree learning is used to generate classifiers independently on the samples of the training data. In the testing stage, the classifiers make the independent predictions that are combined towards predicting the final classification through equal voting. The detailed procedure of Random Forests is illustrated in Fig.6. As concluded in the literature [21], the approach for learning random forests is robust because of the reduction of the variance for decision tree learning algorithms.

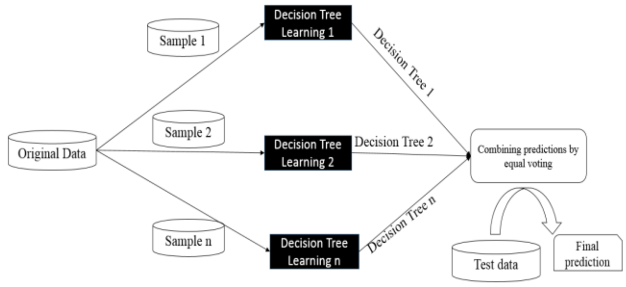

Figure 6 Random Forests [22]

C. *Collaborative Rule Learning*

The collaborative rule learning approach is developed in [27]. It involves collaborations among algorithms in training stage. The essence of this approach is based on the procedure of the separate and conquer rule learning approach. In particular, a single rule is generated in each of the iterations of rule learning. The above approach has all the chosen rule learning algorithms involved in the iteration towards generating a rule; each of the rule learning algorithms may also be assisted by some pruning algorithms depending on the setup of experiments; in the next step, all of the rules, each of which is generated by using a particular rule learning algorithm, are compared in terms of their quality; finally, only the rule with the highest quality is selected and added into the rule set. This process is repeated until all of instances have been deleted from the training set as instructed in the separate and conquer approach. The detailed procedure of the collaborative rule learning approach is illustrated in Fig.7.

Iteration 1:

three rules generated: $rule_{11,}$ $rule_{12}$ and $rule_{13}$

the one with the highest quality: $rule_{13}$

The one added into the rule set: $rule_{13}$

Iteration 2:

three rules generated: $rule_{21,}$ $rule_{22}$ and $rule_{23}$

the one with the highest quality: $rule_{21}$

The one added into the rule set: $rule_{21}$

…………….

Finally, the generated rule set comprises: {$rule_{13,}$ $rule_{21}$ ………………}.

Figure 7 Collaborative rule generation procedure [27]

V. MUTIPLE RULE BASED SYSTEMS

Multiple rule based systems generally mean a special type of rule based systems has multiple rule bases as illustrated in Fig.8. This type of systems is often described by cascaded rule bases, and its most common forms are referred to as Chained Fuzzy System (CFS) or Hierarchical Fuzzy System (HFS) in this section since they are described in the context of fuzzy logic. A HFS is characterized by a white-box nature whereby the inputs are mapped to the outputs by means of some intermediate variables. The operation of a CFS/HFS is based on multiple FID sequences whereby each intermediate variable links the FID sequences for two adjacent rule bases.

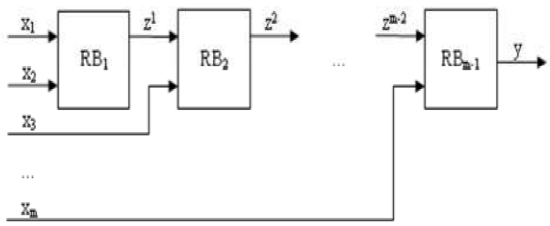

Figure 8 Chained Rule Base [12]

A CFS is usually used as a detailed presentation of a SFS (Standard Fuzzy Systems) for the purpose of improving transparency by explicitly taking into account all subsystems and the interactions among them. Also, some efficiency is gained because of the smaller number of inputs to the individual rule bases. The same positive effect is observed for feasibility which is enhanced by the ability to reflect better the simultaneous influence of the reduced number of inputs to the individual rule bases. However, some accuracy is lost due to the accumulation of errors as a result of the repetitive application of fuzzification, inference and defuzzification within the multiple FID sequences.

