Neural Networks for Feature-Extraction in Multi-Target Classification

Brendon Gouveia Cambuí

Brendon Gouveia Cambuí

Orientador: ***Prof. Dr. Ricardo Cerri***

Dissertação apresentada ao Departamento de Computação da Universidade Federal de São Carlos - DC/UFSCar, como parte dos requisitos necessários à obtenção do título de Mestre em Ciências da Computação.

DC/UFSCar - São Carlos
Agosto/2020

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Defesa de Dissertação de Mestrado do candidato Brendon Gouveia Cambuí, realizada em 21/08/2020.

Comissão Julgadora:

Prof. Dr. Ricardo Cerri (UFSCar)

Prof. Dr. Diego Furtado Silva (UFSCar)

Prof. Dr. João Luis Garcia Rosa (USP)

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

O Relatório de Defesa assinado pelos membros da Comissão Julgadora encontra-se arquivado junto ao Programa de Pós-Graduação em Ciência da Computação.

Agradecimentos

Em primeiro lugar, agradeço a Deus e à minha família, por tornar possível este momento. Considero-os como grandes responsáveis por mais essa conquista. Em especial, à minha mãe, Priscila. Só nós dois sabemos o que já passamos, e não desistimos. Se hoje eu estou tendo esta oportunidade, eu devo a você.

Eu agradeço também à minha namorada, Caroline, por estar ao meu lado em todos os momentos difíceis, por acreditar em mim mais do que eu mesmo, e principalmente por ter me ajudado a superar todos os desafios que tivemos que enfrentar. Obrigado por ser a minha companheira de todas as horas. Você é a minha inspiração.

Agradeço aos meus colegas do DC e do laboratório BIOMaL, por todos os momentos que passamos juntos. Em especial ao Joel, que me ajudou muito no começo da minha pesquisa e me auxiliou a encontrar o norte que eu precisava em vários momentos.

Também agradeço imensamente ao meu orientador, prof. Dr. Ricardo Cerri, pela orientação e pelo apoio durante todo o meu período do Mestrado. Lembro-me que a primeira aula que tive em São Carlos foi com você, e quando terminou você me disse para eu pesquisar sobre o que era multi-target, pois conversaríamos depois. Mal sabia eu que essa pesquisa mudaria a minha vida. Obrigado por ter confiado em um garoto recém-formado de 21 anos que não possuía nenhuma experiência. Você me ensinou muito, e me deu um ótimo direcionamento. Mais do que tudo, obrigado por acreditar em mim quando nem eu mais acreditava, por não me deixar desistir. Eu devo muito a você.

Por fim, estendo este agradecimento a todo o corpo docente e funcionários do DC/PPGCC-UFSCar que diretamente ou indiretamente contribuíram para que este momento se tornasse possível.

Abstract

Multi-target learning is a prediction task where each data example is associated with multiple target-variables (outputs) simultaneously. One of the challenges in this research field is related to the high dimensionality of data present in multi-target datasets, and also the high number of target variables having dependencies among themselves. In such scenarios, it is crucial to extract lower-dimensional representations from the original input-space, such that these can be provided as input to other multi-target predictors. In this research, we proposed the use of Auto-Encoders and Restricted Boltzmann Machines as feature extractors in several multi-target classification datasets publicly available. Results were evaluated considering state-of-the-art multi-target classification methods and evaluation measures in the literature. The experiments showed that the neural networks were able to keep the predictive performance even when the extracted features corresponded to a dimension size equivalent to 10% of the original number of features and, in some cases, getting better results than the original datasets.

Keywords: Multi-Target Classification, Auto-Encoders, Restricted Boltzmann Machine, Feature-Extraction, Dimensionality Reduction.

Table of Contents

List of Abbreviations

AE Auto-Encoder

BR Binary Relevance

CC Classifier Chains

CP Class Powerset

CR Class Relevance

CRS Class Relevance Stacking

dAE Denoising Auto-Encoder

D-AE Deep Auto-Encoder

D-dAE Deep-Denoising Auto-Encoder

DBN Deep Belief Network

ECC Ensemble of Classifier Chains

EMR Exact Match Ratio

HL Hamming Loss

HS Hamming Score

IC Independent Classifiers

LSTM Long-Short Term Memory

MSE Mean Squared Error

MLC Multi-Label Classification

MT Multi-Target

MTC Multi-Target Classification

PCA Principal Component Analysis

RBM Restricted Boltzmann Machine

RNN Recurrent Neural Network

SCC Super-Class Classifier

ST Single-Target

VAE Variational Auto-Encoder

List of Figures

List of Tables

Chapter 1

Introduction

Classification is one of the fundamental tasks in the machine learning field. We can define a classification problem as: given a dataset with n examples $(\mathbf{x}_i, y_j)$, where $\mathbf{x}_i$ is the input attribute vector that describes the example (features) and y_j its associated class (target), define a function that maps $\mathbf{x}_i$ to its associated class y_j, where $1 \leq i \leq n$ and $1 \leq j \leq k$, being k the number of possible classes in the dataset.

Traditionally, classification problems have two or more possible classes, where each example is classified to only one of these classes. When there are two possible classes in the dataset, we call it as a binary classification problem, and multi-class when the dataset presents more than two possible classes. Whenever a classification problem is binary or multi-class, we denominate it as a Single-Target (ST) problem, since its predictions will produce a single output value.

However, there are more complex classification tasks, where each example can be classified into multiple classes simultaneously. This task is denominated as Multi-Label Classification (MLC). In order to label the example to multiple classes, multi-label problems use a vector $\mathbf{y}$ of k binary outputs, where each value corresponds to a class of the dataset. Thus, a multi-label prediction will produce multiple output variables, so we can define it as a Multi-Target (MT) problem. In fact, every classification task involving multiple outputs is a MT task.

Although multi-label classification problems have multiple binary targets, there are classification problems where each target can have more than two possible values, and these problems are denominated as Multi-Target Classification (MTC). It is important to point out that MLC can be viewed as a MTC task, since it has multiple binary targets. But the inverse is not true. The MT term is considered as a generalization of the MLC task, being the MLC term applied when the prediction involves binary targets only.

Being a generalization of the MLC task, the fundamental goal of MTC is the same: to the relationships (dependencies) among features and targets, and to deal with the computational complexity of such task. If the classes are completely unrelated, it should be enough to create a separate independent model for each class, i.e Single-Target (ST). However, this is unlikely to occur.

Also, MT methods, most of the times, derive from MLC, adapting a specific learning approach (e.g. k-nearest neighbors, decision trees, support vector machines) for directly handle MT data (Jia and Zhang, 2020c), or transforming the MT task into one or more ST tasks that can be solved with off-the-shelf learning algorithms (Tsoumakas et al., 2009c). Commonly used approaches can be categorized into those that model single labels, pairs of labels and sets of labels (Yan et al., 2017). Approaches that model single labels include the one-versus-all (also known as binary relevance) approach, methods based on stacked generalization (Tsoumakas et al., 2009a) and the classifier chains algorithm (Read et al., 2011).

Moreover, in this research, we used Restricted Boltzmann Machines (RBMs) (Smolensky, 1986) and Auto-Encoders (AEs) (Cottrell et al., 1987) as feature extractors in the above described multi-target scenarios. These neural networks were already successfully used as feature extractors in different high-dimensional applications (Hinton and Salakhutdinov, 2006, Lange and Riedmiller, 2010, Zabalza et al., 2016).

The rest of this chapter is organized as follows: we discuss the motivation for this research in Section 1.1, and then we present our hypothesis and objectives in Section 1.2. Finally, in Section 1.3, we present how this document is organized.

1.1 Motivation

Traditional prediction problems deal with a set of instances that have a single-target value associated with them. However, several real-life problems include a set of targets: instead of a single property, one is interested in predicting multiple properties. The multi-target learning is also known in the literature as multi-output or multivariate learning (Spyromitros-Xioufis et al., 2012). Applications include the prediction of river water quality parameters from bio-indicator data (Džeroski et al., 2000), olfaction prediction in molecules (Keller et al., 2017), estimation of energy performance in residential buildings (Tsanas and Xifara, 2012) and modelling drivers' behaviour for autonomous vehicles (An et al., 2020).

The multi-label learning has a very similar setting, which can be described as a particular case of the multi-target problem that only involves binary targets, considered as labels that can be 1 or 0 for any data instance (Kocev et al., 2013). Applications include protein function prediction (Cerri et al., 2016), protein sub-cellular localization (Wan et al., 2012) and audio classification (Briggs et al., 2013).

We also highlight the difficulty to extract compact representations keeping the predictive power, since multiple targets are predicted using these representations. Therefore, there exists the need of investigating different architectures of Auto-Encoders (AEs) and Restricted Boltzmann Machines (RBMs) and tuning their hyper-parameters for each scenario.

1.2 Hypothesis and Objectives

The main goal of this research is to take advantage of neural networks to extract features from the high-dimensional datasets and fed then to the state-of-the-art multi-target classification algorithms. For this purpose, were selected three methods, Principal Component Analysis (PCA) (Jolliffe, 2011), Auto-Encoder (AE) (Cottrell et al., 1987) and Restricted Boltzmann Machine (RBM) (Smolensky, 1986), for feature extraction, in order to compare which one achieves the best performance in the final classification.

Besides PCA, which is a direct linear algorithm, neural networks were already successfully used as feature extractors in different applications (Hinton and Salakhutdinov, 2006, Lange and Riedmiller, 2010, Zabalza et al., 2016, Higgins et al., 2017), but not considered in multi-target scenarios. In our work, the extracted representations were used to feed the state-of-the-art methods for multi-target classification. All methods were evaluated with evaluation measures specifically designed for multi-target scenarios. We also used statistical tests to verify if our results were statistically significant. A framework is proposed in Chapter 5, where details of our methods and experiments are presented.

1.3 Document Organization

The remainder of this document is organized as follows. First, we review the dimensionality reduction concept to understand how the feature extraction techniques work (Chapter 2). Then, we present an overview of the neural network architectures that are used in this research (Chapter 3). Moreover, a literature overview in multi-target classification methods is presented (Chapter 4). In Chapter 5 we present the framework proposed in this research, while Chapter 6 presents the experiments and discussions. Finally, we present some conclusions and future research directions (Chapter 7).

CHAPTER 2

Fundamentals of Dimensionality Reduction

Many real-world data, such as noise signals, images, or corporation data, typically have a high dimensionality characteristic. Keeping in mind the goal of dealing with such information satisfactorily, it is desirable to reduce data dimensions in many situations. Therefore, dimensionality reduction is the action of representing high-dimensional data into equivalent lower-dimensional data (Van Der Maaten et al., 2009).

In a perfect scenario, such reduced representation should have a dimensionality that compares to the intrinsic dimensionality of the information. The notion of dimensionality of data can be defined as the base number of parameters expected to represent the observed properties of an original dataset. Dimensionality reduction is vital in numerous areas, since it mitigates several undesired properties of high-dimensional spaces.

Thus, dimensionality reduction encourages classification, visual representation, and compression of high-dimensional information. Traditionally, such technique is performed utilizing direct strategies, for example, Principal Component Analysis (PCA). There are situations however, where these direct strategies can not satisfactorily handle complex nonlinear information.

In the remainder this chapter we introduce the basic concepts of dimensionality reduction, presenting the differences between feature extraction and feature selection.

2.1 Basic Concepts

As introduced in the beginning of this chapter, dimensionality reduction consists in the task of transforming some high-dimensional data into an equivalent lower-dimensional representation, ensuring that compressed data represents the similar information concisely. Methods based on this principle are typically used in machine learning problems in order to obtain more representative features, easing data visualization, and helping in the classification and regression tasks (Hinton and Salakhutdinov, 2006).

For further definitions of dimensionality reduction methods, a concept that needs to be comprehended is the intrinsic dimensionality. The idea behind this concept is to represent a manifold of a higher-dimensional dataset that represents a smaller portion of data (Van Der Maaten et al., 2009).

The dimensionality reduction task can be described as follows: Consider a dataset $\mathbf{D}$ containing n instances, with m attributes, constituting the matrix with dimensions $d = n \times m$. Also, assume that $\mathbf{D}$ has an intrinsic dimension d' (where $d' < d$) representing the points in n that are in (or near) a manifold with dimension d', contained inside the d-dimensional space. So, the dimensionality reduction task consists in the transformation of the dataset $\mathbf{D}$ with dimension d into a dataset $\mathbf{D}'$, with dimension d'. Each attribute of the original dataset is denoted as $\mathbf{x}_i$, with $1 \leq i \leq m$.

An example of dimensionality reduction based on the intrinsic dimension (highlighted in bold) is presented in Figure 2.1. In this scenario, the intrinsic dimension represents a lower number of attributes and instances. Recall that in our research the dimensionality reduction task is focused in just dealing with the number of attributes, and not the number of instances, since we use neural networks and these algorithms present better performances with a large number of instances (Hinton and Salakhutdinov, 2006).

$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	...	$\mathbf{x}_m$
3	-1	3		1
-2	4	4		1
1	**-2**	**3**		0
0	**4**	**4**		1
-1	**4**	**-1**		0
5	**-2**	**1**		1
1	**1**	**-1**		1
-2	-1	-2		0

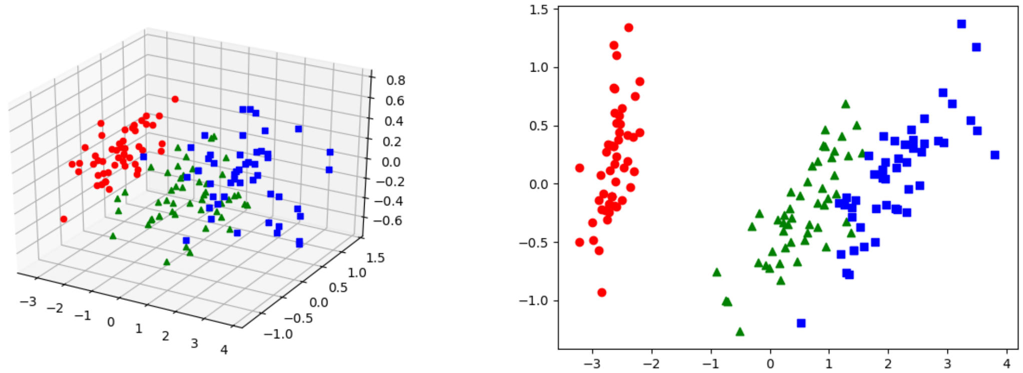
$\Longrightarrow$

$\mathbf{x}'_1$	$\mathbf{x}'_2$	$\mathbf{x}'_3$
2	**5**	**3**
4	**3**	**2**
-1	**1**	**0**
-2	**-2**	**-1**
0	**-4**	**1**

Figure 2.1: Example of dimensionality reduction based on the intrinsic dimension (denoted in bold).

In machine learning, problems usually have a set of input and output variables, where the objective is to map a function that assimilates these input values to the output values. We can therefore define a dataset that represents a machine learning problem as $\mathbf{D} = (\mathbf{X}, \mathbf{Y})$, where $\mathbf{X}$ represents the input variables and $\mathbf{Y}$ represents the output variables. We call the input variables of a machine learning problem as features, and the output variables as targets. Likewise, $\mathbf{X}$ and $\mathbf{Y}$ are also called feature and target spaces, respectively. In our research, as we are dealing with multi-target problems, i.e., problems that have several output variables, we will apply the concept of dimensionality reduction only into the feature space.

An example of dimensionality reduction applied into the feature space is illustrated in Figure 2.2. The figure shows two representations obtained from the Iris dataset (Blake and Merz, 1998), which is originally composed by 5 attributes (being 4 input variables and 1 output variables). The two representations show a 3-dimensional and 2-dimensional portrayal of the features present in the Iris dataset. Besides reducing the number of features, these reduced representations provide better data visualization. Therefore, in Sections 2.2 and 2.3 we will present two well-known dimensionality reduction methods which are focused on reducing the number of attributes.



(a) 3-dimensional representation

(b) 2-dimensional representation

Figure 2.2: Representations of the data obtained after executing feature extraction using the PCA method on the Iris dataset, which originally contains 4 features.

2.2 Feature Selection

Feature selection is a dimensionality reduction technique used in machine learning and statistics. Otherwise called variable selection or attribute selection, it can be described as the way toward choosing a subset of relevant attributes for use in the development of a prediction model (Guyon and Elisseeff, 2003).

A feature selection algorithm can be viewed as the combination of a search system for proposing new feature subsets, alongside an evaluation measure which scores the diverse feature subsets. The basic principle of such technique is to test every possible subset of features finding the one which limits the error rate in some task.

In Statistics, one of the most well known types of feature selection is step-wise regression, which is a wrapper system. It works in a greedy fashion, including the best feature (or erases the most noticeably bad feature) at each iteration. The main issue is choosing when to stop the process. This is regularly done by optimizing some objective function. Also, more robust strategies have been investigated, for example, branch and bound and piece-wise system (Zhang, 2016).

In brief, feature selection only removes the most “unnecessary” features, which is not exactly the purpose of this research, since we want to represent all the features present in a dataset in a reduced dimension. For this reason, a generative model is needed, which creates a new representation of the features, instead of removing the less significant ones. In Section 2.3, we present a generative model, called feature extraction.

2.3 Feature Extraction

As discussed in the beginning of this chapter, modern applications typically involve huge amounts of data, sometimes redundant, or too large to be processed. At that point, it may be necessary to transform the original data into a lower-dimensional data, reducing the number of features. Also described in Section 2.2, selecting a subset of the relevant features is called feature selection. The highlighted features are likely to contain the significant data from the dataset, with the goal that a given task can be performed by utilizing the reduced representation of the data rather than the entire original data.

On the other hand, feature extraction consists in decreasing the number of features required to represent a larger arrangement of data by generating new features. When handling complex data, one of the significant problems comes from the number of features, which generally requires a lot of memory and processing time. Additionally, for prediction tasks, it might generate over-fitting on training instances and predict inadequately to new instances. Feature extraction is a general term for strategies for building combinations of the attributes to overcome these problems while yet portraying the information with exactness (Guyon and Elisseeff, 2006).

Generally, a feature extraction technique begins from a starting set of estimated information and generates derived values (features) proposed to be instructive and non-repetitive, encouraging learning and generalization, also providing, sometimes, better human readability (Hinton and Salakhutdinov, 2006).

In Chapter 3, we will present how neural networks, one of the most well-known techniques for reducing dimensionality, are used to perform feature extraction by reconstructing the input data it receives.

Chapter 3

Neural Networks for Feature Extraction

As mentioned in the Introduction (Chapter 1), the main objective of this research is to use Auto-Encoders (AEs) (Cottrell et al., 1987) and Restricted Boltzmann Machines (RBMs) (Smolensky, 1986) as feature extractors in previously described multi-target classification scenarios. These neural networks were already successfully used as feature extractors in different applications (Hinton and Salakhutdinov, 2006, Lange and Riedmiller, 2010, Zabalza et al., 2016).

One of the reasons for using neural networks as feature extractors is the attempt to reach better results than direct methods. In this research, we compared the results obtained with extracted features by neural networks with the ones obtained when extracting features using Principal Component Analysis (PCA).

3.1 Restricted Boltzmann Machine

First presented in the literature as Harmonium (Smolensky, 1986), a Restricted Boltzmann Machine (RBM) is an energy-based stochastic system to model the probabilistic distribution of data, generally used into pattern recognition and classification. The RBM is also one of the most well known techniques in the feature learning field (Hinton and Salakhutdinov, 2006). An RBM consists of a stochastic neural network, composed by 2 layers (visible and hidden), and its learning process consists in estimating the weights between the visible and the hidden layers.

Following this definition, the architecture of an RBM is composed by a visible layer $\mathbf{x}$ with n units, and a hidden layer $\mathbf{h}$ with m units, interconnected by a weight matrix $\mathbf{W}_{n \times m}$, where W_{ij} references the weight between the visible unit x_i and the hidden unit h_j. Both visible and hidden layers are binary units, meaning that $\mathbf{x} \in \{0, 1\}^n$ and $\mathbf{h} \in \{0, 1\}^m$.

Equation 3.1 gives the energy function of an RBM, where c and b represent the biases of the visible and hidden layer respectively. An illustration of a RBM is presented in Figure 3.1

$$E(\mathbf{x},\mathbf{h}) = \sum_{i=1}^{m} c_i x_i - \sum_{j=1}^{n} b_j h_j - \sum_{i=1}^{m}\sum_{j=1}^{n} x_i h_i w_{ij} \tag{3.1}$$

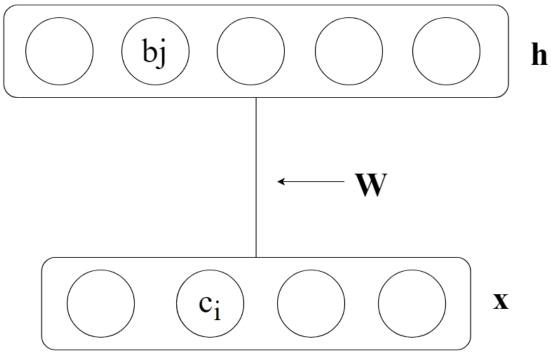


Figure 3.1: Illustration of a Restricted Boltzmann Machine, where c_i and b_j represent the biases of the visible and hidden units.

Given the energy function, a RBM computes the probability distribution, also called as Gibbs distribution, $P(\mathbf{x},\mathbf{h})$, which is obtained by Equation 3.2, where Z represents the sum of all possible states of the machine, also known as partition function.

$$P(\mathbf{x},\mathbf{h}) = \frac{exp(-E(\mathbf{x},\mathbf{h}))}{Z} \tag{3.2}$$

Since a RBM is a bipartite graph, the activations of both visible and hidden units are mutually independent, thus leading to the following conditional probabilities

$$P(\mathbf{x}|\mathbf{h}) = \prod_{i=1}^{m} P(x_i|\mathbf{h}) \Rightarrow P(x_i = 1|\mathbf{h}) = sigm\left(\sum_{j=1}^{n} w_{ij} h_j + c_i\right) \tag{3.3}$$

and

$$P(\mathbf{h}|\mathbf{x}) = \prod_{j=1}^{n} P(h_j|\mathbf{x}) \Rightarrow P(h_j = 1|\mathbf{x}) = sigm\left(\sum_{i=1}^{m} w_{ij} x_j + b_j\right) \tag{3.4}$$

in the forward-phase and in the backward-phase, respectively.

