

# Bi-Level Associative Classifier using Automatic Learning on Rules

Nitakshi Sood<sup>1,2</sup>, Leepakshi Bindra<sup>1</sup> and Osmar Zaiane<sup>1,2</sup>[0000–0002–0060–5988]

<sup>1</sup> University of Alberta, Edmonton, Canada

<sup>2</sup> Alberta Machine Intelligence Institute

{nitakshi, leepaksh, zaiane}@ualberta.ca

**Abstract.** The power of associative classifiers is to determine patterns from the data and perform classification based on the features that are most indicative of prediction. Although they have emerged as competitive classification systems, associative classifiers suffer from limitations such as cumbersome thresholds requiring prior knowledge which varies with the dataset. Furthermore, ranking discovered rules during inference rely on arbitrary heuristics using functions such as sum, average, minimum, or maximum of confidence of the rules. Therefore, in this study, we propose a two-stage classification model that implements automatic learning to discover rules and to select rules. In the first stage of learning, statistically significant classification association rules are derived through association rule mining. Further, in the second stage of learning, we employ a machine learning-based algorithm which automatically learns the weights of the rules for classification during inference. We use the p-value obtained from Fisher’s exact test to determine the statistical significance of rules. The machine learning-based classifiers like Neural Network, SVM and rule-based classifiers like RIPPER help in classifying the rules automatically in the second stage of learning, instead of forcing the use of a specific heuristic for the same. The rules obtained from the first stage form meaningful features to be used in the second stage of learning. Our approach, BiLevCSS (**Bi-Level Classification using Statistically Significant Rules**) outperforms various state-of-the-art classifiers in terms of classification accuracy.

**Keywords:** Associative classification · Classification rules · Statistical significance.

## 1 Introduction

Classification is the process of organizing and categorizing data into distinct classes. It involves various tasks like building a model based on the distribution of the data in consideration and further using this model for identification of the class label of new data. An associative classifier is a kind of supervised classification model that learns on association rules that attribute features with class labels. The association rule mining identifies patterns in the data by extracting

associations between items in a dataset. The class association rules (CARs) obtained from mining are represented in the form,  $X \rightarrow Y$ , where  $X$  and  $Y$  are the antecedent and consequent respectively. For an associative classifier, we choose the consequent to be the class label while the antecedent set includes the set of items that are highly indicative of their association with the class label based on association rules.

Most of the previously proposed associative classification algorithms like CMAR [16], CBA [17] and CPAR [19] have different rule discovery, rule pruning, rule prediction and evaluation methods. However, a predefined weighting scheme is required, for each of these methods in order to predict the class from the association rules. Heuristics like maximum/minimum of confidence, average of confidence or sum of confidence of the rules for the classes can be used to decide the predicted value for the new samples. However, the weighting scheme may differ for various applications when using associative classifiers. Deciding the heuristics to select rules to apply during inference and therefore to predict the class from the derived classification rules is a challenging task, and is typically fixed as part of the algorithm.

This form of classification offered by associative classifiers is easily understandable, flexible and does not assume independence between attributes, however, it requires prior knowledge to choose appropriate support and confidence threshold values for rule mining. Moreover, they contain a large number of noisy rules which are redundant, uninteresting and lead to longer classification time. Various pruning techniques have been designed to deal with this limitation, for instance, removing the low ranked specialized rules, removing conflicting rules or using database coverage based pruning strategy. A two level classification method was initially proposed by Antonie and Zaiane in [4], where the first stage used Apriori-based approach [1] to generate associative rule classification model which is followed by a stage of machine learning classifiers to learn the weights for classification in the second stage. We extended their work and compare the performance of SVM [10], Neural Networks [6] and RIPPER [9] in the second stage of learning. Although, this automatic approach of learning to use the rules is expected to give better classification results, it suffers with certain limitations. Firstly, the setting up of an optimal support and confidence threshold values to mine the rules in the first stage is a cumbersome task. Secondly, the rules generated using the former approach may contain noisy, non statistically significant rules and may not cover all the important features in the selected rules.

Therefore, in order to address the above given limitations, we propose BiLevCSS (**Bi-Level Classification using Statistically Significant Rules**), which uses statistically significant rules generated from a first stage, to form features that are made full use of, for classification in the second stage of learning. We follow the approach proposed by Li and Zaiane in [15] for generation of statistically significant CARs. We also use Fisher's exact test to obtain the p-value which is used to determine the statistical significance of the association rules. We further extract features from these significant association rules and then train the supervised

learning classifiers like Neural Network, SVM and RIPPER on them. Finally, the trained model from the second stage is used to find the class label for a new data point.

Traditional association rules mining methods mostly prune the infrequent items on the basis of frequency of the itemset and thereafter calculate the strength of the rule in the form of its confidence values. This also ignores the statistically significant rules. Although most of the associative classifiers deal with this limitation by setting up small minimum threshold values, however, this leads to the generation of a huge number of insignificant rules. Therefore, in our proposed model, we use the instance-centric pruning strategy as used in SigDirect [15] to find globally optimal CAR (Class Association Rules) for each instance in the training dataset without compromising the classification accuracy.

