2023 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

CFAR++: Enhancing Rule Based Classifier

Md Rayhan Kabir
University of Alberta
Edmonton, Canada
rayhan.kabir@ualberta.ca
0009-0002-6898-1805

Seeratpal Jaura
University of Alberta
Edmonton, Canada
seeratpa@ualberta.ca

Osmar R. Zaiane
Alberta Machine Intelligence Institute
University of Alberta
Edmonton, Canada
zaiane@cs.ualberta.ca
0000-0002-0060-5988

***Abstract*—Over the last few years, associative classifiers have shown massive success in mining patterns using association rules. These rule-based classifiers offer a level of human interpretability, addressing a common concern stemming from several deep learning models. Various associative classifiers have been proposed over the past that have shown state-of-the-art performance. However, those classifiers suffer the limitation of requiring parametric values which vary across different datasets. Furthermore, those frameworks do not consider the statistical significance of the rules. Recently, some works have addressed this limitation by proposing an associative classifier that incorporates the idea of using statistical significance to mine association classification rules. Though the recent associative classifiers show good performance, their performance is greatly affected by the dimension of the data. In this study, we explore the weakness of the recent associative classification models and experiment with using ensemble models to overcome such limitations, particularly on aggregating the ensemble models in a concise but effective predictor. We use 10 UCI datasets for evaluation of our new approach. From our study, we find the results based on the ensemble model with a delayed pruning are very competitive and can better handle large dimensional data spaces.**

***Index Terms*—Rule based Classification, Ensemble Model, Interpretable Model.**

I. Introduction

Classification is a supervised machine-learning task where labels of the instances are predicted for the given set of inputs [1]. Most machine learning or deep learning models for classification act as a black box [2] which means the decision process is difficult to interpret for a certain set of input data. Interpretability becomes a critical concern in specific domains where understanding how a particular decision is reached holds significant importance. Furthermore, these deep learning models use an extensive amount of learning parameters which requires huge computation time. Associative classifiers are a type of rule-based classifiers that overcome to some degree the limitation of interpretability. Associative classifiers have been exploited in various areas including health care domain [3] [5] [6], the financial industry for fraud detection from credit cards [7] etc. Associative classification derives from a common and canonical data mining task for discovering associative patterns in transaction data, known as association rule mining. Association rule mining is the process of finding the relationship between the items in the dataset [4]. An association rule is a statement in the form $X \rightarrow Y$, where X is an antecedent set and Y is a consequent set both subsets of the set of all items. An associative classification rule is an association rule on which some constraints are imposed, namely that the consequent Y is a class label [11]. Since associative classifiers provide interpretable results, in the past, different studies have explored various models of associative classification. Associative classifiers such as CBA [12] make use of the Apriori algorithm to generate a set of rules for classification. Similarly, CMAR also uses the FP growth method to mine association rules [13]. Although the above classifiers resolve the issue of interpretability, they rely on prior knowledge of the parametric values such as support and confidence. These parameters would be dataset-specific and are difficult to set or tune. Recently, to overcome this limitation, Li and Zaiane [14] proposed an associative classifier named SigDirect that uses statistical significance to generate rules instead of relying on support values. The work of Li and Zaiane [14] leverages the Kingfisher algorithm [16] to find rules based on statistical dependencies. The classifier shows competitive results, however, it may produce noisy rules that could impact classifier performance. Sood and Zaiane [11] extended the work of SigDirect [14] by proposing SigD2 where a two-stage pruning strategy was introduced to prune the redundant rules to effectively reduce noise. SigD2 generates less rules, allowing better interpretability [11] and in some cases provides better accuracy compared to SigDirect. Although both SigDirect and SigD2 provide competitive accuracy, these classifiers have performance issues in high-dimensional spaces. We speculate that the limitation comes from the implementation of the Kingfisher algorithm using a one hot vector representation [16]. Since both classifiers use the Kingfisher algorithm, they also inherit the limitations of Kingfisher in terms of dimensionality restrictions.

