A meta-heuristic approach for enhancing performance of associative classification

Abstract: **Associative Classification is an interesting approach in data mining to create more accurate and easily interpretable predictive systems. This approach is often built on both association rule mining and classification techniques, to find a set of rules called association rules for classification (CAR) of label attributes. There are many kinds of associative classification such as CPAR, CBA, CMAR but the accuracy is still low on large datasets, and the running time is not reasonable as well. This paper proposes an meta-heuristic approach to significantly enhance the performance of Associative Classification algorithms in running time, reducing the rule set, and accuracy on large data. Experimental results show that meta-heuristic searching the optimal data set makes the associative classification more useful on big data, and reasonable in practice.**

Keywords: *Associative Classification, Meta-heuristic, CAR, Feature Selection*

I. INTRODUCTION

The association classifier is a supervised learning model that uses association rules to assign labels. The model uses association rules to label new data. Thus, the model can be seen as a list of "if-then" clauses: if a new data meets the left-hand attributes of the rule, then the data will be classified according to the right-hand value of the rule [1, 7, 8, 9]. Most classifiers combine sequential scanning of the set of rules and labeling new data by matching the attributes on the left-hand side of rules. Association classifier rules inherit several metrics from association rules, such as Support or Confidence, that can be used to rank or filter the rules in the model and evaluate their quality. There are many different associative classification methods such as CBA, CMAR, and CPAR.

CBA uses techniques in association rules to classify data, this method has higher performance than traditional classification techniques. The limitation of this method is that the number of rules generated is too large in case the support threshold is low. CMAR applies FP-tree efficiently, consuming less memory and space than the CBA method. However, FP-tree will meet limitations if the data has a large number of attributes, and the memory capacity is not enough to fit. CPAR is a type of association classifier based on predicted association rules. This method combines the advantages of association classification and traditional rule-based classification by generating rules directly during the training phase to avoid missing important rules.

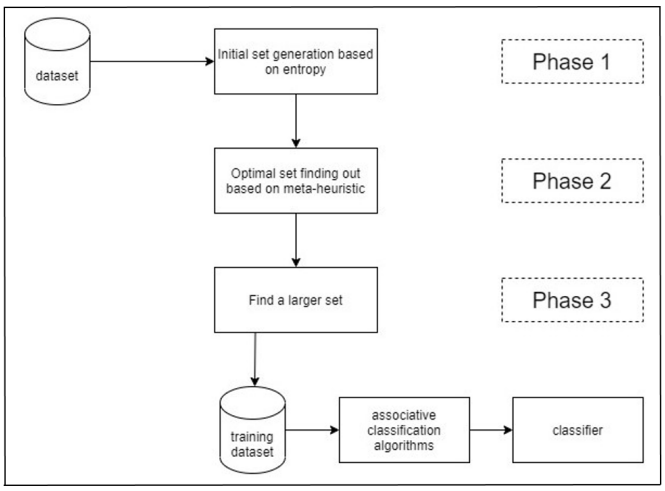

Figure 1. Three phases of meta-heuristic approach

However, association classifier still faces many limitations in terms of time and accuracy rate of classification in case of the attributes is large[5, 6, 9]. This paper proposes a metha-heuristic approach to remove unimportant attributes to enhance the associative clasification method. Our approach works in three stages as shown in Figure 1. This improvement helps descrease running time while significantly increasing accuracy rate of classification.

Phase 1 selects the initial attribute set according to the Entropy. At this stage, attributes are eliminated based on the Entropy relationship of that attribute with the label attribute. The Entropy represents a correlation to check whether the attributes play an important role with the label attribute in classification and decide whether or not to remove those attributes.

Phase 2 finds the optimal set of attributes derived from the initial set in phase 1. The original attribute set is divided into subsets and the associative classification (AC) algorithm is run based on these subsets of data. Based on the accuracy of the model, we can add or remove

randomly some attributes on the current set to obtain a new subset, and run AC repeatedly until getting the best subset of attributes. This method create subsets by using a greedy approach and evaluates the goodness of all subsets derived from the initial set in phase 1 instead of testing all combinations of possible attributes.