Furthermore, we use Neural networks [6] and Support Vector Machines [10] in our approach as they are strong machine learning classifiers, that have proved their worth in various applications. With the aim to build an efficient classification strategy, we train them using meaningful features obtained from the first stage of learning. However, many real time applications specifically in health-care and medicine require explainable models in order to interpret the results post classification. In our proposed strategy, although the statistically significant rules and derived features obtained in the first stage form an explainable model, Neural Network and SVM used in the second stage for classification might make the results un-explainable for such applications. Therefore, in order to make our approach interpretable, we explored the applicability of a rule-based classifier like RIPPER in the second stage for classification of derived features. Ripper [9] is a rule-based classifier which was found to produce a minimal set of explainable classification rules when given meaningful features in the second stage of our proposed approach, without compromising on the classification accuracy.

Therefore, in our study we propose a novel bi-level classification model, which uses the association rule mining to produce statistically significant rules. Further these rules are used to form more meaningful and non redundant features to be given as input in the second stage of learning comprised of a second classifier. The proposed algorithm helps in automatic learning of non noisy, statistically significant rules and further, it leads to a higher classification accuracy. The main contributions of this work are:

- We propose BiLevCSS, (**Bi-Level Classification using Statistically Significant Rules**), which is an effective two stage learning model. In the first stage of learning, we build an associative rules classifier (ARC) model based on statistically significant rules, followed by a supervised learning classifier in the second stage of learning for classification.
- We evaluate the performance of Neural Networks and SVM against rule-based classifier RIPPER to compare their accuracy and suitability for different datasets when used in the second phase of BiLevCSS.
- We evaluate the proposed algorithm BiLevCSS on 10 UCI datasets and with other commonly used classifiers on the basis of classification accuracy. The

results show that our classifier gives better classification accuracy than various state-of-the-art classifiers.

The rest of the paper is organized as follows: Section 2 gives a literature review about some previously proposed associative classifiers; Section 3 explains the methodologies we have adapted in our algorithm; Section 4 shows the evaluation results of our proposed classifier on UCI Datasets; and lastly, Section 5 gives the conclusion and directions about our future work.

## 2 Related Work

The idea of associative classifiers was first presented by Liu et al. [17], while the concept of using association rules as CARs was proposed earlier by Bayardo Jr. [5]. Liu et al. proposed CBA, an approach to perform classification using the class association rules in [17]. The proposed work used Apriori based rule generation algorithm, involving the cumbersome process of tuning support and confidence values. Furthermore, CBA applies the paradigm of “database coverage” for rule pruning and uses highest ranked matching rules as the heuristic for classification. This work paved the way for the associative classification. Li et al. proposed another associative classifier called CMAR in [16]. CMAR uses FP-growth [12] which is a frequent pattern mining based approach to produce a set of association rules. The authors also use a novel data structure called CR-tree to store the CARs. Furthermore, CMAR determines the class label based on the set of matching rules using weighted chi-square measure. Antonie and Zaiane propose an associative rule-based classifier by category for automatic text categorization called ARC-BC [2]. ARC-BC forms association rules grouped by the category for each set of documents. The average confidence value is calculated for each category and finally the class label of the group with highest confidence value is considered as the predicted category. The proposed algorithm works for both single and multi class.

Antonie and Zaiane further proposed the first associative classifier that uses both the positive and negative CARs in [3]. They use Pearson’s coefficient as the interestingness measure to mine positively and negatively correlated CARs. They were able to prove that a much smaller set of positive and negative CARs was efficient enough to compete and outperform various other categorization systems. The classification is made by using an average confidence heuristic.

Coenen and Leng have reviewed three case satisfaction mechanisms namely, Best First Rule, Best K Rules and All Rules in [8] and various alternative rule ordering strategies. The authors have evaluated these case satisfactions as they have been commonly used in numerous Classification Association Rule Mining (CARM) algorithms to use the classifier thus formed, for the prediction task.

A two stage classification model called 2SCARC was proposed in [4], which automatically learns to use the rules for classification. Antonie and Zaiane used an Apriori based algorithm in the first stage to generate features from class association rules, which are given to the next stage for training a Neural Network to automatically learn the weights for classification. The main aim of this work

was to overcome the cumbersome task of tuning support and confidence values for every dataset. Although, the results obtained are interesting, however they are not convincing as they tend to ignore the statistical significance of the rules. Noisy and meaningless rules produced in the first stage might mislead the classification in the second phase. This forms the baseline of our work as described in further sections.

