Remedial Actions Recommendation via multi-Label Classification: A Course Learning Improvement Method

Ammar Elhassan, Ilyes Jenhani, and Ghassen Ben Brahim

***Abstract*— In this paper we present RARS - Remedial Actions Recommender System that is based on a multi-label classification approach to recommend remedial actions to address student performance shortcomings in Learning Outcome Attainment Rates. A dataset of rubric instances is constructed where each instance is characterized by a set of features (e.g. course domain, course level, etc.). Classes labeling the training instances correspond to the remedial actions that have been proposed by instructors and Quality Assurance Experts over several semesters. Experiments carried out on the constructed dataset showed that the use of wrapper multi-label classification approaches as a basis of RARS and especially the classifier chains method with decision trees as a base classifier provides useful remedial actions recommendations.**

***Index Terms*— Educational data mining, Multi-label Classification, Recommender Systems, Remedial Action Recommendations.**

I. Introduction

Due to the exponential growth of data collected from processes and sensors both online and offline, it has become increasingly difficult for decision makers to analyze this data sufficiently to produce information using traditional database or office automation tools, hence it is necessary to deploy Artificial Intelligence (AI)-based approaches to parse and analyze huge volumes of data and produce information that aid the decision making process. Recommender Systems are suitable for analyzing data describing patterns of behavior and making recommendations in business, marketing and commerce, education and government, security as well as medicine.

This paper introduces a multi-label classification-based recommender system (RARS) that addresses shortcomings in student learning attainment. In a previous work [1], the authors created a rubrics dataset built from distinct sources for multiple semesters. In [1], each rubric instance was labeled by a unique remedial action as the "best" action to take to address student performance shortcomings in Learning Outcome Attainment Rates. Standard classification techniques were applied and evaluated using the generated dataset. In this paper, RARS will learn from rubric instances labeled by more than one remedial action (instead of a single remedial action as previously done in [1]) and consequently will recommend a set of remedial actions to address specific shortcomings for each unseen rubric instance. In fact, during the instance labeling process in [1], it was difficult for the instructors to provide a single class label (i.e. a remedial action) for a given rubric instance because in many cases, instructors judged that more than one remedial action are applicable. Hence, in the current work, we are handling the problem as a multi-label classification problem [2] in which each rubric instance could be labeled by more than one remedial action. RARS will be based on a multi-label classification process wherein (multiple) recommendations are made by the system for each rubric instance based on a set of relevant features like section size, topic, course learning outcome (CLO), etc.

The paper is organized as follows: Section II overviews works that inspire this research and highlights special elements that are unique to the work undertaken here. Section III describes the main idea and design of the process of multi-remedial actions recommendation. The structure of the dataset that is generated for this research is described in Section IV. Section V analyzes results obtained from the application of different multi-label classification approaches to the generated dataset. Finally, conclusions and future work are presented in section VI.

II. Background

Most Recommender Systems works in the field of education focus on one of two aspects: (i) finding correlations between student profiles and the academic support activities to be recommended or (ii) identifying suggestions that will enhance the effectiveness of the learning process [3]. RARS, partially inspired by the research work in [4] and a continuation of [1], contributes to the area of Recommender Systems providing recommendations to enhance the effectiveness of the learning process. The authors in [4] focused on a specific set of fundamental "computer programming" skills which require strong reasoning logic and practice, and attempted to assess the students' performance with the objective of improving the learning process. They developed a recommender system that suggests classes of activities based on the learners' needs. The authors introduced the concept of "student profile" which captures the student performance in a particular programming activity. The task of recommending activities was considered as a multi-label classification task aiming to map classes to each instance of student's profile resulting in student data instances being mapped to multidimensional profiles that are, in turn, mapped to classes of programming activities.

The system also has the ability to make recommendations of classes of activities to new learners which are based on existing and similar profiles. The system is based on the

Manuscript received July 28, 2018; revised September 29, 2018.

