applied sciences

MDPI

Article

Multi-Label Classification Based on Associations

Raed Alazaidah [1], Ghassan Samara [1], Sattam Almatarneh [1,*], Mohammad Hassan [1], Mohammad Aljaidi [1] and Hasan Mansur [2]

[1] Faculty of Information Technology, Zarqa University, Zarqa 13110, Jordan
[2] Department of Artificial Intelligence in Accounting, Applied Science Private University, Amman 11931, Jordan
* Correspondence: salmatarneh@zu.edu.jo; Tel.: +962-777-440-610

Abstract: Associative classification (AC) has been shown to outperform other methods of single-label classification for over 20 years. In order to create rules that are both more precise and simpler to grasp, AC combines the rules of mining associations with the task of classification. However, the current state of knowledge and the views of various specialists indicate that the issue of multi-label classification (MLC) cannot be solved by any AC method. Since this is the case, adapting or using an AC algorithm to manage multi-label datasets is one of the most pressing issues. To solve the MLC issue, this research proposes modifying the classification based on associations (msCBA) method by extending its capabilities to consider more than one class label in the consequent of its rules and modifying its rules order procedure to fit the nature of the multi-label dataset. The proposed algorithm outperforms several other MLC algorithms from various learning techniques across a variety of performance measuresand using six datasets with different domains. The main findings of this research are the significance of utilizing the local dependencies among labels compared to global dependencies, and the important rule of AC in solving the problem of MLC.

Keywords: associative classification; classification; machine learning; multi-label classification; prediction

check for updates

Citation: Alazaidah, R.; Samara, G.; Almatarneh, S.; Hassan, M.; Aljaidi, M.; Mansur, H. Multi-Label Classification Based on Associations. *Appl. Sci.* **2023**, *13*, 5081. https://doi.org/10.3390/app13085081

Academic Editor: Christos Bouras

Received: 12 February 2023
Revised: 2 March 2023
Accepted: 6 March 2023
Published: 19 April 2023

1. Introduction

In data mining, classification is a common activity. The goal is to properly anticipate the class label of unseen instances using the rules or functions learned from a labeled set, or training set [1,2]. Many researchers [3–8] have been attracted to classification in recent decades, and have used a wide variety of learning approaches and strategies, including decision trees, neural networks, fuzzy logic, Bayesian and statistical approaches, rule-set induction, and more to create highly accurate classifiers [9]. In categorization, there are three major categories [10]. Each data point in the first two categories must match only one of the predefined classes. The third category [11], on the other hand, enables numerous class labels to be assigned to specific dataset instances. The first, referred to as a "binary classification" has just two class labels, but the second, referred to as a "multi-class classification" contains more than two [12,13]. The more general multi-label classification (MLC) system [11,14] is the third classification scheme. This study focuses on a particular categorization strategy that employs a single-label classification (SLC) to handle the multi-label problem.

Associative classification (AC) is one of the primary approaches that has been actively used in addressing the classification problem [15]. AC is a rule-set induction approach that uses the Association Rule Mining (ARM) task to solve the cassification issue [1]. In general, the AC approach has several distinguishable features over other learning approaches, such as the highly accurate rules produced by AC algorithms, the simplicity of representing the learned rules through the "IF-THEN" format, and its applicability to a wide range of real-life classification problems, i.e., medical diagnosis, e-mail phishing, fraud detection, and software defects [16]. Most AC-based methods have only been used for binary and multi-class classification problems [17]. In contrast, only a few efforts have been presented to apply AC in a broader form of classification termed MLC [16].

This research presents an update to the classification based on associations (msCBA) [18] algorithm. The improved prediction phase is a result of the new version's usage of the local positive dependencies among labels to reduce the large size of the problem search space. Multi-label classification based on associations (ML-CBA) is the new name for the improved system.ML-CBA is one of the first methods to employ AC to address the MLC problem by exploiting local labels' dependencies.

The paper is organized as follows: the next section briefly overviews the main concepts related to the AC approach and surveys some of the algorithms that have attempted to utilize AC in MLC. Section 3 describes the proposed ML-CBA algorithm and the results of comparing it to several other MLC algorithms that use different learning strategies. Finally, Section 4 concludes and introduces significant future work.

2. Literature Review

A brief general overview of MLC is described in Section 2.1. Few efforts have been presented to implement AC in MLC, which are described in Section 2.2. Section 2.3 describes the original CBA and msCBA algorithms.

2.1. MLC Overview

MLC is a general classification type with distinguishable features over conventional single-label classification (binary and multi-class classification) [19–21]. First, in MLC, an instance could be associated with more than one class label simultaneously, whereas single-label classification requires each instance to be associated with only one class label [22]. Second, because more than one class label could apply to the same instance simultaneously, the labels in MLC are not mutually exclusive to each other as they are in single-label classification [22]. Finally, the complexity of SLC is very low compared with MLC [23]. MLC has recently attracted the interest of numerous researchers due to its applicability to a wide variety of contemporary domains, including video and image annotation [24–26], classifying songs based on the invoked emotions [27], prediction of gene functionality [28–30], protein functionality detection [31,32], drug discovery [33], mining social networks [34–36], direct marketing [37], and Web mining [38]. Two main strategies are being used to address the MLC issue. The first strategy involves converting the input multi-label dataset into a single-label dataset or several single-label datasets. The modified dataset(s) are then used to train single-label classification algorithm [23]. This strategy has been referred to as the problem transformation method (PTM). Very few AC-based algorithms have been utilized as a basis classifier in this method, according to the literature [15]. The second method [6] extends a classification algorithm for an SLC to a dataset with multiple labels. This strategy is known as the algorithm adaptation method (AAM). Several single-label classification algorithms, including C4.5 [38], k-nearest neighbor (KNN) [39], back propagation [40], AdaBoost [41], and naive Bayes (NV) [42], have been modified to address the MLC issue. Unfortunately, according to the literature [15], no AC-based algorithm has been modified to address the MLC issue.