Phase 3 expands the optimal attribute set to increase classification accuracy. Expand the optimal set of attributes in phase 2 by adding each remaining attribute to find a new optimal set with higher accuracy.

The main contributions of the paper focus on part 2. The content of the paper is arranged as follows: part 2 describes our method in three phases, and three algorithms correspondingly, the experimental results and evidences of effectiveness are described in part 3, and the conclusion is presented in part 4.

II. OUR PROPOSED METHOD

Algorithm 1 Phase1(FSAC)

Input: minSup, minConf, minAllConf; D: dataset (D_{train}, D_{test});

Output: $model_e$(all best attributes)

1: e := entropy threshold ;
2: D_e := attrribute a_i from D_{train} where entropy(a_i, label) ≤ e;
3: neighbor=[e,e ± △];
4: e' := random(neighbor);
5: $D_{e'}$:= attrribute a_i from D_{train} where entropy(a_i, label) ≤ e;
6: **repeat**
7: $model_e$:= ACAC-RG(D_e);
8: $accuracy_e$:= $model_e$.predict(D_{test});
9: neighbor=[e,e ± △];
10: e' := random(neighbor);
11: $model_{e'}$:= ACAC-RG(D_e);
12: $accuracy_{e'}$:= $model_{e'}$.predict(D_{test});
13: **if** ($accuracy_e$ ≤ $accuracy_{e'}$) **then**
14: e:=e'
15: **end if**
16: **until** $accuracy_e \leq accuracy_{e'}$
17: **return** $model_e$

With large data sets and many attributes, the cost of selecting an optimal attribute set is huge. The proposed method combines the Entropy measure and the Meta-Heuristic approach to select features with three phases. Phase 1 is mainly filtering unimportant attributes according to the Entropy measure (the Entropy value between an attribute and the class attribute) and feature selection method in the form of Hill Climbing. The accuracy of the model after removing attributes was tested again using the cross-validation method. The selected attribute set will be the optimal attribute set for classification.

The Entropy value is in the range [0,1], attributes with large Entropy values that will not contribute much information to the classification should be eliminated. In data sets with a few attributes, we can calculate all the entropies of the attributes and arrange them from highest to lowest. Then, each attribute is checked repeatedly for removing it from high to low entropy and evaluate again the accuracy of the remaining set of attributes. If the accuracy is equal to or greater than the old value (without removing the attribute), then the removed attribute should really be removed. Because this is a bottom-up approach, the cost of eliminating the initial large data set is not feasible. We proposes an initial Entropy threshold, instead of testing Entropy in order from high to low as in the Phase 1 algorithm.

Below is FSAC[11] algorithm for phase 1, line 13 is an associative clasification algorithm, line 14 calculates the accuracy of the model obtained from line 13. The code from lines 8 to 18 repeats the improvement until the model's accuracy reaches its peak. In each iteration, based on the current entropy neighbor (lines 11, 12), and the entropy level e', the algorithm determines the attribute set De including the attributes whose entropy are less than e'. In other words, in this iteration the algorithm tried to remove attributes which are redundant from the training set. Line 15 checks whether the removal is correct or not, if so, updates the better value. After exiting the loop, the algorithm finds a relatively good set of attributes like line number 18. In line 2, we can see D is the data set divided into 2 groups Dtrain and Dtest according to the ratio 80/20. Based on Dtrain to find a set of classification rules, and Dtest is used to test the accuracy of the set of classification rules.

This task iterates until finding the best set with the same number of attributes compared to the attribute set obtained from phase 1. In each iteration, the new attribute set generated from the GenAttributes function in line 3 is tried, based on the delta value (neighborhood parameter) to give a delta number of any random attribute in the bestAttr set, and add the delta number of random attributes from the subAttr set to the bestAttr set. If the new attribute set has better results than the current bestAttr attribute set, update it again.