Considering the parameters of a RBM being $\rho = (\mathbf{W}, c, b)$, the learning algorithm has the objective of maximizing the product of probabilities given a training data. However, the represented scenario is considered to be a naive RBM architecture, due the fact that computing all states of the system can be intractable.

With that in mind, the learning process can be executed based on the Contrastive Divergence algorithm (Hinton, 2002). The idea of this training method to initialize the visible units with a training sample using Equation 3.4 (forward-phase), and then compute the states of the visible unit using Equation 3.3 (backward-phase, also called as reconstruction). The idea behind this algorithm is to substitute E by an estimated $\widetilde{\mathbf{x}}$ obtained by Gibbs sampling, and then start the sampling by the state which the visible layer is encountered. This process is repeated k times, always aiming to lowering E, until the system reaches a stable point.

Assuming $E[\mathbf{hx}]$ as the data learned by the system, it can be computed as:

$$E[\mathbf{hx}]^{model} = P(\widetilde{\mathbf{h}}|\widetilde{\mathbf{x}})\widetilde{\mathbf{x}}^T \tag{3.5}$$

Therefore, Equation 3.6 leads to a learning rule for updating the weight matrix $\mathbf{W}_{m \times n}$, as follows:

$$\begin{aligned} \mathbf{W}^{t+1} &= \mathbf{W}^t + \eta(E[\mathbf{hx}]^{data} - E[\mathbf{hx}]^{model}) \\ &= \mathbf{W}^t + \eta(P(\mathbf{h}|\mathbf{x})\mathbf{x}^T - P(\widetilde{\mathbf{h}}|\widetilde{\mathbf{x}})\widetilde{\mathbf{x}}^T) \end{aligned} \tag{3.6}$$

where $\mathbf{W}^t$ represents the weight matrix at the moment t and η represents the learning rate. It is also necessary to update the biases as defined in the following formulas:

$$\begin{aligned} \mathbf{c}^{t+1} &= \mathbf{c}^t + \eta(\mathbf{x} - E[\mathbf{x}]^{model}) \\ &= \mathbf{c}^t + \eta(\mathbf{x} - \widetilde{\mathbf{x}}) \end{aligned} \tag{3.7}$$

and

$$\begin{aligned} \mathbf{b}^{t+1} &= \mathbf{b}^t + \eta(E[\mathbf{x}]^{data} - E[\mathbf{x}]^{model}) \\ &= \mathbf{b}^t + \eta(P(\mathbf{h}|\mathbf{x}) - P(\widetilde{\mathbf{h}}|\widetilde{\mathbf{x}})) \end{aligned} \tag{3.8}$$

where $\mathbf{c}^t$ and $\mathbf{b}^t$ represent the biases of the visible and hidden units at the moment t.

3.1.1 Deep Belief Network

A Deep Belief Network (DBN) is composed of stacked RBMs trained in a greedy fashion using the Contrastive Divergence algorithm presented in Section 3.1. This means that each RBM is trained independently, and each layer does not consider the other during the training process. Therefore, a DBN with L layers will have L weight matrices $\mathbf{W}_i$, each one from the RBM at layer i. The hidden units of the RBM at layer i will become the input for the visible units of the RBM at layer $i + 1$, and so on.

The approach proposed by Hinton and Salakhutdinov (2006) for training DBNs also considers a fine-tuning as a final step after the training of each RBM. Such procedure can be performed by means of a back-propagation or gradient descent algorithm, for instance, in order to adjust the matrices $\mathbf{W}_i$, $1 \leq i \leq L$. Figure 3.2 presents an illustration of a Deep Belief Network with three RBMs.

3.2 Auto-Encoder

Traditionally, a feed-forward neural network is capable of classifying examples by comparing the output generated by the model with the desired output and computing an error. This error is then used to adjust the weights with the back-propagation and gradient descent algorithms (Goodfellow et al., 2016). In practice, given a dataset $\mathbf{D}$ with n examples, a neural network tries

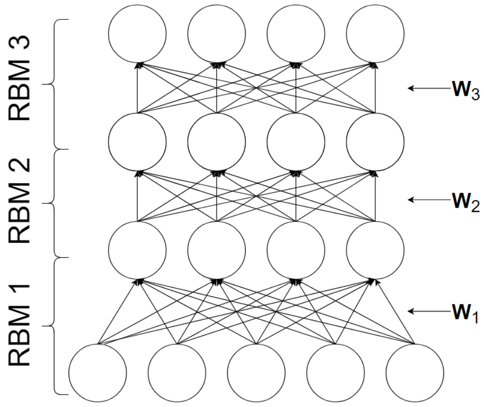


Figure 3.2: Illustration of a Deep Belief Network composed by one visible layer and two hidden layers.

to learn a function $f(\mathbf{x}_i) \approx \mathbf{y}_i$, where $1 \leq i \leq n$, being $\mathbf{x}_i$ the input data, and $\mathbf{y}_i$ the correct output (label) for example $\mathbf{x}_i$.

An Auto-Encoder (AE), on the other hand, is a neural network architecture that learns to reconstruct the input data fed to the algorithm in an unsupervised fashion. In other words, the goal of an Auto-Encoder (AE) is not to classify examples, but to reproduce its own input data (Cottrell et al., 1987). AEs are capable of achieving this by comparing the output generated from the model with the original input data, making the network learn to predict its own input data, i.e, the Auto-Encoder tries to learn $f(\mathbf{x}_i) \approx \mathbf{x}_i$.

For further definitions in this section, consider $\hat{\mathbf{x}}_i$ the reconstruction of the example $\mathbf{x}_i$. Consider also $\mathbf{W}$ the weight matrix between the input layer $\mathbf{x}$ and the code layer $\mathbf{h}(\mathbf{x})$, and $\mathbf{W}'$ the weights between the code layer and the output layer $\hat{\mathbf{x}}$. In addition, consider $\mathbf{b}$ and $\mathbf{c}$ the biases for the code and output layers, respectively. Figure 3.3 gives an illustrative example of an Auto-Encoder.

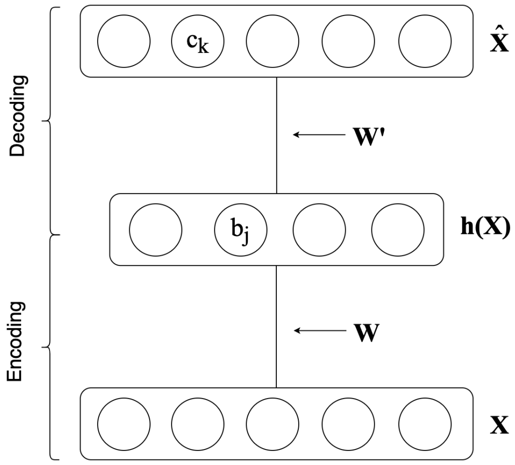


Figure 3.3: Illustration of an Auto-Encoder architecture and the encoding-decoding process.

The learning process of an Auto-Encoder is basically composed by two steps: encoding and decoding (that's why it was named Auto-Encoder). The encoding process occurs in the activation from the input layer to the code layer, as usually occurs in a traditional feed-forward neural network. This is given by Equation 3.9, where the data from the input layer is stored in the code layer after the activation. The equation uses the logistic sigmoid (*sigm*) function, but others could be also used for this purpose.

$$h(\mathbf{x}) = sigm(b + \mathbf{W}\mathbf{x}) \tag{3.9}$$

The decoding step consists in reconstructing the input data based on what the neural network had learned in the code layer. The process to obtain the output is also similar to an usual feed-forward neural network, given by Equation 3.10.

$$\hat{\mathbf{x}} = sigm(c + \mathbf{W}\mathbf{h}(\mathbf{x})) \tag{3.10}$$

As previously mentioned in this section, Auto-Encoders (AEs) are capable of reconstructing the input data by comparing the obtained output with the original example. In order to evaluate this reconstruction, AEs implement a loss function. Two of the most well known functions suitable for this purpose (Goodfellow et al., 2016) are the cross-entropy for binary data, and the squared Euclidean distance for real-valued data. They are given by Equations 3.11 and 3.12 respectively, where k represents the logical units in the input and output layers.

$$l(\hat{\mathbf{x}}) = -\sum_{k} (\mathbf{x}_k \, log(\hat{\mathbf{x}}_k) + (1 - \mathbf{x}_k) \, log(1 - \hat{\mathbf{x}}_k)) \tag{3.11}$$

$$l(\hat{\mathbf{x}}) = \frac{1}{2} \sum_{k} (\hat{\mathbf{x}}_k \; - \; \mathbf{x}_k)^2 \tag{3.12}$$

After defining the loss function, it is possible to adjust the weights **W** and **W′**, in an process similar to a feed-forward network. Thus, the back-propagation algorithm (Goodfellow et al., 2016) can be used for such task, where the gradient can be defined as

$$\nabla_{\mathbf{x}(t)} {}^{l(\hat{\mathbf{x}}^{(t)})} = \hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)} \tag{3.13}$$

where t represents the neural network layer states at moment t.

It is worth pointing out that when the code layer contains less logic units than the input layer, the Auto-Encoder will naturally learn a compact representation of the input data. Therefore, that is the way Auto-Encoders are used for feature extraction. Thus, by applying Equation 3.9, the result will be the compressed representation of the original input data $\mathbf{x_i}$.

In the next section, we discuss how deep learning techniques can improve the performance of Auto-Encoders, likewise occurs in multi-layer perceptron networks.

3.2.1 Applicability of Deep Learning Techniques in Feature Extraction

The application of deep learning techniques in Auto-Encoders is common. Since AEs are naturally feed-forward neural networks, the corresponding deep architectures are known as deep auto-encoders.

One of the benefits of using deep neural networks is the possibility of combining other neural network techniques and architectures to improve performance. This was proposed by Hinton and Salakhutdinov (2006), where they pre-trained a deep Auto-Encoder with a DBN, increasing predictive performance and reducing the computational time. Although LeCun and Ranzato (2013) showed that the use of techniques such dropout (Hinton et al., 2012) and rectification (ReLU) (Agarap, 2018) present more competitive results, these researches show the potential of combining Auto-Encoder and deep learning for feature extraction.

It is worth mentioning, as presented in (LeCun and Ranzato, 2013), that these novel methods drastically improved the performance of Auto-Encoders on dealing with datasets with a huge number of features. There are multi-target regression datasets whose use is currently prohibitive with conventional neural networks, since they have over forty thousand features (Tsoumakas et al., 2011).

Thus, deep neural networks have the potential to outperform traditional neural network architectures when dealing with regression datasets. This is why we used some deep variations in our experiments. Some future research can explore Variational Auto-Encoders (VAEs) and the effectiveness of Recurrent Neural Networks (RNNs), specially Long-Short Term Memory (LSTM), on such tasks.

In Chapter 4 we discuss about the multi-target classification paradigm, as well as the classifiers that will be used in this research, which will be fed by the features extracted by the Auto-Encoders (AEs) and the Restricted Boltzmann Machines (RBMs).

CHAPTER 4

Multi-Label and Multi-Target Classification

In machine learning, classification problems can be categorized in four different ways: binary, multi-class, multi-label and multi-target classification. In order to distinguish them, it is important to understand other concepts: single and multi-target. In supervised learning, classifiers receive a dataset, where each example has input and output values. The objective of the classifier is, through the input data, to predict which are the correct output values. We call a problem as single-target when the example needs to be assimilated with only one output value (i.e. only one target). In contrast, we call multi-target those problems that have multiple output values simultaneously predicted.

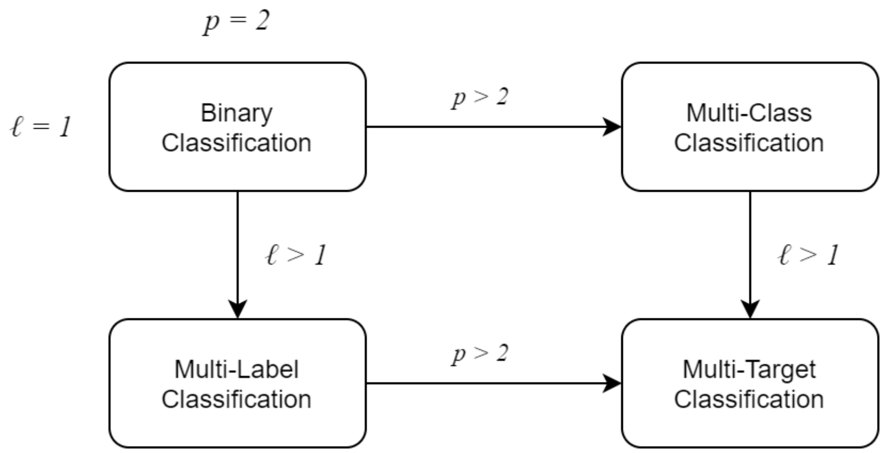

$p = 2$
$\ell = 1$
Binary Classification
$p > 2$
Multi-Class Classification
$\ell > 1$
$\ell > 1$
Multi-Label Classification
$p > 2$
Multi-Target Classification

Figure 4.1: Illustration of the different classification paradigms, where ℓ is the number of targets and p is the number of possible values that each target may take. Adapted from (Read et al., 2014).

Among the four categories of classification problems, we consider the binary and multi-class problems as single-target, and the multi-label and multi-target classification problems as multi-target.

Now that we have differentiated the classification paradigms according to the number of output variables, there is another factor that also differentiates them: the number of possible values per target. When a problem consists in predicting between only two output values, we consider it to be binary. On the other hand, a problem that has multiple possible output values is considered to be multi-class.

Thus, we can say that the problem of multi-label classification is a problem with multiple binary targets, and multi-target classification is a problem with multiple targets where each of these can have multiple values. We illustrate in Figure 4.1 how these four classification paradigms are categorized.

A multi-target classification problem can be formally described as: given a **D** dataset with n examples and their multiple nominal output values, let the matrix $\mathbf{X} \in \mathbf{D}$ consisting of m input variables be the feature space, and the matrix $\mathbf{Y} \in \mathbf{D}$ with ℓ output variables be the target space. For each example $\mathbf{d}^{(i)} = (\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$, let the input space (features) be $\mathbf{x}^{(i)}$ and the output variables (targets) be $\mathbf{y}^{(i)}$, being $1 \leq i \leq n$. The objective is to learn a model $h(\mathbf{X}) \rightarrow \mathbf{Y}$ that assimilates the vector $\mathbf{x}^{(i)}$ to the vector $\mathbf{y}^{(i)}$ which contains ℓ instances, where each one can assume p possible values. This notation will be adopted in the next sections of this chapter, alongside the definitions summarized in the Table 4.1.

Definition	Notation
Number of examples	n
Number of features	m
Feature space	$\mathbf{X} = \{\mathbf{X}_1, ..., \mathbf{X}_j, ..., \mathbf{X}_m\}, 1 \leq j \leq m$
Feature instance of the *i*th example	$\mathbf{x}^{(i)} = (x_1^{(i)}, ..., x_m^{(i)}) \in \mathbf{X}, 1 \leq i \leq n$
Number of targets	ℓ
Number of possible values per target	p
Target space	$\mathbf{Y} = \{\mathbf{Y}_1, ..., \mathbf{Y}_k, ..., \mathbf{Y}_\ell\}, 1 \leq k \leq \ell$
Target instance of the *i*th example	$\mathbf{y}^{(i)} = (y_1^{(i)}, ..., y_\ell^{(i)}) \in \mathbf{Y}, 1 \leq i \leq n$
Full Multi-Target (MT) dataset	$\mathbf{D} = \{(x_1^{(1)}, y_1^{(1)}), ..., (x_m^{(n)}, y_\ell^{(n)})\}$
Single-Target (ST) dataset with the *k*th target	$\mathbf{D}_k = \{(x_1^{(1)}, y_k^{(1)}), ..., (x_m^{(n)}, y_k^{(n)})\}$
MT instance of the *i*th example	$\mathbf{d}^{(i)} = (\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \in \mathbf{D}$
ST instance of the *i*th example with the *k*th target	$\mathbf{d}_k^{(i)} = (\mathbf{x}^{(i)}, y^{(i)}) \in \mathbf{D}_k$
MT classification model	$h(\mathbf{X}) \rightarrow \mathbf{Y}$
ST classification model for the *k*th target	$h_k(\mathbf{X}) \rightarrow \mathbf{Y}_k$
MT prediction	$\mathbf{Z} = h(\mathbf{X}) \in \mathbf{Y}$
ST prediction for the *k*th target	$\mathbf{Z}_k = h_k(\mathbf{X}) \in \mathbf{Y}_k$

Table 4.1: Notation used to formally describe the multi-target classification methods.

In the next sections, we review the multi-target classification literature and its proximity to the multi-target classification research field (Section 4.1), as well as the selected multi-target methods to be used in this research (Sections 4.2, 4.3, 4.4 and 4.5).

4.1 Drawing parallels between Multi-Label and Multi-Target Classification

In machine learning, the task of classifying an example consists of, through a vector of input attributes, which we call features, assimilate the example to an output variable, which we call target. Consequently, we call the traditional classification as Single-Target (ST), because there is only one output variable, i.e., only one target. However, there are classification problems that can be assimilated to several output variables simultaneously. These problems, therefore, we call as Multi-Target (MT). In Figure 4.1 we represented how the existing classification paradigms are categorized.

The research fields of multi-label and multi-target are very close. The term multi-target emerged as a generalization of the term multi-label (Spyromitros-Xioufis et al., 2012, Tsoumakas et al., 2014a). What differentiates these two is: a) multi-label is the term used when the targets of the problem are binary, that is, each target can only be assimilated to two possible values (1 or 0); b) multi-target is the term used when targets can be assimilated to multiple values.

Thus, we can say that the term multi-label should be used in more specific cases, where the multiple output variables can only assume binary values, and the multi-target should be used in more general problems, when we want to explain that the problem has several output variables, and each output variable can assume multiple values.

There is still no standard established in the multi-target literature, and this term is also known as multi-output (Zhang et al., 2012), multi-dimensional (Jia and Zhang, 2020b, Read et al., 2014) or multivariate (Borchani et al., 2015).

The research field of multi-target learning, because it is a multi-label generalization, benefits greatly from multi-label classification researches, where methods can be adapted almost straightforward to multi-target (Tsoumakas et al., 2014a, Borchani et al., 2015), and when not, are often used as a baseline for the development of novel methods (Read et al., 2014).

The multi-label and multi-target classification methods are also categorized in the same way: problem transformation or algorithm adaptation. In Sections 4.1.1 and 4.1.2 we will present both approaches, and also talk about methods that have not yet been suited to the multi-target context.

4.1.1 Problem Transformation Methods

Handling multiple target variables at once can be a very complex and computational expensive task. An alternative to deal with these complex problems is to turn them into smaller, less complex ones. A problem transformation method, as the name implies, consists of transforming multi-target problems into several single-target problems.

In addition to lowering the computational cost to deal with multi-target problems, the idea behind problem transformation methods is also related to the fact that these methods transform multi-target problems in a way that allows predictive models to be built using traditional single-target classifiers.

Among the problem transformation methods, there are two methods that are very popular in multi-label literature, and are used in multi-target problems (Spyromitros-Xioufis et al., 2012). The first is Class Relevance (CR) (more details in Section 4.2), which is the simplest and most intuitive method of all, which consists of creating separate predictive models for each target. The other well-known method is the Classifier Chains (CC) (more details in Section 4.3), which already takes into account that there may be relationships among the targets, and builds its predictive model by chaining the predictions of each target in the feature space. Both methods were used in this research to evaluate our results.

There is also another well-known method in the multi-target literature called Label Powerset, which was converted by Read et al. (2014) to multi-target under the name of Class Powerset (CP). This method consists of creating a predictive model for all possible target combinations, i.e., it considers that there are relations among all targets unconditionally. However, this ends up generating high computational complexity (Cherman et al., 2011). The study published in (Read et al., 2014) shows that the results of the CP in the multi-target context are compared to the CC, which ends up not justifying its high computational cost, and for this reason the use of CP in multi-target is discouraged. Therefore, among the most well-known baseline methods in the multi-label literature, according to our knowledge, this is the only one that was not used in our experiments.

Moreover, there exists another approach for multi-target classification that aims to deal directly with the multi-target problem, without breaking it down into smaller single-target problems. In Section 4.1.2, we will talk about these methods, which are categorized as algorithm adaptation.

4.1.2 Algorithm Adaptation Methods

One of the problems with the problem transformation approach is that several separate classification models are built to solve a multi-target classification problem, which can result in some relationship among the targets not being taken into account, which can lead to poor predictive performance. Algorithm adaptation methods rely on the concept of simultaneously predicting all the targets, considering the relationships among them, and being able to capture all dependencies and relationships using a single model. Hence, this technique presents several advantages over the problem transformation methods, since it usually ensures a better predictive performance.

Multi-target classification methods based on the algorithm adaptation approach, as the name says, consist of applying traditional classification algorithms (such decision trees, support vector machines, k-nearest neighbors) specifically for the multi-target domain, so that the prediction is made through a single model.

At the time we had conducted our experiments, to our knowledge, there were no algorithm adaptation methods for multi-target classification and, therefore, we do not have any of these methods present in our experiments. The closest to an algorithm adaptation of the methods presented in this research is the Super-Class Classifier (SCC) (more details in Section 4.5), which uses an algorithm based on Bayesian networks to map the dependency among the targets, however it is still a problem transformation method.

Recently, an algorithm adaptation method for multi-target classification has been proposed (Jia and Zhang, 2020c), which consists of using the already known k-Nearest Neighbors to solve the multi-target problem using only a predictive model. However, as this is a very recent research, unfortunately we were unable to consider it in our experiments in time.

It is important to mention that, in addition to the multi-target classification, there is also the multi-target regression research field, which deals with problems that have multiple output variables that assume continuous values (instead of binary or nominal). In multi-target regression, there are several implementations of algorithm adaptation methods that showed a predictive performance superior to the problem transformation ones (Spyromitros-Xioufis et al., 2016, Tsoumakas et al., 2014b, Jia and Zhang, 2020c).