2.2. Utilizing AC in MLC

According to the previous studies, relatively few efforts to solve the MLC issue have used AC. Multi-class multi-label associative classification (MMAC) is among the first methods [43] to try to use AC in MLC. MMAC turns the original multi-label dataset into a single-label one by replicating each instance associated with more than one class label a number of times equals to the number of the class label it is associated with, using or without using a weight. Hence, the dataset becomes SL dataset but, with more instances than the original one. After that, MMAC applies any SL classifier such as CBA or msCBA on the newly transformed dataset as described in Section 2.3. MMAC then generates its rules by combining the outcomes of single-label rules with the same antecedent ending with multi-label rules. Unfortunately, MMAC has only been tested on datasets with single label, and it may be too complicated if the original dataset has many labels as well as high number of instances [44]. A novel multi-label method based on AC is presented in [45]. The multi-label classifier based on associative Classification (MCAC) developed a revolutionary rule discovery approach that creates multi-label rules from a single-label dataset without the need for learning. These multi-label rules reflect important information that most earlier AC algorithms often disregard. The correlative lazy associative classifier (CLAC) method, described in [46], is a hybrid algorithm that combines the principles of AC and lazy learning. CLAC generates classification association rules (CARs) that are graded according to their support and confidence ratings. Each class predicted by CLAC is immediately modified as a new characteristic to predict a different class. In comparison to the BoosTexter method, CLAC performed well on three textual datasets. The authors of [47] presented an identical AC-based method to the MMAC algorithm. In contrast to MMAC, the suggested method has been examined using one multi-label dataset (Scene) and emphasizes the importance of adopting AC in addressing the MLC issue.

2.3. CBA and msCBA Algorithms

CBA is one of the earliest algorithms that merge the ARM and classification tasks. CBA was introduced in [48]. Since then, several more techniques based on the combination of ARM and classification have been presented. The MMAC algorithm [43] and the multi-class associative classification (MAC) algorithm [49] are examples of algorithms that adhere to the AC methodology. CBA employs the a priori method in a classification dataset by the use of three key phases. At first, all continuous attributes are discretized.discretization is the step of converting any continuous variable or attribute into a discrete one. This step is compulsory for any AC-based classifier. Then, CARs are generated. CARs consider rules with arbitrary combinations of elements on antecedent (the left-hand side) and a single class on the consequent (the right-hand side). CARs are chosen using two metrics (support and confidence). The objective of the final phase is to construct a classifier using the best CARs [50]. CBA was subsequently enhanced in [18] by removing two flaws in the original CBA algorithm. The first problem is the use of a single minsup (minimum support) threshold value, which may result in an unbalanced class distribution. Using several minsup criteria, the modified version has addressed this problem. The exponential increase in the number of rules issued by CBA is the second flaw of the original CBA. This problem was fixed by combining CBA to a decision tree, as in C4.5, resulting in more precise rules. The modified version of CBA is referred to as CBA2 or msCBA, which is short for multiple support classification based on associations. Algorithm 1 illustrates the first CBA algorithm.

Although msCBA demonstrated higher performance in single-label classification compared to other classifiers from different learning strategies [16], it is incapable of handling multi-label datasets. The msCBA method assumes that each instance input has a single class label associated with it. Hence, it generates single-label rules with a single class label as the rule's consequence. When extending the msCBA method to accommodate multi-label datasets, this assumption should thus be discarded. In addition, the msCBA method captures the global relationships between features (attributes) and class labels, despite the fact that local dependencies and associations outperform global dependencies and associations [51,52].

Algorithm 1 CBA algorithm.

$F_1 = \{large1 - ruleitems\}$;
$CAR_k = genRules(F_1)$;
$prCAR_1 = pruneRules(CAR_1)$;
for $(k = 2; F_{k-1} \neq \phi; k++)$ **do**
    $C_k = candidateGent(F_{k-1})$;
    for each data case $d \in \mathcal{D}$ **do**
        $\mathcal{C}$=ruleSubset(C_k,d);
    for each candidate $c \in \mathcal{C}$ **do**
        c.condsupCont++;
    if $d.class = c.class$ **then**
        c.rulesupCount++;
    end if
    end
    end
    $F_k = \{\mathcal{C} \in C_k | c.rulesupCount \geq minsup\}$;
    $CAR_k = genRules(F_k)$;
    $prCAR_k = pruneRules(CAR_k)$;
end for
$CAR_s = U_k CAR_k$;
$prCAR_s = U_k prCAR_k$;

3. ML-CBA Algorithm

This section describes the planned ML-CBA. ML-CBA employs AC to address the MLC issue. To accommodate multi-label datasets, the classification based on associations (msCBA) method has been modified. The msCBA algorithm was selected for a number of reasons. First, to address one of the most pressing difficulties in the field of automatic classification, namely the construction or adaptation of an AC based classifier to classify datasets with multi-label and create multi-label rules, given the paucity of research in this particular area [15]. Second, msCBA was one of the first classification systems to use the association rules revealed by the a priori method. Interestingly, it has never been modified to support MLC. In addition, msCBA generates a classifier in the form of "IF-THEN" rules, which makes it simpler for experts and normal users to comprehend and use. Finally, AC algorithms are adept at uncovering latent dependencies between various objects, which increases the information acquired during the training phase and, as a result, improves the prediction phase of the learnt classifier. Specifically, two significant enhancements for msCBA algorithm are given to improve its capacity to suit MLC. Initially, the single-label CARs learnt through msCBA should be transformed into multi-label CARs using the captured local dependencies among labels. Second, the technique for sorting the learnt CARs should be modified to account for the operation of MLC, in which each classification rule may result in many class labels. Figure 1 depicts ML-CBA algorithm main stages (transformation stage, number of class labels prediction stage, constructing sub-datasets stage).

Figure 2 shows the transformation step. This step aims to generate the complete set of the CARs and is accomplished through three main substeps: firstly, transform the input multi-label dataset into a single-label dataset using the HSDF (high standard deviation first) transformation method [52]. Then, apply the Bayesian-D [53] discretization technique on the transformed dataset, in order to convert the continuous attributes into categorical attributes. Finally, classify the transformed single-label dataset using msCBA algorithm. Both HSDF and Bayesian-D have been chosen after a comprehensive evaluation where they showed the best results compared to other PTMs and discretization techniques.