Furthermore, Li et al. presented a novel associative classifier which is built upon both positive and negative association classification rules in [14]. The proposed classifier incorporates, rule generation where statistically significant positive and negative CARs are discovered and a rule pruning phase where irrelevant rules are pruned. Further, these rules are used for the prediction of the unlabeled data. They propose a very efficient rule pruning strategy so as to prune both negative and positive CARs simultaneously. Li et al. concluded that summing up the confidence values of all matching rules and accordingly making the class label prediction proved to be the best classification strategy. Li et al. have also presented an associative classifier called SigDirect [15] which produces statistically significant and meaningful rules for classification. The authors have obtained globally optimal association rules using a novel instance-centric rule pruning strategy instead of more prevalent pruning strategy like database coverage. Li et al. evaluate various heuristics for the classification and infer that SigDirect, with a specific heuristic, gives high classification accuracy using a minimum set of association rules.

### 3 Methodology

In this section, we introduce the details about the proposed Bi-Level classification technique. We initially describe the baseline technique of developing a two level classifier by using the Apriori algorithm for building the ARC model in the first level. However, this technique was found to suffer limitations with regard to selecting the optimum support and confidence threshold values for different datasets. Therefore, we extended our baseline to include the approach proposed by Li et al. [15]. In our proposed method we use statistically significant CARs to obtain rule features that are used in the second stage of learning.

#### 3.1 Notations and Definitions

**Definition 1.** Dependency of a CAR [15]

*If a transaction database  $T$  consists of a set of items  $I = \{i_1, i_2, \dots, i_m\}$  and a set of class labels  $C = \{c_1, c_2, \dots, c_L\}$ , a transaction  $X$  in  $T$  consists of a set of items  $A = \{a_1, a_2, \dots, a_n\}$  and a particular class label  $c_k$  such that  $A \subseteq I$  and  $c_k \in C$ . A CAR  $R$  in the form of  $A \rightarrow_{c_k}$  is called dependent if the antecedent part and the consequent class label of the CAR satisfy  $P(A, c_k) \neq P(A)P(c_k)$ , where  $P(A)$  denotes the probability of occurrence of itemset  $A$ .*

**Definition 2.** Fisher's exact test [14]

*Consider a null hypothesis in which  $A$  and  $c_k$  are assumed to be independent*

of each other. The dependency of the CAR  $A \rightarrow c_k$  is said to be statistically significant at level  $\alpha$ , if the probability  $p$  of obtaining an equal or stronger dependency in a dataset complying with a null hypothesis is not greater than  $\alpha$ . The probability  $p$ , i.e.,  $p$ -value, can be calculated by Fisher's exact test:

$$p_f(A \rightarrow c_k) = \sum_{i=0}^{\min\{\sigma(A, \neg c_k), \sigma(\neg A, c_k)\}} \frac{\binom{\sigma(A)}{\sigma(A, c_k)+i} \binom{\sigma(\neg A)}{\sigma(\neg A, \neg c_k)+i}}{\binom{|T|}{\sigma(c_k)}} \quad (1)$$

where  $\sigma(X)$  denotes the support count of  $X$ . The significance level  $\alpha$  is usually set to be 0.05.

**Definition 3.** Potentially Statistically Significant [15]

The CAR  $A \rightarrow c_k$  is defined as "Potentially Statistically Significant" (**PSS**), if it meets either of the following conditions:

(1)  $\sigma(A) \leq \sigma(c_k)$  holds, and the lower bound  $\frac{\sigma(\neg A)! \sigma(c_k)!}{|T|! (\sigma(c_k) - \sigma(A))!}$  is smaller than or equal to  $\alpha$ ;

(2)  $\sigma(A) > \sigma(c_k)$  holds.

where  $A \subseteq I_{Remaining}$  and  $c_k \in \{c_1, c_2, \dots, c_L\}$

If a CAR is **PSS**, we need to calculate the exact  $p$ -value to see if it is indeed statistically significant.

### 3.2 Method 1

The aim of associative classification is to find knowledge from data in the form of association rules associating features and class labels. During inference one or a set of rules are selected and used to predict the class label. This selection is typically based on heuristics for ranking rules.

Using the proposed approach of two stage classification in [4], we have implemented the same technique for building a model which would learn to select and use the discovered rules automatically rather than relying on heuristics to select them. In brief, the first stage is to learn an associative classifier and the second stage is to extract features from the learned rules to learn a second predictor predicting which rule is best to use during inference. The initial training dataset is split into two parts, one used to derive rules with association rule mining and the second part to extract features for the second training level. These two sets are disjoint in order to avoid overfitting. On the TrainSet 1, the first stage of learning is performed. Here, our algorithm uses a constrained form of Apriori [1] to perform association rule mining to obtain a set of rules that have features on the left and class labels on the right side of the rule and that are above the minimum threshold values for support and confidence. This ARC Model is used to collect a set of features from the samples present in TrainSet 2. As proposed in [4], we have used two approaches namely, the class based and the rules based feature extraction, to get the set of features and class labels from the ARC model.

