



# Generation of Efficient Rules for Associative Classification

Chartwut Thanajiranthorn<sup>(✉)</sup> and Panida Songram

Department of Computer Science, Faculty of Informatics Mahasarakham University,  
Mahasarakham, Thailand  
{60011260501,panida.s}@msu.ac.th

**Abstract.** Associative classification is a mining technique that integrates classification and association rule mining for classifying unseen data. Associative classification has been proved that it gives more accurate than traditional classifiers and generates useful rules which are easy to understand by a human. Due to inheriting from association rule mining, associative classification has to face a sensitive of minimum support threshold that a huge number of rules are generated when a low minimum support threshold is given. Some of the rules are not used for classification and need to be pruned. To eliminate unnecessary rules, this paper proposes a new algorithm to find efficient rules for classification. The proposed algorithm directly generates efficient rules. A vertical data representation technique is adopted to avoid the generation of unnecessary rules. Our experiments are conducted to compare the proposed algorithm with well-known algorithms, CBA and FACA. The experimental results show that the proposed algorithm is more accurate than CBA and FACA.

**Keywords:** Associative classification · Class Association Rule · A vertical data representation · Classification

## 1 Introduction

Associative classification (AC) is a supervised learning classification technique which was proposed by Lui et al. [11]. It integrates two important data mining techniques together, association rules mining and classification. Association rules mining aims to find a relationship between data. Classification aims to predict unseen data and assigns a class label. AC focus to find a Class Association Rules (CARs) in form  $itemset \rightarrow c$  where  $itemset$  is a set of attribute values,  $c$  is a class label. AC has been improved that it produces more accurate than other traditional classifiers [19]. In addition, a CAR is an If-Then rule which is easy to understand by users. Therefore, AC is adapted in many fields i.e. phishing website detection [1, 2, 7], heart disease prediction [9, 16], groundwater detection [8], and detection of low quality information in social networks [20].

Li et al. [10] presented the Classification Based on Multiple Association Rules algorithm (CMAR). The CMAR algorithm generates CARs based on the FP-Growth algorithm. It adopted the Frequent Pattern Tree structure (FP-Tree) to store frequent items and classes. First, it scans a database to find the frequent items and then sorts them in F-list. Each transaction is scanned to create a FP-Tree. The CMAR algorithm divides subset in the FP-Tree to search frequent ruleitems and then add the frequent ruleitems to CR-Tree according to their frequencies. It classifies based on the weighted Chi-square analysis. It prunes unnecessary ruleitems based on confidence, correlation and database coverage. Based on the experimental results, they are shown that the CMAR algorithm is more efficient and scalable than other the CBA and C4.5 algorithms.

Although AC is widely used in many fields and efficient classification, it has to face the generation of a large number of CARs because of association rule mining inheritance. Moreover, some of CARs are unnecessary for classification. Some of algorithms, CBA [11], CAR-Miner [13], PCAR [17] and FCBA [5], attempt to create rules that are effective for classification, but they must also create candidate rules to determine whether the rules can be used for classification. Creating candidate rules spend a lot of computation times and a large number of rules are generated. Hadi et al. [8] have proved that more than 100 million candidate 3-ruleitems are generated from 1,000 frequent 2-ruleitems. Nguyen and Nguyen [13] showed that the number of 4 million candidate ruleitems have been generated when the minimum support threshold is set to 1%. Furthermore, Abdelhamid et al. [3] reported that the minimal process of candidate generation is still a challenging work because it has effective in regards to training time, I/O overheads and memory usage.

In this paper, we propose a new algorithm, called the Efficient Class Association Rule Generation Algorithm (ECARG). The ECARG algorithm is proposed a new method to directly generate a small number of efficient CARs for classification. A dataset is represented in a vertical data format [14] which each ruleitem contains transaction ids. Support and confidence values are easily calculated by using the vertical data format. An efficient CAR is directly found and added to a classifier. Then, transaction ids containing the CAR are not used to find the next CARs so that redundant CARs are not generated. Therefore, the ECARG algorithm early prunes ruleitems during CARs generation process.

The remaining sections of the paper are organized as follows. Section 2 presents related work of AC. Section 3 presents the basic definitions. The proposed algorithm is introduced in Sect. 4. Section 5 discusses the experimental results. The paper is concluded in Sect. 6.