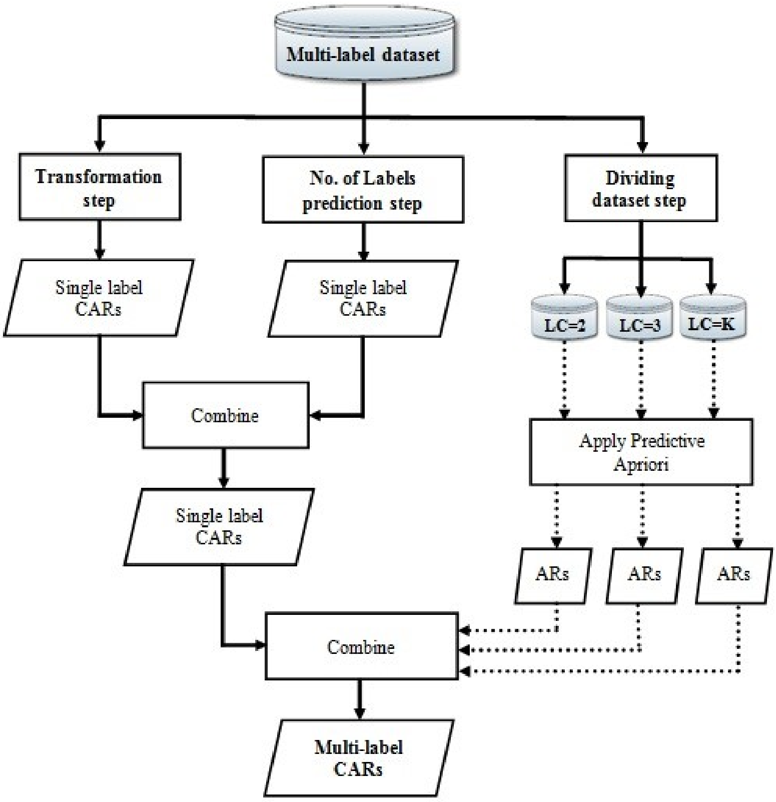

Figure 1. ML-CBA primary stages.

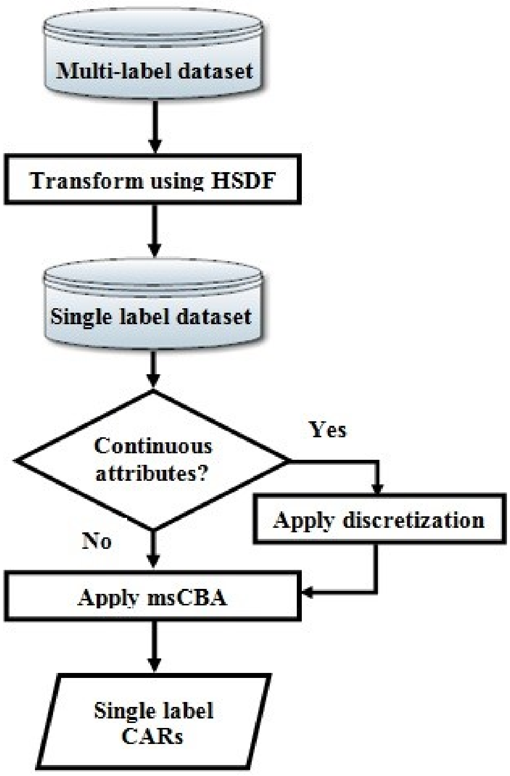

Figure 2. ML-CBA algorithm transformation stage.

Figure 3 illustrates the phase of predicting the number of class labels that might be linked to an example (instance). More information regarding this stage is provided in the second step of Algorithm 2.

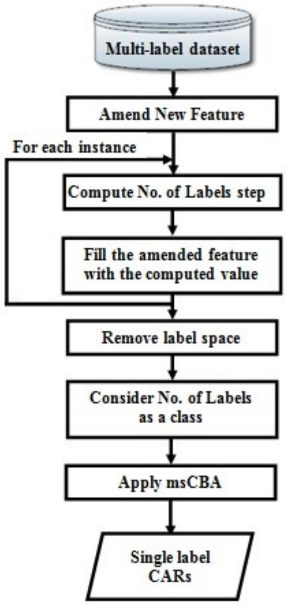

Figure 3. Number of labels associated with a stage of instance learning.

Algorithm 2 ML-CBA algorithm.

Input: Multi-label dataset (D), minsup, minconf, minacc.

Output : Multi-label CARs

begin:
Step 1:
1.1 Transform (D) into single label dataset (S) using HSDF.
1.2 Convert continuous attributes (if any) into categorical attributes, by applying Bayesian-D discretization technique.
1.3 Construct the single label CARs for the transformed dataset that satisfy minsup and minconf thresholds, by applying the msCBA algorithm.

Step 2:
2.1 Amend a new feature to the dataset to represent the total number of labels associated with each instance.
2.2 For each instance in the training set, compute the total number of labels associated with this instance, and amend it to the new feature.
2.3 Remove the label space from the dataset, and consider the last feature as a class.
2.4 Classify the dataset using msCBA algorithm.

Step 3:
3.1 Extract the label space of the input multi-label dataset.
3.2 Divide the extracted label space into (K) subsets, where k = the maximum number of labels that are associated with the instances - 1.
3.3 For each subset, capture all the positive local correlations among labels, with respect to the HSDF transformation order, and the *minacc* threshold (50%)these correlations are considered as local; since they have been captured among a smaller subset of the dataset, and used only when the predicted number of class labels matches the subset with this number of class labels.
3.4 For each label, merge all the captured positive local correlations in the previous step, with respect to the Accuracy of the association rules.

Step 4: Amend all classes that have significant positive associations with the class under processing, to the consequent of the selected single label CAR, with respect to the predicted number of labels.

Step 5: Sort the new multi-label rules according to Algorithm 3.
Step 6: Use the sorted multi-label rule resulted from Step 5 to classify any new instance.
End.

Figure 4 illustrates the step of constructing several sub-datasets in order to simplify the capturing of positive local dependencies among labels, considering the predicted total number of class labels associated to an example. More information regarding this step is given in Algorithm 2, Step 3.

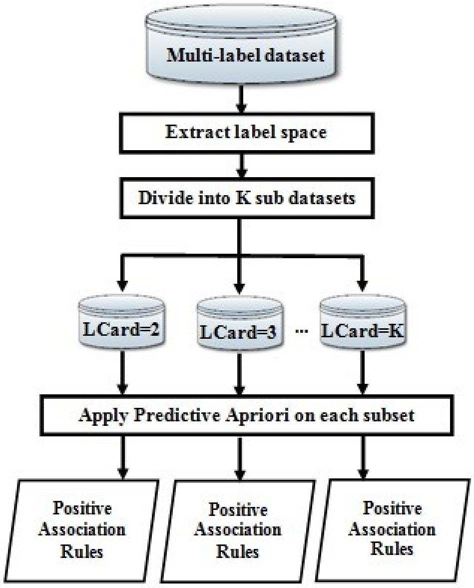

Figure 4. Dividing the input dataset into several subsets.

Algorithm 2 illustrates the phase of predicting the number of class labels that might be linked to an example (instance). Figure 4 illustrates the step of constructing several sub-datasets in order to simplify the capturing of positive local dependencies among labels, considering the predicted total number of class labels associated to an example.

After the construction of the multi-label rules, these rules are ordered and sorted, especially the rules with the same consequences.

If more than one single-label CARs foretell the same class label, then, the one which has the greater confidence will be applied first, as depicted in Algorithm 3. If there are several single-label CARs with equal confidence, then, the multi-label rule which has the highest average of the association rules used to produce its consequent is fired. If a tie still exists, the single-label CAR rule with the highest support will be chosen. The rule with the highest cardinality will be removed if more than one rule has equivalent values for the aforementioned criteria. In the end, if the scores are still tied, the fired rule will be determined by a coin toss.