To remedy the performance issue, we attempt to solve the high dimensionality problem by dividing the large feature space in a clever ensemble procedure. Very recently, Kabir and Zaiane [17] attempted to deal with the issue of high-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ASONAM '23, November 6-9, 2023, Kusadasi, Turkey
© 2023 Association for Computing Machinery.
ACM ISBN 979-8-4007-0409-3/23/11…$15.00
https://doi.org/10.1145/3625007.3627497

dimensional feature processing in the Kingfisher by proposing CFAR, an ensemble model that considers a subset of feature vectors during training using a base learner. We are inspired by the CFAR model and propose CFAR++ which attempts to find the significance of rules using the entire training dataset and delays the rule pruning after aggregation to benefit from the entire rule set. The main contributions of this study are as follows:

1) We exploit the idea of CFAR [17] by changing the pruning strategy to improve the performance of previous associative classifier in terms of accuracy and number of rules.
2) The approach shows competitive performance in terms of accuracy, memory consumption and run-time.

The rest of the paper is organized as follows: Section II discusses the related works, Section III provides detail of our proposed model, Section IV provides the specific description of the dataset. Section V discusses the results of the proposed model in contrast to other state-of-the-art models, and finally, Section VI provides our concluding remarks and future research direction.

II. Related work

Over the last few years, the idea of developing rule-based classifiers has been given a boost due to their performance and interpretability. The work done by Bayardo [20] used association rule mining to discover classification rules using a brute force technique. Liu et al. [12] investigated a classifier named CBA that performs classification based on the set of association rules referred to as class association rules (CARs). The classifier utilizes the Apriori algorithm to generate CARs and then prune the redundant ones. Followed by their work, many different associative classifiers were introduced. CMAR [13] uses the FP growth method to mine association rules. The CMAR pruning is performed using different parameters like confidence value, the coverage in the database, and its correlation. Later, CPAR: Classification based on Predictive Association Rules was introduced in the work of Yin and Han [15]. This work focuses on two limitations: first, a large number of rule generations, and second, the comparison based on the confidence value. Their approach makes use of both associative classifiers and rule-based techniques to exploit the training set using a greedy approach during rule generation. Another work by Zaiane and Antonie [21] also attempted to investigate the limitation of associative classifiers which generate a large number of rules by introducing a new pruning strategy.

One of the limitations of the above classifiers is that they require the use of parametric values like support and confidence and these values are data-specific and very difficult to tune. To overcome this limitation, Li and Zaiane [14] introduced an associative classifier named SigDirect that mines rules based on statistical significance. In their work, they stress that a strong support value does not necessarily indicate strong significance. To find the significance of rules, the authors extend the work of Hamalainen [16] to mine rules based on significance and not based on the frequency of the rule during the rule generation process. Though their model showed good performance, it was observed that SigDirect produces a large number of rules that might add noise to the final classification. Sood and Zaiane [11] introduced two-stage pruning where they add an additional pruning step in SigDirect. The work of Sood and Zaiane did extensive pruning by significantly reducing the number of rules in a few datasets without compromising the accuracy of the classifier. Very recently, Kabir and Zaiane proposed Classification by Frequent Association Rules (CFAR) [17] where they focused on dealing with the performance issue arising due to high dimensional features. Their approach has three main phases. The first step follows the classic ensemble approach with SigD2 used as the base classifier. The base classifier in this step performs both rule generation and pruning. In the next steps, the rules are aggregated based on their relative frequency and compared against a certain frequency threshold. These steps consider finding the optimal threshold value automatically. The final step performs prediction based on the joined rules . Their results indicate that an ensemble-based approach could be a potential way to resolve the performance issues of the Kingfisher algorithm in the SigD2 classifier - ie. limited to about 30 dimensions. From this we are inspired to extend the CFAR model and propose CFAR++.

III. Proposed Method

We explore a different approach to deal with the performance issue by articulating that the issue arises with high dimensional data in the Kingfisher algorithm.

From the literature we found that an ensemble method can solve the performance issue of associative classifier in the case of dataset with high dimension [17], [18], [19]. In the case of using an ensemble of Associative classifiers for rule based classification, one can collect all the rules learned from the base learners to form the final model. However, selecting all the rules can include noisy rules which affects the performance of the model. To solve this they selected the rules which are very frequent among the base learners. Getting inspired from their work, we adopt and exploit the same idea. For selecting the frequent rules, they introduced the term Relative Frequency Ratio(RFR). We also use this value. In addition, we use Fisher's exact test [30] for selecting rules.