## 2 Related Work

Various algorithms have been proposed for AC. Lui et al. introduced the CBA algorithm which integrated association rule mining and classification. The process of the CBA algorithm is divided into two steps. First, CARs are generated based on the Apriori algorithm [4]. Second, they are sorted and the redundant

rules are pruned. The second step is an important process for selecting efficient CARs in a classifier. The CBA algorithm was proved that it produces a lower error rate than C4.5 [15]. Unfortunately, the CBA algorithm has to face the problem of candidate generation due to base on Apriori manner. It finds CARs from all possible candidate rules at each level so that a large number of candidate rules is generated. After associative classification was introduced, many different algorithms were proposed such as CMAR, eMCAC, FACA, PCAR, FCBA and ACPRISM.

Abdelhamid [1] proposed an Enhanced Multi-label Classifiers based Associative Classification (eMCAC) for phishing websites detection. The eMCAC algorithm enhances the MCAC algorithm [12] by applies a vertical data format to represented datasets. The eMCAC algorithm generates multi-label rules that attribute values are connected with more than one class. The support and the confidence and support values for a multi-label rule are calculated based on the average confidence and support values of all classes. The rules in classifier are sorted to ensure that the rules with high confidence and support values are given higher priority. The class is assigned to the test data if attribute values are fully matched with the rule's antecedent. The accuracy is evaluated and shown that the eMCAC algorithm outperforms CBA, PART, C4.5, jRip and MCAR.

Hadi et al. [7] proposed the FACA algorithm to detect phishing websites. It applies a Diffset [21] to discover CARs. The FACA algorithm discovers k-ruleitems by extending frequent (k-1)-ruleitems. Then, ruleitems are ranked according to the number of itemset in ruleitem, confidence, support, and occurrence. To predict unseen data, the FACA algorithm utilizes All Exact Match Prediction Method. The method matches unseen data with all CARs in classifier. Next, unseen data is assigned to class label with the highest count. The FACA algorithm outperforms CBA, CMAR, MCAR, and ECAR [6].

Song and Lee introduced Predictability-Based Collective Class Association Rule algorithm (PCAR). The PCAR algorithm uses a cross-validation between test dataset and train dataset to calculate a predictability value of CARs. First, PCAR adapts the Eclat algorithm to discover CARs from a dataset. Next, it calculates an average support of CARs from each fold of testing dataset. Then, CARs are ranked according to predictive value, confidence, support, rule antecedent length and rule occurrences, respectively. Finally, the full-matching method is applied to assigns a class label for unseen data. The PCAR algorithm is compared with C4.5, Ripper, CBA and MCAR on the accuracy and shown that it outperforms the others.

Alwidian et al. proposed the WCBA algorithm to overcome the problem of most associative classification algorithms which determines the importance of rules based on support and confidence values. The WCBA assumes that every attributes are equally important regardless of real-world application. For example, medical work determines information affecting the prediction results. The weight of an attribute is calculated to indicate its importance. In addition, CARs are prior sorted by using the harmonic mean which is an average value between support and confidence. The WCBA algorithm is more significantly accurate

than CBA, CMAR, MCAR, FACA, and ECBA. However, the WCBA algorithm generates CARs based on Apriori that scan database many times.

From the previous algorithms, they were proposed for associative classification with high ability of rules for prediction. However, most of the algorithms produce k-ruleitems from (k-1)-ruleitems. They have to calculate support when a new ruleitems is found. To calculate support, they have to scan database with all transactions. Moreover, a huge number of candidate CARs are generated and needed for pruning process to reduce unnecessary CARs. Unlike the previous algorithm, we propose a method to directly generate efficient CARs for prediction so that the pruning and sorting processes do not need in the proposed algorithm. Moreover, a simple set theories, intersection and set different are adapted to calculate easily support, confidence and remove unnecessary ruleitems.

### 3 Basic Definitions

Let  $A = \{a_1, a_2, \dots, a_m\}$  is a finite set of all attributes in dataset.  $C = \{c_1, c_2, \dots, c_n\}$  is a set of classes,  $g(x)$  is a set of transactions containing itemset  $x$  and  $|g(x)|$  is the number of transactions containing  $x$ .