Algorithm 3 Rules ordering algorithm.

Input: Set of multi-label CARs

Output: Sorted multi-label CARs

For any two given rules r1 and r2, r1 precedes r2 if:

1. The confidence of r1 is higher than that of r2.
2. Both rules have the same confidence value, but the average accuracy of association rules that form the consequent of r1 is higher than that of r2
3. Both rules have the same confidence value, and the same association rules accuracy average, but r1 has a higher support than that of r2.
4. Both rules have the same confidence value, the same association rules accuracy average, the same support value, but r1 has a lower cardinality than that of r2.
5. Chose randomly when the four previous conditions are the same for r1 and r2

3.1. Classification Phase in ML-CBA

The ML-CBA prediction method works as follows: when a test case is being processed, and before determining the expected class label, it first considers all SL rules learned by msCBA algorithm during the transformation phase. Second, it estimates the possible number of class labels linked to a test example using the classifier learnt in the previous step. ML-CBA determines the subset and resorts all local positive dependencies with the expected class according to the consequence of the triggered rule from the transformation phase, using the predicted class and the predicted total number of class labels linked to an instance.

3.2. Evaluation of the Proposed ML-CBA Algorithm

In this subsection, we will discuss the testing procedure of the suggested ML-CBA approach. The proposed strategy has been programmed using Java. high standard deviation first (HSDF) has been chosen as PTM. HSDF is a new PTM that attempts to maximize the capturing and the exploitation of the positive pairwise correlation among labels. this method works as follows: it starts with extracting the feature space of the dataset and considering the class label as a transactional dataset. Then, using predictive a priori, HSDF captures all the positive pairwise among labels. After that, it ranks the class labels according to the standard deviation of the accuracy of its correlation in a descendent fashion. The obtained rank is used to transform the original multi-label dataset to SL one. More information regarding HSDF and other PTMs could be found in [52]. Furthermore, predictive a priori and msCBA algorithms have been used as they have been implemented and programmed in KEEL with their default settings. KEEL which is short for knowledge extraction for evolutionary learning is an open source java based library for a large number of learning strategies and models in machine learning [54]. In the evaluation phase, ML-CBA has been compared to other MLC algorithms which take into account both global and local dependencies and come from a wide range of learning approaches. Currently, four forms of evaluation have been used (accuracy, Hamming loss, exact match, and one-error). Averaged across all instances, an accuracy metric indicates the fraction of correct predictions made for a given set of labels. Here is the formula that determines *accuracy*:

$$Accuracy = \frac{1}{t}\sum_{i=1}^{t}\frac{|(Zi \cap Yi)|}{|(Zi \cup Yi)}| \tag{1}$$

where:

Zi: the predicted label set
Yi: the ground truth label set

The Hamming Loss measures the typical amount of incorrectly labeled instances across all labels in a multi-label dataset. Inaccurate label predictions and missed labels are also accounted for in this metric. The lower this parameter's value, the better the classifier will perform. If we take the symmetric difference between the grounded truth label set and the expected set, we obtain an expression for the *Hamming loss*.

$$Hamming\ Loss = \frac{1}{t}\sum_{i=1}^{t}\frac{1}{q}[Z_i \Delta Y_i] \tag{2}$$

where:

q: total number of labels
t: total number of instances.

The Exact Match measure is particularly limiting since it gives equal weight to accurate and incorrect answers. In order to get this measure, the number of situations when the predicted label and the grounded truth label match-up is averaged. Maximizing the following equation will result in the best possible *exact match*:

$$Exact\ Match = \frac{1}{t}\sum_{i=1}^{t}[Z_i = Y_i] \tag{3}$$

Finally, the number of times the most preferred label was not included in the final collection of projected labels is calculated using the one-error metric. Since this metric only considers the most prominent label and disregards the others, it is clear that it is insufficient for the MLC problem. The following formula may be used to determine how to compute the *one-error metric*:

$$One\text{-}Error = \frac{1}{t}\sum_{i=1}^{t}[argmin\tau_i(\lambda) \notin Y_i, \lambda \in \mathcal{L}] \tag{4}$$

Six different datasets that belong to MLC with unique features are being used in this paper; four are of typical dataset size (yeast, scene, emotions, and flags), while the other two are of big dataset size (Genbase and TMC2007). Table 1 provides a description of the six datasetsLCard is short for label cardinality and represents the average number of class labels per instance in the datasets.

Tables 2–9 depict a comparison between the proposed ML-CBA algorithm and other MLC algorithms, using several evaluation metrics. The compared algorithms have been chosen to represent the three main MLC approaches. The first order approach which ignores any correlations among labels has been represented by two algorithms (BR and ML-KNN [39]). The second order approach which considers pairwise correlations only has been represented by two algorithms (BP-MLL [40] and CLR [55]). Finally, the high order approach which considers high order correlations among labels has been represented by eight algorithms (LP [56], RAKEL [57], CC [58], PS [59], ECC [58], EPS, ML-LOC [51], and BR+). Further, the chosen algorithms belong to both PTMs (BR, CLR, LP, RAKEL, CC, PS, ECC, EPS, and BR+), and AAMs (ML-KNN and BP-MLL).

Table 1. Multi-label datasets characteristics.

Dataset	Instances	Attributes	Labels	LCard
Yeast	2417	103	14	4.327
Scene	2712	294	6	1.074
Emotions	593	72	6	1.868
Flags	194	19	7	3.392
Genbase	662	1186	27	1.252
TMC2007	28,596	500	22	2.16

Furthermore, the chosen algorithms capture both types of correlations: local correlations (ML-LOC and LPLC [3]), and global correlations (LP, RAkEL, CC, PS, EPS, ECC, and BR+). Finally, it is worth mentioning that the Bayesian discretizer [60] has been used as a discretization technique in the ML-CBA algorithm.

3.2.1. An Analysis of the Proposed ML-CBA Algorithm Utilizing Datasets of Typical Size

Table 2 shows how the proposed ML-CBA algorithm stacks up against other MLC techniques in terms of accuracy. Table 2 shows that out of the thirteen algorithms considered, the ML-CBA technique has the greatest accuracy value. Finally, ML-CBA beats the other two approaches for capturing local dependencies between labels (ML-LOC and LPLC). Furthermore, when the cardinality of the dataset is high such as in flags and yeast, the advantages of discovering and exploiting the local positive dependencies among labels become more obvious. “NG” denotes “Not Given” in the tables below,since the original paper where the considered algorithm in these tables did not examine the evaluation metrics or datasets in this paper.