One of the major problems with algorithm adaptation methods, as mentioned in (Read et al., 2014), is the fact that applicability depends highly on the type of data that the algorithm or technique can handle. The authors mention that a great example of this is that there are several methods and evaluation measures that have not yet been adapted from multi-label classification, which is a problem that also deals with nominal targets (but can only assume two possible values), for the context of multi-target classification.

While we are not aware of the existence of any other algorithm adaptation methods for the multi-target classification context, our experiments were conducted with the problem transformation methods presented in Sections 4.2, 4.3, 4.4 and 4.5.

4.2 Class Relevance

When dealing with a multi-target classification problem, one of the most straightforward ways to handle it is to create a separate model for each output variable and make separate predictions. According to (Last et al., 2011), Class Relevance (CR) defines exactly this approach, which consists of dividing the multi-target problem into several single-target problems.

This method tackles a multi-target problem by dividing the multi-target dataset $\mathbf{D} = (\mathbf{X}, \mathbf{Y})$, where $\mathbf{X}$ and $\mathbf{Y}$ are the feature and target space respectively, into several single-target datasets $\mathbf{D}_k = (\mathbf{X}, \mathbf{Y}_k)$, being $1 \leq k \leq \ell$ and ℓ the number of targets present in this dataset, where each $\mathbf{D}_k$ represents the kth target assimilated to the feature space $\mathbf{X}$. For each of these $\mathbf{D}_k$ single-target datasets, this method creates ℓ separate classification models $h_k : \mathbf{X} \rightarrow \mathbf{Y}_k$ using a traditional multi-class classifier, as shown in Equation 4.1:

$$\begin{aligned} h_1 &: \mathbf{X}_1, ..., \mathbf{X}_m \rightarrow \mathbf{Y}_1 \\ h_2 &: \mathbf{X}_1, ..., \mathbf{X}_m \rightarrow \mathbf{Y}_2 \\ &\vdots \\ h_\ell &: \mathbf{X}_1, ..., \mathbf{X}_m \rightarrow \mathbf{Y}_\ell \end{aligned} \tag{4.1}$$

and then, we can define the predictive model as the combination of all the single-target models generated separately for each target, as shown in Equation 4.2:

$$\mathbf{h} = (h_1, ..., h_\ell) \tag{4.2}$$

so, the prediction values $\mathbf{Z}$ are obtained by applying $\mathbf{h}$ on the testing data, as we show on Equation 4.3:

$$\mathbf{Z} = \mathbf{h}(\mathbf{X}) \tag{4.3}$$

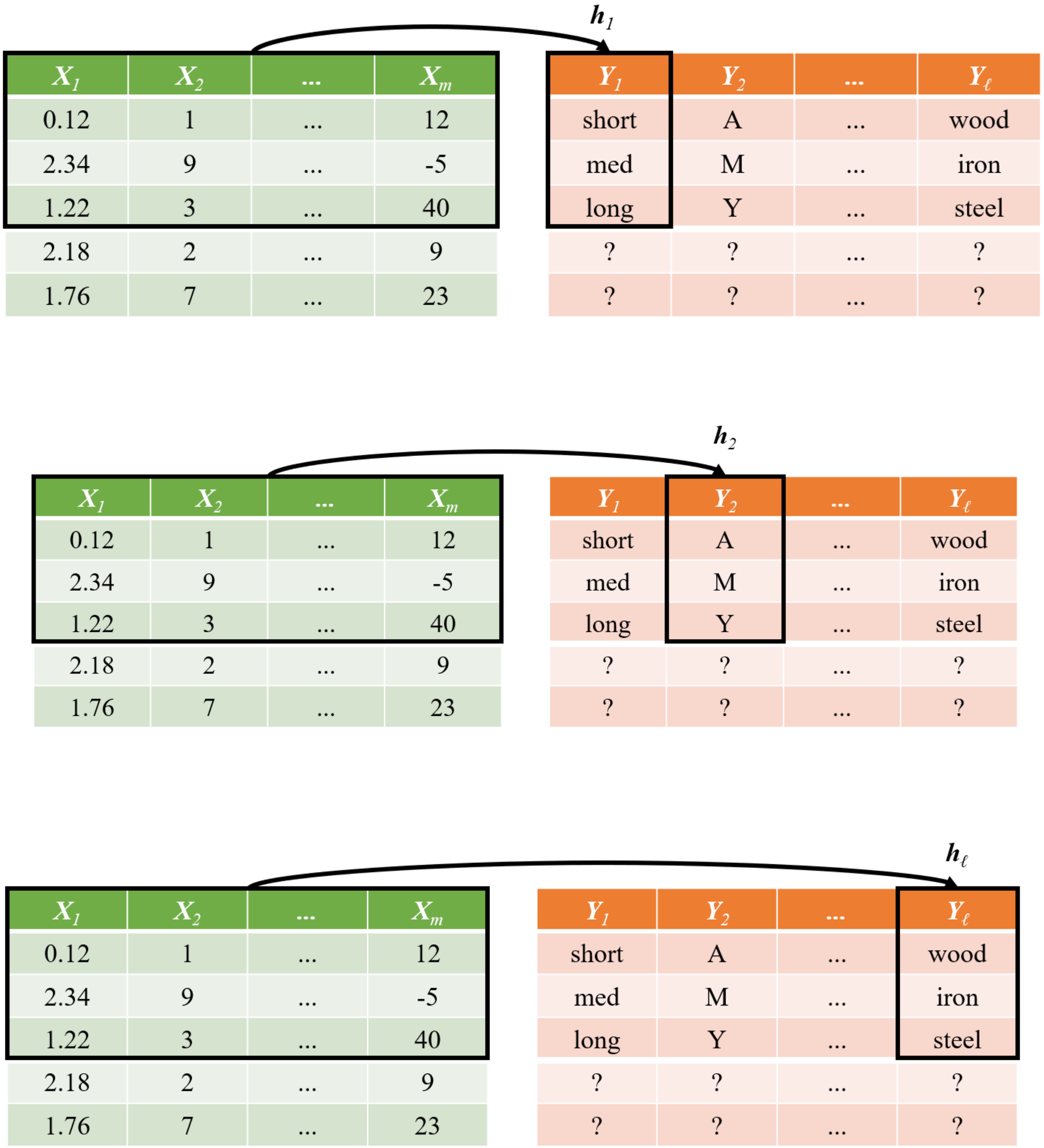
As in class relevance each target has a separate model, we can describe the predictions $\mathbf{Z}_k$ independently for each target as shown in Equation 4.4:

$$\begin{aligned} \mathbf{Z}_1 &= \mathbf{h}_1(\mathbf{X}) \Rightarrow h_1 : \mathbf{X} \rightarrow \mathbf{Y}_1 \\ \mathbf{Z}_2 &= \mathbf{h}_2(\mathbf{X}) \Rightarrow h_2 : \mathbf{X} \rightarrow \mathbf{Y}_2 \\ &\vdots \\ \mathbf{Z}_\ell &= \mathbf{h}_\ell(\mathbf{X}) \Rightarrow h_\ell : \mathbf{X} \rightarrow \mathbf{Y}_\ell \end{aligned} \tag{4.4}$$

As we discussed in Section 4.1, class relevance is a problem-transformation method, where learning the classification model consists of dividing the multi-target problem into several single-target problems, as illustrated in Figure 4.2. We also illustrate in this figure how is the learning process in a multi-target problem using this method, where it is possible to see that the relationship among the targets are completely ignored.

Class relevance is a method inspired by the Binary Relevance (BR), from multi-label literature. The big difference is that class relevance is designed to handle multiple values per target, while binary relevance handles only outputs with two possible values. These methods are also known in literature as Independent Classifiers (IC) (Read et al., 2014), given the characteristic of the learning in this method. It is considered as the simplest strategy to tackle multi-target problems. However, it is more effective than it may seem at first sight (Luaces et al., 2012), since having a multi-target problem does not always mean that there is a necessary relationship among the targets.

This relationship between CR and BR emphasizes the proximity between the multi-label and multi-target classification research fields. We will also see this in Section 4.3, where we present the Classifier Chains (CC), a method also inspired by the multi-label literature.



X_1	X_2	...	X_m
0.12	1	...	12
2.34	9	...	-5
1.22	3	...	40
2.18	2	...	9
1.76	7	...	23

Y_1	Y_2	...	Y_ℓ
short	A	...	wood
med	M	...	iron
long	Y	...	steel
?	?	...	?
?	?	...	?

X_1	X_2	...	X_m
0.12	1	...	12
2.34	9	...	-5
1.22	3	...	40
2.18	2	...	9
1.76	7	...	23

Y_1	Y_2	...	Y_ℓ
short	A	...	wood
med	M	...	iron
long	Y	...	steel
?	?	...	?
?	?	...	?

X_1	X_2	...	X_m
0.12	1	...	12
2.34	9	...	-5
1.22	3	...	40
2.18	2	...	9
1.76	7	...	23

Y_1	Y_2	...	Y_ℓ
short	A	...	wood
med	M	...	iron
long	Y	...	steel
?	?	...	?
?	?	...	?

Figure 4.2: Illustration of the predictive model built in the Class Relevance (CR) method. Adapted from (Tsoumakas et al., 2014a).

4.3 Classifier Chains

One of the fundamental characteristics of Class Relevance (CR) is that its predictive model disregards any possible relationship among targets, as we discussed in Section 4.2. In contrast, Classifier Chains (CC) is a method designed to explore the relationships among the targets, and for that reason, it is a slightly more sophisticated method than CR.

The CC tackles the multi-target classification problem by expanding the feature-space. As the name says, this is done in a chained way, where the previous predictions become part of the feature-space throughout the learning. The strategy of CC is to divide the multi-target dataset $\mathbf{D} = (\mathbf{X}, \mathbf{Y})$, where $\mathbf{X}$ and $\mathbf{Y}$ are the feature and target space respectively, into several single-target $\mathbf{D}_k$ datasets, where $1 \leq k \leq \ell$ and ℓ is the number of targets present in this dataset, similarly as occurs in CR.

However, a major difference when composing these single-target datasets is that, for each $\mathbf{D}_k$ where $k > 1$, the prediction of the last target is concatenated on the feature space $\mathbf{X}$ subsequently. When chaining the predictions into feature-space for each iteration of k, this method is able to take into account the relationships among the targets even using separate classification models for each target. Therefore, ℓ separate classification models $h_k : \mathbf{X} \rightarrow \mathbf{Y}_k$ are learned using traditional multi-class classifiers, as shown in Equation 4.5:

$$\begin{aligned}
&h_1 \colon \mathbf{X}_1, ..., \mathbf{X}_m \rightarrow \mathbf{Y}_1 \\
&h_2 \colon \mathbf{X}_1, ..., \mathbf{X}_m, h_1(\mathbf{X}) \rightarrow \mathbf{Y}_2 \\
&h_3 \colon \mathbf{X}_1, ..., \mathbf{X}_m, h_1(\mathbf{X}), h_2(\mathbf{X}) \rightarrow \mathbf{Y}_3 \\
&\ \vdots \\
&h_\ell \colon \mathbf{X}_1, ..., \mathbf{X}_m, h_1(\mathbf{X}), ..., h_{\ell-1}(\mathbf{X}) \rightarrow \mathbf{Y}_\ell
\end{aligned} \tag{4.5}$$

In this way, we can define the predictive model $\mathbf{h}$ combining the models generated for each target, and then, obtain the predictions $\mathbf{Z}$ by applying $\mathbf{h}$ on the data, similarly as in CR, as shown in Equations 4.6 and 4.7, respectively:

$$\mathbf{h} \colon (h_1, ..., h_\ell) \tag{4.6}$$

$$\mathbf{Z} = \mathbf{h}(\mathbf{X}) \tag{4.7}$$

Also, we can describe each target prediction $\mathbf{Z}_k$ as shown in Equation 4.8:

$$\begin{aligned}
&\mathbf{Z}_1 = \mathbf{h}_1(\mathbf{X}) \Rightarrow h_1 : \mathbf{X}_1, ..., \mathbf{X}_m \rightarrow \mathbf{Y}_1 \\
&\mathbf{Z}_2 = \mathbf{h}_2(\mathbf{X}) \Rightarrow h_2 : \mathbf{X}_1, ..., \mathbf{X}_m, h_1(\mathbf{X}) \rightarrow \mathbf{Y}_2 \\
&\mathbf{Z}_3 = \mathbf{h}_3(\mathbf{X}) \Rightarrow h_3 : \mathbf{X}_1, ..., \mathbf{X}_m, h_1(\mathbf{X}), h_2(\mathbf{X}) \rightarrow \mathbf{Y}_3 \\
&\ \vdots \\
&\mathbf{Z}_\ell = \mathbf{h}_\ell(\mathbf{X}) \Rightarrow h_\ell : \mathbf{X}_1, ..., \mathbf{X}_m, h_1(\mathbf{X}), ..., h_{\ell-1}(\mathbf{X}) \rightarrow \mathbf{Y}_\ell
\end{aligned} \tag{4.8}$$

The multi-target implementation of CC is inspired by the CC from multi-label literature (Sorower, 2010). As in BR, the CC in the multi-label research field is designed for binary classification, while the multi-target specification can handle multiple nominal values per target.

Also, CC is another problem transformation method, as we described in Section 4.1. It means that the CC classification strategy consists of dividing the multi-target problem into several single-target problems. We also provided in Figure 4.3 a step-by-step of the learning process, where it is possible to see how CC tackles a multi-target problem by learning several single-target models. In this figure, it is also possible to see more clearly how CC is able to take into account the relationships among the targets in a chained way.

However, not all of the possible relationships are taken into consideration in the CC prediction. In Equation 4.5 and in Figure 4.3 it is possible to observe that, when learning a target $\mathbf{Y}_k$ with $k < \ell$, the correlation with the next targets are not explored. Thus, the only target that CC explores all the possible correlations with the others is the last one, $\mathbf{Y}_\ell$.

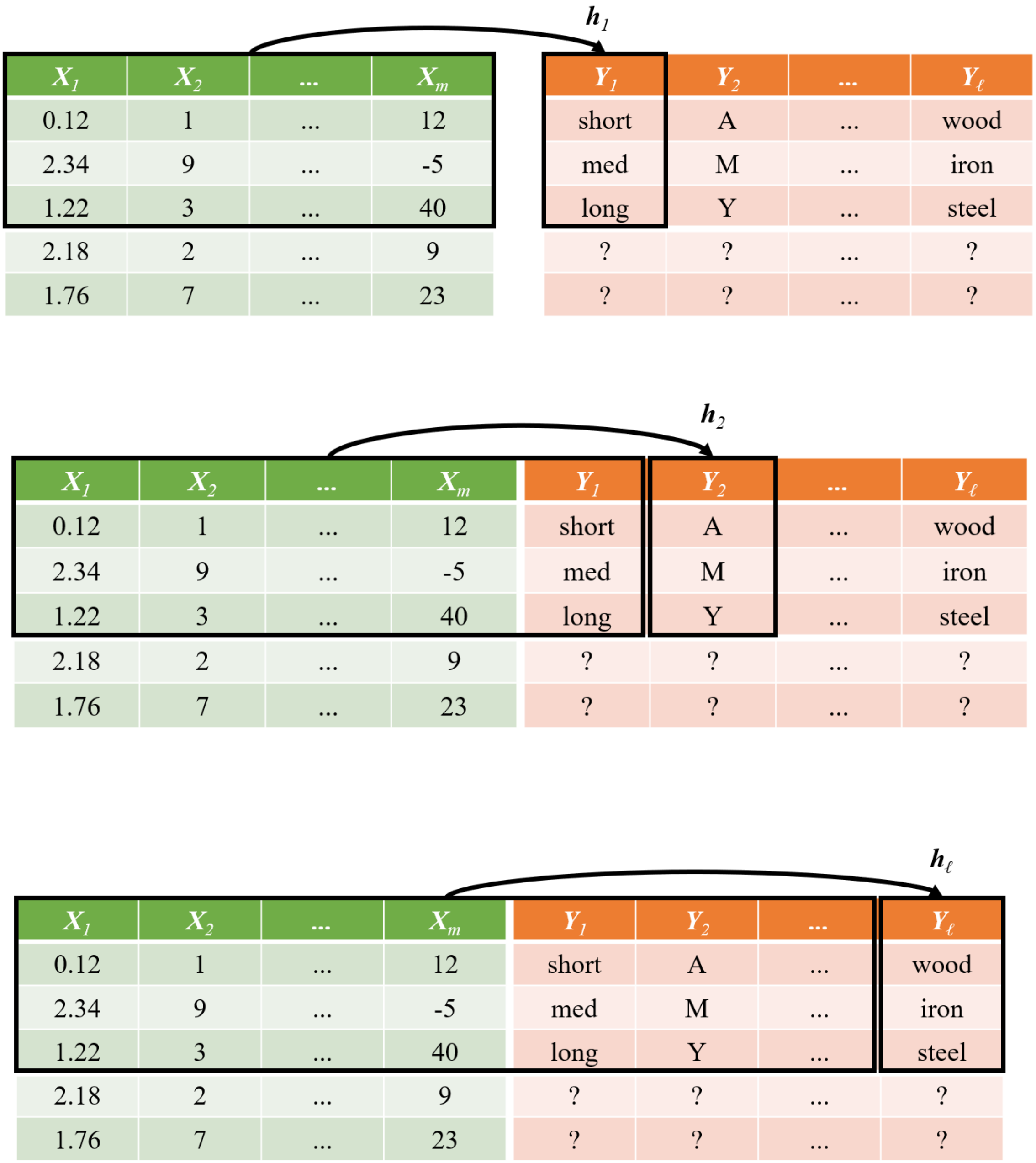
4.3.1 Ensemble of Classifier Chains

One of the problems that CC has is that not all targets are taken into account when creating the predictive model. The chained way in which this method builds the classifier makes only the last target to take into consideration the information present in all the others. Ensemble of Classifier Chains (ECC) consists of a variant of CC presented by Read et al. (2011), where several CCs are generated with random chaining orders and random subsets of the original training data. The idea behind this is to predict each classifier separately, and then count which target values are most predicted using a voting method. The target values that have the number of votes greater than a defined threshold will be part of the final classification model.

As the name says, the ECC creates an emsemble with d trained CC classifiers $C_1, ..., C_d$, where each C_d classifier is trained with chained targets at random. The predictions are counted for each target, so that the target values that get the most votes will compose the final prediction model. Each prediction model C_d has its predicted values stored in a voting matrix $\mathbf{W} = (\mathbf{w}_1, ..., \mathbf{w}_k)$, where $\mathbf{w}$ represents a vector containing ℓ positions, where each $\mathbf{w}_j$ represents the classes that each target $\mathbf{Y}_k$ can have. In this way, each target value that has the number of votes in $\mathbf{W}_k$ higher than a threshold t across all the trained models, is selected to be part of the final model.

The results presented in other studies (Read et al., 2014, Jia and Zhang, 2020a) show that ECC, despite being a relatively simple proposal based on CC, proved to be competitive compared to other multi-target classification methods.

In Section 4.4, we present the Class Relevance Stacking (CRS), a problem transformation method based on CR that also tries to take into account the relationship among all targets into the predictive model.



X_1	X_2	...	X_m	Y_1	Y_2	...	Y_ℓ
0.12	1	...	12	short	A	...	wood
2.34	9	...	-5	med	M	...	iron
1.22	3	...	40	long	Y	...	steel
2.18	2	...	9	?	?	...	?
1.76	7	...	23	?	?	...	?

X_1	X_2	...	X_m	Y_1	Y_2	...	Y_ℓ
0.12	1	...	12	short	A	...	wood
2.34	9	...	-5	med	M	...	iron
1.22	3	...	40	long	Y	...	steel
2.18	2	...	9	?	?	...	?
1.76	7	...	23	?	?	...	?

X_1	X_2	...	X_m	Y_1	Y_2	...	Y_ℓ
0.12	1	...	12	short	A	...	wood
2.34	9	...	-5	med	M	...	iron
1.22	3	...	40	long	Y	...	steel
2.18	2	...	9	?	?	...	?
1.76	7	...	23	?	?	...	?

Figure 4.3: Illustration of how the predictive model is built in the Classifier Chains (CC) method. Adapted from (Tsoumakas et al., 2014a).

4.4 Class Relevance Stacking

As discussed in Section 4.2, the CR method does not take into account any possible relationship among the targets of a multi-target problem. The CC, in the other hand, takes into account the possible relationships among targets, however, as shown in Section 4.3, it does not take into account all possible correlations among the targets. In an attempt to solve this problem, another approach besides the ECC can be used. According to this study (Tsoumakas et al., 2009b) from the multi-label literature, it is possible to use the stacking technique to take into account all targets in the construction of the classification model. Here, we describe how the Class Relevance Stacking (CRS) method can be applied in a multi-target classification problem.

We saw in the CC that for each target a classification model is created based on the previous target, i.e., there is a chaining that modifies the feature-space, so that only the last target has learned all the information among the targets. CRS also modifies the feature-space, but in a different way: it first composes an intermediate model that has information from all targets, and uses this to build a final model, which has all the target information. Thus, its predictions take into account the information of all targets.