**Definition 1.** An item can be described as an attribute  $a_i$  containing a value  $v_j$ , denote as  $\langle(a_i, v_j)\rangle$ .

**Definition 2.** An itemset is the set of items, denoted as  $(a_{i1}, v_{i1}), (a_{i2}, v_{i2}), \dots, (a_{ik}, v_{ik})$ .

**Definition 3.** A ruleitem is of the form  $\langle\text{itemset}, c_j\rangle$ , which represents an association between itemsets and class in dataset, basically it is represented in form  $\text{itemset} \rightarrow c_j$ .

**Definition 4.** The length of ruleitem is the number of items, denoted as  $k$  – ruleitem.

**Definition 5.** Absolute support of ruleitem  $r$  is the number of transactions containing  $r$ , denoted as  $\text{sup}(r)$ . The support of  $r$  can be found from Eq. 1.

$$\text{sup}(r) = |g(r)| \quad (1)$$

**Definition 6.** Confidence of ruleitem  $\langle\text{itemset}, c_j\rangle$  is the ratio of the number of transactions that contain itemset in  $c_j$  and the number of transaction contain the itemset, and is calculated by Eq. 2

$$\text{conf}(\langle\text{itemset}, c_j\rangle) = \frac{|g(\langle\text{itemset}, c_j\rangle)|}{|g(\text{itemset})|} \times 100 \quad (2)$$

**Definition 7.** Frequent ruleitem is a ruleitem whose support is not less than the minimum support threshold ( $\text{minsup}$ ).

**Definition 8.** Class association rule (CAR) is a frequent ruleitem whose confidence is not less than the minimum confidence threshold ( $\text{minconf}$ ).

## 4 The Proposed Algorithm

In this section, we propose a new algorithm for associative classification. The proposed algorithm includes 4 phases: (1) 1-ruleitem generation, (2) redundant rules removal, (3) ruleitem extension, and (4) default rule creation. Each step will be explained in the subsection with an example. To show an example, we use the weather dataset, as shown in Table 1. The minimum support and confidence thresholds are set to 3 and 90%, respectively.

**Table 1.** Weather dataset.

Transaction ID	Outlook	Temperature	Humidity	Windy	Play
1	Sunny	Hot	High	False	No
2	Sunny	Hot	High	True	No
3	Overcast	Hot	High	False	Yes
4	Rainy	Mild	High	False	Yes
5	Rainy	Cool	Normal	False	Yes
6	Rainy	Cool	Normal	True	No
7	Overcast	Cool	Normal	True	Yes
8	Sunny	Mild	High	False	No
9	Sunny	Cool	Normal	False	Yes
10	Rainy	Mild	Normal	False	Yes
11	Sunny	Mild	Normal	True	Yes
12	Overcast	Mild	High	True	Yes
13	Overcast	Hot	Normal	False	Yes
14	Rainy	Mild	High	True	No

### 4.1 Efficient 1-Ruleitem Generation

In the first phase, 1-ruleitem is generated on vertical data format which is easy to calculate support and confidence. The vertical data format [14] represents associated transaction ids of 1-ruleitems as shown in Table 2. The set of the transaction ids of 1-ruleitems easily find from  $|g(itemset) \cap g(c_k)|$ , which is the support. If the support of 1-ruleitem is less than the minimum support, the 1-ruleitem will be not further extended with other items. The last 2 rows of Table 2 show the support and confidence of ruleitems, respectively.

From Table 1, sunny value in outlook occurs in transaction ids 1, 2, 8, 9 and 11, denoted as  $g(\langle Outlook, Sunny \rangle) = \{1, 2, 8, 9, 11\}$ . Class Yes is in transaction ids 3,4,5,7,9,10,11,12, and 13, denoted as  $g(Yes) = \{3, 4, 5, 7, 9, 10, 11, 12, 13\}$  while class No is in transaction ids 1, 2, 6, 8, and 14, denoted as  $g(No) = \{1, 2, 6, 8, 14\}$ . Transaction ids containing  $\langle Outlook, Sunny \rangle \rightarrow Yes$  is  $g(\langle Outlook, Sunny \rangle) \cap g(Yes) = \{1, 2, 8, 9, 11\} \cap \{3, 4, 5, 7, 9, 10, 11, 12, 13\} =$

{9, 11}, so the supports of  $\langle Outlook, Sunny \rangle \rightarrow Yes$  is 2. Transaction ids containing  $\langle Outlook, Sunny \rangle \rightarrow No$  is  $g(\langle Outlook, Sunny \rangle) \cap g(No) = \{1, 2, 8, 9, 11\} \cap \{1, 2, 6, 8, 14\} = \{1, 2, 8\}$ , so the supports of  $\langle Outlook, Sunny \rangle \rightarrow No$  is 3. We can see that, the rule  $\langle Outlook, Sunny \rangle \rightarrow Yes$  will not be extended because its support is less than the minimum support threshold.