Table 2. Evaluation of the proposed ML-CBA algorithm on the regular-sized datasets using accuracy metric, with respect to different MLC algorithms.

Correlations Type	Approach	Algorithm	Yeast	Scene	Emotions	Flags
		ML-CBA	**0.584**	**0.977**	**0.744**	**0.694**
	1st Order	**BR**	0.52	0.643	0.551	0.576
		ML-KNN	0.52	0.691	0.366	0.555
Global Correlations	2nd Order	**BP-MLL**	0.185	0.212	0.276	NG
		CLR	0.514	0.695	0.557	NG
	High Order	**LP**	0.53	0.735	0.584	NG
		RAKEL	0.493	0.694	0.592	NG
		CC	0.521	0.736	0.584	NG
		PS	0.533	0.751	0.599	NG
		ECC	0.299	0.27	0.282	NG
		EPS	0.537	0.751	0.599	NG
		BR+	0.4838	0.5744	0.5537	NG
Local Correlations		**ML-LOC**	0.51	NG	0.497	0.568
		LPLC	0.542	NG	0.565	0.607

Table 3 depicts the Hamming loss results of the proposed ML-CBA algorithm, with respect to several other MLC algorithms.

Table 3. Evaluation of the proposed ML-CBA algorithm on the regular-sized datasets using Hamming loss metric, with respect to different MLC algorithms.

Correlations Type	Approach	Algorithm	Yeast	Scene	Emotions	Flags
		ML-CBA	**0.078**	**0.006**	**0.09**	**0.118**
	1st Order	**BR**	0.193	0.009	0.188	0.274
		ML-KNN	0.193	0.085	0.262	0.284
Global Correlations	2nd Order	**BP-MLL**	0.322	0.057	0.433	NG
		CLR	0.226	0.101	0.214	NG
	High Order	**LP**	0.206	0.09	0.198	NG
		RAKEL	0.207	0.095	0.186	NG
		CC	0.211	0.1	0.197	NG
		PS	0.205	0.084	0.192	NG
		ECC	0.619	0.47	0.63	NG
		EPS	0.207	0.085	0.193	NG
		BR+	0.222	0.258	0.226	NG
Local Correlations		**ML-LOC**	0.193	NG	0.21	0.262
		LPLC	0.202	NG	0.197	0.279

The results for the Hamming loss evaluation show that ML-CBA algorithm has a superior performance on the four regular-sizes datasets (yeast, scene, emotions, and flags). Table 4 depicts the exact match results of the proposed ML-CBA algorithm, with respect to several other MLC algorithms. Table 4 shows the superior performance of the proposed ML-CBA algorithm, comparing with variety of different MLC algorithms that follow different learning approaches using the exact match metric. ML-CBA overcomes all other algorithms on the four regular-sizes datasets.

Table 4. The exact match results of the proposed ML-CBA algorithm on the regular-sized datasets with respect to several other MLC algorithms.

Correlations Type	Approach	Algorithm	Yeast	Scene	Emotions	Flags
		ML-CBA	**0.276**	**0.97**	**0.638**	**0.513**
	1st Order	**BR**	0.146	0.617	0.307	0.076
		ML-KNN	0.189	0.643	0.143	0.098
Global Correlations	2nd Order	**BP-MLL**	0.185	0.212	0.276	NG
		CLR	NG	NG	NG	NG
	High Order	**LP**	0.194	0.696	0.351	0.123
		RAKEL	0.163	0.662	0.341	NG
		CC	0.196	0.669	0.349	NG
		PS	0.258	0.717	0.367	NG
		ECC	0.243	0.007	0.022	0.191
		EPS	0.253	0.715	0.366	NG
Local Correlations		**ML-LOC**	0.199	NG	0.261	0.115
		LPLC	0.186	NG	0.303	0.123

Table 5 depicts the one-error results of the proposed ML-CBA algorithm, with respect to several other MLC algorithms.

Table 5 shows clearly that the proposed ML-CBA algorithm has acceptable one-error values compared with several MLC algorithms. Nevertheless, the accuracy of the ML-CBA algorithm is higher than the Accuracy of all other MLC algorithms as depicted in Table 2. This indicates the high benefits of capturing the local positive correlations against capturing global correlations. Furthermore, this is a strong evidence that local correlations are more accurate than global correlations, and thus, have a high influence on the predictive performance of the classification task.

Table 5. The one-error results of the proposed ML-CBA algorithm on the regular-sized datasets with respect to several other MLC algorithms.

Correlations Type	Approach	Algorithm	Yeast	Scene	Emotions	Flags
		ML-CBA	0.258	**0.009**	**0.123**	**0.145**
	1st Order	**BR**	0.227	0.262	0.256	NG
		ML-KNN	0.228	0.219	0.263	NG
Global Correlations	2nd Order	**BP-MLL**	0.235	0.821	0.318	NG
		CLR	0.241	0.323	0.291	NG
	High Order	**LP**	0.267	0.246	0.31	NG
		RAKEL	0.255	0.237	0.26	NG
		CC	0.256	0.268	0.283	NG
		PS	0.321	0.287	0.427	NG
		ECC	0.685	0.775	0.802	NG
		EPS	0.265	0.225	0.3	NG
Local Correlations		**ML-LOC**	**0.216**	0.179	NG	NG
		LPLC	NG	NG	NG	NG

3.2.2. Evaluation of the Proposed ML-CBA Algorithm on the Large-sized Datasets

In this subsection, an evaluation of the proposed ML-CBA algorithm on the large-sized multi-label datasets is presented. Two large-sizes datasets (Genbase and TMC2007) are considered in this paper. Four evaluation metrics have been considered in this evaluation (accuracy, Hamming loss, exact match, and one-error). Tables 6–9 shows the evaluation results of the proposed ML-CBA algorithm on the two large-sizes datasets using the previously mentioned evaluation metrics. Table 6 depicts the accuracy evaluation results of the proposed ML-CBA algorithm compared against several other MLC algorithms on the two large-sizes multi-label datasets.

From Table 6, it can be seen that ML-CBA has a superior accuracy on TMC2007 dataset, while it has a fair Accuracy on Genbase dataset. Genbase has a very low LCard, and only 19 local positive correlations have been captured in this dataset. Table 7 depicts the Hamming loss evaluation results of the proposed ML-CBA algorithm compared against several other MLC algorithms on the two large-sizes multi-label datasets.

Table 6. The accuracy results of the proposed ML-CBA algorithm on the large-sized datasets, with respect to several other MLC algorithms.