**Class Based Features** For the class based feature extraction technique, we derive rules from TrainSet 1 and we match the features from our TrainSet 2 with the antecedents of the rules in the ARC Model. A rule is said to be *applicable* to a new instance of TrainSet2 if the antecedent of the rule is a subset of the features of the instance. Using the set of rules that apply to the instances in TrainSet 2, we count the number of rules that match for each class. Using this approach we derive a transformed feature set as shown in Table 1, where we state the average confidence and the count of all the matching rules for an example of three given class labels. This dataset of *class-based features* is given to the next level of learning in order to train a classification model that selects rules.

Table 1: Example for transformed set of features in Class based

Class1		Class2		Class3	
Avg Conf	#Rules	Avg Conf	#Rules	Avg Conf	#Rules
85	1	81.6	3	80	2

Table 2: Example for transformed set of features in Rule based

R1			R2			R3		
Conf	Sup	Match	Conf	Sup	Match	Conf	Sup	Match
80	10	0	90	10	1	85	15	1

**Rule Based Features** For the rule based approach, we use the characteristics of the rules derived from TrainSet 1 to create a new feature space. For each instance in the dataset TrainSet 2, we check if each of the rules in the ARC model apply or not, that is we match the features from the sample with the antecedents of the rule. This feature is denoted by a boolean value 1 to represent a match, 0 for absent. Along with this, information of support and confidence is added as features in the new set. An example is shown in Table 2, where one row in the dataset is taken and a new feature is generated for 3 rules of the ARC Model.

The features derived using the ARC model are further given as a training input to the next level, consisting of the classifier, which learns how to use the rules in the prediction process. In the second level, machine learning based classifiers like Neural network (NN) and Support Vector Machine(SVM) or rule based classifier like RIPPER, are used to automatically learn on rules to determine the weighting scheme for classification and obtain the final model.

For testing, we use the ARC model to derive the set of features for the Test dataset. Further, these features are given to the trained model of Neural network, SVM or RIPPER to classify the new samples. The ARC model and the trained model in the second level together predict the class for any new sample given for classification.

### 3.3 Method 2

In our second approach, we extend the bi-level classification technique by using statistically significant CARs. For this purpose, we derive positive and negative rules which are statistically significant [14]. Li et al. proposed to use Fisher’s exact test to extract the statistical significance of rules. The proposed algorithm determines non-redundant association rules for classification which show statistical dependency between the antecedent items and the consequent labels by using the p-value.

We split our training dataset into two parts as illustrated in Algorithm 1. On the TrainSet 1, the first level of learning is performed. The association rule mining is done by building Apriori like tree to form the ARC model, which gives us the set of association classification rules. The rules described by this ARC model are statistically significant, giving us the p-value for each rule. The rules obtained in the first level are used to extract the transformed feature set from the TrainSet 2. We used rule-based approach as described in Method 1 to extract features for this classifier as well. This is because the rule-based features in Method 1 shows better results than class-based features, as will be discussed in Section 4.

As proposed in [15], initially, all the impossible items are removed. An item is termed as impossible to appear in a statistically significant CAR if it has support value below  $\gamma|T|$ , where  $\gamma \leq 0.5$  and  $T$  is the transaction database. These items are removed and thereafter all the left over items ( $I_{\text{Remaining}}$ ) are sorted in the ascending order of their support values. Further the tree is enumerated to generate class association rules and only those with one antecedent are listed. These rules are then checked for their PSS value (Definition 3). Rules that do not satisfy either of the PSS conditions are pruned and the other rules are checked for statistical significance. From PSS 1-itemset rules, PSS 2-itemset rules are generated considering the property that if a rule is PSS, then its parent rule will also be PSS, i.e. if CAR  $A \rightarrow_{c_k}$  is PSS, then any of its parent rule  $B \rightarrow_{c_k}$  is also PSS, where  $B \subsetneq A$  and  $|B| = |A| - 1$ . The process repeats until no PSS rules are generated at a certain level. Also, if a rule is marked as minimal, the expansion from this rule is stopped because all of its children rules can not get a lower p-value.

The number of rules generated by the above approach may be large and might contain some unnecessary rules as well. In order to make the classification efficient and to obtain globally best rules from the training dataset, we use the proposed instance-centric rule pruning approach [15]. These pruned rules form the ARC model for this method.

We further apply the rule based approach to extract the features for the TrainSet 2 using this ARC model. An example for rule-based feature extraction for Method 2 is shown in Table 3 with just two rules. For each sample in TrainSet 2, we take the boolean value representing whether the rule matches the sample or not. Along with this, we take the characteristics of the rule as features in the transformed feature set. These include support value of the rule, its confidence and the log of the p-value. The lower the p-value, the better the rule, and sum-

ming up the p-value is not a suitable heuristic for a set of rules. Hence, we take the log value of p-value in order to generalize the process for rule-based and class based feature extraction. The features are extracted for each row in the testing dataset using the ARC model and the learnt classification model predicts the class label for each data point.