The confidences of the remaining rules are found and easily calculated from  $\frac{|g(itemset \rightarrow c_k)|}{|g(itemset)|} \times 100$ . If the confidence of the rule is 100%, the rule will be added to the classifier. If not, it will be considered to extend in the next phase. For example, the confidence of  $\langle Outlook, Sunny \rangle \rightarrow No$  can easily find from  $\frac{|g(1,2,8)|}{|g(1,2,8,9,11)|} \times 100 = \frac{3}{5} \times 100 = 60\%$ . The confidence of the rule  $\langle Outlook, Sunny \rangle \rightarrow No$  is not 100% so it will be extended. While the confidence of  $\langle Outlook, Overcast \rangle \rightarrow Yes$  is  $\frac{|g(3,7,12,13)|}{|g(3,7,12,13)|} \times 100 = \frac{5}{5} \times 100 = 100\%$  so it is the first CAR adding to the classifier.

**Table 2.** The rules which passed minimum support threshold (white background cell).

	Outlook						Temperature						Humidity				Windy			
	Sunny		Overcast		Rainy		Hot		Mild		Cool		High		Normal		True		False	
	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N
	9	1	3		4	9	3	1	4	8	5	6	3	1	5	6	7	2	3	1
	11	2	7		5		13	2	10	14	7		4	2	7		11	6	4	8
		8	12		10				11		9		12	8	9		12	14	5	
			13						12					14	10				9	
															11				10	
															13				13	
sup	2	3	4	0	3	1	2	2	4	2	3	1	3	4	6	1	3	3	6	2
conf		60	100		60				67		75		43	57	86		50	50	75	

### 4.2 Redundant Rule Removal

After finding a ruleitem with 100% of confidence, the transaction ids containing the ruleitem will be removed to reduce the unnecessary CARs generation. For example, Table 1 we can see that if  $\langle Outlook, Overcast \rangle$  is found, the class will be actually Yes. Therefore we do not need to consider any item occurs in the same transaction ids of  $\langle Outlook, Overcast \rangle$ . To remove unnecessary transaction ids, set difference is adopted. Let  $r_i$  is a CAR with 100% of confidence and  $T$  is a set of ruleitems in the same class of  $r_i$ . For all  $r_j \in T$ , the new transaction ids of  $r_j$  is  $g(r_j) = g(r_j) - g(r_i)$ . For example, in Table 2, the CAR  $g(\langle Outlook, Overcast \rangle \rightarrow Yes) = \{3, 7, 12, 13\}$  and  $g(\langle Humidity, High \rangle \rightarrow Yes) = \{3, 4, 12\}$ . The new transaction ids of  $g(\langle Humidity, High \rangle \rightarrow Yes) = g(\langle Humidity, High \rangle \rightarrow Yes) - g(\langle Outlook, Overcast \rangle \rightarrow Yes) = \{3, 4, 12\} - \{3, 7, 12, 13\} = \{4\}$ . The new transaction ids of all ruleitems are shown in Table 3.

**Table 3.** The remained transaction ids after generating the first CAR.

	Outlook						Temperature						Humidity				Windy			
	Sunny		Overcast		Rainy		Hot		Mild		Cool		High		Normal		True		False	
	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N
	9	1			4	9		1	4	8	5	6	4	1	5	6	11	2	4	1
	11	2			5		2	10		9				2	9			6	5	8
		8			10			11						8	10			14	9	
														14	11				10	
sup	2	3	0	0	3	1	2	2	3	2	2	1	1	4	4	1	1	3	4	2
conf		60			60				60					80	80			75	67	

### 4.3 Ruleitem Extension

The ruleitem  $r$  with highest confidence will be first considered to extend in breadth first search manner. It will be combined with other ruleitem in the same class until the new rule has 100% of confidence. If  $r_i$  is extended with  $r_j$  to be  $r_{new}$  and  $g(r_j) \subseteq g(r_i)$  then  $conf(r_{new}) = 100\%$ .