We can define how Class Relevance Stacking (CRS) addresses a multi-target problem as follows: given a multi-target dataset $\mathbf{D} = (\mathbf{X}, \mathbf{Y})$, where $\mathbf{X}$ and $\mathbf{Y}$ are the feature and target space respectively, we can divide $\mathbf{D}$ into several single-target datasets $\mathbf{D}_k = (\mathbf{X}, \mathbf{Y}_k)$, being $1 \leq k \leq \ell$ and ℓ the number of targets present in this dataset, where each $\mathbf{Y}_k$ represents the kth target assimilated to the feature space $\mathbf{X}$. First, each of these $\mathbf{D}_k$ single-target datasets will produce ℓ separate classification models $h_k : \mathbf{X} \rightarrow \mathbf{Y}_k$ using a traditional multi-class classifier, exactly as occurs in CR, as we show in Equation 4.9:

$$\begin{aligned} h_1 &: \mathbf{X}_1, ..., \mathbf{X}_m \rightarrow \mathbf{Y}_1 \\ h_2 &: \mathbf{X}_1, ..., \mathbf{X}_m \rightarrow \mathbf{Y}_2 \\ &\vdots \\ h_\ell &: \mathbf{X}_1, ..., \mathbf{X}_m \rightarrow \mathbf{Y}_\ell \end{aligned} \tag{4.9}$$

The model is built as Equation 4.10:

$$\mathbf{h} : (h_1, ..., h_\ell) \tag{4.10}$$

but, unlikely as occurs in CR, the classification is not performed based on this model $\mathbf{h}$. Indeed, CRS does the prediction, but it uses this output to generate an intermediate feature-space $\mathbf{Y}'$, as shown on Equation 4.11:

$$\mathbf{Y}' = \mathbf{h}(\mathbf{X}) \tag{4.11}$$

and then, it builds the final prediction models using CR with $\mathbf{X}$ and $\mathbf{Y}'$ as the feature-space:

$$\begin{aligned}
&h'_1 : \mathbf{X}_1, ..., \mathbf{X}_m, \mathbf{Y}'_1, ..., \mathbf{Y}'_\ell \rightarrow \mathbf{Y}_1 \\
&h'_2 : \mathbf{X}_1, ..., \mathbf{X}_m, \mathbf{Y}'_1, ..., \mathbf{Y}'_\ell \rightarrow \mathbf{Y}_2 \\
&\quad\vdots \\
&h'_\ell : \mathbf{X}_1, ..., \mathbf{X}_m, \mathbf{Y}'{}_1, ..., \mathbf{Y}'_\ell \rightarrow \mathbf{Y}_\ell
\end{aligned} \tag{4.12}$$

thus composing the model $\mathbf{h}'$ (Equation 4.13) which will be used to obtain the final predictions $\mathbf{Z}$ (Equation 4.14):

$$\mathbf{h}' : (h'_1, ..., h'_\ell) \tag{4.13}$$

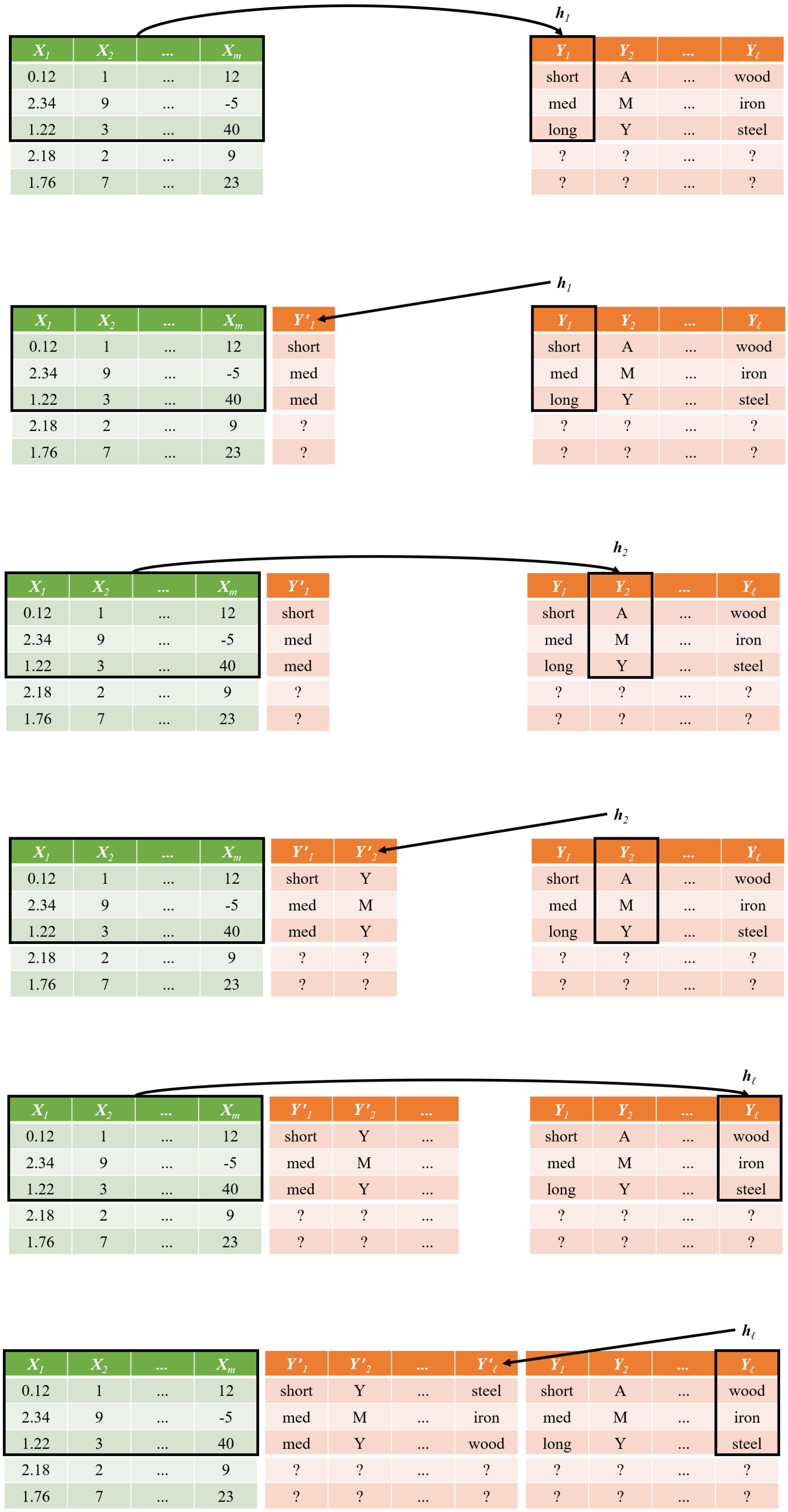
$$\mathbf{Z} = \mathbf{h}'(\mathbf{X}) \tag{4.14}$$

We can also describe each target prediction $\mathbf{Z}_k$ as:

$$\begin{aligned}
&\mathbf{Z}_1 = \mathbf{h}'_1(\mathbf{X}) \Rightarrow h'_1 : \mathbf{X}_1, ..., \mathbf{X}_m, \mathbf{Y}'_1, ..., \mathbf{Y}'_\ell \rightarrow \mathbf{Y}_1 \\
&\mathbf{Z}_2 = \mathbf{h}'_2(\mathbf{X}) \Rightarrow h'_2 : \mathbf{X}_1, ..., \mathbf{X}_m, \mathbf{Y}'_1, ..., \mathbf{Y}'_\ell \rightarrow \mathbf{Y}_2 \\
&\ldots \\
&\mathbf{Z}_\ell = \mathbf{h}'_\ell(\mathbf{X}) \Rightarrow h'_\ell : \mathbf{X}_1, ..., \mathbf{X}_m, \mathbf{Y}'_1, ..., \mathbf{Y}'_\ell \rightarrow \mathbf{Y}_\ell
\end{aligned} \tag{4.15}$$

This method is based on the Binary Relevance (BR) stacking proposal, presented in (Tsoumakas et al., 2009b), and was straightforward converted to the multi-target classification. Its application is relatively simple, and consists of using the BR twice, once to obtain the intermediate prediction model (Figure 4.4), and them to make the final predictions (Figure 4.5). The idea behind this is that this method can extract information from all targets to make the final prediction. The experiments carried out in (Tsoumakas et al., 2009b) show that the method was able to obtain better results than the BR, which is probably due to the fact that there were relationships among the targets of the data used for evaluation.

However, CRS arbitrarily assumes that there are relationships among all targets, which may not be true. In Section 4.5, a method will be presented that assesses the possible relationships among the targets before making the prediction.



X_1	X_2	...	X_m
0.12	1	...	12
2.34	9	...	-5
1.22	3	...	40
2.18	2	...	9
1.76	7	...	23

Y_1	Y_2	...	Y_ℓ
short	A	...	wood
med	M	...	iron
long	Y	...	steel
?	?	...	?
?	?	...	?

X_1	X_2	...	X_m
0.12	1	...	12
2.34	9	...	-5
1.22	3	...	40
2.18	2	...	9
1.76	7	...	23

Y'_1
short
med
med
?
?

Y_1	Y_2	...	Y_ℓ
short	A	...	wood
med	M	...	iron
long	Y	...	steel
?	?	...	?
?	?	...	?

X_1	X_2	...	X_m
0.12	1	...	12
2.34	9	...	-5
1.22	3	...	40
2.18	2	...	9
1.76	7	...	23

Y'_1
short
med
med
?
?

Y_1	Y_2	...	Y_ℓ
short	A	...	wood
med	M	...	iron
long	Y	...	steel
?	?	...	?
?	?	...	?

X_1	X_2	...	X_m
0.12	1	...	12
2.34	9	...	-5
1.22	3	...	40
2.18	2	...	9
1.76	7	...	23

Y'_1	Y'_2
short	Y
med	M
med	Y
?	?
?	?

Y_1	Y_2	...	Y_ℓ
short	A	...	wood
med	M	...	iron
long	Y	...	steel
?	?	...	?
?	?	...	?

X_1	X_2	...	X_m
0.12	1	...	12
2.34	9	...	-5
1.22	3	...	40
2.18	2	...	9
1.76	7	...	23

Y'_1	Y'_2	...
short	Y	...
med	M	...
med	Y	...
?	?	...
?	?	...

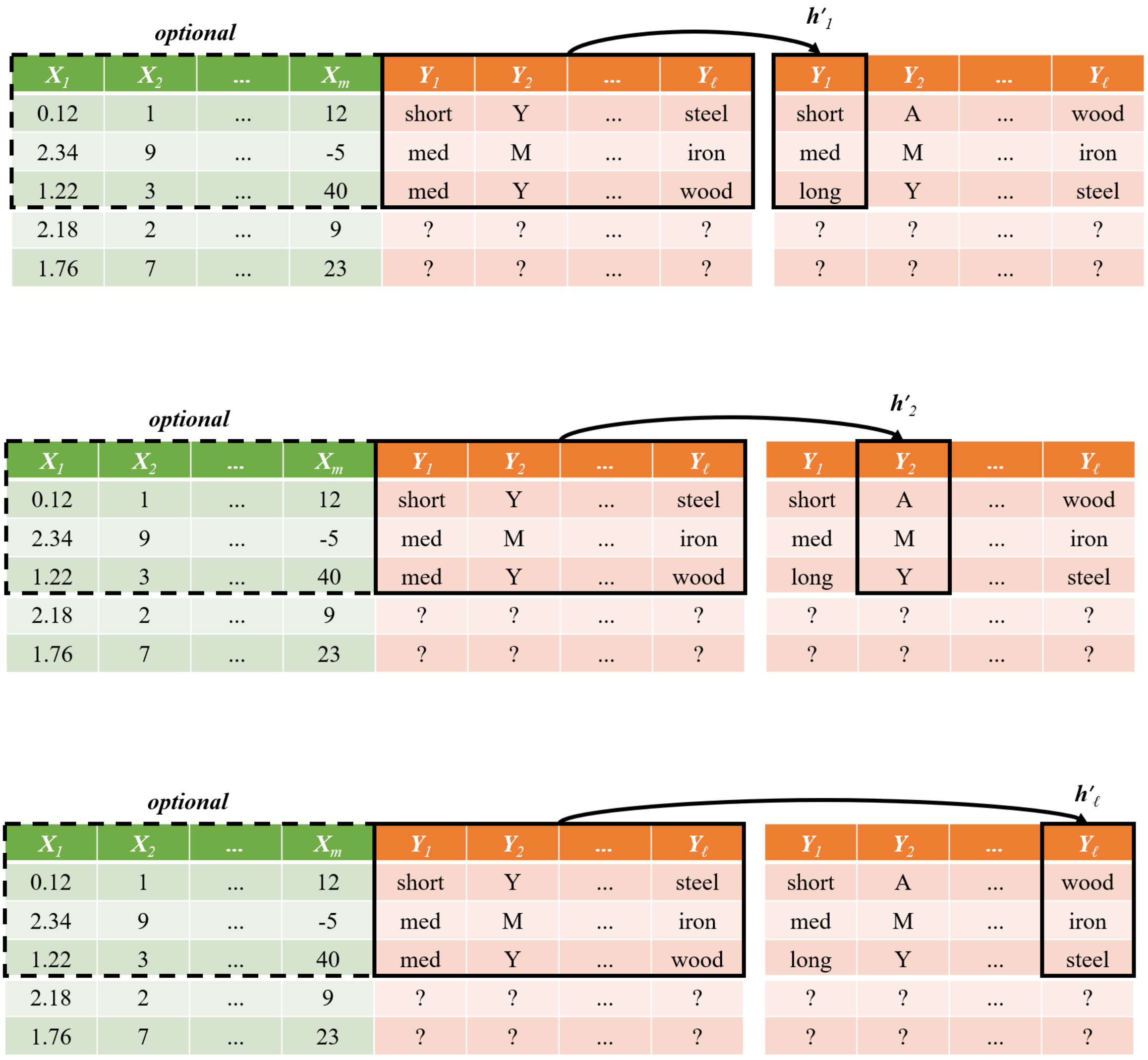
Y_1	Y_2	...	Y_ℓ
short	A	...	wood
med	M	...	iron
long	Y	...	steel
?	?	...	?
?	?	...	?

X_1	X_2	...	X_m
0.12	1	...	12
2.34	9	...	-5
1.22	3	...	40
2.18	2	...	9
1.76	7	...	23

Y'_1	Y'_2	...	Y'_ℓ
short	Y	...	steel
med	M	...	iron
med	Y	...	wood
?	?	...	?
?	?	...	?

Y_1	Y_2	...	Y_ℓ
short	A	...	wood
med	M	...	iron
long	Y	...	steel
?	?	...	?
?	?	...	?

Figure 4.4: Illustration of the first phase of learning at Class Relevance Stacking (CRS), where Class Relevance (CR) is applied to obtain the intermediate predictions. Adapted from (Tsoumakas et al., 2014a).



X_1	X_2	...	X_m	Y_1	Y_2	...	Y_ℓ	Y_1	Y_2	...	Y_ℓ
0.12	1	...	12	short	Y	...	steel	short	A	...	wood
2.34	9	...	-5	med	M	...	iron	med	M	...	iron
1.22	3	...	40	med	Y	...	wood	long	Y	...	steel
2.18	2	...	9	?	?	...	?	?	?	...	?
1.76	7	...	23	?	?	...	?	?	?	...	?

X_1	X_2	...	X_m	Y_1	Y_2	...	Y_ℓ	Y_1	Y_2	...	Y_ℓ
0.12	1	...	12	short	Y	...	steel	short	A	...	wood
2.34	9	...	-5	med	M	...	iron	med	M	...	iron
1.22	3	...	40	med	Y	...	wood	long	Y	...	steel
2.18	2	...	9	?	?	...	?	?	?	...	?
1.76	7	...	23	?	?	...	?	?	?	...	?

X_1	X_2	...	X_m	Y_1	Y_2	...	Y_ℓ	Y_1	Y_2	...	Y_ℓ
0.12	1	...	12	short	Y	...	steel	short	A	...	wood
2.34	9	...	-5	med	M	...	iron	med	M	...	iron
1.22	3	...	40	med	Y	...	wood	long	Y	...	steel
2.18	2	...	9	?	?	...	?	?	?	...	?
1.76	7	...	23	?	?	...	?	?	?	...	?

Figure 4.5: Illustration of how the Class Relevance Stacking (CRS) uses the intermediate predictions to build the final prediction model. Adapted from (Tsoumakas et al., 2014a).

4.5 Super-Class Classifier

Regarding the predictive model created through the methods presented in Sections 4.2, 4.3 and 4.4, there are several peculiarities regarding the relationships among the targets. Briefly, we can summarize:

- Class Relevance (CR) is the simplest approach of all. However, it completely disregards any relationship among the targets;
- Classifier Chains (CC) already takes into account that there exists relationships among the targets, but does not consider all possibilities. The Ensemble of Classifier Chains (ECC) tries to mitigate that in an arbitrary way;
- Class Relevance Stacking (CRS) takes into account that there is relationships among all the targets, regardless of whether those targets are related or not.

Regarding the methods above, it is possible to state that they all assume in an arbitrary way the relationships that exist between the output variables. Proposed by Read et al. (2014), Super-Class Classifier (SCC) is a method for multi-target classification that aims to explore how the targets are related, defining the dependencies among them, before building the predictive model.

The authors' motivation to propose the method came when they faced a real-world multi-target classification problem, and there were no specific methods for this domain that explored how targets were related, and the existing problem adaptation algorithms from the multi-label literature that could be easily suited to multi-target, had their predictions in an arbitrary way.

Still according to the authors, they did not want to decide between CR (Independent Classifiers (IC) as they called at the time) and CP. The reasons were: the CR does not take into account any relationship among the targets, and the CP has high computational complexity, as pointed out in Section 4.1.1.

In order to address the possible relationships among the targets, the authors decided to design a method that was capable of making this evaluation before creating the predictive model. The idea behind this is to create models for each set of targets that are more related to each other.

For example, let's assume that we have the following dataset to perform a multi-target classification: $\mathbf{D} = (\mathbf{X}, \mathbf{Y})$ with n examples and $\ell = 3$ targets, where each of these targets can assume $p = 3$ possible different values, being $\mathbf{Y}_1 \in \{1, 2, 3\}$, $\mathbf{Y}_2 \in \{4, 5, 6\}$ and $\mathbf{Y}_3 \in \{7, 8, 9\}$.

We can say that CR would generate the following models:

$$\begin{aligned} h_1 &: \mathbf{X}_1, ..., \mathbf{X}_m \rightarrow \{1, 2, 3\} \\ h_2 &: \mathbf{X}_1, ..., \mathbf{X}_m \rightarrow \{4, 5, 6\} \\ h_3 &: \mathbf{X}_1, ..., \mathbf{X}_m \rightarrow \{7, 8, 9\} \end{aligned} \tag{4.16}$$

and CP would generate the following models:

$$\mathbf{h} = \mathbf{X}_1, ..., \mathbf{X}_m \rightarrow \text{DISTINCT}\{\mathbf{y}^{(1)}, ..., \mathbf{y}^{(n)}\} \tag{4.17}$$

It is possible to see in Equations 4.16 and 4.17 that the CR learns $p \times \ell = 9$ possible classes, because for each model there are 3 different possible classes. On the other hand, the CP learns $p^{\ell} = 27$ distinct classes, due to the fact that CP makes the cartesian product in the target space to obtain all possible target combinations.

Based on the distribution of data in this dataset $\mathbf{D}$, we can say that the probability of an existent dependence between targets $\mathbf{Y}_1$ and $\mathbf{Y}_2$ is high (let's say it is $P(\mathbf{Y}_1|\mathbf{Y}_2) \approx 1$ and $P(\mathbf{Y}_2|\mathbf{Y}_1) \approx 1$), while $\mathbf{Y}_3$ is independent of $\mathbf{Y}_1$ and $\mathbf{Y}_2$ (where $P(\mathbf{Y}_3|\mathbf{Y}_1, \mathbf{Y}_2) \approx 0$). Therefore, it is possible to say that both CR and CP violate the constraints of this relationship.

According to the authors, the ideal model would be defined as:

$$\begin{aligned} \mathbf{h}_{1,2}\colon \mathbf{X}_1, ..., \mathbf{X}_m &\rightarrow \text{DISTINCT}\{\mathbf{y}_{1,2}^{(1)}, ..., \mathbf{y}_{1,2}^{(n)}\} \\ h_3\colon \mathbf{X}_1, ..., \mathbf{X}_m &\rightarrow \{7, 8, 9\} \end{aligned} \tag{4.18}$$

where $\mathbf{y}_{1,2}^{(n)} \equiv (y_1^{(n)}, y_2^{(n)})$. In this example, therefore, two predictive models were created satisfying the constraints of the relationship. Basically, this is a super-class classifier. Therefore, the composite prediction model can be generally defined as:

$$\mathbf{h}_{\theta}\colon (\mathbf{h}_{S_1}, ..., \mathbf{h}_{S_{|\theta|}}) \tag{4.19}$$

where S is the super-class space, and θ a partition of classes, defined as:

$$\theta = \{S_1, ..., S_{|\theta|}\} \tag{4.20}$$

Thus, for the example we described, the partition would be:

$$\theta = \{(\mathbf{Y}_1, \mathbf{Y}_2), \mathbf{Y}_3\} \tag{4.21}$$

To obtain the partition θ, it is necessary to map a dependency matrix, that is responsible for mapping which are the potentially related targets for the prediction. According to the authors, there exists two types of target dependencies: a) unconditional dependency, where the target relates to another regardless of what the $\mathbf{X}$ values are; b) conditional dependency, where the relationships among the targets can change according to $\mathbf{X}$.

The authors state, however, that based on the study presented in (Dembszynski et al., 2010), as much as unconditional and conditional dependencies may be related, it is impossible to guarantee that in fact they are, and on top of that, conditional dependency is more relevant when it comes to classification.

Therefore, to map the dependency matrix, SCC internally uses the CR to obtain the predictive models using the training data. To find the conditional dependencies, for each classification model obtained by the CR, the prediction is made with the test data and the error is calculated as follows:

$$\boldsymbol{\epsilon}^{(i)} = \mathbf{I}(\mathbf{y}^{(i)}, \mathbf{z}^{(i)}) \quad \Rightarrow \begin{cases} \epsilon_1^{(i)} = \mathcal{I}(y_1^{(i)}, z_1^{(i)}) \\ \vdots \\ \epsilon_\ell^{(i)} = \mathcal{I}(y_\ell^{(i)}, z_\ell^{(i)}) \end{cases} \tag{4.22}$$

where $1 \leq i \leq n$, and $\mathcal{I}$ is the indicator function (which returns if $y_k^{(i)} = z_k^{(i)}$ for the kth target, similarly as presented in Equation 6.3). Being a and b indexes for two different targets, if ϵ_a is related to ϵ_b, then we can say that there is a conditional dependency between the ath and bth targets.

After obtaining θ based on the conditional dependency among the targets, the SCC should learn the predictive models as demonstrated in Equation 4.18. If a target $\mathbf{Y}_a$ is independent of the others, then a predictive model $h(\mathbf{X}) \rightarrow \mathbf{Y}_a$ will be built. In contrast, if there are b targets in a partition that have conditional dependency, that is, the value of $\mathbf{X}$ can influence the prediction of these targets, then there will be b predictive models $\mathbf{h}(\mathbf{X}) \rightarrow \text{DISTINCT}\{\mathbf{Y}_b\}$, where $\mathbf{Y}_b$ corresponds to the b targets present in this partition.