While designing CFAR, the authors speculated that the base learner may generate many rules which might not add information in the classification process rather introduce noise in the process. For this reason they present a simple pruning approach based on the frequency of the rules among the base learners. However, in their approach the statistical significance of the rules over the whole dataset was not considered while selecting the rules. We further speculate that there can be many rules which might be frequent among the base learners but not statistically significant for the entire dataset. Since each of the base learners selects a subset of the features, it is possible that one pair of features might be selected often among base learners and therefore many base learners may produce the

same rule with these features. For this reason we introduce an additional pruning strategy where we consider the statistical significance of the rules on the entire dataset with their frequency which might further enhance the performance of the ensemble model by removing noisy rules. In our approach there are four main steps:

1) Training base learners: For the Rule generation phase we split the entire dataset into 80%-20% where we use 80% of the data for training and the rest 20% for validation. Then we again split the training data into 80%-20% and now we use the 20% data for testing purpose. To train each base learner we take a random subset of the feature vector of size 30. This subsample of the features are randomly selected with replacement and shuffle. After training 100 base learners with the training data, we gather all the rules generated by the base learners.

2) Find optimal T value: Instead of keeping all the rules gathered from the whole ensemble, the idea is to keep only the most popular rules among base learners and discard the rest. In this step we select rules based on frequency first. For each rule, the relative frequency ratio is calculated to select only those rules which are more frequent. The Relative Frequency Ratio of a rule is compared against the threshold value T to identify if the rule is frequent or not. We calculate the RFR of any rule R by Equation 1.

$$Relative\_frequency\_ratio(R) = \frac{Frequency(R)}{max\_frequency} \quad (1)$$

where $Frequency(R)$ is the number of base learners that derived R and $max\_frequency$ is the largest frequency of any rule. If the RFR of a rule is greater than T, then the rule is selected otherwise it is discarded for the next step. To empirically find the optimal value of T, we need to do experiments attributing different values for T. From our experiment we find there is no specific value for this threshold which provides optimum result for all the dataset. We plot the performance of the model for different values of T in Figure 2. From Figure 2, we see for different datasets different values of T provide best results. Thus in this step we test the model with different values of T starting from 0.0 to 1.0 and select the value which provides the best result on test data. We provide the algorithm of finding optimal T value in Algorithm 1 which we adopted from [17] and made necessary changes. We also provide in Table II a sample of selected rules using threshold value T for one of the dataset that we used. From Table II we can see, the number of selected rules are only 13 for the final prediction whereas from our experiment we find the total generated rules by the base learners was 1848. This makes a huge advantage for both classification process in terms of memory and run time and for the interpretation of result. Indeed, to understand decision process of the classifier, we only need to analyze 13 rules. From Table II we can see it is very convenient with a small number of rules to interpret the learned model and understand the decision of the classification process.

Algorithm 1 Rules selection based on RFR

Input: Rules: all rules generated by base learners; **T:** Threshold value to select Rules.
Output: Selected_rules: rules with RFR $\geq$ T

```
Selected_rules ← []
for r in Rules do:
    if RFR(r) ≥ T:
        Selected_rules.append(r)
    end if
end for
return Selected_rules
```

TABLE I
UCI DATASET INFORMATION

Dataset	Transactions	Features	Class
Anneal	898	67	5
Adult	48842	95	2
Ionosphere	351	155	2
Heart	303	47	5
Horse	368	83	2
Hepatitis	155	54	2
Glass	214	41	6
PageBlocks	5473	41	5
Pima	768	36	2
Wine	178	65	3

TABLE II
RANKED SELECTED RULES IN STEP 2 FROM THE HORSE DATASET WITH THEIR FREQUENCY AND RFR. EACH RULE IS IN THE FORM OF "ANTECEDENT → CLASS LABEL;(SUPPORT, CONFIDENCE, -LN(P-VALUE))" WHERE ANTECEDENT IS A CONJUNCTION OF TOKENIZED FEATURES. FOR READABILITY, TOKENS COULD BE MAPPED BACK TO FEATURES (ATTRIBUTE-VALUE PAIRS)

	Selected Rules	Frequency	RFR
1	0 → 0;(0.5277,0.873,-54.379)	40	1
2	63 → 1;(0.0340,0.889,-6.068)	39	0.975
3	49 → 0;(0.0894,0.840,-4.528)	33	0.825
4	1 → 1;(0.3064,0.791,-57.129)	31	0.775
5	64 → 0;(0.1447,0.944,-13.211)	26	0.65
6	55 → 0;(0.1191,0.903,-8.601)	25	0.625
7	60 → 0;(0.1915,0.738,-4.093)	25	0.625
8	15 → 0;(0.4000,0.718,-8.093)	25	0.625
9	48 → 0;(0.1957,0.821,-8.635)	24	0.6
10	54 → 0;(0.1064,0.833,-5.093)	24	0.6
11	78 → 0;(0.1447,0.919,-11.476)	24	0.6
12	62 → 1;(0.0681,0.800,-9.304)	24	0.6
13	12 → 1;(0.0936,0.595,-5.567)	24	0.6