Ammar Elhassan is with Princess Sumaya University for Technology, Amman, Jordan (e-mail: ammarelhassan@gmail.com).

Ilyes Jenhani and Ghassen Ben Brahim are with Prince Mohammad Bin Fahd University, Khobar, KSA (e-mail: ijenhani@pmu.edu.sa, gbrahim@pmu.edu.sa).

doi: 10.18178/ijmlc.2018.8.6.749

ML-kNN algorithm [5] and was able to achieve results matching human recommendations most of the time. The strategy proposed in [4] focuses on the C language programming skills and hence, is limited by narrow scope in two ways: the first one being the language-specific nature of the proposal and secondly, the skills being assessed within the programming language, also highly specific. Furthermore, the performance of the methodology was assessed with respect to a single algorithm: (ML-kNN) rather than conducting comparative performance analyses for many multi-label classification algorithms.

RARS applies to a similar application domain as in [4] and applies multi-label based recommendations with distinguishing features that enhance coverage scope: all courses/topics in 3 degree programs, label design: remedial actions are drawn from a master pool of 39 actions that is built by instructors and quality assurance experts (QAE's) over several academic semesters with no exclusive mappings between the remedial actions and specific courses/domains, Multiple algorithms/performance metrics: The accuracy of the recommended actions is assessed according to several error/loss metrics.

III. Process Design

The proposed RARS system learns from a multi-labeled dataset and recommends remedial actions from a pool of remedial actions (RA) based on the recognized patterns of attribute values; this type of recommender system is based on collaborative filtering [6]. The process operates on rubric assessments wherein rubric instances that require remedial actions are identified and added to a rubrics dataset which subsequently forms the input for the recommendation and assessment processes.

A. *Main phases*

The recommendation process works in 2 phases: (i) RARS engine will be used to assess the performance of a set of wrapper method-classifier combinations, the input to this phase is the Rubric dataset where each instance corresponds to a rubric which is characterized by a set of feature values and is labeled by a set of remedial actions. These different multi-label classification approaches will be tested based on a separate testing set in order to identify the nominated optimum wrapper method-classifier combination to be used for producing suitable recommendations. (ii) The classroom application step where the nominated wrapper method-classifier combination will be used by RARS to recommend the remedial actions that an instructor could undertake to address student performance shortcomings in learning outcome attainment rates.

B. *Remedial Actions*

The master set of Remedial Actions (RA) is built incrementally and iteratively over a period of several academic semesters using expertise of academic instructors and quality assurance experts who recommend appropriate tutorials and additional materials to students based on the results of formative or summative assessments and taking into consideration several factors including Course Level, Learning Outcome, Labs, Section Size, etc. The remedial actions (RA0,..RAn), that are associated with each instance in the dataset are suggested by instructors and QAE. These include but are not limited to: "Provide extra assignments", "Provide extra quizzes", "Provide extra practical assignments", "Cover more examples", "Revise concepts", "Vary exam question types", "Provide supplement textbook and material", "Fortify Pre-Requisites", "Promote class discussion", "Adopt demo-based tutorials". "Visit support center", "Improve class/lab coordination", "Update course content", "Introduce research assignments", "Revise basic math", "Modify course structure", etc.

C. *Feature Engineering*

To prepare the data for RARS, it was necessary to remap some attribute (feature) values from their original format/values thus introducing the new features. This process is described in Table I, all other attributes remain unchanged.