There is also an optional step during the initial phase of SCC, which aims to improve the creation of θ. Through an user-defined parameter $\mathcal{T}$, the SCC uses an algorithm based on Bayesian networks, inspired by Zhang and Zhang (2010), which perform estimates (similarly as in Equation 4.22) looking for new conditional dependencies between targets. The SCC will execute this algorithm $\mathcal{T}$ times, and at each iteration it mutates θ by inserting the new relationships found among the targets.

Indeed, as much as this additional step may generate a better θ, it can lead to a high number of partitions created in θ. If we look carefully at the example presented in Equation 4.18, the more dependencies there are among the targets, the more SCC's predictive models approaches CP. The authors make this clear, the greater $\mathcal{T}$ is, the greater the number of dependencies among targets on θ can be. This can make the computational complexity of the SCC potentially closer to the CP.

The experiments carried out in (Read et al., 2014) confirm that the higher the $\mathcal{T}$, the closer the SCC approaches the CR with respect to the time needed to perform the training, as well as an increase in predictive performance. Also, the results of SCC proved to be competitive in relation to other problem transformation methods. However, in that study, the results used for the comparison are only when SCC was configured with $\mathcal{T} = 1000$. Then, in order to maintain the same basis of comparison, we will use this same parameter to compare the results, as explained in Section 6.3.

In Chapter 5, we will present an overview of the proposed framework for applying feature extraction to multi-target datasets, how we did the tests on the selected multi-target classifiers, and how we planned to evaluate the performance of the classifiers according with the feature extraction methods we used.

CHAPTER 5

Feature Extraction for Multi-Target Classification

As mentioned in the Introduction (Chapter 1), the main goal of this research is to extract features from high-dimensional multi-target datasets. The datasets with extracted features are then given as input to state-of-the-art multi-target classification algorithms. We want to see if the classifiers can obtain better or competitive results with extracted features when compared to the use of the same datasets with the original features.

We propose a framework which is composed by a two-step process. The first step is responsible for parse and standardize the dataset, and for the execution of the feature extraction methods Restricted Boltzmann Machine (RBM), Auto-Encoder (AE), Principal Component Analysis (PCA), and some of its deep learning variations. The second step executes the multi-target classification algorithms with all the datasets obtained by the execution of the first step, and also evaluates the prediction performances obtained by the classifiers. Figure 5.1 presents an overview of the proposed framework. In Sections 5.1 and 5.2 we present the steps of the proposed framework in details.

5.1 Parsing Data and Extracting Features

This section discusses in details the first step of the proposed framework. Initially, the proposed framework parses and standardizes the dataset. Since the feature extraction process is unsupervised, we remove the targets from the datasets in order to execute the feature extraction algorithms.

We used three feature extraction algorithms in our experiments: a) Principal Component Analysis (PCA), which is executed under the scikit-learn library (Pedregosa et al., 2011); b) Restricted Boltzmann Machines (RBMs), including Deep Belief Networks (DBNs), which is

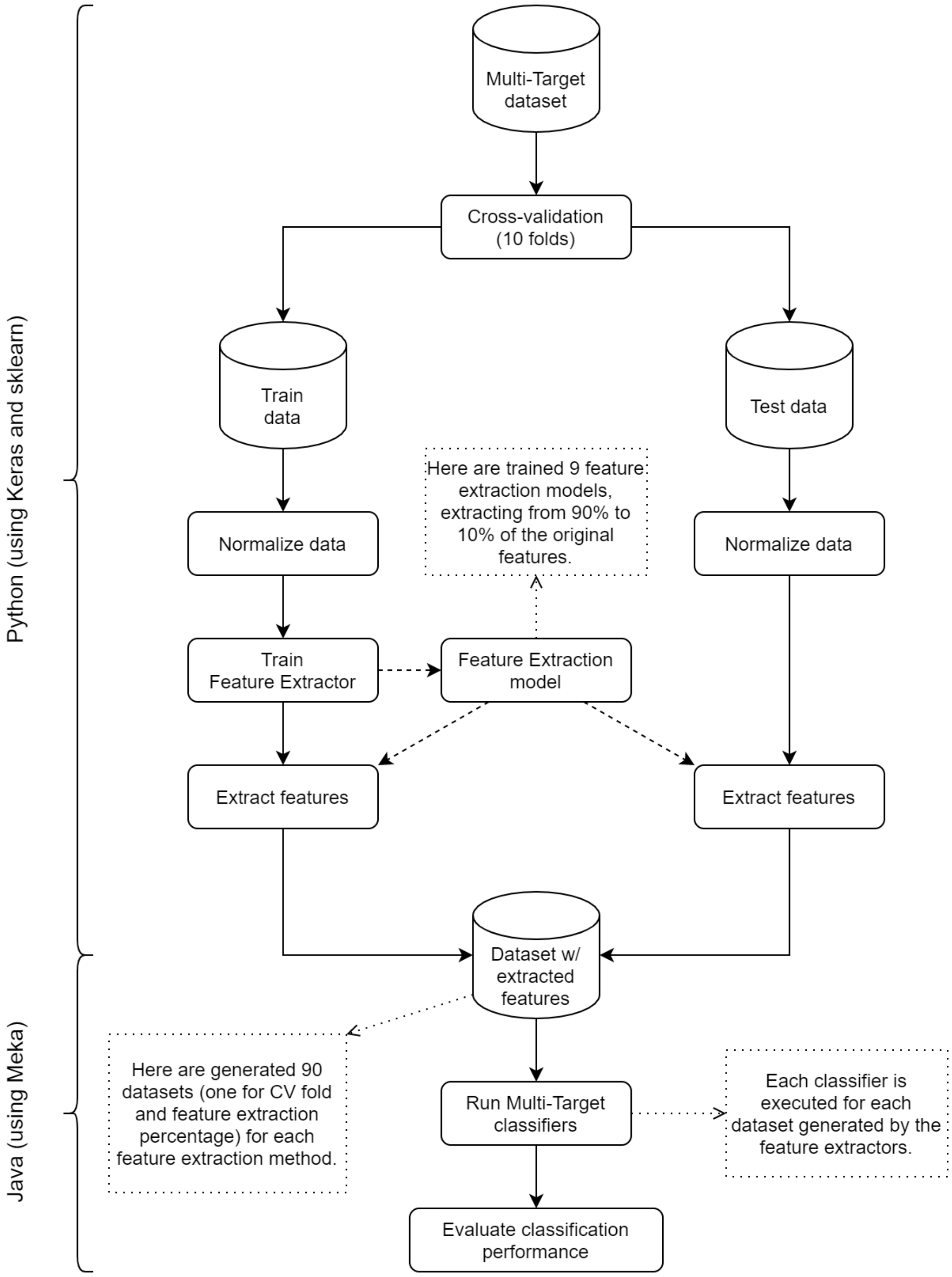

Python (using Keras and sklearn)
Java (using Meka)
Multi-Target dataset
Cross-validation (10 folds)
Train data
Test data
Here are trained 9 feature extraction models, extracting from 90% to 10% of the original features.
Normalize data
Normalize data
Train Feature Extractor
Feature Extraction model
Extract features
Extract features
Dataset w/ extracted features
Here are generated 90 datasets (one for CV fold and feature extraction percentage) for each feature extraction method.
Run Multi-Target classifiers
Each classifier is executed for each dataset generated by the feature extractors.
Evaluate classification performance

Figure 5.1: Framework overview: the framework is composed by a two-step process, where the first is responsible for the feature extraction and the second is responsible for the evaluation.

also executed under the scikit-learn library; and c) Auto-Encoders (AEs), including also its deep-learning variations, executed under the Keras library (Chollet et al., 2015).

To evaluate the performances of the classifiers after the feature extraction and, moreover, to evaluate the overall prediction decay rate after extraction, we tried different number of extracted features. For each dataset and for each feature extraction method, we extracted 90% of the features, than 80%, than 70%, until the number of generated features reaches 10% of the original number of features.

Each set of newly generated features is concatenated back with the original targets, generating new (feature extracted) datasets, which were fed to the multi-target classification methods, as discussed in the following section.

5.1.1 Tuning the neural networks hyper-parameters

Due to the fact that the datasets we selected (Table 6.1) contain nominal features, we used the one-hot encoding technique (Harris and Harris, 2010), which transforms the nominal features into a subset of binary features containing the number of the possible values for each feature. This considerably increased the number of features that were fed to the feature extractors. This technique is necessary when dealing with nominal features, since we cannot just turn them into sequential numbers, which would insert a magnitude between these nominal values (e.g. two is bigger than one, but a nominal value is neither bigger nor smaller than other nominal value).

These datasets also contain numeric features. When using neural networks, it is common to standardize numeric values in order to help in the convergence process. We standardized the numeric features individually, using the method presented by Pedregosa et al. (2011), where each value has the mean value subtracted, and then divided by the standard deviation of the feature to which it belongs. Also, as we are using the hyperbolic tangent (Nwankpa et al., 2018) activation function in our output layer, in the reconstruction process our outputs units are represented with values between -1 and 1.

There are several activation functions that could be used, but since we are dealing with reconstruction loss functions (measuring the error when reconstructing data), not all activation functions are applicable. In our preliminary tests, we used the sigmoid logistic and hyperbolic tangent functions, the ReLU (Agarap, 2018), and the softmax (Kanai et al., 2018) (combining them with different training algorithms which will be further discussed). The hyperbolic tangent presented better results when reconstructing data.

In order to train our feature extractors, we used the Adam algorithm (Kingma and Ba, 2014). In our tests, training our feature extractors with Adam increased considerably the reconstruction performance, also hugely decreasing our training time over the standard gradient descent. The Adam parameters used in our experiments are $learning\ rate = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$, and $\epsilon = 1e - 07$.

During our experiments, we also tried using AMSGrad (Reddi et al., 2019), but it decreased considerably our prediction performances, since the algorithm wasn't able to converge in datasets with a high number of features. We also tested the dropout technique (Srivastava et al., 2014) in many ways in our tests, but it did not increase our prediction performances.

Also, some datasets have missing values in both training and testing data. When a missing value was numeric, we defined it as the mean of the observed feature in the training data. When the missing value was nominal, we used the statistical mode.

In order to measure the performances of our models, we monitored the loss (Equation 3.12) using the Mean Squared Error (MSE) for every epoch, and defined $\delta = 0.001$ as a minimum change to qualify as an improvement, i.e. an absolute change of less than δ was counted as no improvement. Our neural networks were trained until the prediction performance stopped improving for 10 epochs. This allowed us to avoid over-fitting in our models, and we believe it is due the combination of the standardization and the Adam algorithm.

5.2 Executing State-of-the-Art Multi-Target Classifiers

The second step of the framework consists of executing the multi-target classifiers and evaluating their predictive performances. All the multi-target classification methods in our experiments were performed under the Meka library (Read et al., 2016), which contains specific implementations for multi-label and multi-target classification.

As we are using cross validation, for each fold a dataset was generated with the features obtained by the previous step. These datasets are separated into training and test data, so we feed the multi-target classifiers in this way, indicating which are the training and test instances. We emphasize that the training and test samples are separated from the very beginning of the process, this means that neither the feature extractor nor the classifier had access to the test instances during training, and vice versa.

With that said, for each feature extractor and for each percentage of features extracted, 10 folds were generated. We then executed the multi-target classifier with each of these folds, and stored the evaluation statistics for each fold. At the end of the execution of these 10 folds, we calculate the mean and standard deviation of all the results obtained.

In Chapter 6 we will describe how we conducted our experiment, what were the evaluation measures used, as well as the results obtained using the framework that was presented here.

Chapter 6

Experiments

In this chapter we present how we conducted all of our experiments. In Section 6.1, we presented the evaluation measures used, and in Section 6.2 we showed the selected datasets for the tests. In sequence, in Section 6.3 we evaluate the results of the multi-target classifiers in the original datasets, i.e., without extracting features, in order to have a baseline for comparison. In Section 6.4, we present the results obtained by the multi-target classifiers and compare them with the feature extraction methods we used in this research.

6.1 Evaluation Measures

To capture the notion of partially correct predictions, one strategy is to evaluate the average differences between the predicted targets and the actual targets for each test example, and then average differences over all examples in the test set. This strategy is called "example-based" evaluation. In a similar manner, one could define a "target-based" evaluation where each target is evaluated first and then averaged over all targets. It is important to note that any target-based method would fail to directly address the correlations among different targets (Last et al., 2011).

Although different evaluation measures have been proposed to evaluate multi-label classifiers, this is not true for multi-target scenarios. Many of the evaluation measures commonly used in Multi-Label Classification (MLC), such as multi-label F-measure (Sorower, 2010), have not yet been suited for Multi-Target Classification (MTC).

The next sections present the two evaluation measures used in our experiments. They are proposed for multi-label classification, but can be used to evaluate multi-target methods. To present the evaluation measures, consider the following notation. Let $\mathbf{D}$ be a multi-target dataset with n examples $(\mathbf{x}_i, \mathbf{y}_i)$, with $1 \leq i \leq n$, being $\mathbf{x}_i$ a feature vector and $\mathbf{y}_i$ the vector representing the targets of example $\mathbf{x}_i$. Let f be a classifier and $\mathbf{z}_i = f(\mathbf{x}_i)$ the vector with the prediction obtained by f for example $\mathbf{x}_i$.

6.1.1 Exact Match Ratio

As previously described, the evaluation of MTC algorithms is difficult mostly because MT predictions may be partially correct. This means that a classifier can correctly predict some of the targets while missing other targets. Thus, it is not correct to totally penalize a classification when part of the predictions is correct. However, there is an evaluation measure that does exactly this, i.e., ignores partially correct predictions, considering them as incorrect. In this way, considering a prediction as totally correct or totally wrong, we can evaluate multi-target classifiers as if we were using an accuracy used in single-label problems. This evaluation measure is called Exact Match Ratio (EMR) (Read et al., 2014), also known in literature as example accuracy (Sorower, 2010). It is shown in Equation 6.1.

$$EMR = \frac{1}{n}\sum_{i=1}^{n} \mathcal{I}(\mathbf{y}_i = \mathbf{z}_i) \tag{6.1}$$

In Equation 6.1, $\mathcal{I}$ is the indicator function, and $\mathbf{y}_i$ and $\mathbf{z}_i$ are, respectively, the correct and predicted target vectors for example $\mathbf{x}_i$. Ideally, it is expected that $EMR = 1$, which indicates a perfect model. The bigger the value of EMR, the better the performance of the classifier. Clearly, a disadvantage of this measure is that it does not distinguish between complete incorrect and partially correct predictions.

Recently, a study presented a new evaluation measure for multi-target classification, called Sub-Exact Match (Jia and Zhang, 2020c), which consists in: for each example, if there is only one target wrongly classified, the example is also computed as correctly classified. This measure was presented with the intention of not being so harsh when dealing with partially correct predictions. Unfortunately, this measure was presented in a very recent research and we were unable to insert it in our experiments.

6.1.2 Hamming Loss and Hamming Score

Hamming Loss (HL) (Sorower, 2010), also known in the literature as class accuracy (Read et al., 2014), reports how many times on average, the relevance of an example to a target is incorrectly predicted. Therefore, it takes into account the prediction error (an incorrect target is predicted) and the missing error (a relevant target not predicted), normalized over the total number of targets and the total number of examples. The evaluation measure is given by Equation 6.2, where k is the number of targets in the dataset, $\mathcal{I}$ is the indicator function, $\mathbf{y}_i$ is the correct target vector, and $\mathbf{z}_i$ is the predicted target vector.

$$HL = \frac{1}{kn}\sum_{i=1}^{n}\sum_{j=1}^{k} \mathcal{I}(y_{ij} \neq z_{ij}) \tag{6.2}$$

Ideally, it is expected that $HL = 0$, which would imply no error. The smaller the value of HL, the better the performance of the classifier.

The Hamming Score (HS) is the opposite of HL, i.e., it reports how many times on average, the relevance of an example to a target is correctly predicted. HS is commonly used in the multi-target classification literature to present results consistently together with EMR. It is presented in Equation 6.3.

$$HS = 1 - HL \tag{6.3}$$

6.2 Datasets

Table 6.1 presents the characteristics of the datasets used in our experiments. We used three MTC datasets (Solar Flare, Bridges, Thyroid) and seven MLC datasets (Music, Scene, Yeast, ENRON, Medical, Slashdot, LLOG).

	N	d	K	m
Music	593	6	2	72n
Scene	2407	6	2	294n
Yeast	2417	14	2	103n
ENRON	1702	53	2	1001b
Medical	978	45	2	1449b
Slashdot	3782	22	2	1079b
LLOG	1460	75	2	1004b
Solar Flare	323	3	5	10m
Bridges	107	5	2-6	7m
Thyroid	9172	7	2-5	7n, 20b, 1m

Table 6.1: Characteristics of the MTC and MLC datasets used in our experiments. N: number of examples; d: number of targets; K: number of values per target; m number of attributes, with n, b, and m corresponding to numeric, binary, and nominal attributes, respectively. Recall that, for the multi-label datasets, $K = 2$.

We included some multi-label datasets given the difficulty in finding well-formatted public data for multi-target classification. These datasets are commonly used in the multi-label literature (Sorower, 2010). Below we describe the domain of each dataset:

- Music: pieces of music are associated with various emotions;
- Scene: image annotation dataset, where images are labeled with scene concepts;
- Yeast: biological dataset where genes are associated with multiple biological functions;
- ENRON: a collection of email messages from the ENRON corpus, categorized into 53 topics, such as company strategy, humor, legal advice;
- Medical: free-text clinical reports labelled with one or more out of 45 disease codes;
- Slashdot: article titles with subject categories that represent the label space;
- LLOG: dataset compiled from the Language Log Forum, which discussed several topics related to the language and 75 topics;

- Solar Flare: categorizes types of solar flares, having three target variables corresponding to types of solar flares seen in a 24-hour period;
- Bridges: prediction of bridge design properties based on certain constraints;
- Thyroid: medical dataset for the prediction of types of thyroid problems given patient attributes.

6.3 Results Before Feature Extraction

As a baseline to start our experiments, we wanted to see how these multi-target classification methods perform on the selected datasets using their original features. The results are presented in Table 6.2.

	CR	CC	ECC	CRS	SCC
Hamming Score (HS)					
Music	0.733	0.709	**0.771**	0.761	0.712
Scene	0.857	0.853	**0.897**	0.868	0.845
Yeast	0.741	0.735	**0.793**	0.789	0.724
ENRON	0.936	0.935	**0.942**	0.941	0.923
Medical	0.988	0.988	**0.989**	**0.989**	**0.989**
Slashdot	**0.952**	0.938	0.951	0.923	—
LLOG	0.979	0.980	**0.983**	0.953	—
Solar Flare	**0.905**	**0.905**	**0.905**	**0.905**	**0.905**
Bridges	**0.622**	**0.622**	0.590	0.580	0.570
Thyroid	0.986	**0.987**	0.986	0.986	0.986
Exact Match Ratio (EMR)					
Music	0.165	0.169	0.245	**0.249**	0.186
Scene	0.404	0.523	**0.573**	0.519	0.522
Yeast	0.068	0.132	**0.175**	0.117	0.134
ENRON	0.023	0.048	**0.069**	0.050	0.026
Medical	0.615	0.630	0.640	0.625	**0.645**
Slashdot	0.239	**0.309**	0.230	0.247	—
LLOG	0.207	**0.214**	0.197	0.205	—
Solar Flare	**0.785**	**0.785**	**0.785**	**0.785**	**0.785**
Bridges	0.140	0.140	0.116	0.116	**0.163**
Thyroid	0.914	0.920	0.918	0.916	**0.924**

Table 6.2: Results without Feature Extraction

To perform the experiments of this research, we selected the following multi-target methods, using the J48 decision tree induction algorithm as base classifier. They are all implemented in the MEKA framework (Read et al., 2016), and we used their default hyperparameter values.

- Ensemble of Classifier Chains (ECC): It was executed with the default parameters present in (Read et al., 2016), where the number of trained classifier-chains in the ensemble is $d = 10$, and the threshold is $t = 0.67$;
- Super-Class Classifier (SCC): It was also executed with the default parameters present in (Read et al., 2016), where the number of iterations to build the super-class partition is $T = 1000$;
- Class Relevance (CR), Classifier Chains (CC), Class Relevance Stacking (CRS): these methods do not require additional execution parameters.

As we discussed in Section 6.1, the Hamming Score (HS) is used to measure how many targets were classified correctly on average. For this reason, methods that evaluate target relationships in an arbitrary way can obtain a better score on it. The Exact Match Ratio (EMR) calculates how many examples on average had all targets correctly predicted. In other words, it is more likely that methods that take into account the relationships among all targets will be highlighted when evaluated by EMR.

Starting with the CR, the simplest approach, which consists in predicting each target of the problem independently. Thus, it is expected that its best results would be measured by HS. As mentioned in (Read et al., 2011), as simple as it may be, this method can achieve competitive results, especially because there are no guarantee that necessarily exist relationships among targets. We can see that in the Slashdot dataset it got the best result, but very close to the ECC. This may indicate that most of the targets present in this dataset are unrelated. Another dataset that the CR had better results was Bridges, next to the CC. This dataset is a specific case. We can see in Bridges that all multi-target classifiers correctly classified just over half of the targets, which indicates that perhaps it does not have enough examples, or its features do not have enough characteristics to build a good multi-target classification model.

The CC, on the other hand, when evaluated through HS, it did not stood out in any dataset, while remained close to the performance of the CR. It only got a slightly higher score on Thyroid, but it correctly predicted only 26 targets more than the ECC, the second best, out of a total of 25683 targets. However, if we look at the EMR, CC surprisingly had better results than the other methods in the Slashdot and LLOG datasets. This may indicate that there is some relationships related to the order in which the targets appear in these datasets, since the CC "accumulates" the information across the targets in a chained way, following the order that they are presented. Another method that could have satisfactory results in these two datasets would be the SCC, and we will discuss further in this section why it was not executed in both of these datasets.

As we discussed at the beginning of this section, and also pointed out by other studies (Read et al., 2014, Jia and Zhang, 2020c), the ECC was expected to obtain the best results when evaluated by HS, and the only datasets that did not have better results was Slashdot, by an insignificant margin, and Bridges, which we justified earlier in this section. ECC also performed well on EMR for the multi-label datasets Scene, Yeast and ENRON, being slightly behind in Music and Medical. Based on these results, this method was also expected to excel during the experiments proposed in this research, and we present this in Section 6.4.