3) Pruning: After selecting rules in the previous step, we consider testing each of the selected rules for statistical significance on the entire dataset. If any of the selected rules is not statistically significant, we consider those rules as noisy rules and prune them. For this we adopted the work of Li and Zaiane [14] where they measured the statistical significance

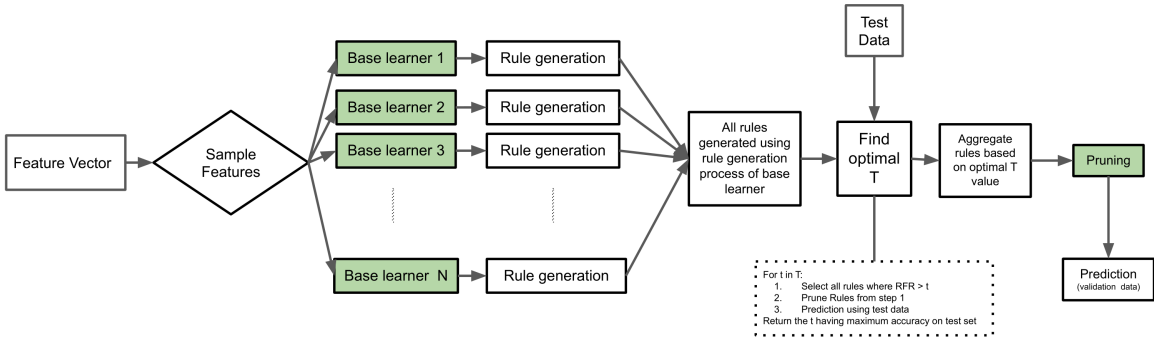

Fig. 1. Proposed model of CFAR++

TABLE III
COMPARISON BASED ON THE ACCURACY

Dataset	SigD2	CFAR using SigDirect	CFAR++ using SigDirect	CFAR using SigD2	CFAR++ using SigD2
Anneal	0.9238	0.8165	0.9562*	0.8261	0.9516*
Adult	0.8363	0.8282	0.8293	0.8118	0.8354
Ionosphere	0.9038	0.9275	0.9233	0.9404	0.9184
Heart	0.5572	0.5294	0.5119	0.5368	0.5157
Horse	0.7568	0.8187	0.7802	0.8325	0.7818
Hepatitis	0.7718	0.7767	0.8182*	0.7819	0.8056*
Glass	0.5562	0.5777	0.5903*	0.5834	0.5979
PageBlocks	0.9045	0.3164	0.9155*	0.4058	0.9155*
Pima	0.7154	0.7818	0.8015*	0.7936	0.8064
Wine	0.8637	0.8928	0.8226	0.9114	0.8329
Average	0.7789	0.7265	0.7949	0.7424	0.7961

* In paired t-test, improvement of CFAR++ over CFAR is statistically significant

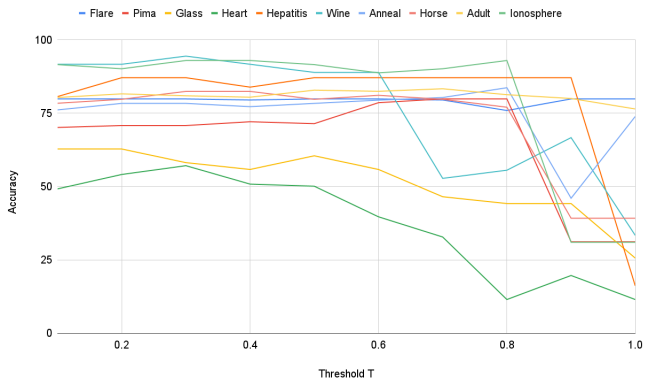

Fig. 2. Performance of the model for different values of Threshold T on 10 different datasets

of each of the class association rules on the dataset. We use an instance centring approach for this purpose. We count how many instances from the training dataset select each of our rules. With this count value we calculate the P_F value of each of the rules. If P_F is greater than equal to a certain threshold values i.e $P_F \geq \alpha$ we prune that rule. In our case we use $\alpha = .05$, the conventional P-value used in most statistical research. With this value of α we prune rules from the set which were selected in the previous step.