For example, in Table 3,  $g(\langle Humidity, High \rangle \rightarrow No)$  has the highest confidence and the  $g(\langle Outlook, Sunny \rangle \rightarrow No) = \{1, 2, 8\}$  is subset of  $g(\langle Humidity, High \rangle \rightarrow No) = \{1, 2, 8, 14\}$  so that the new rule  $g(\langle Outlook, Sunny \rangle, \langle Humidity, High \rangle \rightarrow No)$  is found with 100% of confidence. Then  $g(\langle Outlook, Sunny \rangle, \langle Humidity, High \rangle \rightarrow No)$  is stopped to extend. For 2-ruleitem extended from  $\langle Outlook, Sunny \rangle \rightarrow No$ , there is only one rule with 100% of confidence and it is added to the classifier as the second CAR.

After the second CAR is added to the classifier, the transaction ids associated with the CAR will be removed as explained in Sect. 4.2. The remaining transaction ids are shown in Table 4.

**Table 4.** Transaction ids after generating the second CAR.

	Outlook						Temperature						Humidity				Windy			
	Sunny		Overcast		Rainy		Hot		Mild		Cool		High		Normal		True		False	
	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N
	9				4	9			4	14	5	6	4	14	5	6	11	6	4	
	11				5			10						9					9	
					10			11						10					9	
														11					10	
sup	2				3	1			3	1	2	1	1	1	4	1	1	2	4	
conf					75				75						80					100

In Table 4, the rule  $\langle Windy, False \rangle \rightarrow Yes$  has 100% of confidence and it is added to classifier as the third CAR and removes associated transaction ids. Finally, there no any ruleitem pass minimum support threshold as shown in Table 5, the CAR generation will be stopped.

**Table 5.** Transaction ids after generating the third CAR.

	Outlook						Temperature						Humidity						Windy			
	Sunny		Overcast		Rainy		Hot		Mild		Cool		High		Normal		True		False			
	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N		
	11							11	14		6		14	11	6	11	6					
sup	1							1	1		1		1	1	1	1	2					
conf																						

### 4.4 Default Rule Creation

The proposed algorithm continues to build a default CAR for adding to the classifier. In this step, the class that appears the most relevant transaction ids is selected as the default CAR. From Table 5, the remaining transaction ids is relevant to class ‘No’ the most, so that the default class is No. Finally, all CARs in the classifier is shown in Table 6.

**Table 6.** All CARs.

Rule ID	Rule
R1	$\langle Outlook, Overcast \rangle \rightarrow Yes$
R2	$(\langle Humidity, High \rangle, \langle Outlook, Sunny \rangle) \rightarrow No$
R3	$\langle Windy, False \rangle \rightarrow Yes$
Default	No

## 5 Experimental Setting and Result

All the experiments are performed on a 2.3 GHz Intel Core i3-6100u processor with 8 GB DDR4 main memory, running Microsoft Windows 10. We implemented FACA and ECARG algorithms in java and used the CBA algorithm in WEKA data mining software. We have performed an experiment to evaluate the accuracy and number of CARs. Our algorithm is compared with the CBA and FACA algorithms. In the experiments, the minimum support threshold is set to 2% and the minimum confidence threshold is set to 50%. The thresholds are a set according from [5, 8, 18]. Three algorithms are tested with 13 datasets from UCI Machine Learning Repository. The characteristics of the experimental datasets are shown in Table 7. Table 7 shows the number of attributes (exclude a class label), the number of class labels and the number of instances in each data set.

The experimental results are shown in Tables 8 and 9. Table 8 reports the accuracy of the CBA, FACA, and ECARG algorithms. It is clear that the ECARG algorithm performs on average well when comparing to the CBA and FACA algorithms. It gives higher accuracy than CBA and FACA by 4.48% and



**Table 7.** Characteristics of experiment dataset.

Data sets	#of attributes	#of classes	Instances
Breast	11	2	699
Cars	6	4	1,728
Contact-lenses	4	3	24
Diabetes	7	2	768
Iris	4	3	150
Labor	17	2	57
Lymph	18	4	148
Mushroom	22	2	8,214
Post-operative	9	4	90
Tic-tac-toe	9	2	958
Vote	16	2	435
Wined	13	3	178
Zoo	17	7	101

4.22% respectively. This gain has been resulted from the methodology to find the most efficient rule in each iteration and eliminate redundant rules simultaneously.