Correlations Type	Approach	Algorithm	Genbase	TMC2007
	1st Order	**ML-CBA**	0.978	**0.685**
		BR	0.962	0.541
		ML-KNN	0.948	0.531
Global Correlations	2nd Order	**BP-MLL**	0.632	0.652
		CLR	0.561	0.506
	High Order	**RAKEL**	**0.982**	0.549
		ECC	0.978	0.517
		EPS	0.945	0.549
Local Correlations		**ML-LOC**	NG	NG
		LPLC	NG	NG

Table 7 clearly shows that the ML-CBA algorithm has a superior performance on the two large-sizes datasets, especially on TMC2007 dataset. Table 8 depicts the exact match evaluation results of the proposed ML-CBA algorithm, with respect to several other MLC algorithms on the two large-sizes multi-label datasets.

Table 7. The Hamming loss results of the proposed ML-CBA algorithm on the large-sized datasets, with respect to several other MLC algorithms.

Correlations Type	Approach	Algorithm	Genbase	TMC2007
		ML-CBA	**0.001**	**0.027**
	1st Order	**BR**	**0.001**	0.071
		ML-KNN	0.005	0.073
Global Correlations	2nd Order	**BP-MLL**	0.004	0.098
		CLR	0.004	0.068
	High Order	**RAKEL**	0.003	0.068
		LIFT	0.003	NG
		ECC	0.002	0.068
		EPS	0.007	0.069
Local Correlations		**ML-LOC**	**0.001**	NG
		LPLC	NG	NG
		LEAD	0.002	0.063

Table 9 depicts the one-error evaluation results of the proposed ML-CBA algorithm with respect to several MLC algorithms on the two large-sizes datasets.

To summarize this section, the evaluation phase of the proposed ML-CBA algorithm shows a superior performance over other MLC algorithms that capture local and global correlations among labels on most datasets considered in this paper and using the four evaluation metrics. The main reason for the superior performance of the ML-CBA algorithm is the capturing of the positive local correlations among labels, which have been proven to be more accurate, and thus, have a strong positive influence on the final classification step of the proposed ML-CBA algorithm. Furthermore, one of the distinguishable feature that causes the superior performance of the ML-CBA algorithm is the strong capabilities of the msCBA algorithm as a base classifier. msCBA is capable to capture hidden information that help to improve the accuracy of the msCBA algorithm, and consequently, improve the predictive performance of the ML-CBA algorithm.

Table 8. The exact match results of the proposed ML-CBA algorithm on the large-sized datasets with respect to several other MLC algorithms.

Correlations Type	Approach	Algorithm	Genbase	TMC2007
		ML-CBA	**0.978**	**0.52**
	1st Order	**BR**	0.48	0.26
		ML-KNN	NG	NG
Global Correlations	2nd Order	**BP-MLL**	NG	NG
		CLR	0.884	0.147
	High Order	**RAKEL**	0.964	0.256
		LIFT	NG	NG
		ECC	0.592	0.233
		EPS	0.894	0.26
Local Correlations		**ML-LOC**	NG	NG
		LPLC	NG	NG
		LEAD	NG	NG

Table 9. The one-error results of the proposed ML-CBA algorithm on the large-sized datasets with respect to several MLC algorithms.

Correlations Type	Approach	Algorithm	Genbase	TMC2007
		ML-CBA	0.022	**0.167**
	1st Order	**BR**	0.037	0.342
		ML-KNN	0.055	0.32
Global Correlations	2nd Order	**BP-MLL**	0.368	0.445
		CLR	0.439	0.425
	High Order	**RAKEL**	NG	0.253
		LIFT	**0**	0.213
		ECC	0.001	0.232
Local Correlations		**ML-LOC**	0.004	NG
		LPLC	NG	NG
		LEAD	0.007	0.226

4. Conclusions and Future Work

The AC learning approach has been proven to generate more accurate classifiers than other learning approaches. Furthermore, AC algorithms usually capture hidden information that could not be discovered by other learning approaches, and represent the discovered knowledge through "IF-Then" rules, which make it easier to understand by all types of users.

In this paper, an adaptation of the popular msCBA algorithm has been presented. The adapted algorithm has been compared against several other MLC algorithms from different leaning strategies, and using several evaluation metrics, where the adapted algorithm (ML-CBA) showed a superior performance.

As a future work, much more research should be conducted on adapting other AC algorithms to handle the problem of MLC, and considering different discretization techniques.