TABLE I: Attribute Definition/Mapping

Attribute	Values	Source of the attribute	Process/ Comment
Domain	Programming, Networking, System Configuration	Course Code	Course code has no semantic weight in the process of label selection. Uses a Course to Domain mapping
NQF	Knowledge Cognitive Communication Psychomotor Interpersonal	Course Learning Outcome (CLO)	Course Learning Outcomes total well over 140 although they tend to fall into around 6 Competency/ Knowledge Framework categories
Level	Beginner Intermediate Advanced	Course Code	Course code 1st numerical digit determines the year of the course offering, hence Year1 courses are grouped as Beginner, Year2, Year3 as Intermediate and Year4 as Advanced
Size	Small Medium Large	The section size (number of students)	The size value is changed to categorical value: Large: >= 15, Medium: > 5, Small: <=5
HasLab	Yes No	Lab attribute from the Course entity	Determines if the course has a lab or not
UCat	Green Amber Red	(U)nsatisfactory , rubric percentage value	Amber: >= 15 Red: >= 20 Green: All other values
MCat	Green Amber Red	(M)inimal rubric percentage value	Amber: >= 15 Red: >= 20 Green: All other values

Rubrics are used as a basis for all assessment methods. All students' grades are grouped into four possible rubrics ((U)nsatisfactory, (M)inimal, (A)dequate and (E)xemplary). For each section s, for each assessment a and for each course learning outcome o, we will have the percentage of students who belong to each rubric category based on the grades they obtained for o in a. No remedial actions are needed if students in s performed well in learning outcome o (i.e. belong to either Adequate or Exemplary rubric categories). However, remedial actions need to be taken for different alarm levels within the Unsatisfactory and Minimal rubric categories. In fact, in each one of these low-performance categories, a red alarm will be triggered when 20% of the students or more did not well in learning outcome o and an amber alarm will be triggered when the percentage of students is between 15% and 20%. If the percentage is less than 15%, no alarms will triggered hence no remedial actions need to be taken.

D. Testing the Approach with MEKA

The dataset including the QAE's recommended remedial actions is split as 70%-30% Training-Testing ratio; a frequently applied testing strategy. Different tests using several multi-label classifiers were conducted using MEKA [7]. Different loss/accuracy metrics were collected for each test. The (wrapper) method-classifier combination with the highest accuracy (lowest loss) is nominated as the candidate approach for RARS.

IV. Dataset Description

Because there is no limit imposed on the number of labels (remedial actions) academics can suggest, the training data contains varying sets of recommendations as shown in Table II, with the detailed occurrence distribution shown in Fig. 1.

From Table II, we can see that instances in our multi-labeled dataset are almost equally distributed in terms of number of class labels each one of these instances is labeled with. The number of instances labeled with few labels (between 1 and 5) is almost equal to the number of instances having more labels (between 6 and 11). From Fig. 1, we can also see that most of the instances have 9 labels then 3 labels, then 2 labels, etc.

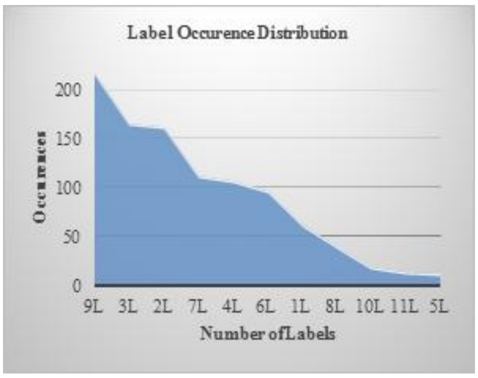

Fig. 1. Label Occurrences per instance in the dataset (detailed).

The original number of records in the full assessment database is over 15000 records which are grouped into Rubric Lines. Within the set of all rubric lines, the training dataset is selected based on the threshold of the "Unsatisfactory" and "Minimal Performance" categories (i.e. UCat and MCat) described in Table I.

TABLE II: Class Label Occurrences in the Dataset (Summary)

Number of Class Labels (Remedial Actions) per instance	Occurrences
Fewer than 3	224
Between 3 and 5	286
Between 6 and 8	250
More than 8	248

Each instance in the final dataset corresponds to a rubric line which is characterized by the 7 features described in Table I and which is labeled by 1 or more remedial action (from the set of remedial actions listed in Section III.B). Table III provides more details about the final dataset.