Algorithm 2 Phase2.1(Hill Climbing)

Input: The training dataset, initValid, delta
Output: bestAttr(all best attributes)

```
1: for i ← 0; i ≤ n; i++ do
2:     newAttr := GenAttributes(bestAttr, subAttr, delta);
3:     validation := AC(newAttr);;
4:     if validation > initValid  thenif
5:         initValid := validation;
6:         i := 0;
7:         bestAttr := newAttr
8:         subAttr := allAttr - bestAttr
9:     end if
10: end for
11: return bestAttr
```

Algorithm 3 Phase2.2 (Simulated Annealing)

Input: The training dataset, initValid, delta, T, k
Output: bestAttr

```
1: for i ← 0; i ≤ n; i++ do
2:     newAttr := GenAttributes(bestAttr, subAttr, delta);
3:     validation := AC(newAttr);;
4:     if validation > initValid  thenif
5:         initValid := validation;
6:         bestAttr := newAttr
7:         subAttr := allAttr - bestAttr
8:         i := 0;
9:     else
10:        △f=|validation - initValid|
11:        r := random r(0,1)
12:        if r > exp(-△f/kT)  then
13:            initValid := validation;
14:            bestAttr := newAttr
15:            subAttr := allAttr - bestAttr
16:            i := 0;
17:        end if
18:    end if
19: end for
20: return bestAttr
```

After implementing Phase 1, we obtain an initial set of attributes that are relatively good but not really optimal because of the limitations of the Filter method for each attribute. Phase 2 bases on the Meta-Heuristic approach, the starting solution is the initial set of attributes obtained from phase 1. At this phase, attributes are divided into two attribute groups (bestAttr and subAttr). The bestAttr attribute set contains the attributes obtained from phase 1, temporarily considered a good solution.

The subAttr set contains the remaining attributes. The Meta-Heuristic algorithms in Phase 2 find out whether there is any set that is better than the current bestAttr set. If so, the bestAttr set will be updated. This task is from lines 3 to 9 in the Phase 2 (Hill-Climbing) algorithm, and is from lines 3 to 16 in the Phase 2 (Simulated Annealing) algorithm.

After completing phase 2, we have the optimal attribute set with the same number of attributes as in phase 1. Phase 3 is the expanding the number of attribute sets to find a better value than the set of attributes obtained in phase 2. Lines 2 to 7 in the Phase 3 algorithm is a task that tries to find a more optimal extended set by combining the set obtained in phase 3. with each remaining attribute in the subAttr set.

Algorithm 4 Phase 3

Input: The training dataset, initValid
Output: bestAttr

```
1: bestAttr :=[];
2: for i ← 0; i ≤ n; i++ do
3:     newAttr := initAttr + subAttr [i];
4:     validation := AC(newAttr);
5:     if validation > initValid then
6:         initValid := validation;
7:         bestAttr := newAttrs;
8:     end if
9: end for
10: return bestAttrs
```

III. Experiments

The experimental environment is processed centrally on a computer configured with Intel(R), Core(TM) i7-6820HQ CPU @ 2.70GHz (8 CPUs), 2.7GHz and Windows 10 operating system. In this paper, the experiment has 3 phases as follows:

- Phase 1: Choose the entropy threshold and initial attribute set
- Phase 2: Find the optimal set of attributes by using HC and SA algorithms
- Phase 3: Find a larger attribute set with better results than the original attribute set

1. Simple Dataset

The first experiment is implemented on Mushroom data [2]. Because this data is used by the original ACAC[1, 10] algorithm, it has 8125 rows and 23 columns. In the data, the first attribute is a target attribute named 'class' to classify mushrooms as poisonous or non-poisonous (edible=e, poisonous=p).

In figure 2, vertical axis describes the accuracy of model, the horizontal axis describes the entropy of each removed attribute. The graph peaks at accuracy of 99.51% correspond to the entropy score of 0.757, with 8 attributes. However, the attributes are removed continuously with entropy values (0.745, 0.727, 0.714), and the accuracy still keeps at 99.51%. The classification performance will go down if one more attribute is removed. So the the entropy value of 0.714 is the best value, and gains the highest accuracy of 99.51%.