A HFS is also usually used as a simplified presentation of a SFS for the purpose of improving efficiency and transparency. Efficiency is improved by the reduction of the overall number of rules which is a linear function of the number of inputs to the subsystems and the number of linguistic terms per input. Transparency is also improved by explicitly taking into all subsystems and the interactions among them. The same applies to feasibility which is facilitated by the small number of inputs to the individual rule bases. However, these improvements are at the expense of losing accuracy for the same reason as in the case of a CFS.

VI. NETWORKED RULE BASED SYSTEMS

Networked rule based systems generally mean a special type of rule based systems that has networked rule bases as illustrated in Fig.9. This system is referred to as Networked Fuzzy System (NFS) in this section since it is described in the context of fuzzy logic. A NFS is characterized by a white-box nature whereby the inputs are mapped to the outputs by means of some intermediate variables. Each subsystem in a NFS is represented by a node whereas the interactions among subsystems are the connections among these nodes.

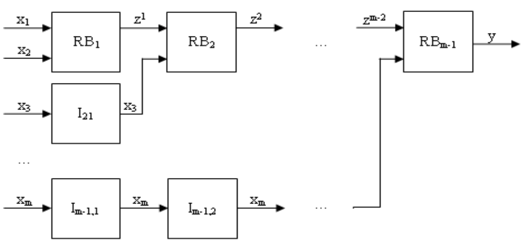

Figure 9 Networked Rule Base [12]

A NFS is a hybrid between a SFS and a CFS/HFS. On one hand, the structure of a NFS is similar to the structure of a

CFS/HFS due to the explicit presentation of subsystems and the interactions among them. On the other hand, the operation of a NFS resembles the operation of a SFS due to the possibility of simplifying the original multiple rule bases to a linguistically equivalent single rule base. This simplification is based on a linguistic composition approach central to a NFS.

Being a hybrid concept, a NFS potentially has some of the advantages and the disadvantages of a SFS and a CFS/HFS. In this respect, on the positive side, a NFS could be as feasible and transparent as a CFS/HFS due to the original multiple rule base presentation. However, on the negative side, a NFS could also be as efficient as a SFS due to the equivalent single rule base presentation. As far as accuracy is concerned, a NFS could have either advantages or disadvantages in relation to a SFS or a CFS/HFS. For example, a NFS could be more accurate than a CFS/HFS due to the single application of a FID sequence but less accurate than a SFS due to the approximation effect of the linguistic composition approach.

VII. CONCLUSION

This paper has presented several deterministic approaches for classification tasks and featured several fuzzy approaches for modelling tasks. In particular, a unified framework is presented towards design of single rule based systems with high level of accuracy, efficiency and interpretability. Also, several approaches of ensemble learning are presented towards design of ensemble rule based systems leading to improvement of overall accuracy in classification. In addition, two types of rule bases are presented towards effective management of complexity in rule based systems and have been applied in two case studies: mortgage assessment and product pricing [12]. The approaches presented in this paper for design of rule based systems can be advanced through adopting granular computing concepts and techniques. Specific directions have been suggested in [13].

ACKNOWLEDGMENT

This paper summarizes conceptually the contents of a tutorial with the same title to be presented at the IEEE International Conference on Intelligent Systems 2016. The authors would like to thank the conference organizers for inviting them to present research results from some of their previous works in this tutorial.

REFERENCES

[1] H. Liu, A. Gegov and F. Stahl, "Categorization and Construction of Rule Based Systems," in *International Conference on Engineering Applications of Neural Networks* , Sofia, 2014.

[2] H. Liu, A. Gegov and F. Stahl, "Unified Framework for Construction of Rule Based Classification Systems," in *Information Granularity, Big Data, and Computqational Intelligtence*, W. Pedrycz and S. M. Chen, Eds., Switzerland, Springer, 2015, pp. 209-230.

[3] R. Quinlan, C4.5: programs for machine learning, Morgan Kaufman, 1993.