TABLE III: Dataset Description

Total number of Instances	1008
Total number of Class labels (remedial actions)	39
Largest set of labels	11
Smallest set of labels	1
Average number of labels per instance	6
Percentage of Rare Labels Occurring in less than 50 instances	6/39 (16%)
Number of Attributes (Features)	7

As shown in Table III, we ended up with a total of 1008 instances. The pool of remedial actions contains 39 different actions. Subsets of this pool will be used as labels for each instance. The range of the number of remedial actions labeling each instance is [1...11] with an average of 6 labels per instance (which is reasonable and easy to manage by the instructor). Only few remedial actions (6 out of 39) are rarely used in the final dataset.

V. Experimental Results

In order to recommend the appropriate set of remedial actions for a given subset of rubric instances via RARS, the process requires iteration through various MEKA supported methods and classifiers to identify the optimum method-classifier combination; this involves generating result sets and then adjusting the MEKA parameters to obtain the highest accuracy.

A. Evaluation Strategy

Our evaluation process is based on the percentage split testing strategy; 70% of the data is used for training and 30% is used for testing. We will apply the following Wrapper methods: Binary Relevance (BR) [2], Classifier Chains (CC) [8], RAndom-klabEL set (RAkEL) [9] and Rank+Threshold (RT) [10]. For the wrapper methods, we will use the following base classifiers: a top-down divide and conquer (recursive) classifier: C4.5 Decision Tree, an instance-based learner: K-Nearest Neighbor classifier and a simple probabilistic approach: Naïve Bayes classifier.

These wrapper-classifier combinations will be assessed using the Hamming Loss [10], Zero-One Loss [11], One-Error [4], Accuracy and Rank Loss [12] performance metrics. We will generate and contrast accuracy results to nominate an optimum method-classifier combination for our dataset specifically and the domain of formative/summative assessments generally.

B. Loss/Accuracy Results and Analysis

The loss and accuracy results of the experimental tests are discussed below.

1) Hamming loss

The Hamming Loss results in Table IV show that Binary Relevance-Decision Tree, Binary Relevance-Nearest Neighbor, Classifier Chains-Decision Tree, Classifier Chains-Nearest Neighbor and Rank Threshold-Nearest Neighbor produced the lowest loss (highest performance) in this metric which identifies misplaced, missing, or over-placed labels within each set of predicted labels for each test instance.

TABLE IV: HAMMING LOSS PERFORMANCE RESULTS

Method-Base Classifier		Hamming Loss
Binary Relevance (BR)	Decision Trees	0.008
	1-Nearest Neighbors	0.010
	Naïve Bayes	0.070
Classifier Chains (CC)	Decision Trees	0.005
	1-Nearest Neighbors	0.007
	Naïve Bayes	0.097
Rakel	Decision Trees	0.174
	1-Nearest Neighbors	0.178
	Naïve Bayes	0.200
Rank+Threshold (RT)	Decision Trees	0.305
	1-Nearest Neighbors	0.010
	Naïve Bayes	0.104

The results show that: CC ~> BR ~>RT ~> R where (~> indicates "marginally better than").

As for classifiers performance within wrapper methods, we note that: DT ~> 1-NN ~> NB with the exception of RT where DT is showing the poorest performance. This is because the RT (Ranking + Threshold) wrapper method duplicates each multi-labeled instance into instances with one label each then trains a base classifier, and uses a threshold to reconstitute a multi-label classification. Combined with the DT base classifier which is a robust classifier for symmetric label noise [13], this duplication-based approach will create asymmetric noise (due to different numbers of labels per instance) in the new generated dataset thus leading to impure sub-partitions for most decision nodes and leaves. This will make the attribute selection and the leaf labeling very hard to achieve hence producing random results in many cases.