Author Contributions: Conceptualization, R.A. and G.S.; methodology, R.A.; software, R.A.; validation, H.M., S.A. and M.H.; formal analysis, R.A.; investigation, R.A.; resources, R.A.; data curation, R.A.; writing—original draft preparation, R.A.; writing—review and editing, R.A. and S.A.; visualization, M.A.; supervision, R.A.; project administration, R.A.; funding acquisition, G.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by Zarqa University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data supporting reported results are available on request from the corresponding authors.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hadi, W.; Al-Radaideh, Q.A.; Alhawari, S. Integrating associative rule-based classification with Naïve Bayes for text classification. *Appl. Soft Comput.* **2018**, *69*, 344–356. [CrossRef]
2. Zeng, C.; Zhou, W.; Li, T.; Shwartz, L.; Grabarnik, G.Y. Knowledge guided hierarchical multi-label classification over ticket data. *IEEE Trans. Netw. Serv. Manag.* **2017**, *14*, 246–260. [CrossRef]
3. Huang, J.; Li, G.; Wang, S.; Xue, Z.; Huang, Q. Multi-label classification by exploiting local positive and negative pairwise label correlation. *Neurocomputing* **2017**, *257*, 164–174. [CrossRef]
4. Mohana, G.; Chitra, S. Design and development of an efficient hierarchical approach for multi-label protein function prediction. *Biomed. Res. Health Sci. Bio Converg. Technol. Ed. II* **2017**, 370–379. Available online: https://www.semanticscholar.org/paper/Design-and-development-of-an-efficient-hierarchical-MohanaPrabha-Chitra/a8b4c905f2d083801b2a7b06356eed9ad49be797 (accessed on 11 February 2023).
5. Sousa, R.; Gama, J. Multi-label classification from high-speed data streams with adaptive model rules and random rules. *Prog. Artif. Intell.* **2018**, *7*, 177–187. [CrossRef]
6. Xu, S.; Yang, X.; Yu, H.; Yu, D.J.; Yang, J.; Tsang, E.C. Multi-label learning with label-specific feature reduction. *Knowl.-Based Syst.* **2016**, *104*, 52–61. [CrossRef]
7. Gamallo, P.; Almatarneh, S. Naive-Bayesian Classification for Bot Detection in Twitter. In Proceedings of the CLEF, Lugano, Switzerland, 9–12 September 2019.
8. Almatarneh, S.; Gamallo, P.; ALshargabi, B.; Al-Khassawneh, Y.; Alzubi, R. Comparing traditional machine learning methods for COVID-19 fake news. In Proceedings of the 2021 22nd International Arab Conference on Information Technology (ACIT), Muscat, Oman, 21–23 December 2021; IEEE: New York, NY, USA, 2021; pp. 1–4.
9. Lin, Q.; Man, Z.; Cao, Y.; Wang, H. Automated Classification of Whole-Body SPECT Bone Scan Images with VGG-Based Deep Networks. *Int. Arab. J. Inf. Technol.* **2023**, *20*, 1–8. [CrossRef]
10. Alazaidah, R.; Thabtah, F.; Al-Radaideh, Q. A multi-label classification approach based on correlations among labels. *Int. J. Adv. Comput. Sci. Appl.* **2015**, *6*, 52–59. [CrossRef]
11. Gibaja, E.; Ventura, S. A tutorial on multilabel learning. *ACM Comput. Surv.* **2015**, *47*, 1–38. [CrossRef]
12. Suri, J.S.; Bhagawati, M.; Paul, S.; Protogerou, A.D.; Sfikakis, P.P.; Kitas, G.D.; Khanna, N.N.; Ruzsa, Z.; Sharma, A.M.; Saxena, S.; et al. A powerful paradigm for cardiovascular risk stratification using multiclass, multi-label, and ensemble-based machine learning paradigms: A narrative review. *Diagnostics* **2022**, *12*, 722. [CrossRef]
13. Hegazy, H.I.; Tag Eldien, A.S.; Tantawy, M.M.; Fouda, M.M.; TagElDien, H.A. Real-time locational detection of stealthy false data injection attack in smart grid: Using multivariate-based multi-label classification approach. *Energies* **2022**, *15*, 5312. [CrossRef]
14. El-Hasnony, I.M.; Elzeki, O.M.; Alshehri, A.; Salem, H. Multi-label active learning-based machine learning model for heart disease prediction. *Sensors* **2022**, *22*, 1184. [CrossRef] [PubMed]
15. Abdelhamid, N.; Jabbar, A.A.; Thabtah, F. Associative classification common research challenges. In Proceedings of the 2016 45th International Conference on Parallel Processing Workshops (ICPPW), Philadelphia, PA, USA, 16–19 August 2016; IEEE: New York, NY, USA, 2016; pp. 432–437.
16. Abdelhamid, N.; Thabtah, F. Associative classification approaches: Review and comparison. *J. Inf. Knowl. Manag.* **2014**, *13*, 1450027. [CrossRef]
17. Li, B.; Li, H.; Wu, M.; Li, P. Multi-label Classification based on Association Rules with Application to Scene Classification. In Proceedings of the 2008 The 9th International Conference for Young Computer Scientists, Hunan, China, 18–21 November 2008; pp. 36–41. [CrossRef]
18. Liu, B.; Ma, Y.; Wong, C.K. Improving an association rule based classifier. In Proceedings of the Principles of Data Mining and Knowledge Discovery: 4th European Conference, PKDD 2000, Lyon, France, 13–16 September 2000; Springer: Berlin/Heidelberg, Germany, 2000; pp. 504–509.

19. Alazaidah, R.; Ahmad, F.K.; Mohsen, M.F.M. A comparative analysis between the three main approaches that are being used to. *Int. J. Soft Comput.* **2017**, *12*, 218–223.
20. Massidda, L.; Marrocu, M.; Manca, S. Non-intrusive load disaggregation by convolutional neural network and multilabel classification. *Appl. Sci.* **2020**, *10*, 1454. [CrossRef]
21. Wu, X.; Gao, Y.; Jiao, D. Multi-label classification based on random forest algorithm for non-intrusive load monitoring system. *Processes* **2019**, *7*, 337. [CrossRef]
22. Alluwaici, M.; Junoh, A.K.; Alazaidah, R. New problem transformation method based on the local positive pairwise dependencies among labels. *J. Inf. Knowl. Manag.* **2020**, *19*, 2040017. [CrossRef]
23. Alluwaici, M.; Junoh, A.K.; Ahmad, F.K.; Mohsen, M.F.M.; Alazaidah, R. Open research directions for multi label learning. In Proceedings of the 2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), Penang Island, Malaysia, 28–29 April 2018; pp. 125–128. [CrossRef]
24. Dimou, A.; Tsoumakas, G.; Mezaris, V.; Kompatsiaris, I.; Vlahavas, I. An empirical study of multi-label learning methods for video annotation. In Proceedings of the 2009 Seventh International Workshop on Content-Based Multimedia Indexing, Crete, Greece, 3–5 June 2009; IEEE: New York, NY, USA, 2009; pp. 19–24.
25. Peters, S.; Denoyer, L.; Gallinari, P. Iterative annotation of multi-relational social networks. In Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining, Odense, Denmark, 9–11 August 2010; IEEE: New York, NY, USA, 2010; pp. 96–103.
26. Wang, J.; Neskovic, P.; Cooper, L.N. Improving nearest neighbor rule with a simple adaptive distance measure. *Pattern Recognit. Lett.* **2007**, *28*, 207–213. [CrossRef]
27. Trohidis, K.; Tsoumakas, G.; Kalliris, G.; Vlahavas, I.P. Multi-label classification of music into emotions. In Proceedings of the ISMIR, Philadelphia, PA, USA, 14–18 September 2008; Volume 8, pp. 325–330.
28. Barutcuoglu, Z.; Schapire, R.E.; Troyanskaya, O.G. Hierarchical multi-label prediction of gene function. *Bioinformatics* **2006**, *22*, 830–836. [CrossRef]
29. Elisseeff, A.; Weston, J. A kernel method for multi-labelled classification. In *Advances in Neural Information Processing Systems 14 (NIPS 2001)*; Dietterich, T., Becker, S., Ghahramani, Z., Eds.; The MIT Press: Cambridge, MA, USA, 2001; Volume 14.
30. Skabar, A.; Wollersheim, D.; Whitfort, T. Multi-label classification of gene function using MLPs. In Proceedings of the 2006 IEEE International Joint Conference on Neural Network Proceedings, Vancouver, BC, Canada, 16–21 July 2006; IEEE: New York, NY, USA, 2006; pp. 2234–2240.
31. Chan, A.; Freitas, A.A. A new ant colony algorithm for multi-label classification with applications in bioinfomatics. In Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, Seattle, WA, USA, 8–12 July 2006; pp. 27–34.
32. Diplaris, S.; Tsoumakas, G.; Mitkas, P.A.; Vlahavas, I. Protein classification with multiple algorithms. In Proceedings of the Advances in Informatics: 10th Panhellenic Conference on Informatics, PCI 2005, Volas, Greece, 11–13 November 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 448–456.
33. Kawai, Y.; Fujii, Y.; Akimoto, K.; Takahashi, M. Evaluation of Serum Protein Binding by Using in Vitro Pharmacological Activity for the Effective Pharmacokinetics Profiling in Drug Discovery. *Chem. Pharm. Bull.* **2010**, *58*, 1051–1056. [CrossRef]
34. Krohn-Grimberghe, A.; Drumond, L.; Freudenthaler, C.; Schmidt-Thieme, L. Multi-relational matrix factorization using bayesian personalized ranking for social network data. In Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, Seattle, WA, USA, 8–12 February 2012; pp. 173–182.
35. Tang, L.; Liu, H. *Community Detection and Mining in Social Media*; Morgan & Claypool Publishers: San Rafael, CA, USA, 2010.
36. Soonsiripanichkul, B.; Murata, T. Domination dependency analysis of sales marketing based on multi-label classification using label ordering and cycle chain classification. In Proceedings of the 2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), Kumamoto, Japan, 10–14 July 2016; IEEE: New York, NY, USA, 2016; pp. 1048–1053.
37. Nassar, O.A.; Al Saiyd, N.A. The integrating between web usage mining and data mining techniques. In Proceedings of the 2013 5th International Conference on Computer Science and Information Technology, Amman, Jordan, 27–28 March 2013; IEEE: New York, NY, USA, 2013; pp. 243–247.
38. Quinlan, J.R. Combining instance-based and model-based learning. In Proceedings of the Tenth International Conference on Machine Learning, Amherst, MA, USA, 27–29 July 1993; pp. 236–243.
39. Zhang, M.L.; Zhou, Z.H. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognit.* **2007**, *40*, 2038–2048. [CrossRef]
40. Zhang, M.L.; Zhou, Z.H. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Trans. Knowl. Data Eng.* **2006**, *18*, 1338–1351. [CrossRef]
41. Freund, Y.; Schapire, R.E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [CrossRef]
42. Zhang, M.L.; Peña, J.M.; Robles, V. Feature selection for multi-label naive Bayes classification. *Inf. Sci.* **2009**, *179*, 3218–3229. [CrossRef]
43. Thabtah, F.A.; Cowling, P.; Peng, Y. MMAC: A new multi-class, multi-label associative classification approach. In Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04), Brighton, UK, 1–4 November 2004; IEEE: New York, NY, USA, 2004; pp. 217–224.