Table 3: Example for Transformed Set of Features for Method 2

R1				R2			
Conf	Sup	ln(p-value)	Match	Conf	Sup	ln(p-value)	Match
80	10	-10.6	0	90	10	-5.1	1

We also evaluate the BiLevCSS with SigDirect associative classifier in the second level. However, SigDirect is found to have a limitation of not being able to work well with very high dimensional datasets. For some datasets when using BiLevCSS, the features extracted for the second phase are found to have a large dimensionality due to a sizeable number of generated rules. This greatly increases the runtime of the SigDirect algorithm when used in the second phase. Therefore, we do not report the results of SigDirect as a predictor in the second stage.

## 4 Experimental Results

We have evaluated our algorithm on 10 UCI datasets to compare the classification accuracy with other rule based and machine learning based algorithms that exist in the literature. We report the average of the results obtained for every dataset on the 10 fold cross validation in our experiments. We compare the performance with common machine learning techniques like SVM and Neural networks, rule-based classifiers like C4.5 and RIPPER and previously proposed associative classifiers like CBA, CMAR and CPAR. We also compare our baseline approaches 2SARC1 (NN) [4], 2SARC2 (NN) [4], 2SARC1 (SVM), 2SARC2 (SVM), 2SARC1 (RIPPER) and 2SARC2 (RIPPER) with these classifiers.

### 4.1 Classification Accuracy

We compare our proposed model BiLevCSS with the above stated contenders on the basis of classification accuracy. We evaluate the performance of BiLevCSS model with three different classifiers in the second level; Neural Network at the second stage (regarded as BiLevCSS (NN)), RIPPER in the second stage (regarded as BiLevCSS (RIPPER)) and SVM in the second stage (regarded as BiLevCSS (SVM)).

We follow the default parameter values for SVM [10], C4.5 [18], CBA [17], CMAR [16], CPAR [19] as stated in the original papers. For RIPPER as a standalone rule based classifier, we have used default parameters from Weka [13] which are also stated to be the best by the authors in [9]. For vanilla Neural

---

**Algorithm 1:** Algorithm for BiLevCSS
 

---

**Data:** **Train Dataset:** Initial training dataset. **Test Dataset:** Initial testing dataset. **TransformedTestSet:** Testing dataset for classification model.  
**TrainSet1:** Training set used to build the ARC Model. **TrainSet2:** Training set used to build features using the ARC model and train the classification model.

**Result:** Predict class label of each instance in **TestSet**.

- 1 Use TrainSet1 to generate all statistically significant CARs  $A \rightarrow c_k$  ;  
     ▷ Follow the Algorithm 1 and 2 in [15]
- 2 classLabelsSet  $\leftarrow$  Unique set of class labels in dataset
- 3 ARC Model = { CARs  $A \rightarrow c_k \mid c_k \in \text{classLabelSet}$ }
- 4 **for** each instance  $T$  in TrainSet2 **do**
- 5     NewFeature=[]
- 6     **for** each rule  $R$  in ARC model **do**
- 7         match( $T, R$ ) ; ▷ Determine if instance  $T$  matched the antecedent of rule  $R$ .
- 8         **if**  $\text{match}(T, R) == \text{True}$  **then**
- 9             NewFeature.append(Conf( $R$ ), Support( $R$ ), log(P-value( $R$ )), 1);
- 10         **else**
- 11             NewFeature.append(Conf( $R$ ), Support( $R$ ), log(P-value( $R$ )), 0);
- 12         **end**
- 13     **end**
- 14     TransformedTrainSet.append(NewFeature);
- 15 **end**
- 16 Train a supervised learning model using TransformedTrainSet dataset for classification.
- 17 Repeat steps 4 to 15, to extract features from Test Dataset using ARC model to build TransformedTestSet for second stage of learning.
- 18 Derive the accuracy of the classification model using the Test dataset.

---

Network, we use a single hidden layer with the number of nodes to be the average of the number of input and output nodes and we also tune ReLU or sigmoid activation functions with a learning rate of 0.1.

For our baseline Method 1, we perform experiments using Apriori [1] based rule generation in the first level learning. Further, we test the accuracy of the rule-based feature extraction approach to build the bi-level classifier with Neural Network, SVM or RIPPER in the second stage. Similarly, we also measure the accuracy, of the bi-level classifier, which uses class-based features. For Apriori, we use a range of support values from 5% to 30% depending on the size of the dataset. The threshold value for confidence is set around 50%. In Table 4, we report the accuracy obtained for the 10 UCI datasets using our baseline approach. Along with the classification accuracy values, the name of the dataset and the number of records have also been reported. As can be seen from Table 4, the overall accuracy does not follow a pattern and nothing conclusive could be derived from the results aforementioned. However, the results from Method 1 showed that, for most of the UCI datasets, the rule-based feature extraction

Table 4: Comparison of classification accuracy using Rule-based and Class-based Features extraction in Method 1