2) Zero-One loss

Although the Zero-one Loss metric is too strict (whenever at least 1 incorrect class is recommended the loss is 1) for the nature of the problem domain we are dealing with, as with the Hamming Loss results reported above, the Zero-One Loss results in Table V show that Binary Relevance-Decision Tree, Binary Relevance-Nearest Neighbor, Classifier Chains-Decision Tree, Classifier Chains-Nearest Neighbor and Rank Threshold-Nearest Neighbor produced the lowest loss (highest performance) in this metric which produces a 1 value whenever the classifier predicted at least one incorrect element in the set of labels for each test instance.

TABLE V: ZERO-ONE LOSS PERFORMANCE RESULTS

Method-Base Classifier		Zero-One Loss
Binary Relevance (BR)	Decision Trees	0.105
	1-Nearest Neighbors	0.085
	Naïve Bayes	0.601
Classifier Chains (CC)	Decision Trees	0.055
	1-Nearest Neighbors	0.067
	Naïve Bayes	0.615
Rakel	Decision Trees	1.000
	1-Nearest Neighbors	1.000
	Naïve Bayes	1.000
Rank+Threshold (RT)	Decision Trees	1.000
	1-Nearest Neighbors	0.085
	Naïve Bayes	0.863

In summary, CC ~> BR ~>RT ~> R. As for classifiers performance within each wrapper method, we note that: 1-NN ~> DT ~> NB.

3) One-error loss

The One-Error Loss results reported in Table VI show that Binary Relevance and Rank Threshold algorithms (regardless of the used base classifier) produced the lowest loss (highest performance) in this metric which measures the frequency of missing "high priority" labels for each test instance. We note that BR = RT ~> R ~> CC and in terms of the performance of the classifiers within the CC wrapper method, we note that the classifier performance is: 1-NN ~> DT ~> NB.

TABLE VI: ONE-ERROR LOSS PERFORMANCE RESULTS

Method-Base Classifier		One Error
Binary Relevance (BR)	Decision Trees	0.000
	1-Nearest Neighbors	0.000
	Naïve Bayes	0.000
Classifier Chains (CC)	Decision Trees	0.023
	1-Nearest Neighbors	0.017
	Naïve Bayes	0.044
Rakel	Decision Trees	0.000
	1-Nearest Neighbors	0.012
	Naïve Bayes	0.041
Rank+Threshold (RT)	Decision Trees	0.000
	1-Nearest Neighbors	0.000
	Naïve Bayes	0.000

4) Rank loss

TABLE VII: RANK LOSS PERFORMANCE RESULTS

Method-Base Classifier		Rank Loss
Binary Relevance (BR)	Decision Trees	0.000
	1-Nearest Neighbors	0.000
	Naïve Bayes	0.004
Classifier Chains (CC)	Decision Trees	0.005
	1-Nearest Neighbors	0.008
	Naïve Bayes	0.092
Rakel	Decision Trees	0.189
	1-Nearest Neighbors	0.196
	Naïve Bayes	0.215
Rank+Threshold (RT)	Decision Trees	0.148
	1-Nearest Neighbors	0.000
	Naïve Bayes	0.007

The Rank Loss results in Table VII show that Binary Relevance-Decision Tree, Binary Relevance–Nearest Neighbor, Classifier Chains-Decision Tree; Classifier Chains–Nearest Neighbor and Rank Threshold – Nearest

Neighbor produced the lowest loss (highest performance) in this metric which assesses accuracy of the ordinal position (priority) of predicted labels within their respective set for each test instance.

We note that CC ~> BR ~> RT ~> R and in terms of the performance of the classifiers, we note that DT ~> 1-NN ~> NB. The Rank Loss metric is a good indicator for the problem domain presented here as it assesses the ordinal position and instructor priorities within the recommended remedial actions without dismissing the whole set at first error.

5) Average accuracy

Similarly to the Hamming Loss results (Section V.II.A), the Average accuracy results in Table VIII show that Binary Relevance-Decision Tree, Binary Relevance – Nearest Neighbor, Classifier Chains-Decision Tree, Classifier Chains – Nearest Neighbor and Rank Threshold–Nearest Neighbor produced the highest performance in this metric which assesses the average of all the accuracies of each testing instance predicted class/label set, worked out as the difference in set size of the predicted set of classes and the correct set of classes against the size of the correct class.