Next, we have the CRS, which was a straightforward adaptation of the multi-label literature method (Tsoumakas et al., 2009b) for multi-target, often used as a baseline method for comparison. It was expected that, as it is an adaptation of the CR that takes into account all targets, that this method would obtain better results in EMR rather than HS. Therefore, if we evaluate the CRS taking into account the HS, we can see that it did not stand out in any dataset, but some of its results remained relatively close to the ECC, which was not expected. On the other hand, we can confirm that the CRS obtained better results than the CR when compared by the EMR.

The only exception, again, was the Bridges dataset, as we justified earlier.

Finally, we evaluate the SCC. As seen in Table 6.2, the Slashdot and LLOG datasets were not executed in this method. This is due to the fact that both datasets have some unlabeled targets, and the SCC is not able to deal with this situation, which we did not expect, because at no time this is mentioned by the study in (Read et al., 2014). However, as mentioned in Section 6.2, these datasets were selected from the multi-label literature just because of the lack of existing datasets for multi-target classification, and the SCC is a method that was developed and tested exclusively for this task, and ultimately, the Slashdot and LLOG datasets were not used to evaluate the SCC in the study in which it was presented.

Regarding the results of SCC, as discussed in Section 4.5 and mentioned by Read et al. (2014), SCC is expected to obtain competitive results in EMR more than in HS. This is due to the fact that the method will create its predictive model based on the dependency among the targets, which makes it more likely to correctly predict all targets simultaneously. The results show that this, in fact, happened. In all multi-target datasets, SCC obtained the highest score when evaluated by EMR, specially in Thyroid.

One of the datasets that we did not went into detail earlier in this section was Solar Flare. As can be seen in Table 6.2, all methods obtained exactly the same results during the evaluations, both in HS and in EMR. In addition to the low number of features, Solar Flare is the dataset with the lowest number of targets, despite being a specific multi-target classification dataset. For those reasons, it is possible that the data present into this dataset does not characterize it as a complex problem for multi-target classification methods.

In Section 6.4 we present the results of our research, how the feature extraction methods were performed and, mainly, how the multi-target classifiers reacted to the features generated by these extractors.

6.4 Results After Feature Extraction

In this section we present the results obtained after extracting features and executing all classifiers. We also explain some of the particularities encountered when analysing the data obtained from the executions. The results are presented in Tables 6.3, 6.4, 6.5, 6.6 and 6.7.

As described in Chapter 5, the goal of the proposed framework is to make use of neural networks, in our case Auto-Encoders (AEs) and Restricted Boltzmann Machines (RBMs), to extract features from multi-target datasets. Based on the proposed framework, we list below the selected methods for extracting features and the parameters used for the first step of the framework:

- Auto-Encoder (AE): An AE with a hidden layer and an output layer. The number of units in the hidden layer of the Auto-Encoder is the same number of original features, and the number of units in the output layer is proportional to the percentage of features extracted;
- Denoising Auto-Encoder (dAE): The configuration of the layers is the same as in AE. We added a Gaussian noise in the input layer of 20%, with the intention that the neural

network obtains a better generalization performance (Vincent et al., 2008), and be able to achive better reconstruction when the input contains missing data, which is the case of the Bridges and Thyroid datasets (Section 6.2);

- Deep Auto-Encoder (D-AE): An AE containing three hidden layers and an output layer. The number of units in the hidden layers corresponds to the average of the number of original features and the number of features to be extracted. For example, if the number of original features is 100, and the number of features to extract is 50, the number of units in the hidden layer will be 75. The output layer has the same configuration as in AE;

- Deep-Denoising Auto-Encoder (D-dAE): Consists of the combination of dAE and D-AE. The configuration of the layers is the same as that one used in the D-AE, and here we also add the same noise configuration that is present in the dAE;

- Restricted Boltzmann Machine (RBM): RBMs do not have a hidden layer, so the number of units in the input layer is equal to the number of original features, and the number of units in the output layer corresponds to the percentage of features that we want to extract;

- Deep Belief Network (DBN): Composed of two stacked RBMs, forming a neural network with a hidden layer and an output layer. The hidden layer (which is the output of the first RBM), has the number of units calculated in the same way as in the hidden layer of the D-AE, and the output layer is the same as the RBM;

- Principal Component Analysis (PCA): It was executed under the implementation present in (Pedregosa et al., 2011).

Recall that the hyperparameters used in neural networks are present in Section 5.1. After extracting the features from the original datasets, we executed the multi-target classifiers. The selected classifiers as well as their parameters were the same as presented on Section 6.3.

The results obtained in our experiments are presented on Tables 6.3, 6.4, 6.5, 6.6 and 6.7, and they are organized as follows: each column represents the feature extraction method used, and each row represents one of the datasets we used. The values present in these tables correspond to the average obtained for HS and EMR across all the 10 folds from the cross-validation setup (as presented in Chapter 5), and the values in parentheses correspond to the standard deviations. In bold, are the feature extraction methods that obtained the best results in each dataset. The last column represents the results obtained with the original dataset, without extracting features. We use an asterisk ($*$) to highlight when a result of a multi-target classifier with extracted features was the same or better than that obtained with the original dataset.

Before we start presenting the results, it is important to mention two peculiarities found during our experiments, which can be seen in Tables 6.3, 6.4, 6.5, 6.6 and 6.7. The first is that the SCC is not able to deal with unlabeled targets (as discussed in Section 6.3) , which makes it impossible to execute in the Slashdot and LLOG datasets. The second peculiarity is related to the PCA, since it is not possible to execute the PCA in datasets where the number of features is greater than the number of examples. This meant that the PCA could not have its results computed for the Medical, Slashdot and LLOG datasets, due to the fact that during the cross-validation used in our framework (Section 5), these datasets ended up having the number of examples smaller than than the number of features they have.

We start our discussion with the feature extractions at 10% of the original number of features present in the datasets. First, it is important to clarify why the Bridges dataset has not been evaluated. This is due to the fact that this dataset has 7 features, i.e. it is not possible to extract 10% of the features in this dataset. For this reason, later in this section, we will also comment on the feature extraction at 20%, with emphasis on the features extracted for the Bridges dataset.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.706 (0.021)	**0.712 (0.015)**	0.702 (0.017)	0.705 (0.014)	0.658 (0.010)	0.656 (0.007)	0.638 (0.017)	0.733
Scene	0.788 (0.011)	0.785 (0.014)	0.795 (0.006)	0.790 (0.008)	**0.803 (0.011)**	0.791 (0.015)	0.759 (0.007)	0.857
Yeast	0.768 (0.002)*	0.769 (0.002)*	0.769 (0.001)*	0.769 (0.001)*	**0.772 (0.003)***	0.768 (0.005)*	0.730 (0.003)	0.741
ENRON	0.924 (0.003)	0.924 (0.001)	0.924 (0.002)	0.926 (0.003)	0.928 (0.004)	**0.936 (0.002)***	0.879 (0.002)	0.936
Medical	0.963 (0.003)	0.963 (0.002)	0.963 (0.001)	0.964 (0.002)	0.963 (0.003)	**0.971 (0.002)**	—	0.988
Slashdot	**0.945 (0.001)**	0.944 (0.000)	**0.945 (0.001)**	0.944 (0.001)	0.934 (0.003)	0.943 (0.005)	—	0.952
LLOG	**0.983 (0.001)***	0.982 (0.001)*	**0.983 (0.000)***	**0.983 (0.000)***	**0.983 (0.001)***	**0.983 (0.001)***	—	0.979
Solar Flare	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	0.860 (0.014)	0.905
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.959 (0.001)	0.986
				Exact Match Ratio (EMR)				
Music	0.149 (0.050)	0.157 (0.030)	0.149 (0.022)	**0.163 (0.021)**	0.100 (0.023)	0.103 (0.022)	0.080 (0.027)	0.165
Scene	0.236 (0.046)	0.253 (0.016)	0.250 (0.011)	0.244 (0.010)	**0.300 (0.018)**	0.158 (0.042)	0.226 (0.022)	0.404
Yeast	0.018 (0.002)	0.018 (0.002)	0.018 (0.002)	0.018 (0.002)	**0.054 (0.008)**	0.046 (0.018)	0.035 (0.009)	0.068
ENRON	0.003 (0.003)	0.003 (0.002)	0.003 (0.002)	0.002 (0.002)	0.026 (0.005)*	**0.030 (0.011)***	0.015 (0.005)	0.023
Medical	0.026 (0.009)	0.020 (0.014)	0.030 (0.012)	0.022 (0.013)	0.164 (0.021)	**0.171 (0.030)**	—	0.615
Slashdot	0.002 (0.003)	0.004 (0.004)	0.002 (0.001)	0.003 (0.001)	**0.147 (0.012)**	0.091 (0.008)	—	0.239
LLOG	0.144 (0.006)	0.138 (0.011)	0.144 (0.007)	0.144 (0.009)	**0.167 (0.007)**	0.161 (0.013)	—	0.207
Solar Flare	**0.796 (0.013)***	**0.796 (0.013)***	**0.796 (0.013)***	**0.796 (0.013)***	**0.796 (0.013)***	**0.796 (0.013)***	0.783 (0.024)	0.785
Thyroid	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.003)**	0.914

Table 6.3: Results using CR with 10% of the original features.

In Table 6.3 we present the results obtained through the feature-extraction in the CR. Regarding the performance of the AEs, we can see that the dAE variant stood out in the Music dataset, when we analyzed HS. This was not expected, because as we discussed at the beginning of this section, dAE was inserted in our experiments in order to obtain better results in datasets that have missing data, which is not the case with Music. If we compare the standard deviation of the dAE among the other feature-extraction methods, we can see that it is relatively low, so this probably should not be related to any outlier, but rather to the characteristics of the data present in Music.

Still regarding the AEs in CR, we can identify that the variants D-AE and D-dAE, although they did not stand out among any other method, were the ones that had less variation between the results obtained across all the 10 folds, which can indicate that the various hidden layers kept the model more stable. We emphasize that all the variants of AE managed to obtain better predictions when compared to the original datasets in Yeast, LLOG and Solar Flare. However, it is possible to observe that the variants of AE did not obtain good results when evaluated by EMR, with the exception of D-dAE for the Music dataset, again, which reinforces the thesis that the characteristics of data present in Music favor the features generated through the denoising variants of AEs.

Unlike the AEs, both RBM and DBN achieved better results in the CR when evaluated by EMR. Under this measure, RBM obtained better results in the Scene and Yeast datasets by a relatively good margin, especially when compared to AEs, and the features generated through RBM in the ENRON dataset obtained an even better prediction result than the original dataset. Looking at HS, RBM also stood out, but this time by a smaller margin, and its features also achieved better results than the original datasets in Yeast, ENRON and LLOG. The DBN variant obtained better results than the other methods for both HS and EMR in the ENRON and Medical datasets, and the features obtained in ENRON also brought better results than the original dataset, in the two evaluation measures.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.687 (0.058)	**0.706 (0.012)**	0.699 (0.020)	**0.706 (0.012)**	0.601 (0.011)	0.599 (0.017)	0.612 (0.020)	0.709
Scene	0.785 (0.016)	0.788 (0.010)	0.792 (0.007)	0.787 (0.004)	**0.807 (0.005)**	0.789 (0.012)	0.753 (0.008)	0.853
Yeast	0.661 (0.017)	0.661 (0.022)	0.673 (0.019)	0.651 (0.021)	**0.678 (0.016)**	0.677 (0.005)	0.643 (0.008)	0.735
ENRON	0.915 (0.001)	0.915 (0.002)	0.917 (0.002)	0.917 (0.002)	0.926 (0.004)	**0.932 (0.002)**	0.878 (0.002)	0.935
Medical	0.961 (0.001)	0.962 (0.002)	0.961 (0.002)	0.962 (0.001)	0.962 (0.004)	**0.967 (0.003)**	—	0.988
Slashdot	0.908 (0.001)	0.907 (0.002)	0.907 (0.001)	0.907 (0.001)	**0.921 (0.004)**	0.920 (0.003)	—	0.938
LLOG	0.980 (0.002)*	0.981 (0.001)*	0.981 (0.001)*	**0.982 (0.002)***	0.980 (0.003)*	0.978 (0.003)	—	0.980
Solar Flare	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	0.864 (0.008)	0.905
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.959 (0.001)	0.987
				Exact Match Ratio (EMR)				
Music	**0.212 (0.038)***	0.206 (0.022)*	0.197 (0.030)*	0.209 (0.020)*	0.149 (0.015)	0.144 (0.017)	0.151 (0.028)	0.169
Scene	0.350 (0.042)	0.360 (0.025)	0.366 (0.020)	0.355 (0.014)	**0.408 (0.016)**	0.357 (0.026)	0.366 (0.023)	0.523
Yeast	0.012 (0.007)	0.018 (0.011)	0.012 (0.006)	0.011 (0.004)	0.059 (0.015)	0.056 (0.005)	**0.060 (0.007)**	0.132
ENRON	0.013 (0.005)	0.016 (0.003)	0.019 (0.006)	0.014 (0.006)	0.036 (0.008)	**0.046 (0.012)**	0.035 (0.008)	0.048
Medical	0.046 (0.020)	0.040 (0.013)	0.053 (0.014)	0.053 (0.020)	0.198 (0.031)	**0.205 (0.031)**	—	0.630
Slashdot	0.061 (0.006)	0.059 (0.009)	0.057 (0.003)	0.058 (0.008)	**0.192 (0.017)**	0.163 (0.011)	—	0.309
LLOG	0.124 (0.018)	0.126 (0.014)	0.123 (0.016)	0.133 (0.018)	**0.186 (0.020)**	0.181 (0.016)	—	0.214
Solar Flare	**0.796 (0.013)***	**0.796 (0.013)***	**0.796 (0.013)***	**0.796 (0.013)***	**0.796 (0.013)***	**0.796 (0.013)***	0.791 (0.017)*	0.785
Thyroid	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.003)**	0.920

Table 6.4: Results using CC with 10% of the original features.

In Table 6.4 we present the results obtained through the feature-extraction in CC. Unlike as happened in CR, in CC all variants of AE generated features that brought a better result than the original Music dataset. When we look at HS, the features obtained through dAE also brought better results in Music.

While observing the results for RBM and DBN in the CC, it is possible to verify again that both methods stood out from the others when evaluated by EMR. The features generated by RBM in the Scene, Slashdot and LLOG datasets achieved better results than the other methods, and the DBN features reached better results in the ENRON and Medical datasets.

In general, the HS for the feature extraction methods in CC were similar to those of CR. However, when evaluating the performances of the CC with EMR, only the features generated by the variants of the AE brought results superior to those obtained by the original dataset, which did not occur in the CR. But it is worth noting that the dataset in which the AEs obtained the best result was Music, and this happened in all the classifiers we tested. Therefore, this may not indicate that the AE was better, but that the CC obtains better results in HS than in EMR, as discussed in Section 6.3.

In Table 6.5 we present the results obtained through feature-extraction in ECC. As the ECC was proposed as an improvement for the CC, where its predictive model takes into account

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.715 (0.022)	0.723 (0.016)	0.717 (0.015)	**0.725 (0.018)**	0.635 (0.022)	0.597 (0.012)	0.623 (0.026)	0.771
Scene	0.822 (0.015)	0.825 (0.006)	**0.832 (0.009)**	0.826 (0.008)	0.822 (0.015)	0.776 (0.009)	0.779 (0.006)	0.897
Yeast	0.736 (0.002)	0.735 (0.003)	**0.738 (0.003)**	0.734 (0.001)	0.734 (0.007)	0.731 (0.007)	0.708 (0.004)	0.793
ENRON	0.933 (0.001)	0.933 (0.001)	0.933 (0.001)	0.933 (0.001)	0.940 (0.002)	**0.941 (0.002)**	0.892 (0.001)	0.942
Medical	0.972 (0.001)	0.972 (0.000)	0.971 (0.000)	0.971 (0.001)	0.970 (0.004)	**0.974 (0.002)**	—	0.989
Slashdot	0.946 (0.000)	0.946 (0.000)	0.946 (0.000)	0.946 (0.000)	0.941 (0.001)	**0.947 (0.003)**	—	0.951
LLOG	**0.984 (0.000)***	**0.984 (0.000)***	**0.984 (0.000)***	**0.984 (0.000)***	0.983 (0.001)*	0.983 (0.001)*	—	0.983
Solar Flare	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	0.866 (0.007)	0.905
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.959 (0.001)	0.986
				Exact Match Ratio (EMR)				
Music	0.192 (0.065)	**0.228 (0.024)**	0.207 (0.028)	0.218 (0.024)	0.152 (0.020)	0.122 (0.022)	0.124 (0.023)	0.245
Scene	0.339 (0.080)	0.368 (0.017)	0.349 (0.025)	0.338 (0.015)	**0.401 (0.027)**	0.275 (0.028)	0.338 (0.014)	0.573
Yeast	0.032 (0.006)	0.031 (0.004)	0.032 (0.005)	0.027 (0.004)	0.081 (0.007)	**0.086 (0.006)**	0.056 (0.007)	0.175
ENRON	0.002 (0.002)	0.003 (0.002)	0.002 (0.002)	0.003 (0.003)	0.034 (0.006)	**0.036 (0.011)**	0.026 (0.004)	0.069
Medical	0.005 (0.005)	0.004 (0.004)	0.009 (0.009)	0.008 (0.005)	**0.191 (0.035)**	0.182 (0.029)	—	0.640
Slashdot	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	**0.145 (0.008)**	0.090 (0.009)	—	0.230
LLOG	0.154 (0.005)	0.154 (0.004)	0.154 (0.005)	0.154 (0.005)	**0.168 (0.006)**	0.158 (0.016)	—	0.197
Solar Flare	0.796 (0.013)*	0.796 (0.013)*	0.796 (0.013)*	0.796 (0.013)*	0.796 (0.013)*	0.796 (0.013)*	**0.797 (0.012)***	0.785
Thyroid	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.003)**	0.918

Table 6.5: Results using ECC with 10% of the original features.

the relationships among the targets in a better way, this method was expected to obtain better results than CC when evaluated through EMR. Counter intuitively, just the opposite happened, the EMR results were lower in ECC than in CC. It was expected that ECC would obtain better EMR results than the CC, which did not happen.

However, ECC had the best results overall in HS. This classifier obtained the best results with the features extracted from the AEs, with D-AE obtaining the best results in the Scene and Yeast datasets, and D-dAE once again obtaining the best result in the Music dataset. On the other hand, ECC was the classifier in which AEs had the worst results when evaluated by EMR, with the results being zero (or very close to zero) in the ENRON, Medical and Slashdot datasets. This can be seen as the reflex of the ECC results in EMR in general.

In ECC, we can again observe that the feature-extraction methods RBM and DBN were better than the others when evaluated by EMR. In general, the datasets for which RBM and DBN had the best results in EMR were the same as in CC. However, if we compare the results of EMR with CC, those obtained in ECC were lower.

In Table 6.6 are the results obtained through the feature-extraction in the CRS. Unlike to what happened between the CC and the ECC, which are two very related methods that presented different results regarding the EMR, the results between CR and CRS were as expected. The results presented in the CRS were, in general, better than those presented by the CR, both when evaluated by HS and by EMR.

In Table 6.7 we present the results obtained through feature-extraction in SCC. As the SCC authors mentions (Read et al., 2014), this method tends to have more competitive results in EMR rather than in HS. And it is possible in this table, that SCC results went as expected. When analyzing the results for HS, we can say that, in general, all results were lower than those present in the other classifiers, especially when compared to ECC. However, the EMR results were the best among all the classifiers that we used in our experiments.