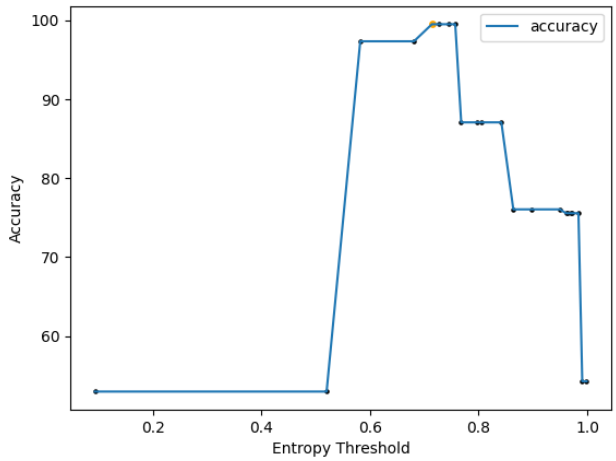

Figure 2. The accuracy of classification

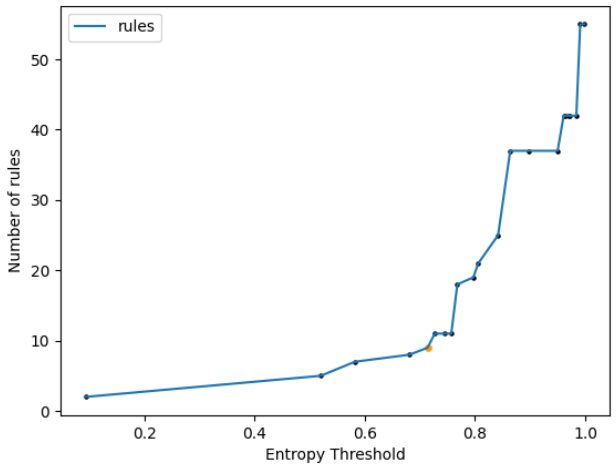

Figure 3. The optimal CAR set

After phase 1, the prediction model achieves very high accuracy in small datatsets (we can see the performance of CPAR, CBA, CMAR on 5 attributes of Mushroom in table 1), so the phase 2 and phase 3 is not necessary to implement more. Our research focuses on the challenge of associative classification problem, the large dataset with

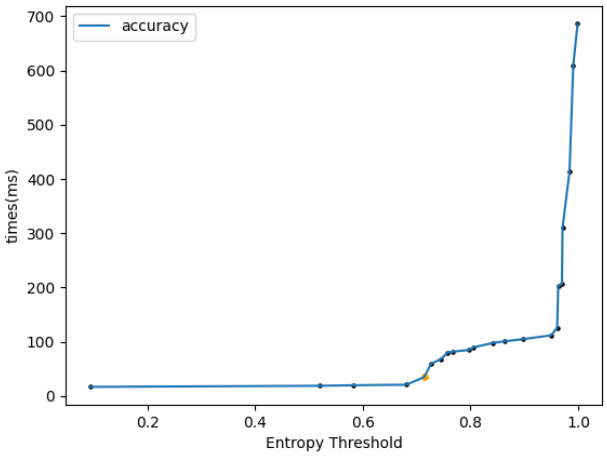

Figure 4. Running time of processing

many attributes, so we do experiments on more challenge dataset, SpamEmail.

TABLE I
REASONABLE VALUES ON MUSHROOM

	CPAR	CBA	CMAR
Accuracy	0.9951	0.9397	0.8876
Rules	9	8	14
Times(ms)	169	146	200

2. Large Dataset

The Spam Emails Dataset [3] is selected for experiment, it is a rather complex data set, with 4602 rows and 58 columns. In particular, the last attribute is a label attribute called 'spam' to classify emails as spam or spam (spam=1, spam=0). This data is also used in the ACAC algorithm. B.1 Phase 1: Selecting the Entropy threshold, the initial set. Among the remaining 57 attributes, there are many attributes that have no value in classification and need to be eliminated. To eliminate, we uses entropy to determine the ability of each attribute to contribute to the classification. Any attribute with a high entropy value will be removed from the data set, the new data set will be tried again using the ACAC method for classification. This model will be tested using cross-validation method (80% - 5000 lines of data used for training, 20% - 1001 lines of data used for testing). If the precision increases then the attribute removal is correct, and this is done until the precision reaches its highest value. In other words, work stops when accuracy is degraded.