TABLE VIII: AVERAGE ACCURACY PERFORMANCE RESULTS

Method-Base Classifier		Accuracy
Binary Relevance (BR)	Decision Trees	0.984
	1-Nearest Neighbors	0.978
	Na ïve Bayes	0.858
Classifier Chains (CC)	Decision Trees	0.989
	1-Nearest Neighbors	0.984
	Na ïve Bayes	0.817
Rakel	Decision Trees	0.580
	1-Nearest Neighbors	0.571
	Na ïve Bayes	0.530
Rank+Threshold (RT)	Decision Trees	0.520
	1-Nearest Neighbors	0.980
	Na ïve Bayes	0.786

Accuracy wise, CC ~> BR ~> RT ~> R and within each wrapper method, the classifier relative performance, we note that 1-NN ~> DT ~> NB except in RT (DT) for the same reason explained in the Hamming Loss case above.

C. Optimum Algorithm Candidate

Based on the above performances analyses, we can conclude that, on average, most of the tested methods perform well in recommending multiple remedial actions despite the relatively high average number of actions per instance (6), over the five used metrics and independently of the used base classifier. We also note that the Classifier Chains (CC) multi-label transformation approach has provided the best performance in all four metrics, and that CC provided best results for the main desirable metrics: Hamming Loss, Ranking Loss and Accuracy). As such, the nomination for the optimum multi-label approach for RARS correspond to the Classier Chains combined with the Decision Tree base classifier (CC-DT).

VI. CONCLUSIONS AND FUTURE WORK

This research set out to analyze (formative and summative) outcome based student academic assessment data that is grouped into rubric instances with flags identifying shortcomings in student outcome attainment. Using a pool of possible remedial actions that is progressively and reiteratively constructed over several academic semesters by instructors and quality assurance experts, a multi-label recommender system approach was applied to this data.

Five Loss and Accuracy performance metrics were then considered to measure the accuracy of the used multi-label approaches with results obtained from the MEKA toolset. The applied evaluation strategies implemented using a fairly comprehensive method-classifier combinations showed positive and consistent results. These results are close to .09 for loss functions and above 85% for accuracy functions, the performance results indicate that the Classier Chains wrapper method combined with the Decision Tree base classifier (CC-DT) is ideal for the type of problem domain and dataset we are experimenting with.

As the assessment database expands further tests will be conducted using RARS in conjunction with the expected incremental semantic expert knowledge of instructors and QAE's; this may indicate a need to switch to other (current or newly developed) method–classifier combinations should the accuracy performance results indicate this. In addition to the expected expansion of the data above over the next academic semesters, it is worth noting that the current dataset pertains to a single academic college, hence the dataset size has been a limiting factor in terms of the strength of the learning algorithm and a larger dataset incorporating other college assessments.

The proposed approach has room for enhancement in three main aspects:

(i) To produce even more consistent and accurate results, we are working on a new dataset version with additional data collected from other universities, aiming for a larger dataset with less university-specific features to improve the accuracy of RARS and make it as a generic system which could be used in any university.

(ii) One of the recurring issues in this process is the Labels Ratings from the QAE's perspective, it is expected that the incorporation of QAE attributes such as experience, qualifications and specialty as new features will play a significant role in the prioritization of classes within recommended actions and hence, affect the loss and accuracy performance.

(iii) Because the heuristics for identifying a student outcome shortcoming from the rubrics are rule-based, there is a high chance that a fuzzy logic or a possibilistic logic [14] based approach may be more appropriate to the problem domain and hence will produce a more representative dataset.

REFERENCES

[1] I. Jenhani, G. B. Brahim, and A. Elhassan, "Course learning outcome performance improvement: A remedial action classification based approach," in *Proc. International Conference on Machine Learning and Applications*, 2016, Los Angeles, USA, pp. 408-413.