Comparing the results obtained among all the multi-target classifiers, we can say that AEs,

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.711 (0.018)	**0.712 (0.015)**	0.702 (0.017)	0.705 (0.014)	0.658 (0.010)	0.656 (0.007)	0.638 (0.017)	0.785
Scene	0.789 (0.007)	0.785 (0.014)	0.795 (0.006)	0.790 (0.008)	**0.803 (0.011)**	0.791 (0.015)	0.759 (0.007)	0.898
Yeast	0.771 (0.005)	0.773 (0.004)	0.771 (0.004)	**0.776 (0.003)**	0.772 (0.004)	0.769 (0.003)	0.730 (0.003)	0.789
ENRON	0.924 (0.003)	0.924 (0.001)	0.924 (0.002)	0.926 (0.003)	0.928 (0.004)	**0.936 (0.002)**	0.879 (0.002)	0.941
Medical	0.963 (0.003)	0.963 (0.002)	0.963 (0.001)	0.964 (0.002)	0.963 (0.003)	**0.971 (0.002)**	—	0.989
Slashdot	**0.945 (0.001)**	0.944 (0.000)	**0.945 (0.001)**	0.944 (0.001)	0.934 (0.003)	0.943 (0.005)	—	0.953
LLOG	**0.983 (0.001)***	0.982 (0.001)	**0.983 (0.000)***	**0.983 (0.000)***	**0.983 (0.001)***	**0.983 (0.001)***	—	0.983
Solar Flare	**0.912 (0.004)***	**0.912 (0.004)***	0.907 (0.015)*	**0.912 (0.004)***	0.910 (0.002)*	**0.912 (0.004)***	0.860 (0.014)	0.905
Thyroid	0.958 (0.010)	0.955 (0.012)	0.958 (0.010)	0.942 (0.016)	**0.961 (0.001)**	**0.961 (0.001)**	0.959 (0.001)	0.986
				Exact Match Ratio (EMR)				
Music	**0.169 (0.018)**	0.157 (0.030)	0.149 (0.022)	0.163 (0.021)	0.100 (0.023)	0.103 (0.022)	0.080 (0.027)	0.249
Scene	0.252 (0.012)	0.253 (0.016)	0.250 (0.011)	0.244 (0.010)	**0.300 (0.018)**	0.158 (0.042)	0.226 (0.022)	0.519
Yeast	**0.081 (0.007)**	0.077 (0.011)	0.078 (0.009)	**0.081 (0.007)**	0.050 (0.009)	0.040 (0.019)	0.035 (0.009)	0.117
ENRON	0.003 (0.003)	0.003 (0.002)	0.003 (0.002)	0.002 (0.002)	0.026 (0.005)	**0.030 (0.011)**	0.015 (0.005)	0.050
Medical	0.026 (0.009)	0.020 (0.014)	0.030 (0.012)	0.022 (0.013)	0.164 (0.021)	**0.171 (0.030)**	—	0.625
Slashdot	0.002 (0.003)	0.004 (0.004)	0.002 (0.001)	0.003 (0.001)	**0.147 (0.012)**	0.091 (0.008)	—	0.247
LLOG	0.144 (0.006)	0.138 (0.011)	0.144 (0.007)	0.144 (0.009)	**0.167 (0.007)**	0.161 (0.013)	—	0.205
Solar Flare	**0.796 (0.006)***	**0.796 (0.006)***	0.783 (0.038)	**0.796 (0.006)***	0.791 (0.009)*	**0.796 (0.006)***	0.783 (0.024)	0.785
Thyroid	0.726 (0.053)	0.708 (0.071)	0.729 (0.054)	0.637 (0.089)	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	0.916

Table 6.6: Results using CRS with 10% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.665 (0.038)	0.681 (0.017)	0.672 (0.017)	**0.694 (0.017)**	0.623 (0.013)	0.592 (0.016)	0.589 (0.015)	0.712
Scene	0.781 (0.017)	0.786 (0.006)	0.786 (0.011)	0.782 (0.007)	**0.803 (0.008)**	0.763 (0.010)	0.744 (0.006)	0.845
Yeast	0.680 (0.004)	0.679 (0.004)	0.680 (0.005)	0.679 (0.004)	0.697 (0.008)	**0.708 (0.009)**	0.652 (0.003)	0.724
ENRON	0.909 (0.002)	0.908 (0.001)	0.909 (0.001)	0.908 (0.002)	0.917 (0.003)	**0.920 (0.002)**	0.871 (0.002)	0.923
Medical	0.951 (0.001)	0.952 (0.001)	0.951 (0.001)	0.952 (0.002)	0.963 (0.003)	**0.965 (0.004)**	—	0.989
Slashdot	—	—	—	—	—	—	—	—
LLOG	—	—	—	—	—	—	—	—
Solar Flare	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	0.904 (0.023)	0.865 (0.008)	0.905
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.959 (0.001)	0.986
				Exact Match Ratio (EMR)				
Music	0.164 (0.039)	0.175 (0.017)	0.168 (0.019)	**0.201 (0.030)***	0.154 (0.020)	0.126 (0.022)	0.106 (0.018)	0.186
Scene	0.340 (0.047)	0.351 (0.016)	0.352 (0.027)	0.336 (0.019)	**0.395 (0.018)**	0.297 (0.025)	0.342 (0.015)	0.522
Yeast	0.038 (0.004)	0.039 (0.006)	0.042 (0.008)	0.042 (0.007)	0.083 (0.010)	**0.088 (0.004)**	0.053 (0.006)	0.134
ENRON	0.012 (0.004)	0.011 (0.004)	0.009 (0.003)	0.008 (0.003)	0.030 (0.005)*	**0.044 (0.013)***	0.026 (0.004)*	0.026
Medical	0.067 (0.009)	0.067 (0.018)	0.060 (0.012)	0.069 (0.016)	0.186 (0.028)	**0.236 (0.024)**	—	0.645
Slashdot	—	—	—	—	—	—	—	—
LLOG	—	—	—	—	—	—	—	—
Solar Flare	**0.796 (0.013)***	**0.796 (0.013)***	**0.796 (0.013)***	**0.796 (0.013)***	**0.796 (0.013)***	0.775 (0.062)	0.794 (0.012)*	0.785
Thyroid	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.003)**	0.924

Table 6.7: Results using SCC with 10% of the original features.

especially their denoising variants, stood out in the Music dataset, for both HS and EMR. AEs demonstrated competitive performance in relation to RBMs and DBNs when compared to HS. However, RBMs and DBNs achieved results relatively superior to AE, when compared by EMR, the only exception being the Music dataset. We can say that the features generated by DBN were the most successful in EMR.

The PCA, in turn, was used as a baseline method for comparison. It is a statistical method aimed at dimensionality reduction. Considering that it is the simplest method of our comparison, much less sophisticated than neural networks, it presented results above expectations. It is even possible to see that, in the CC, it was the method that achieved the best classification result for the Yeast dataset, when evaluated by EMR.

As mentioned earlier in this section, the Bridges dataset could not be evaluated together with the other datasets when extracting new features at 10%. Therefore, in the next paragraphs we will present how these classifiers performed with the features extracted from Bridges.

In Tables 6.8, 6.9, 6.10, 6.11, and 6.12, we present the results obtained for the classifiers when extracting new features at 20%. We can see that none of the multi-target classifiers in conjunct with any feature extraction method were able to achieve higher results with the Bridges datasets. As mentioned in Section 6.3, all multi-target classifiers had difficulty predicting its classes, which may be due to the fact that this dataset has few examples, or its features do not have enough characteristics to create a good model for multi-target classification. In the HS evaluation, it is possible to notice that just over half of its targets were classified correctly. We can also see that the method for which Bridges had the best results in EMR was PCA, which can be related to the low number of examples present in this dataset.

When analyzing the results obtained with the new features extracted at 20%, it is possible to see that the scores obtained by HS and EMR had lowered, when compared to the same classifiers and feature-extraction methods at 10%. In Section 6.5 we will discuss about it and make the final considerations about our experiments.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.693 (0.017)	0.699 (0.024)	0.701 (0.013)	**0.703 (0.017)**	0.660 (0.006)	0.657 (0.010)	0.618 (0.023)	0.733
Scene	0.783 (0.015)	0.786 (0.011)	0.770 (0.010)	0.787 (0.010)	**0.817 (0.016)**	0.781 (0.010)	0.750 (0.007)	0.857
Yeast	0.768 (0.002)*	0.769 (0.001)*	0.769 (0.002)*	0.768 (0.002)*	**0.774 (0.002)*****	0.771 (0.006)*	0.728 (0.003)	0.741
ENRON	0.916 (0.002)	0.916 (0.003)	0.916 (0.002)	0.920 (0.003)	0.922 (0.005)	**0.933 (0.002)**	0.875 (0.003)	0.936
Medical	0.955 (0.002)	0.958 (0.004)	0.955 (0.002)	0.956 (0.003)	0.950 (0.008)	**0.969 (0.003)**	—	0.988
Slashdot	**0.940 (0.003)**	0.938 (0.003)	0.938 (0.003)	0.939 (0.003)	0.917 (0.008)	0.939 (0.005)	—	0.952
LLOG	0.980 (0.001)*	0.980 (0.001)*	0.980 (0.001)*	0.980 (0.001)*	0.982 (0.002)*	**0.983 (0.001)*****	—	0.979
Solar Flare	**0.911 (0.008)*****	**0.911 (0.008)*****	0.909 (0.007)*	**0.911 (0.008)*****	0.909 (0.007)*	0.902 (0.024)	0.864 (0.007)	0.905
Bridges	0.544 (0.027)	0.544 (0.036)	0.529 (0.044)	0.532 (0.064)	0.470 (0.047)	**0.560 (0.037)**	0.545 (0.019)	0.622
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.959 (0.000)	0.986
				Exact Match Ratio (EMR)				
Music	0.111 (0.035)	0.140 (0.013)	0.128 (0.024)	**0.146 (0.022)**	0.104 (0.013)	0.102 (0.021)	0.078 (0.021)	0.165
Scene	0.209 (0.038)	0.235 (0.017)	0.203 (0.016)	0.219 (0.021)	**0.304 (0.025)**	0.139 (0.035)	0.216 (0.014)	0.404
Yeast	0.017 (0.003)	0.018 (0.001)	0.018 (0.002)	0.018 (0.003)	0.056 (0.003)	**0.060 (0.010)**	0.027 (0.006)	0.068
ENRON	0.003 (0.002)	0.002 (0.002)	0.003 (0.002)	0.004 (0.004)	0.023 (0.004)*	**0.028 (0.008)*****	0.008 (0.004)	0.023
Medical	0.031 (0.010)	0.041 (0.010)	0.032 (0.006)	0.034 (0.010)	**0.181 (0.021)**	0.166 (0.041)	—	0.615
Slashdot	0.011 (0.008)	0.014 (0.006)	0.012 (0.005)	0.013 (0.007)	**0.145 (0.013)**	0.085 (0.014)	—	0.239
LLOG	0.121 (0.010)	0.121 (0.015)	0.123 (0.017)	0.122 (0.020)	0.164 (0.013)	**0.167 (0.007)**	—	0.207
Solar Flare	**0.794 (0.015)*****	**0.794 (0.015)*****	0.790 (0.015)*	**0.794 (0.015)*****	0.792 (0.013)*	0.770 (0.068)	0.791 (0.017)*	0.785
Bridges	0.040 (0.018)	0.028 (0.020)	0.028 (0.020)	0.038 (0.021)	0.016 (0.023)	0.035 (0.024)	**0.068 (0.022)**	0.140
Thyroid	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.003)**	0.914

Table 6.8: Results using CR with 20% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
Hamming Score (HS)								
Music	0.668 (0.052)	**0.698 (0.015)**	0.694 (0.013)	0.690 (0.018)	0.603 (0.012)	0.604 (0.013)	0.581 (0.019)	0.709
Scene	0.772 (0.017)	0.784 (0.006)	0.774 (0.009)	0.780 (0.008)	**0.810 (0.005)**	0.782 (0.010)	0.743 (0.008)	0.853
Yeast	0.653 (0.021)	0.662 (0.023)	0.654 (0.015)	0.658 (0.021)	0.668 (0.013)	**0.676 (0.008)**	0.646 (0.014)	0.735
ENRON	0.914 (0.001)	0.914 (0.002)	0.914 (0.002)	0.915 (0.001)	0.926 (0.005)	**0.930 (0.002)**	0.874 (0.002)	0.935
Medical	0.958 (0.001)	0.959 (0.001)	0.958 (0.002)	0.957 (0.002)	0.952 (0.006)	**0.968 (0.003)**	—	0.988
Slashdot	0.905 (0.001)	0.906 (0.001)	0.906 (0.001)	0.905 (0.001)	0.911 (0.008)	**0.920 (0.001)**	—	0.938
LLOG	0.978 (0.001)	**0.979 (0.001)**	**0.979 (0.001)**	**0.979 (0.001)**	0.978 (0.003)	0.978 (0.003)	—	0.980
Solar Flare	**0.911 (0.008)***	**0.911 (0.008)***	0.909 (0.007)*	**0.911 (0.008)***	**0.911 (0.008)***	**0.911 (0.008)***	0.866 (0.004)	0.905
Bridges	0.557 (0.037)	**0.570 (0.033)**	0.550 (0.045)	0.532 (0.059)	0.506 (0.061)	0.561 (0.035)	0.538 (0.019)	0.622
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.959 (0.000)	0.987
Exact Match Ratio (EMR)								
Music	0.173 (0.025)*	**0.194 (0.019)***	0.178 (0.015)*	0.190 (0.018)*	0.151 (0.012)	0.149 (0.014)	0.118 (0.018)	0.169
Scene	0.318 (0.047)	0.349 (0.017)	0.319 (0.021)	0.337 (0.021)	**0.410 (0.012)**	0.343 (0.022)	0.337 (0.025)	0.523
Yeast	0.013 (0.007)	0.016 (0.010)	0.015 (0.005)	0.015 (0.008)	0.054 (0.005)	0.058 (0.011)	**0.059 (0.011)**	0.132
ENRON	0.013 (0.003)	0.015 (0.002)	0.014 (0.004)	0.015 (0.004)	**0.042 (0.010)**	0.040 (0.012)	0.034 (0.011)	0.048
Medical	0.058 (0.008)	0.062 (0.014)	0.059 (0.011)	0.064 (0.007)	**0.201 (0.022)**	0.200 (0.053)	—	0.630
Slashdot	0.061 (0.008)	0.064 (0.010)	0.062 (0.009)	0.057 (0.008)	**0.188 (0.018)**	0.160 (0.008)	—	0.309
LLOG	0.107 (0.014)	0.111 (0.013)	0.112 (0.017)	0.110 (0.010)	**0.195 (0.014)**	0.185 (0.014)	—	0.214
Solar Flare	0.794 (0.015)*	0.794 (0.015)*	0.790 (0.015)*	0.794 (0.015)*	0.794 (0.015)*	0.794 (0.015)*	**0.796 (0.010)***	0.785
Bridges	0.051 (0.025)	0.044 (0.016)	0.047 (0.015)	0.035 (0.024)	0.016 (0.021)	0.030 (0.026)	**0.063 (0.015)**	0.140
Thyroid	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.003)**	0.920

Table 6.9: Results using CC with 20% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
Hamming Score (HS)								
Music	0.714 (0.016)	0.722 (0.017)	0.715 (0.024)	**0.727 (0.015)**	0.638 (0.011)	0.596 (0.014)	0.609 (0.018)	0.771
Scene	0.823 (0.015)	**0.833 (0.009)**	0.825 (0.007)	0.826 (0.007)	0.824 (0.012)	0.776 (0.004)	0.775 (0.007)	0.897
Yeast	**0.739 (0.003)**	0.736 (0.002)	0.738 (0.003)	0.737 (0.003)	**0.739 (0.007)**	0.725 (0.006)	0.707 (0.004)	0.793
ENRON	0.932 (0.001)	0.932 (0.001)	0.932 (0.000)	0.932 (0.001)	0.938 (0.004)	**0.941 (0.001)**	0.892 (0.004)	0.942
Medical	0.971 (0.000)	0.971 (0.000)	0.970 (0.001)	0.971 (0.000)	0.963 (0.007)	**0.974 (0.002)**	—	0.989
Slashdot	**0.946 (0.000)**	**0.946 (0.000)**	**0.946 (0.000)**	**0.946 (0.000)**	0.933 (0.005)	0.944 (0.005)	—	0.951
LLOG	**0.984 (0.000)***	**0.984 (0.000)***	**0.984 (0.000)***	**0.984 (0.000)***	**0.984 (0.001)***	0.983 (0.001)*	—	0.983
Solar Flare	**0.911 (0.008)***	**0.911 (0.008)***	0.909 (0.007)*	**0.911 (0.008)***	**0.911 (0.008)***	**0.911 (0.008)***	0.866 (0.004)	0.905
Bridges	0.516 (0.035)	0.514 (0.053)	0.507 (0.053)	0.477 (0.053)	0.438 (0.027)	**0.540 (0.038)**	0.522 (0.030)	0.590
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.959 (0.000)	0.986
Exact Match Ratio (EMR)								
Music	0.181 (0.064)	0.214 (0.026)	0.207 (0.035)	**0.216 (0.018)**	0.158 (0.014)	0.120 (0.018)	0.121 (0.022)	0.245
Scene	0.291 (0.072)	0.333 (0.014)	0.307 (0.019)	0.309 (0.013)	**0.404 (0.016)**	0.267 (0.016)	0.317 (0.026)	0.573
Yeast	0.030 (0.005)	0.027 (0.005)	0.030 (0.007)	0.031 (0.006)	**0.085 (0.006)**	0.084 (0.009)	0.050 (0.010)	0.175
ENRON	0.003 (0.002)	0.004 (0.002)	0.004 (0.002)	0.003 (0.003)	0.031 (0.006)	**0.034 (0.008)**	0.019 (0.009)	0.069
Medical	0.018 (0.004)	0.020 (0.005)	0.016 (0.008)	0.018 (0.005)	**0.232 (0.020)**	0.173 (0.045)	—	0.640
Slashdot	0.000 (0.000)	0.000 (0.001)	0.000 (0.001)	0.001 (0.001)	**0.152 (0.009)**	0.086 (0.004)	—	0.230
LLOG	0.154 (0.005)	0.154 (0.005)	0.154 (0.005)	0.154 (0.005)	0.162 (0.011)	**0.166 (0.005)**	—	0.197
Solar Flare	**0.796 (0.016)***	**0.796 (0.016)***	0.791 (0.013)*	**0.796 (0.016)***	**0.796 (0.016)***	**0.796 (0.016)***	**0.796 (0.010)***	0.785
Bridges	0.035 (0.024)	0.030 (0.021)	0.033 (0.019)	0.030 (0.028)	0.021 (0.022)	0.032 (0.028)	**0.054 (0.018)**	0.116
Thyroid	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.003)**	**0.743 (0.002)**	**0.743 (0.003)**	0.918

Table 6.10: Results using ECC with 20% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.696 (0.016)	0.699 (0.024)	0.701 (0.013)	**0.703 (0.017)**	0.660 (0.006)	0.657 (0.010)	0.618 (0.023)	0.785
Scene	0.786 (0.009)	0.786 (0.011)	0.770 (0.010)	0.787 (0.010)	**0.817 (0.016)**	0.781 (0.010)	0.750 (0.007)	0.898
Yeast	0.767 (0.003)	0.765 (0.005)	0.764 (0.005)	0.767 (0.004)	**0.773 (0.001)**	**0.773 (0.003)**	0.728 (0.003)	0.789
ENRON	0.916 (0.002)	0.916 (0.003)	0.916 (0.002)	0.920 (0.003)	0.922 (0.005)	**0.933 (0.002)**	0.875 (0.003)	0.941
Medical	0.955 (0.002)	0.958 (0.004)	0.955 (0.002)	0.956 (0.003)	0.950 (0.008)	**0.969 (0.003)**	—	0.989
Slashdot	**0.940 (0.003)**	0.938 (0.003)	0.938 (0.003)	0.939 (0.003)	0.917 (0.008)	0.939 (0.005)	—	0.953
LLOG	0.980 (0.001)	0.980 (0.001)	0.980 (0.001)	0.980 (0.001)	0.982 (0.002)	**0.983 (0.001)***	—	0.983
Solar Flare	**0.910 (0.009)***	**0.910 (0.009)***	**0.910 (0.009)***	**0.910 (0.009)***	**0.910 (0.009)***	**0.910 (0.009)***	0.864 (0.007)	0.905
Bridges	0.529 (0.052)	0.551 (0.042)	0.516 (0.071)	0.512 (0.097)	0.457 (0.036)	**0.565 (0.030)**	0.545 (0.019)	0.580
Thyroid	0.938 (0.034)	0.947 (0.022)	0.945 (0.021)	0.948 (0.016)	**0.961 (0.000)**	**0.961 (0.000)**	0.959 (0.000)	0.986
				Exact Match Ratio (EMR)				
Music	0.121 (0.022)	0.140 (0.013)	0.128 (0.024)	**0.146 (0.022)**	0.104 (0.013)	0.102 (0.021)	0.078 (0.021)	0.249
Scene	0.219 (0.017)	0.235 (0.017)	0.203 (0.016)	0.219 (0.021)	**0.304 (0.025)**	0.139 (0.035)	0.216 (0.014)	0.519
Yeast	0.071 (0.008)	0.070 (0.007)	0.070 (0.011)	**0.075 (0.009)**	0.057 (0.007)	0.056 (0.006)	0.027 (0.006)	0.117
ENRON	0.003 (0.002)	0.002 (0.002)	0.003 (0.002)	0.004 (0.004)	0.023 (0.004)	**0.028 (0.008)**	0.008 (0.004)	0.050
Medical	0.031 (0.010)	0.041 (0.010)	0.032 (0.006)	0.034 (0.010)	**0.181 (0.021)**	0.166 (0.041)	—	0.625
Slashdot	0.011 (0.008)	0.014 (0.006)	0.012 (0.005)	0.013 (0.007)	**0.145 (0.013)**	0.085 (0.014)	—	0.247
LLOG	0.121 (0.010)	0.121 (0.015)	0.123 (0.017)	0.122 (0.020)	0.164 (0.013)	**0.167 (0.007)**	—	0.205
Solar Flare	**0.794 (0.021)***	0.792 (0.021)*	**0.794 (0.021)***	**0.794 (0.021)***	0.792 (0.020)*	**0.794 (0.021)***	0.791 (0.017)*	0.785
Bridges	0.047 (0.028)	0.040 (0.030)	0.044 (0.035)	0.054 (0.030)	0.016 (0.015)	0.054 (0.015)	**0.068 (0.022)**	0.116
Thyroid	0.673 (0.091)	0.689 (0.084)	0.673 (0.090)	0.673 (0.087)	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.003)**	0.916

Table 6.11: Results using CRS with 20% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.673 (0.026)	0.674 (0.019)	0.676 (0.017)	**0.685 (0.012)**	0.620 (0.014)	0.592 (0.019)	0.581 (0.020)	0.712
Scene	0.773 (0.015)	0.783 (0.008)	0.772 (0.006)	0.774 (0.003)	**0.800 (0.009)**	0.760 (0.004)	0.737 (0.010)	0.845
Yeast	0.680 (0.003)	0.678 (0.005)	0.681 (0.005)	0.681 (0.007)	0.700 (0.006)	**0.701 (0.010)**	0.648 (0.004)	0.724
ENRON	0.909 (0.003)	0.909 (0.002)	0.909 (0.002)	0.909 (0.001)	**0.919 (0.002)**	**0.919 (0.003)**	0.870 (0.003)	0.923
Medical	0.951 (0.001)	0.952 (0.001)	0.951 (0.001)	0.952 (0.001)	0.956 (0.007)	**0.964 (0.005)**	—	0.989
Slashdot	—	—	—	—	—	—	—	—
LLOG	—	—	—	—	—	—	—	—
Solar Flare	0.910 (0.008)*	**0.911 (0.008)***	0.908 (0.009)*	**0.911 (0.008)***	0.910 (0.009)*	**0.911 (0.008)***	0.866 (0.004)	0.905
Bridges	0.492 (0.031)	0.462 (0.052)	0.482 (0.042)	0.450 (0.060)	0.406 (0.027)	**0.511 (0.037)**	0.495 (0.041)	0.570
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.959 (0.000)	0.986
				Exact Match Ratio (EMR)				
Music	0.159 (0.029)	**0.175 (0.019)**	0.161 (0.023)	0.174 (0.019)	0.145 (0.015)	0.124 (0.017)	0.110 (0.020)	0.186
Scene	0.318 (0.041)	0.346 (0.018)	0.315 (0.016)	0.322 (0.008)	**0.391 (0.022)**	0.292 (0.013)	0.323 (0.027)	0.522
Yeast	0.040 (0.008)	0.036 (0.007)	0.039 (0.006)	0.038 (0.008)	0.086 (0.006)	**0.087 (0.009)**	0.049 (0.009)	0.134
ENRON	0.012 (0.005)	0.010 (0.005)	0.013 (0.005)	0.013 (0.004)	0.027 (0.005)*	**0.037 (0.013)***	0.029 (0.011)*	0.026
Medical	0.066 (0.009)	0.063 (0.011)	0.073 (0.016)	0.068 (0.012)	0.202 (0.015)	**0.222 (0.027)**	—	0.645
Slashdot	—	—	—	—	—	—	—	—
LLOG	—	—	—	—	—	—	—	—
Solar Flare	0.792 (0.015)*	0.794 (0.015)*	0.785 (0.024)*	0.794 (0.015)*	0.792 (0.017)*	0.794 (0.015)*	**0.796 (0.010)***	0.785
Bridges	0.030 (0.015)	0.021 (0.022)	0.030 (0.015)	0.026 (0.032)	0.007 (0.011)	0.016 (0.015)	**0.044 (0.020)**	0.163
Thyroid	**0.743 (0.003)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.003)**	0.924

Table 6.12: Results using SCC with 20% of the original features.