4) Class label prediction: The last step is the class label prediction. From the previous step we get the final selected rules (representing the learned model) that we use for our classification process to predict the class label of a given instance. For each of the instances of the validation data, we select the applicable rules and apply those rules to predict the class label for each of the instances. With the predicted class label, we calculate the performance of the model. The whole architecture is shown in Figure 1.

IV. DATASET

We use 10 different UCI datasets [26] to evaluate our proposed model CFAR++. These are the same datasets used in previous studies. Before using a dataset we discretize the numerical values as stated in [27]. We convert the features to a binary feature vector. We used the same vector form of discretized values for all our experiments with all contenders.

TABLE IV
COMPARISON BASED ON THE AVERAGE NUMBER OF RULES SELECTED FOR PREDICTION

Dataset	SigD2	CFAR using SigDirect	CFAR++ using SigDirect	CFAR using SigD2	CFAR++ using SigD2
Anneal	42.0	18.5	35.9	14.9	33.3
Adult	64.0	56.5	57.4	54.7	54.2
Ionosphere	67.0	92.9	33.6	121.1	33.8
Heart	60.0	101.7	28.6	84.25	27.6
Horse	83.0	30.7	31.6	71.8	25.2
Hepatitis	26.0	5.8	21.1	5.2	15.8
Glass	46.0	126.8	39.0	113.8	35.0
PageBlocks	24.0	20.55	25.0	27.9	22.6
Pima	17.0	13.8	11.4	15.05	11.0
Wine	23.0	60.05	18.0	66.2	14.0
Average	45.20	52.73	30.16	57.49	**27.25**

TABLE V
COMPARISON ON MEMORY CONSUMPTION (MEGABYTE)

Dataset	SigD2	CFAR using SigDirect	CFAR++ using SigDirect	CFAR using SigD2	CFAR++ using SigD2
Anneal	140.0	150.25	154.1	153.6	150.3
Adult	572.0	568.85	545.9	561.95	568.65
Ionosphere	523.0	142.7	145.95	145.75	141.75
Heart	176.0	212.15	221.2	286.95	212.1
Horse	440.0	146.15	152.9	148.05	148.5
Hepatitis	201.0	167.3	175.15	182.7	171.25
Glass	105.0	149	157.1	148.3	147.6
PageBlocks	125.0	182.95	208.8	198.3	175.5
Pima	103.0	154.85	157.8	163.2	159.35
Wine	107.0	135.3	137.7	137.5	138.6
Average	249.20	200.95	205.66	212.63	201.36

As we mentioned earlier, to test our model CFAR++, we use 20% of the dataset to validate our model. We further divide the rest 80% of the data into train and test data in the ratio of 80% and 20% respectively. For all other models, we use 80% of the data to train the model and rest 20% is used as test data. We show the information of the dataset in Table I.

V. RESULT ANALYSIS

We experimented with our ensemble approach and compared it against SigD2, and CFAR. We report the result of CFAR and CFAR++ with an average of over 20 runs for each dataset. For CFAR and CFAR++, experiments with both the SigDirect and SigD2 as a base learner as SigDirect uses a one step pruning strategy and SigD2 use a two step pruning strategy. We are interested to observe how our proposed model performs using both of them as base learners. While using SigD2, we use a 30% confidence threshold since in the proposal of SigD2 [28], the authors conducted sensitivity analysis on the threshold and show that a threshold value from 30% to 50% yields the best results. As CFAR has effectively dealt with the performance issue, therefore, one of the main goals of CFAR++ was to make it efficient by reducing the number of rules and improving accuracy or not compromising it significantly.