To be more precise, the proposed algorithm gives the highest accuracy in 9 of 13 datasets. We further analyzed the win-tie-lost records. Based on the Table 8, win-tie-lost records of the proposed algorithm against CBA and FACA in term of accuracy are 12-0-1 and 11-0-2, respectively.

Table 9 shows the average number of CARs from the CBA, FACA and ECARG algorithms using 10-fold cross-validation. The result shows that our algorithm has derived a smaller number of rules than the CBA algorithm. In particular, the ECARG algorithm generates 16 CARs on average against 13 datasets, while the CBA algorithm derives 21 rules on average and the FACA algorithm derives 14 rules on average. We can see that the ECARG algorithm slightly more the average number than the FACA algorithm. However, the ECARG algorithm outperforms the FACA algorithm in term of accuracy rate by 4.22% and win over 11 from 13 datasets.

ECARG discovers rules with 100% of confidence to build the classifier since the high confidence demonstrates the high possibility of class occurrences when occurring an itemset. Therefore, it gives high accuracy. While the CBA and FACA algorithms build classifier from CARs which passed the minimum confidence threshold. Some of CARs have low confidences so they may predict incorrect class and then the accuracies of CBA and FACA are lower than that of the proposed algorithm.

**Table 8.** Accuracies of CAB, FACA and ECARG.

Data sets	CBA	FACA	ECARG
Breast cancer	71.52	72.44	<b>80.00</b>
Cars	76.85	70.77	<b>80.92</b>
Contact-lenses	66.67	68.33	<b>70.83</b>
Diabetes	74.47	<b>73.56</b>	67.32
Iris	92.67	<b>95.33</b>	94
Labor	91.23	91.67	<b>92.67</b>
Lymph	78.37	82.43	<b>88.51</b>
Mushroom	93.4	96.53	<b>99.15</b>
Post-operative	56.67	65.56	<b>70</b>
Tic-tac-toe	<b>98.85</b>	90.23	95.4
Vote	94.94	91.92	<b>95.31</b>
Wined	91.57	92.16	<b>98.87</b>
Zoo	80.63	86	<b>93.07</b>
Average	82.14	82.40	<b>86.62</b>

**Table 9.** The average number of generated rules on UCI datasets.

Data sets	CBA	FACA	ECARG
Breast cancer	16	24	37
Cars	25	14	22
Contact-lenses	9	5	7
Diabetes	46	18	4
Iris	11	7	4
Labor	14	15	9
Lymph	33	15	19
Mushroom	8	16	22
Post-operative	35	12	12
Tic-tac-toe	20	12	46
Vote	30	12	11
Wined	11	11	9
Zoo	10	11	10
Average	21	14	16

## 6 Conclusion

In this paper, we proposed an enhanced associative classification method, namely ECARG. The ECARG algorithm avoids a candidate generation by attempt-

ing to select a first general rule with the highest accuracy. It easily generates CARs based on vertical data representation and set difference. Moreover, it early reduces a search space to discover a CAR based on rule correlation. These improvements guarantee small classifiers size that doesn't overfit the training dataset and maintain their accuracy rate. For this reason, the proposed algorithm different from the traditional algorithms, it no needs sorting and pruning process. The experiments were conducted on 13 UCI datasets and shows that the ECARG algorithm outperformed the CBA and FACA algorithms in term of accuracy. It can gain a higher accuracy rate than the CBA and FACA algorithms by 4.40% and 5.63% respectively. Our future work, we continue to speed up the ECARG algorithm for finding efficient class association rules.