Datasets	#cls	#rec	2SARC2 (NN)	2SARC2 (SVM)	2SARC2 (RIPPER)	2SARC1 (NN)	2SARC1 (SVM)	2SARC1 (RIPPER)
Iris	3	150	93.74	89.74	<b>94.28</b>	<b>94.11</b>	89.3	90.94
Glass	7	214	48.9	<b>52.2</b>	<b>69.17</b>	50	<b>52.2</b>	51.74
Heart	5	303	<b>63.5</b>	54.34	54.95	<b>62.34</b>	57.14	54.02
Hepati	2	155	<b>85</b>	<b>81.25</b>	79.97	70	75	80.48
Pima	2	768	66.45	65.2	<b>72.74</b>	64.39	67.53	<b>70.93</b>
Flare	9	1389	74.5	70.58	<b>84.35</b>	74.39	70.6	<b>83.96</b>
Anneal	6	989	77	82	<b>96.41</b>	79.5	78.04	<b>83.74</b>
Horse	2	368	67.6	63.3	<b>81.40</b>	<b>72.97</b>	70.96	63.75
Breast	2	699	89.7	93	93.14	93.75	<b>98.6</b>	<b>93.78</b>
Wine	3	178	<b>97.18</b>	77.97	85.15	<b>94.92</b>	72.02	53.84
<b>Average</b>			<b>76.35</b>	72.95	<b>81.15</b>	75.63	73.13	72.71

approach is found to give altogether a better average accuracy over the class-based feature extraction approach.

Therefore, in the second method, we adapt the rule-based feature extraction approach to build the bi-level classification model with statistically significant rules. For the following experiments, we discretize the numerical attributes of the datasets as stated in [7]. All the results reported in this section have been performed on the same discretized dataset for fair comparison.

Table 5: Comparison of classification accuracy of BiLevCSS with other state-of-the-art classifiers

Datasets	BilevCSS (RIPPER)	BilevCSS (NN)	BilevCSS (SVM)	RIPPER	NN	SVM	C4.5	CBA	CMAR	CPAR
Iris	95.72	<b>100</b>	<b>98.66</b>	94	98.09	94.6	94	94.67	94	94.7
glass	69.27	<b>86.60</b>	59.52	68.69	70.14	68.6	71.47	73.9	70.1	<b>74.4</b>
Heart	56.51	<b>78.64</b>	52.84	53.97	56.72	55.4	<b>61.5</b>	57.8	56.2	53.8
Hepati	82.57	<b>84.95</b>	<b>88.41</b>	78.06	82.89	79.3	79.25	81.82	80.5	79.4
Pima	73.64	<b>81.24</b>	73.2	66.36	<b>75.95</b>	74	73.7	72.9	75.1	73.8
Flare	84.27	<b>96.1</b>	83.1	72.13	<b>84.61</b>	73.8	82.1	84.2	84.3	63.9
Anneal	<b>96.93</b>	96.96	96.25	95.8	93.96	85	89.87	<b>97.91</b>	97.3	98.4
Horse	83.34	<b>87.78</b>	77.27	84.23	81.321	72.5	<b>85.04</b>	82.36	82.6	84.2
Breast	93.05	94.26	92.80	95.42	<b>96.83</b>	95.7	94.71	96.28	<b>96.4</b>	96
Wine	89	<b>94.94</b>	84.20	91.57	91.66	<b>94.9</b>	71.7	49.6	92.7	88.2
<b>Average</b>	82.43	<b>90.14</b>	80.62	80.02	<b>83.21</b>	79.38	80.33	79.14	82.92	80.68

Moreover, as suggested by Li et al. in [15], we use the Fisher exact test to analyse the statistical significance of the class association rules. The threshold for p-value is set to be 0.05. The use of only statistically significant rules and the addition of p-value value along with support and confidence as a feature in the rule-based classification gives us much better results for Method 2 than the baseline Method 1. For the second layer of both the methods, we use Neural Network with single hidden layer, with 'ReLU' or 'sigmoid' as the activation functions and a learning rate of 0.1. We also tune the hyper parameter values of gamma, kernel and regularization parameters for the SVM classifier. We have

performed 5 fold internal cross validation for SVM and NN to tune their respective hyper parameter values. For RIPPER at the second stage of learning, we use the default best parameters from Weka. It can be observed that, in Table 5, the BiLevCSS model gives the best overall classification accuracy for the considered datasets. Our algorithm BiLevCSS(NN) outperforms all the other classification algorithms in the 10 UCI datasets with highest average accuracy.

We further perform a comparison between BiLevCSS with Neural Network at the second level against the vanilla Neural Network with 1 hidden layer, to validate the efficiency of the model. The results show that the proposed algorithm outperforms the vanilla NN. Similarly, BiLevCSS(SVM) was found to outperform vanilla SVM and BiLevCSS(RIPPER) outperformed the vanilla RIPPER algorithm. Figure 1 illustrates the comparison of results given by the best model BiLevCSS(NN) with vanilla Neural Network.