TABLE VI
COMPARISON OF RUN TIME IN (SECONDS)

Dataset	SigD2	CFAR using SigDirect	CFAR++ using SigDirect	CFAR using SigD2	CFAR++ using SigD2
Anneal	4.41	12.04	16.01	22.23	18.87
Adult	88.96	284.25	175.62	639.91	262.57
Ionosphere	1204.71	8.87	10.19	9.90	11.39
Heart	4.83	164.52	102.40	53.36	94.68
Horse	16.65	10.26	11.66	11.93	12.53
Hepatitis	5.04	12.53	17.12	13.86	13.56
Glass	0.50	11.53	22.37	15.98	23.61
PageBlocks	10.20	73.48	74.17	142.05	75.79
Pima	0.322	15.48	18.01	19.18	17.95
Wine	0.254	4.55	4.71	5.04	4.53

1) Accuracy: Table III shows the different approaches compared against the accuracy measure, We observe, in most cases, the ensemble-based approach surpasses the accuracy test on most of the datasets when compared against the original SigD2. Specifically, comparing CFAR++ and CFAR, we find, CFAR++ performed relatively well on a majority of datasets. For Ionosphere, Horse, and Wine, CFAR performs well, however, CFAR++ still provides competitive performance. On average, given all 10 datasets, CFAR++ performs better equally with SigDirect or SigD2 as base learner.

2) Comparison based on number of rules: With the associative classifier, the number of rules plays a significant role. The more rules constitute the learned model, the less interpretable it is. The goal is to have the least number of rules without compromising the accuracy. Table IV compare the different approaches based on the number of rules used by the classifier for prediction. When CFAR++ is compared against SigD2, in some cases the number of rules is very close. However, for datasets like Ionosphere, Wine, Heart, Horse, and Hepatitis, CFAR++ generates way less rules. When comparing CFAR and CFAR++ from the Table IV, it is to be noted that for some datasets, the number of rules generated by CFAR++ is lower. For example, with the Glass dataset, the number of average rules generated by the CFAR classifier was 126.8/113.8, however, with CFAR++ the number of average rules is 39.0/35.0 which is significantly smaller. Furthermore, by comparing the accuracy of CFAR and CFAR++ models on the Glass dataset in Table III, the CFAR++ has gained major accuracy which indicates that CFAR++ generates more non-

redundant rules and yields better accuracy.

3) Memory analysis: The memory consumption was measured using *psutil* python library. Table V highlights the comparison between the different models SigD2, and two ensemble-based approaches, CFAR and CFAR++, using base learners. When CFAR++ is compared against CFAR, we observe that in most cases the memory consumption of both methods is very close. However, when CFAR++ is compared against the original SigD2, CFAR++ consumes comparatively less memory for datasets like Adult, Ionosphere, Horse, and Hepatitis. This is because despite the overhead generated by the ensemble, each base learners deals only with a subspace of the feature set while SigD2 struggles with large dimensions. These results indicate that ensemble-based is a possible direction to improve performance based on memory.

4) Run Time: It is observed from Table VI that SigD2 seems faster. This is expected because both CFAR and CFAR++ are ensembles that call SIGD2 or SigDirect 100 times. Still, CFAR++ and CFAR have shorter runtime on Horse and Ionosphere. It is to note that the runtime greatly depends on other processes running while experimenting with a certain classifier. In addition to that the runtime that we record in our experiment is on a different device from the work of the previous authors [11] [17]. Thus our reported runtime is slightly different from the one stated in previous works.

VI. CONCLUSION

In this study we persist in improving the idea of an ensemble of associative classifiers to address performance limitations of the Kingfisher algorithm built-in the already excellent SigDirect and SigD2 due to high dimensionality. From our experiment we find that our ensemble approach CFAR++, where we change the pruning strategy of CFAR, also provides a competitive performance. Both of the approaches improve the performance of previous classifier in terms of memory consumption and run time.

REFERENCES

[1] Kotsiantis, S.B., Zaharakis, I., Pintelas, P., et al.: Supervised machine learning: A review of classification techniques. Emerging artificial intelligence applications in computer engineering 160(1), 3–24 (2007)

[2] Lei, D., Chen, X., Zhao, J.: Opening the black box of deep learning. arXiv preprint arXiv:1805.08355 (2018)

[3] Zaiane, O., Antonie, M-L., Coman, A.: Mammography Classification by an Association Rule-Based Classifier, Third Intl. ACM SIGKDD Workshop on Multimedia Data Mining, pp. 62-69, Edmonton, Canada, (2002).

[4] Solanki, S.K., Patel, J.T.: A survey on association rule mining. In: 2015 Fifth International Conference on Advanced Computing Communication Technologies, pp. 212–216 (2015).