44. Alazaidah, R.; Ahmad, F.K. Trending challenges in multi label classification. *Int. J. Adv. Comput. Sci. Appl.* **2016**, *7*, 127–131. [CrossRef]
45. Abdelhamid, N.; Ayesh, A.; Hadi, W. Multi-label rules algorithm based associative classification. *Parallel Process. Lett.* **2014**, *24*, 1450001. [CrossRef]
46. Veloso, A.; Meira, W.; Gonçalves, M.; Zaki, M. Multi-label lazy associative classification. In Proceedings of the Knowledge Discovery in Databases (PKDD 2007: 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, Warsaw, Poland, 17–21 September 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 605–612.
47. Li, X.; Qin, D.; Yu, C. ACCF: Associative classification based on closed frequent itemsets. In Proceedings of the 2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery, Shandong, China, 18–20 October 2008; IEEE: New York, NY, USA, 2008; Volume 2, pp. 380–384.
48. Liu, B.; Hsu, W.; Ma, Y. Integrating classification and association rule mining. In Proceedings of the Kdd, New York, NY, USA, 27–31 August 1998; Volume 98, pp. 80–86.
49. Abdelhamid, N.; Ayesh, A.; Thabtah, F.; Ahmadi, S.; Hadi, W. MAC: A multiclass associative classification algorithm. *J. Inf. Knowl. Manag.* **2012**, *11*, 1250011. [CrossRef]
50. Alazaidah, R.; Almaiah, M.A. Associative classification in multi-label classification: An investigative study. *Jordanian J. Comput. Inf. Technol.* **2021**, *7*. Available online: https://www.proquest.com/openview/9a1e4545ef6dd7deea31b808f011119c/1?pq-origsite=gscholar&cbl=5500744 (accessed on 11 February 2023). [CrossRef]
51. Huang, S.J.; Zhou, Z.H. Multi-label learning by exploiting label correlations locally. In Proceedings of the AAAI Conference on Artificial Intelligence, Toronto, ON, US, 22–26 July 2012; Volume 26, pp. 949–955.
52. Alazaidah, R.; Ahmad, F.K.; Mohsin, M. Multi label ranking based on positive pairwise correlations among labels. *Int. Arab J. Inf. Technol.* **2020**, *17*, 440–449. [CrossRef]
53. Liu, H.; Setiono, R. Feature selection via discretization. *IEEE Trans. Knowl. Data Eng.* **1997**, *9*, 642–645.
54. Triguero, I.; González, S.; Moyano, J.M.; García López, S.; Alcalá Fernández, J.; Luengo Martín, J.; Fernández Hilario, A.L.; Jesús Díaz, M.J.D.; Sánchez, L.; Herrera Triguero, F.; et al. KEEL 3.0: An Open Source Software for Multi-Stage Analysis in Data Mining. 2017. Available online: https://digibug.ugr.es/handle/10481/49780 (accessed on 15 September 2022).
55. Fürnkranz, J.; Hüllermeier, E.; Loza Mencía, E.; Brinker, K. Multilabel classification via calibrated label ranking. *Mach. Learn.* **2008**, *73*, 133–153. [CrossRef]
56. Boutell, M.R.; Luo, J.; Shen, X.; Brown, C.M. Learning multi-label scene classification. *Pattern Recognit.* **2004**, *37*, 1757–1771. [CrossRef]
57. Tsoumakas, G.; Vlahavas, I. Random k-labelsets: An ensemble method for multilabel classification. In Proceedings of the Machine Learning (ECML 2007): 18th European Conference on Machine Learning, Warsaw, Poland, 17–21 September 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 406–417.
58. Read, J.; Pfahringer, B.; Holmes, G.; Frank, E. Classifier chains for multi-label classification. *Mach. Learn.* **2011**, *85*, 333–359. [CrossRef]
59. Read, J.; Pfahringer, B.; Holmes, G. Multi-label classification using ensembles of pruned sets. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; IEEE: New York, NY, USA, 2008; pp. 995–1000.
60. Wu, X. A Bayesian discretizer for real-valued attributes. *Comput. J.* **1996**, *39*, 688–691. [CrossRef]