## References

1. Abdelhamid, N.: Multi-label rules for phishing classification. *Appl. Comput. Inf.* **11**(1), 29–46 (2015). <https://doi.org/10.1016/j.aci.2014.07.002>
2. Abdelhamid, N., Ayesh, A., Thabtah, F.: Phishing detection based associative classification data mining. *Expert Syst. Appl.* **41**(13), 5948–5959 (2014)
3. Abdelhamid, N., Jabbar, A.A., Thabtah, F.: Associative classification common research challenges. In: 2016 45th International Conference on Parallel Processing Workshops (ICPPW), pp. 432–437. IEEE (2016)
4. Agrawal, R., Srikant, R., et al.: Fast algorithms for mining association rules. In: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, vol. 1215, pp. 487–499 (1994)
5. Alwidian, J., Hammo, B., Obeid, N.: FCBA: fast classification based on association rules algorithm. *Int. J. Comput. Sci. Netw. Secur. (IJCSNS)* **16**(12), 117 (2016)
6. Hadi, W.: ECAR: a new enhanced class association rule. *Adv. Comput. Sci. Technol.* **8**(1), 43–52 (2015)
7. Hadi, W., Aburub, F., Alhawari, S.: A new fast associative classification algorithm for detecting phishing websites. *Appl. Soft Comput.* **48**, 729–734 (2016). <https://doi.org/10.1016/j.asoc.2016.08.005>
8. Hadi, W., Issa, G., Ishtaiwi, A.: ACPRISM: associative classification based on PRISM algorithm. *Inf. Sci.* **417**, 287–300 (2017). <https://doi.org/10.1016/j.ins.2017.07.025>
9. Jabbar, M., Deekshatulu, B., Chandra, P.: Heart Disease Prediction System using Associative Classification and Genetic Algorithm. [arXiv:1303.5919](https://arxiv.org/abs/1303.5919) [cs, stat], March 2013
10. Li, W., Han, J., Pei, J.: CMAR: accurate and efficient classification based on multiple class-association rules. In: Proceedings 2001 IEEE International Conference on Data Mining, pp. 369–376. IEEE (2001)
11. Liu, B., Yiming Ma, Hsu, W.: Integrating classification and association rule mining. In: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, August 1998
12. Abdelhamid, N., Ayesh, A., Thabtah, F.: Phishing detection based associative classification data mining. *Expert Syst. Appl.* **41**(13), 5948–5959 (2014). <https://doi.org/10.1016/j.eswa.2014.03.019>. <http://www.sciencedirect.com/science/article/pii/S0957417414001481>

13. Nguyen, L., Nguyen, N.T.: An improved algorithm for mining class association rules using the difference of Obidsets. *Expert Syst. Appl.* **42**(9), 4361–4369 (2015). <https://doi.org/10.1016/j.eswa.2015.01.002>
14. Ogihara, Z.P., Zaki, M., Parthasarathy, S., Ogihara, M., Li, W.: New algorithms for fast discovery of association rules. In: 3rd International Conference on Knowledge Discovery and Data Mining. Citeseer (1997)
15. Quinlan, J.: C4.5: Programs for Machine Learning. Elsevier, Amsterdam (2014)
16. Singh, J., Kamra, A., Singh, H.: Prediction of heart diseases using associative classification. In: 5th International Conference on Wireless Networks and Embedded Systems (WECON), pp. 1–7, October 2016. <https://doi.org/10.1109/WECON.2016.7993480>
17. Song, K., Lee, K.: Predictability-based collective class association rule mining. *Expert Syst. Appl.* **79**(Suppl. C), 1–7 (2017). <https://doi.org/10.1016/j.eswa.2017.02.024>. <http://www.sciencedirect.com/science/article/pii/S0957417417301069>
18. Thabtah, F., Cowling, P., Peng, Y.: MCAR: multi-class classification based on association rule. In: The 3rd ACS/IEEE International Conference on Computer Systems and Applications, January 2005. <https://doi.org/10.1109/AICCSA.2005.1387030>
19. Thabtah, F., Hadi, W., Abdelhamid, N., Issa, A.: Prediction phase in associative classification mining. *Int. J. Softw. Eng. Knowl. Eng.* **21**(06), 855–876 (2011)
20. Wang, D.: Analysis and detection of low quality information in social networks. In: 2014 IEEE 30th International Conference on Data Engineering Workshops, pp. 350–354, March 2014. <https://doi.org/10.1109/ICDEW.2014.6818354>
21. Zaki, M., Gouda, K.: Fast vertical mining using difffsets. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2003, pp. 326–335. ACM, New York (2003). <https://doi.org/10.1145/956750.956788>