[5] Soni, S., Vyas, O.: Using associative classifiers for predictive analysis in health care data mining. International Journal of Computer Applications 4(5), 33–37 (2010)

[6] Anand, T., Pal, R., Dubey, S.K.: Data mining in healthcare informatics: Techniques and applications. In: 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), pp. 4023–4029 (2016)

[7] Kamaruddin, S., Ravi, V.: Credit card fraud detection using big data analytics: Use of psoaann based one-class classification. In: Proceedings of the International Conference on Informatics and Analytics. ICIA-16. Association for Computing Machinery, New York, NY, USA (2016).

[8] Han, J., Cheng, H., Xin, D., Yan, X.: Frequent pattern mining: Current status and future directions. Data Min. Knowl. Discov. 15(1), 55–86 (2007).

[9] Pillai, J., Vyas, O.: User centric approach to itemset utility mining in market basket analysis. Int J Comp Sci & Eng 3(1), 393–400 (2011)

[10] Srikant, R., Agrawal, R.: Mining generalized association rules. Future Generation Computer Systems 13(2), 161–180 (1997).

[11] Sood, N., Zaiane, O.: Building a competitive associative classifier. In: Song, M., Song, I.-Y., Kotsis, G., Tjoa, A.M., Khalil, I. (eds.) Big Data Analytics and Knowledge Discovery, pp. 223–234. Springer, Cham (2020)

[12] Liu, Bing, Wynne Hsu, and Yiming Ma. "Integrating classification and association rule mining." Proceedings of the fourth international conference on knowledge discovery and data mining. 1998.

[13] 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 (2001).

[14] Li, J., Zaiane, O.R.: Exploiting statistically significant dependent rules for associative classification. Intell. Data Anal. 21(5), 1155–1172 (2017).

[15] Yin, X., Han, J.: Cpar: Classification based on predictive association rules. In: SDM (2003)

[16] Hämäläinen, W.: Kingfisher: An efficient algorithm for searching for both positive and negative dependency rules with statistical significance measures. Knowledge and Information Systems - KAIS 32, 1–32 (2011).

[17] Kabir, M.R., Zaiane, O.: Classification by Frequent Association Rules. ACM Symposium on Applied Computing (Data Mining track), Tallinn Estonia (2023)

[18] Kabir, M.R., and Zaiane, O.: "Dynamic Ensemble Associative Learning." 2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). IEEE, 2022. APA

[19] Kabir, Md Rayhan, Vaid, S., Sood, N., and Zaiane, O.: "Deep Associative Classifier." 2022 IEEE International Conference on Knowledge Graph (ICKG). IEEE, 2022.

[20] Bayardo, R.J.: Brute-force mining of high-confidence classification rules. In: Proceedings of the Third International Conference on Knowledge Discovery and Data Mining. KDD'97, pp. 123–126. AAAI Press, (1997)

[21] Zaïane, O.R., Antonie, M.-L.: On pruning and tuning rules for associative classifiers. In: Khosla, R., Howlett, R.J., Jain, L.C. (eds.) Knowledge-Based Intelligent Information and Engineering Systems, pp. 966–973. Springer, Berlin, Heidelberg (2005)

[22] Singh, J., Ram, H., Sodhi, D.J.: Improving efficiency of Apriori algorithm using transaction reduction. International Journal of Scientific and Research Publications 3(1), 1–4 (2013)

[23] Fisher, Ronald A. "Statistical Methods for Research Workers: New York City." Springer,

[24] Hämäläinen, W.: Efficient discovery of the top-k optimal dependency rules with Fisher's exact test of significance. In: 2010 IEEE International Conference on Data Mining, pp. 196–205 (2010).

[25] Webb, G.I., Zhang, S.: K-optimal rule discovery. Data Mining and Knowledge Discovery 10, 39–79 (2005)

[26] Dua, D., Graff, C.: UCI Machine Learning Repository (2017). http://archive.ics.uci.edu/ml

[27] Coenen, F.: The LUCS-KDD software library (2004). https://cgi.csc.liv.ac.uk/%CB%9Cfrans/KDD/Software/LUCS-KDD-DN/lucs-kdd DN.html

[28] Sood, N.: Building a Competitive Associative Classification Model. (2020).

[29] Pearson, K.: On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. Philosophical Magazine 5(50), 157–175 (1900)

[30] Upton, Graham JG. "Fisher's exact test." Journal of the Royal Statistical Society: Series A (Statistics in Society) 155.3 (1992): 395-402.