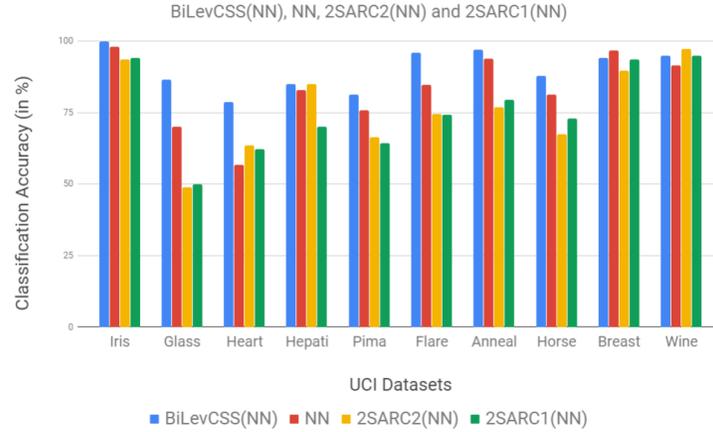


Fig. 1: Comparison of classification accuracy for BiLevCSS(NN) with vanilla Neural Network, 2SARC1(NN) and 2SARC2(NN).

The results shown in Table 5 highlight that the BiLevCSS model outperforms other rule based and associative classifiers on comparison. Next, we compared the three proposed strategies namely, BilevCSS (Ripper), BiLevCSS (NN) and BiLevCSS (SVM) with SigDirect. The results of this comparison are summarized graphically in Figure 2. The graph shows that BiLevCSS (NN) performs better than the rest, which proves that, when meaningful, statistically significant and non-noisy rules are given to Neural Network, the classification accuracy of the classifier improves. The results obtained from BiLevCSS (Ripper) are motivating, however do not beat BiLevCSS (NN) in performance. Therefore, in the future we aim to evaluate more explanatory classification models in the second phase

of learning, for a more explainable model since Neural Networks are more of a black box compared to Ripper.

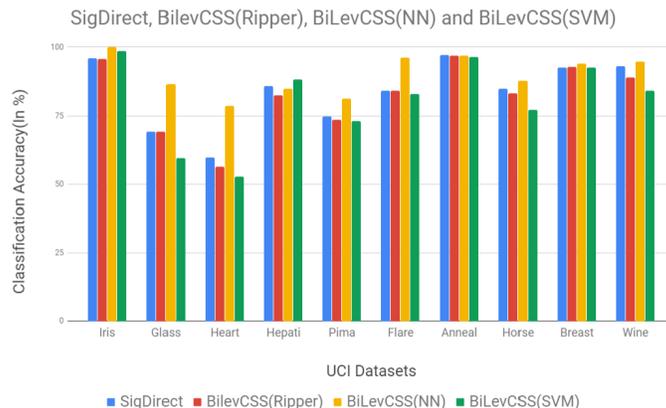


Fig.2: Comparison of classification accuracy for BiLevCSS(NN) with BiLevCSS(RIPPER), BiLevCSS(SVM) and SigDirect.

## 4.2 Statistical Analysis

The accuracy values report that BiLevCSS performs better for most of the datasets. To confirm this statement, we perform statistical analysis as shown in Table 6. We follow Demsar’s study [11] and use Friedman’s test to compare the statistical significance of the results obtained from the comparison of all the algorithms on the basis of classification accuracy. Since the p-value obtained from this test was less than the critical value (alpha) which is equal to 0.05, it proves that the results are statistically significant and the algorithms are significantly different from one another.

Furthermore, to investigate the statistical significant of the proposed algorithm with other contenders pair-wise, we perform another non-parametric test called Wilcoxon signed ranked test [11]. In this test, for every pair of algorithm in consideration, the difference of their classification accuracy,  $D_i$  is calculated to analyse the ranks based on the absolute values of these differences,  $|D_i|$ . Further, positive ranks  $R_i^+$  and negative ranks  $R_i^-$  are calculated based on the original values of  $D_i$  for two algorithms. Adding up all the values of  $R_i^+$  and  $R_i^-$ ,  $W_{stat}$  is calculated as  $\min(\sum R_i^+, \sum R_i^-)$  which gives us the critical value Z. For alpha value equal to 0.05, the corresponding Z-value is -1.96, therefore, the null hypothesis is rejected if the obtained critical value Z is less than -1.96.

Table 6 reports the p-values obtained by comparing the most accurate model, BiLevCSS(NN) against other classifiers using Wilcoxon test. We also compare

the number of times the different algorithms win or lose against BiLevCSS(NN) and if there is a tie between them. The p-values obtained are less than 0.05 which show that BiLevCSS(NN) is statistically significantly better than all the contenders. The results show that the proposed BiLevCSS algorithm with Neural Network at the second stage of learning outperforms the rest of the algorithms by winning in at least 8 out of 10 instances.