Based on table 2, the entropy threshold = 0.91 is chosen to reduce the number of attributes from 57 to 39, thereby reducing space, processing time and improving evaluation indicators. Although the entropy threshold = 0.91 helps

TABLE II
OUTPUT OF PHASE 1 ON SPAMEMAIL

Entropy	Attributes	rules	times(ms)	Accuracy(%)
0.91	39	?	?	?
0.885	33	3108	7.783E+13	94.2
0.88	31	2239	3.5E+11	94.4
0.875	29	1612	11E+9	94.9
0.87	28	1254	261000000	94.06
0.865	27	989	51000000	93.04
0.86	26	692	10000000	92.4
0.85	24	338	2200000	90.99
0.84	22	109	35170	88.93
0.83	17	24	343	87.04
0.81	14	17	151	81.11
0.79	13	13	145	80.61

improve the limitation score compared to the original data set, the number of 39 attributes is still too large for one processing. Therefore, we propose an approach that is to divide the set of 39 attributes into 2 subsets. The first subset (original data set) includes 17 attributes at entropy threshold = 0.81, and the second subset includes the remaining 22 attributes. The reason for choosing the entropy threshold = 0.81 as the initial attribute set for experimentation is because at this threshold, although the perfomance is not the best, it is not bad to be improved and expanded more.

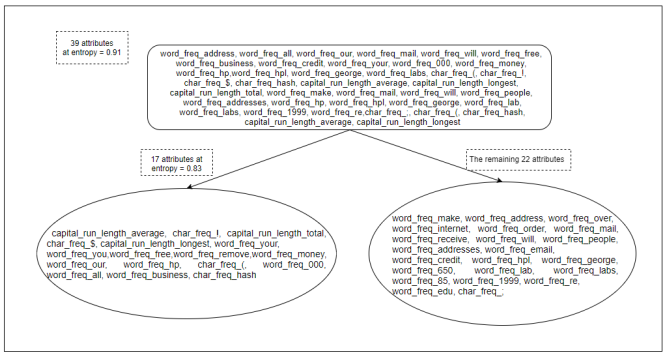

Figure 5. Divide 2 subsets from output of phase 1

a) Phase 2: Find out the optimal set

After selecting the initial attribute set, there are 17 attributes at the entropy threshold = 0.81. The next step is to find a set of 17 attributes that are more optimal than the original 17 attributes. To do that, the experimental part chooses delta values of 1, 2 and 3 respectively, each different delta value will give different results. On this experimental, we select HC and SA to implement in meta-heuristic search.

In Figure 6, it shows the peak changes (accuracy) of the data set over iterations of the program. At delta = 1, the number of peak changes when finding a set with higher accuracy than the current set is 7 times, this number at delta = 3 is 3 times and delta = 2 is 4 times. The following data sets tend to have higher accuracy than the original (0.8704) with each new peak being found, and the highest peak achievable at delta=3 across iterations is 0.9294. For delta = 3, finding the peak with accuracy = 0.9294 takes less time and also has higher accuracy than delta = 2 or delta = 3.

SA (figure 7) is considered an improved approach to improve the disadvantages of Hill Climbing, we can refer charts comparing the indicators of each approach on the same data set (17 attributes) at different delta values.