[2] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, pp. 1–13, 2007.

[3] M. J. Dascalua, C. N. Bodeab, M. N. Mihailescua, E. A. Tanasea, and P. O. de Pablosc, "Educational recommender systems and their application in lifelong learning," *Journal of Behaviour & Information Technology*, vol. 35, issue 4, 2016.

[4] M. C. De Oliveira, P. Gonçalves, C. Marques, and E. Oliveira, "Recommendation of programming activities by multi-label classification for a formative assessment of students," *Expert Systems with Applications*, vol. 40, no. 16, pp. 6641-6651, 2013.

[5] M. L. Zhang and Z. H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern Recognition*, vol. 40, no. 7, pp. 2038-2048, 2007.

[6] P. Resnick, N. Iacovou, M. Suchack, P. Bergstrom, and J. Riedl, "Grouplens: An open architecture for collaborative filtering of netnews," in *Proc. ACM CSW 1994 Conference on Computer. Supported Cooperative Work*, 1994, pp. 175–186.

[7] J. Read, P. Reutemann, B. Pfahringer, and G. Holmes, "MEKA: A Multi-label/Multi-target Extension to WEKA," *Journal of Machine Learning Research*, vol. 17, no. 21, pp. 1−5, 2016.

[8] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification,"*Lecture Notes in Computer Science*, vol. 5782, pp. 254–269, Springer, 2009.

[9] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Random k-Labelsets for multi-label classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 7, pp. 1079–1089, 2011.

[10] M. Boutell, J. Luo, X. Shen, and C. Brown, "Learning multi-label scene classification," *Pattern Recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.

[11] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, New York, Wiley, 1973.

[12] E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker, "Label ranking by learning pairwise preferences," Artificial Intelligence, vol. 172, pp. 1897-1916, 2008.

[13] A. Ghosh, M. Naresh, and P. S. Sastry, "On the robustness of decision tree learning under label noise," arXiv preprint arXiv: 1605.06296, 2016.

[14] L. A. Zadeh, "Fuzzy Logic: A personal perspective," *Fuzzy Sets and Systems*, vol. 281, pp. 4-20, 2015.

Ammar Elhassan is an assistant professor of computer science at Princess Sumaya University for Technology, Jordan (psut.edu.jo). He obtained his doctorate in computing in 2001 from Surrey University, UK (surrey.ac.uk) and his MSc in Information Systems in 1991 from the University of East Anglia, UK (uea.ac.uk). Dr. Elhassan has extensive industrial experience in the area of software engineering combined with academic and administrative experience in quality and accreditation. His research interests include AI, optimization, machine learning, sustainable development and assistive technology. He is very keen on sustainable development events and the utilization of technology for clean energy and waste management.

Ilyes Jenhani obtained his Ph.D. in computer science from Artois University, France in 2010. Since 2014, he is an assistant professor in the Information Technology Department at Prince Mohammad bin Fahd University (KSA). Before joining PMU, he served as a lecturer at the Higher Institute of Management (ISG Tunis) and as an assistant professor at the Faculty of Economic and management sciences (FSEG Tunis). His research interests include machine learning (classification, clustering and feature selection) from uncertain data, uncertain reasoning, artificial immune systems, educational data mining. He has many publications related to these topics in international refereed journals and conferences. In addition, Dr. Jenhani served in the PC of many international conferences and as a reviewer in several international journals.

Ghassen Ben Brahim is an assistant professor of computer science at Prince Mohammed University in Saudi Arabia. He received his BSc degree in computer science from the National School of Computer Science in Tunisia, the MS degree from the University of Missouri-Columbia, and the PhD degree from Western Michigan University. He worked as a systems analyst engineer in the integrated defense systems at boeing and as research visitor at the US Naval Research Lab. His research interests include: machine learning, computer and network security, wireless networks, QoS routing in large scale MANETs, routing in all optical networks, and the design and analysis of network protocols.