[4] J. Furnkranz, "Separate-and-Conquer rule learning," *Artificial Intelligence Review,* vol. 13, pp. 3-54, 1999.

[5] H. Liu, A. Gegov and M. Cocea, "Network Based Rule Representation for Knowledge Discovery and Predictive Modelling," in *IEEE International Conference on Fuzzy Systems*, Istanbul, 2015.

[6] M. S. Aksoy, "A review of rules family of algorithms," *Mathematical and Computational Applications,* vol. 1, no. 13, pp. 51-60, 2008.

[7] R. Quinlan, "Induction, knowledge and expert systems," in *Artificial Intelligence Developments and Applications*, Amsterdam, 1988.

[8] R. S. Michalski, "On the quasi-minimal solution of the general covering problem," in *Proceedings of the Fifth International Symposium on Information Processing*, Bled, Yugoslavia, 1969.

[9] R. Quinlan, "Inductive knowledge acquisition: a case study," in *Applications of Expert Systems*, R. Quinlan, Ed., Turing Institue Press, 1987, pp. 157-173.

[10] T. J. Ross, Fuzzy Logic with Engineering Applications, West Sussex: John Wiley & Sons Ltd, 2004.

[11] A. Gegov, Complexity Management in Fuzzy Systems, Berlin: Springer, 2007.

[12] A. Gegov, Fuzzy Networks for Complex Systems: A Modular Rule Base Approach, Berlin: Springer, 2010.

[13] H. Liu, A. Gegov and M. Cocea, "Rule Based Systems: A Granular Computing Perspective," *Granular Computing,* vol. 1, no. 4, pp. 1-16, 2016.

[14] Y. Yao, "Granular Computing," in *Proceedings of The 4th Chinese National Conference on Rough Sets and Soft Computing*, 2004.

[15] Z. Pawlak, "Rough sets," *International Journal of Computer and Information Sciences,* vol. 11, no. 5, pp. 341-356, 1982.

[16] T. M. Mitchell, Machine Learning, McGraw Hill, 1997.

[17] R. Quinlan, "Induction of Decision Trees," *Machine Learning,* vol. 1, pp. 81-106, 1986.

[18] J. Cendrowska, "PRISM: an algorithm for inducing modular rules," *International Journal of Man-Machine Studies,* vol. 27, p. 349–370, 1987.

[19] H. Liu and A. Gegov, "Induction of Modular Classification Rules by Information Entropy Based Rule Generation," in *Innovative Issues in Intelligent Systems*, vol. 623, V. Sgurev, R. Yager, J. Kacprzyk and V. Jotsov, Eds., Switzerland, Springer, 2016, pp. 217-230.

[20] D. Avison and G. Fitzgerald, Information Systems Development: Methodologies, Techniques and Tools, 4 ed., London: McGraw-Hill Higher Education, 2006, p. 656 .

[21] I. Kononenko and M. Kukar, Machine Learning and Data Mining: Introduction to Principles and Algorithms, Chichester, West Sussex: Horwood Publishing Limmited, 2007.

[22] H. Liu, A. Gegov and M. Cocea, Rule Based Systems for Big Data: A Machine Learning Approach, 1 ed., vol. 13, Switzerland: Springer, 2016.

[23] L. Breiman, "Bagging predictors," *Machine Learning,* vol. 24, no. 2, p. 123–140, 1996.

[24] L. Breiman, "Random Forests," *Machine Learning,* vol. 45, no. 1, pp. 5-32, 2001.

[25] P.-N. Tan, M. Steinbach and V. Kumar, Introduction to Data Mining, New Jersey: Pearson Education, 2006.

[26] J. Li and L. Wong, "Rule Based Data Mining Methods for Classification Problems in Biomedical Domains," in *15th European Conference on Machine Learning and 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, Pisa, 2004.

[27] H. Liu, A. Gegov and M. Cocea, "Collaborative Rule Generation: An Ensemble Learning Approach," *Journal of Intelligent and Fuzzy Systems,* vol. 30, no. 4, pp. 2277-2287, 2016.