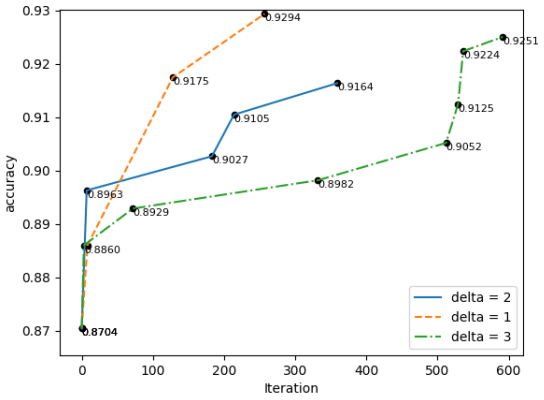

Figure 6. Wrapper with delta neighbor by SA

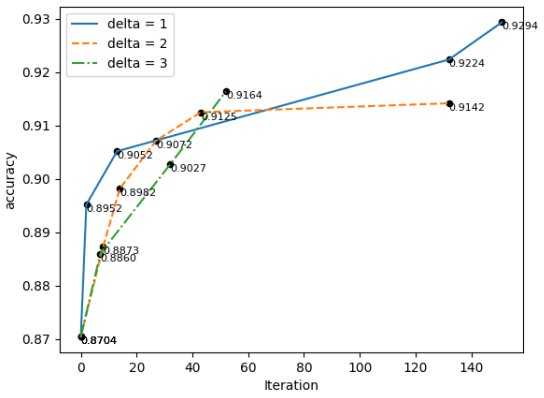

Figure 7. Wrapper with delta neighbor by HC

b) Phase 3: Find a larger attribute set that has better result than the original attribute set

From phase 2, we have selected a set of rules including 17 attributes with accuracy = 0.9294 (highest peak) and the remaining 22 attributes. In phase 3, we will find a set of 18 attributes with accuracy higher than 0.9294.

To concretize the above goal, this experimental part will combine the set of 17 attributes and each of the remaining 22 attributes. Each set of 18 attributes will give different possible results.

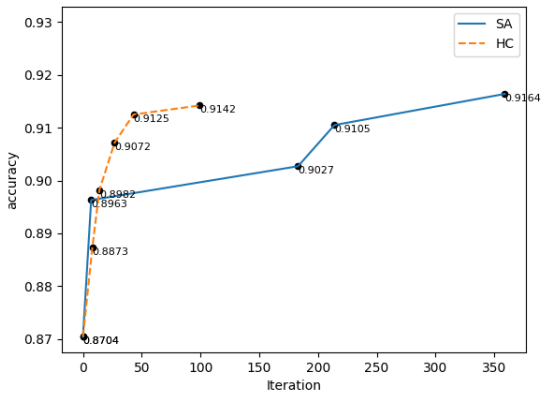

Figure 8. Comparison between HC and SA

3. Phase 3: Find a larger attribute set that has better result than the original attribute set

TABLE III
OUTPUT OF PHASE 3

Entropy	Attributes	times(ms)	rules	Accur(%)
0.308	capital_run_length_a	7862	170	9294
0.589	capital_run_length_t	7165	170	0.9294
0.658	capital_run_length_l	6891	170	0.9294
0.786	word_freq_hp	18390	170	0.9294
0.789	char_freq_(	7029	169	0.9294
0.815	word_freq_all	17124	219	0.9294
0.830	word_freq_george	19007	170	0.9294
0.835	word_freq_mail	19316	332	0.9305
0.836	word_freq_address	18231	225	0.9294
0.836	word_freq_hpl	18023	170	0.9294
0.838	word_freq_will	12262	267	0.9294
0.846	word_freq_email	13710	244	0.9294
0.866	word_freq_credit	**13292**	**278**	**0.9305**
0.871	word_freq_re	12716	194	0.9294
0.877	word_freq_people	17010	185	0.9294
0.878	word_freq_1999	18207	170	0.9294
0.884	char_freq_;	17159	327	0.9305
0.887	word_freq_addresses	17589	204	0.9305
0.892	word_freq_edu	18668	211	0.9294
0.898	word_freq_labs	17062	170	0.9294
0.898	word_freq_85	17108	170	0.9294
0.901	word_freq_650	17400	171	0.9294

From phase 2, we have selected a set of rules including 17 attributes with accuracy = 0.9294 (highest peak) and the remaining 22 attributes. In phase 3, we will find a set of 18 attributes with accuracy higher than 0.9294. To concretize the above goal, this experimental part will combine the set of 17 attributes and each of the remaining 22 attributes. Each set of 18 attributes will give different possible results.In this case chosing word_freq_addresses because of its performnance After go through 3 phase, We have the best attributes set so we can use Associative classfication algorithms to build prediction model.

TABLE IV
HEURISTIC OUTPUT OF SPAMEMAIL

	CPAR	CBA	CMA
Accuracy	0.9305	0.848	0.6916
Rules	278	184	158
Times(ms)	13292	2257889	3496

IV. CONCLUSION

The main advantage of AC is that the set of "If-Then" rules has the ability to infer new knowledge that other classification methods can not do. Another important advantage of AC is the ease of interpretation of labeling for new data. AC algorithms have limitations that often occur when minsupp is set to a very small value.