Table 6: BiLevCSS(NN) compared to the rest of the algorithms on 10 UCI datasets

Classifiers	Wins	Losses	Ties	P-value
BiLevCSS(NN) vs BiLevCSS(SVM)	9	1	0	0.017
BiLevCSS(NN) vs RIPPER	9	1	0	0.007
BiLevCSS(NN) vs NN	9	1	0	0.013
BiLevCSS(NN) vs SVM	9	1	0	0.009
BiLevCSS(NN) vs 2SARC2(NN)	8	2	0	0.013
BiLevCSS(NN) vs 2SARC2(SVM)	10	0	0	0.005
BiLevCSS(NN) vs 2SARC2(RIPPER)	10	0	0	0.005
BiLevCSS(NN) vs 2SARC1(NN)	10	0	0	0.005
BiLevCSS(NN) vs 2SARC1(SVM)	9	1	0	0.007
BiLevCSS(NN) vs 2SARC1(RIPPER)	10	0	0	0.005
BiLevCSS(NN) vs BiLevCSS(RIPPER)	10	0	0	0.05
BiLevCSS(NN) vs C4.5	9	1	0	0.007
BiLevCSS(NN) vs CBA	8	2	0	0.013
BiLevCSS(NN) vs CPAR	8	2	0	0.013
BiLevCSS(NN) vs CMAR	8	2	0	0.013

## 5 Conclusion and Future Work

In this project, we have introduced a novel approach BiLevCSS, a two level classifier built on statistically significant dependent CARs. The proposed classification model consists of four steps of rule generation, rule pruning, transformed feature extraction for the next phase using the obtained rules and finally, the prediction on the learned model using Neural Network in the second stage. Rule generation leads to the generation of all statistically significant rules which are further used to train a second classification model to select appropriate rules. Since, these rules might be noisy with some irrelevant information, they are pruned using the instance-centric rule pruning strategy. Furthermore, the features are extracted using rule based or class based techniques. Finally, the classification is done by using the learned NN, SVM or RIPPER in the second level. The idea of using statistically significant rules has made our algorithm more efficient by selecting only valuable CAR and providing new features for the second stage. The experimental results are very encouraging. The proposed classifier especially BiLevCSS(NN) is found to have achieved better prediction than other state-of-the-art classification algorithms in terms of accuracy.

In the future, we aim to experiment our algorithm by incorporating more features other than support, confidence, lift and p-value. We would also like to

evaluate the performance of our model with explainable associative classifiers in the second stage of learning. We would also extend our work for multi-label classification.

## References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Proceedings of 20th International of very large databases (VLDB). vol. 1215, pp. 487–499 (1994)
2. Antonie, M., Zaiane, O.R.: Text document categorization by term association. In: proceedings of IEEE International Conference on Data Mining. pp. 19–26 (2002)
3. Antonie, M.L., Zaiane, O.R.: An associative classifier based on positive and negative rules. In: Proceedings of the 9th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery. pp. 64–69 (2004)
4. Antonie, M.L., Zaiane, O.R., Holte, R.C.: Learning to use a learned model: A two-stage approach to classification. In: Sixth International Conference on Data Mining (ICDM). pp. 33–42. IEEE (2006)
5. Bayardo Jr, R.J.: Brute-force mining of high-confidence classification rules. In: KDD. vol. 97, pp. 123–126 (1997)
6. Beale, H.D., Demuth, H., Hagan, M.: Neural network design. Pws, Boston (1996)
7. Coenen, F.: The lucs-kdd software library (2004), <http://cgi.csc.liv.ac.uk/frans/KDD/Software/>
8. Coenen, F., Leng, P.: An evaluation of approaches to classification rule selection. In: Fourth IEEE International Conference on Data Mining (ICDM'04). pp. 359–362 (2004)
9. Cohen, W.: Fast effective rule induction. In: International Conference on Machine Learning, pp. 115–123. Elsevier (1995)
10. Cortes, C., Vapnik, V.: Support-vector networks. *Machine learning* **20**(3), 273–297 (1995)
11. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research* **7**, 1–30 (2006)
12. Han, J., Pei, J. and Yin, Y.: Mining frequent patterns without candidate generation. In proceedings of 2000 ACM SIGMOD International Conference on Management of Data **29**, 1–12 (2000)
13. Holmes, G., Donkin, A., Witten, I.: Weka: A machine learning workbench. Proceedings of ANZIIS (1994)
14. Li, J., Zaiane, O.: Associative classification with statistically significant positive and negative rules. In: Proceedings of the 24th ACM International on conference on Information and Knowledge Management. pp. 633–642. ACM (2015)
15. Li, J., Zaiane, O.R.: Exploiting statistically significant dependent rules for associative classification. *Intelligent Data Analysis* **21**(5), 1155–1172 (2017)
16. Li, W., Han, J. and Pei, J.: CMAR: Accurate and efficient classification based on multiple class-association rules. In: IEEE International Conference on Data Mining, ICDM. pp. 369–376 (2001)
17. Liu, B., Hsu, W. and Y.: Integrating classification and association rule mining. In: International Conference on Knowledge Discovery and Data Mining (1998)
18. Quinlan, J.R.: C4.5: Programs for machine learning. *Machine Learning* **16**(3), 235–240 (1994)
19. Yin, X. and Han, J.: Cpar: Classification based on predictive association rules. In: SIAM International Conference on Data Mining. pp. 331–335 (2003)