Experimental results show that AC algorithms on small and simple data such as Mushroom (23 attributes) are very effective. On large and complex data like SpamEmail (58 attributes), the algorithms show limitations (as shown in table 2).

The meta-heuristic approach still maintains almost the same classification accuracy, but running time and rule pruning are more efficient (as shown in table 4 compared to table 2). The comparison on the same classification accuracy (0.9305 vs 0.9304), processing time (13292 vs 51000000) is 3750 times faster, the number of rules is much reduced (278 vs 989). The running time is more important than the number of rules, that why we select 278 rules instead of 204 rules.

REFERENCES

[1] Z. Huang, Z. Zhou, T. He, and X. Wang, "ACAC: Associative Classification Based on All-Confidence," *IEEE International Conference on Granular Computing*, pp. 289–293, 2011.

[2] *Mushroom Classification Dataset*, [Online]. Available: `https://www.kaggle.com/datasets/uciml/mushroom-classification`

[3] *Spam Emails Dataset*, [Online]. Available: `https://www.kaggle.com/datasets/yasserh/spamemailsdataset`

[4] H. F. Ong, C. Y. M. Neoh, V. K. Vijayaraj, Y. X. Low, "Information-Based Rule Ranking for Associative Classification," *ISPACS*, 2022.

[5] M. Abrar, A. Tze and S. Abbas, "Associative Classification using Automata with Structure based Merging," *IJACSAA*, vol. 10, 2019.

[6] D. L. Olson and G. Lauhoff, "Market Basket Analysis" in Descriptive Data Mining," *Springer Singapore*, 2019.

[7] K. D. Rajab, "New Associative Classification Method Based on Rule Pruning for Classification of Datasets," *IEEE Access*, vol. 7, pp. 157783-157795, 2019.

[8] H. F. Ong, N. Mustapha, H. Hamdan, R. Rosli and A. Mustapha, "Informative top-k class associative rule for cancer biomarker discovery on microarray data," *Expert Systems with Applications*, vol. 146, 2020.

[9] Majid Seyf, Yue Xu, Richi Nayak, "DAC: Discriminative Associative Classifcation", SN Computer Science (2023) 4:401
[10] E.R.Omiecinski, "Alternative interest measures for mining associations in databases", IEEE Transactions on Knowledge and Data Engineering, vol 15, pp. 57-69, 2003.
[11] N. Q. Huy, T. A. Tuan, N. T. N. Thanh, "An efficient algorithm that optimizes the classification association rule set", VNICT 26, pp. 13-19, 2023.

Quoc-Huy Nguyen received the B.S. in Information System major from HCM City University of Science, Vietnam in 2002, and he got Doctoral Degree in Computer Science from Japan Advanced Institute of Science and Technology (JAIST), Japan in 2014. From 2008 to now, he is a lecturer in Faculty of Information Technology, Saigon University, Vietnam. His research interests are Data Mining, Computer Game, Computer Vision, Knowledge Based Systems.

Anh-Tuan Tran is currently a 4th year student majoring in Information Technology at Saigon University (SGU), Ho Chi Minh City. His research interests are Data Mining, Meta-heuristic algorithms.

Nhu-Tai Do received the B.S. in Information System major from HCM City University of Foreign Language – Information Technology, Vietnam in 2005, the M.S. in Information System Management from International University, Vietnam National University at HCMC, Viet Nam in 2017, and his Ph.D. degree in Artificial Intelligence Convergence, Chonnam National University, South Korea, in 2021. From 2005 to 2017, he was a lecturer in Faculty of Information Technology, HCM University of Foreign Languages and Information Technology, Vietnam. From 2021 to 2023, he is Postdoc researcher in the Pattern Recognition Lab, Department of Artificial Intelligence Convergence, Chonnam National University, Korea. From 2023 to now, he is the lecturer in University of Economics Ho Chi Minh City-UEH Vietnam. His interests are pattern recognition, deep learning, computer vision, video understanding, and medical analysis.