6.5 Final Considerations

When we started this research, we didn't know what would be the impact of the extracted features on the multi-target classifiers. So the methodology adopted was that we extract the features at different "levels", based on the number of features contained in the original datasets. It was thought that the predictive performance would decrease as the number of features decreased. For example, when we extracted new features at 90% of the original dataset, we were expecting to have the performance of the classification around 90% in relation to the original result. The idea was to use AEs and RBMs so that we could mitigate this. For example, by extracting new features at 90%, we could achieve a classification performance above 90% of the result obtained with the original dataset.

However, as we can see in the results above (Tables 6.3, 6.4, 6.5, 6.6 and 6.7), by extracting new features at 10% of the number of original features in the dataset, it was possible to obtain even better classification results than the original datasets, in some cases.

Until then, we thought that extracting larger numbers of features would result in even better classification results. However, when evaluating these results (Tables 6.8, 6.9, 6.10, 6.11 and 6.12), we found out that maybe it would be more difficult for the classifiers to predict the correct values as the number of features extracted by the AEs increased. In Figures 6.1, 6.2, 6.3 and 6.4 we show this behavior, as the number of features increased in the LLOG, Music, Yeast and Slashdot datasets.

Based on Figures 6.1, 6.2, 6.3 and 6.4, we also hypothesized that the smaller the space in which the AEs extract the features, the better the predictive potential of the multi-target classifier is. Based on the studies (Hinton and Salakhutdinov, 2006, Zabalza et al., 2016), we believe that AEs have a great potential to acquire the characteristics of the data in its code-layer (as we discussed in Section 3.2), and these characteristics are distributed in the number of features that the output will have. Therefore, when we selected the number of features equal to 10% of the original dataset, these neural networks were able to represent all the characteristics learned from the dataset in a much smaller space. As the results indicate, the multi-target classifiers were able to take better advantage of the characteristics present in a smaller number of features than in larger numbers. On the other hand, we can see that the DBNs were able to better distribute these characteristics in a larger number of features.

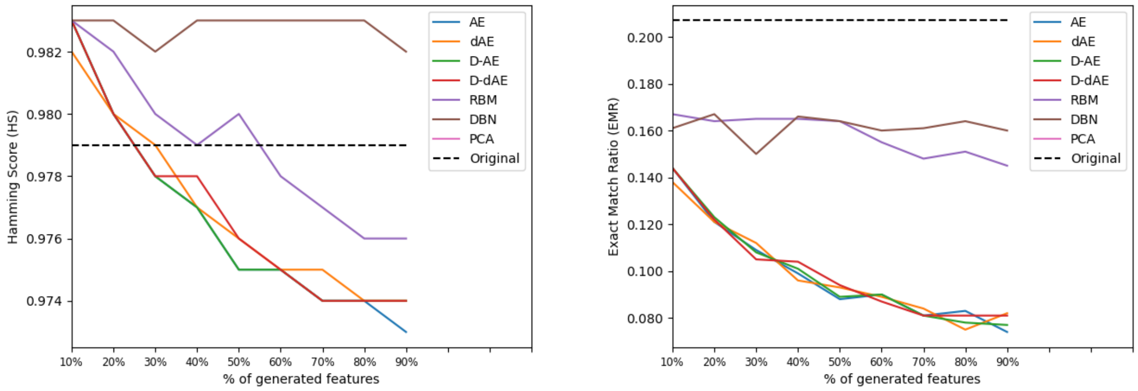
In Figure 6.7 we show how Thyroid reacts with the increase of generated features. We can observe that regardless of the methodology used to extract features, or even the multi-target classifier used, the result of the prediction was always the same. This leads us to believe that in some cases, the feature-extraction depends more on the characteristics of the data in which it is applied, than on the method used to extract features, or even the multi-target classifier.

In general, we can say that the application of neural networks as feature extractors was successful, as we can see in the results obtained in Figures 6.1, 6.2, 6.5 and 6.6, and . We saw that the features extracted by DBN managed to obtain competitive results, and in some cases, even better than executing the multi-target classifiers in the original datasets. The AEs also had shown competitive results when extracted the features at 10%. However, its performance decreased as the number of generated features increased.

In some cases, as can be seen in Figures 6.3, 6.4 and 6.7, the predictive performance obtained with the features extracted by the neural networks were below the results brought by the original datasets. However, in Figures 6.3 and 6.4, we can see that even with lower performances, these results were not distant when compared to the predictive performances of the original datasets. It is worth mentioning that in addition to the predictive performance, there are the benefits obtained from the reduction of dimensionality, as we discussed in Chapter 2. Therefore, depending on the domain that these methods would be applied, the loss of predictive performance may not be so impactful, compared to the problems brought by high dimensionality, since these results were obtained through a portion of the original data.

Finally, in Appendix A we present in details all results that were obtained in our experiments. In Appendix B we also present all graphs that show how the predictive performances of each multi-target classifier was affected according to the number features extracted for each dataset that we selected for our experiments.

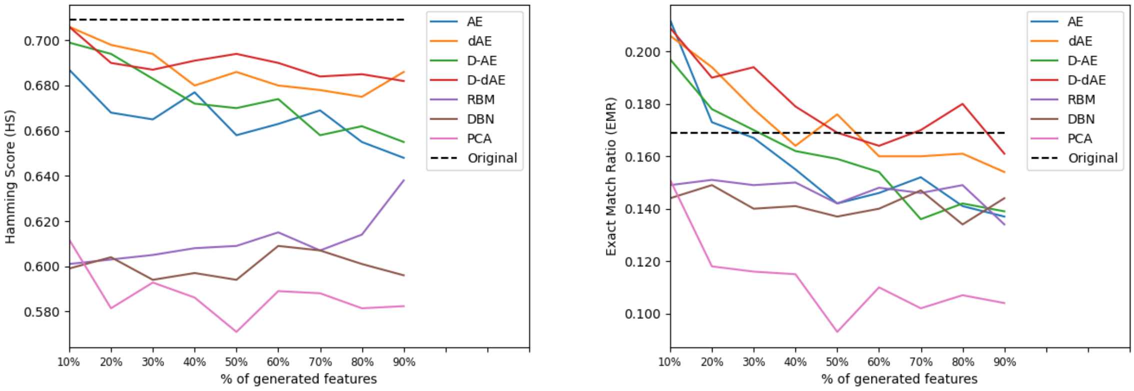
In Chapter 7 we present the conclusions that we obtained from this research, as well as the future works.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

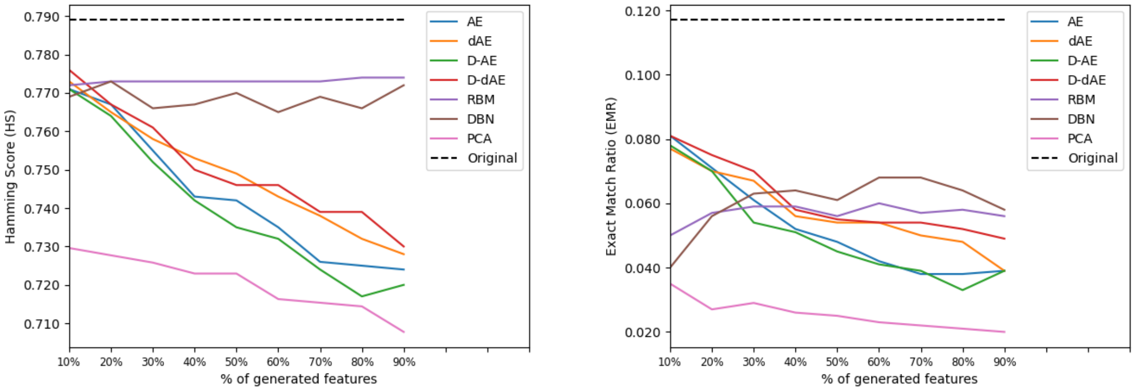
Figure 6.1: How the CR classification performance is affected by the number of extracted features in the LLOG dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

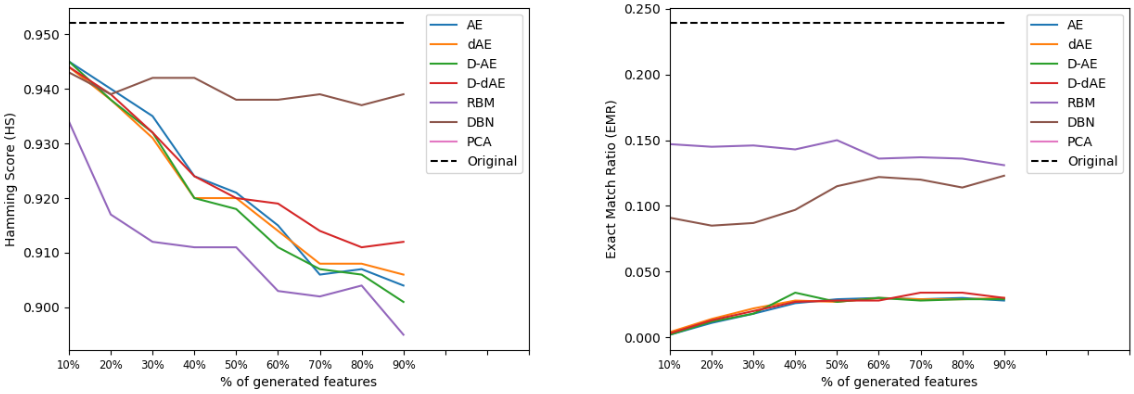
Figure 6.2: How the CC classification performance is affected by the number of extracted features in the Music dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

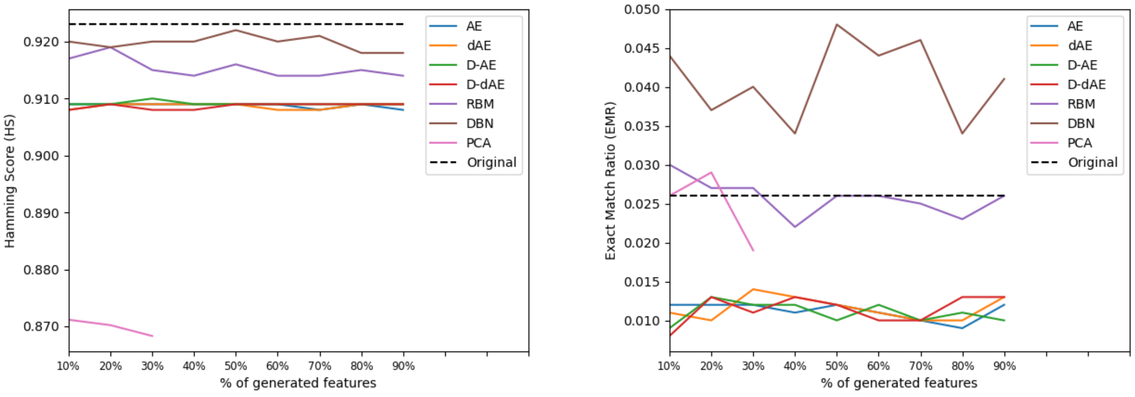
Figure 6.3: How the CRS classification performance is affected by the number of extracted features in the Yeast dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

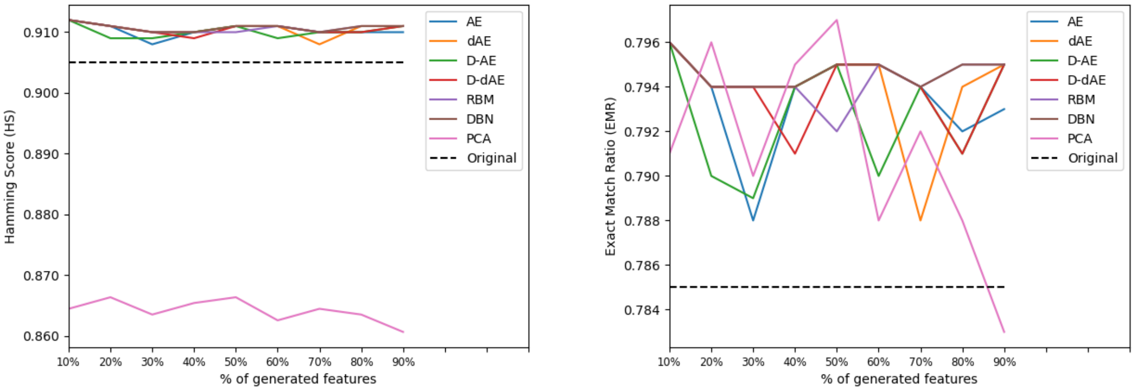
Figure 6.4: How the CR classification performance is affected by the number of extracted features in the Slashdot dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

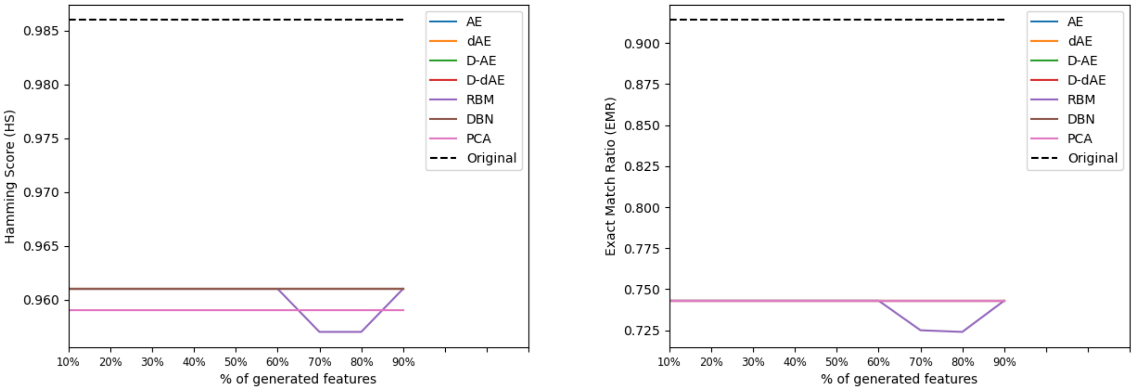
Figure 6.5: How the SCC classification performance is affected by the number of extracted features in the ENRON dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

Figure 6.6: How the CC classification performance is affected by the number of extracted features in the Solar Flare dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

Figure 6.7: How the CR classification performance is affected by the number of extracted features in the Thyroid dataset.

Chapter 7

Conclusions

This research investigated how neural networks, in particular Auto-Encoders (AEs) and Restricted Boltzmann Machines (RBMs) together with some of their deep variations, can be used to extract features from multi-target classification datasets. The objective was to compare the performances of multi-target classifiers when using the original dataset features with their performances when using the same datasets with extracted features. The use of such feature extractors can open new possibilities for the application of state-of-the-art multi-target classifiers.

According to our experiments, we were able to obtain very competitive results using neural networks, keeping the predictive performance even when the extracted features corresponded to a dimension size equivalent to 10% of the original number of features. The Deep Belief Networks (DBNs) were able to obtain better results than the other methods when the evaluation measure takes into account that all targets of a test instance were classified correctly. DBNs were also able to maintain predictive performance as the number of features increased. We also observed that both Auto-Encoders (AEs) and Restricted Boltzmann Machines (RBMs) were able to obtain competitive results when the evaluation measure took into account how many targets in average were correctly predicted. The Denoising Auto-Encoders have also improved predictive performance in some scenarios.

There were several challenges in this research. The first of them was the lack of related works on feature extraction for multi-target problems, which is a very specific niche of research. When we decided to work on this topic, we also faced the lack of multi-target (or multi-dimensional, as the literature also says) publicly available information, in addition to the lack of multi-target datasets. We know that several (if not the majority) real-world problems are, in some form, multi-target. However, many of the proposed methods, such as (Last et al., 2011) and (Read et al., 2014), which are great multi-target implementations, use private data in order to evaluate their proposals.

In the next sections we discuss the contributions we believe this work brings to the multi-target learning field. We also present some future research directions.

7.1 Contributions

In a first moment, even if this research does not come up with a new multi-target classification method, we believe it still can considerably contribute to the multi-target learning literature.

Currently, to the best of our knowledge, there exist no studies in multi-target learning that investigate neural networks as feature extractors in such scenarios, even though these techniques are successfully implemented in other high-dimensional scenarios, as we showed in Chapter 2.

Moreover, it seems we are the first group proposing the investigation of neural networks to build new input representations to be fed to existing multi-target predictors, which also opens the possibility of creating new strategies of multi-target learners.

Lastly, the results obtained by the classifiers using the extracted features were very similar, or were better, than the results obtained when using the original features. Thus, the feature extractors proved to work well with state-of-the-art multi-target learners. This also opens new possibilities and insights on how pre-training or preparing multi-dimensional datasets can improve the performances of the prediction algorithms.

7.2 Future work

As mentioned in Section 4.1.2, there are plenty of algorithm adaptation methods for multi-target regression, and these have shown to be very competitive. Therefore, it seems that this type of approach can be very promising in multi-target classification. Therefore, we are considering creating a new deep-learning multi-target classifier based on how well the neural networks had performed in our experiments. This will certainly be a great contribution to the multi-target research field.

Recently, several researches have appeared in the field of multi-target classification. In one of them (Jia and Zhang, 2020c), an algorithm adaptation method for multi-target classification was presented, which consists of using the k-Nearest Neighbors algorithm to solve the multi-target problem with a single predictive model. Several public datasets have also been adapted to be used for multi-target classification (Jia and Zhang, 2020b). As these are very recent studies, unfortunately we were unable to insert them in our experiments in time. However, we are considering the possibility of evaluating our experiments using these new datasets as well as the new algorithm adaptation method.

Currently, only two evaluation measures (Section 6.1) have been brought from the multi-label literature for the multi-target classification. Despite the study present in (Read et al., 2014) pointing out that these statistics may not be effective in multi-target classification, we believe that there is the possibility of bringing the precision and recall measures, and consequently, F-measure (Sorower, 2010) to evaluate the results obtained in multi-target classification problems. Also, a new evaluation measure specific for multi-target classification was presented in (Jia and

Zhang, 2020c), and it could be used to evaluate future experiments.

Another situation mentioned in Section 3.2.1 is that in datasets where the number of features is very large, extracting features with traditional neural networks as presented in this research is intractable. However, these really high dimensional datasets are only available for multi-target regression. Anyway, deep learning in general has the potential to perform better than traditional neural networks, and therefore, future studies may explore the use of Variational Auto-Encoders (VAEs), Recurrent Neural Networks (RNNs) and specially Long-Short Term Memory (LSTM) in these scenarios.

References

Agarap, A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*. Cited on pages 16 and 37.

An, D., Liu, J., Zhang, M., Chen, X., Chen, M., and Sun, H. (2020). Uncertainty modeling and runtime verification for autonomous vehicles driving control: A machine learning-based approach. *Journal of Systems and Software*, página 110617. Cited on page 4.

Blake, C. L. and Merz, C. J. (1998). Uci repository of machine learning databases [http://www. ics. uci. edu/~ mlearn/mlrepository. html]. irvine, ca: University of california. *Department of Information and Computer Science*, 55. Cited on page 8.

Borchani, H., Varando, G., Bielza, C., and Larrañaga, P. (2015). A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(5):216–233. Cited on page 19.

Briggs, F., Huang, Y., Raich, R., Eftaxias, K., Lei, Z., Cukierski, W., Hadley, S. F., Hadley, A., Betts, M., Fern, X. Z., Irvine, J., Neal, L., Thomas, A., Fodor, G., Tsoumakas, G., Ng, H. W., Nguyen, T. N. T., Huttunen, H., Ruusuvuori, P., Manninen, T., Diment, A., Virtanen, T., Marzat, J., Defretin, J., Callender, D., Hurlburt, C., Larrey, K., and Milakov, M. (2013). The 9th annual mlsp competition: New methods for acoustic classification of multiple simultaneous bird species in a noisy environment. In *2013 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, páginas 1–8. Cited on page 4.

Cerri, R., Barros, R. C., P. L. F. de Carvalho, A. C., and Jin, Y. (2016). Reduction strategies for hierarchical multi-label classification in protein function prediction. *BMC Bioinformatics*, 17(1):373. Cited on page 4.

Cherman, E. A., Monard, M. C., and Metz, J. (2011). Multi-label problem transformation methods: a case study. *CLEI Electronic Journal*, 14(1):4–4. Cited on page 20.

Chollet, F. et al. (2015). Keras. Cited on page 36.

Cottrell, G. W., Munro, P., and Zipser, D. (1987). Learning internal representation from gray-scale images: An example of extensional programming. In *Ninth Annual Conference of the Cognitive Science Society*, páginas 462–473. Cited on pages 4, 5, 11, and 14.

Dembszynski, K., Waegeman, W., Cheng, W., and Hüllermeier, E. (2010). On label dependence in multilabel classification. In *LastCFP: ICML Workshop on Learning from Multi-label data*. Ghent University, KERMIT, Department of Applied Mathematics, Biometrics and .... Cited on page 32.

Džeroski, S., Demšar, D., and Grbović, J. (2000). Predicting chemical parameters of river water quality from bioindicator data. *Applied Intelligence*, 13(1):7–17. Cited on page 4.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org. Cited on pages 13 and 15.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182. Cited on page 9.

Guyon, I. and Elisseeff, A. (2006). An introduction to feature extraction. In *Feature extraction*, páginas 1–25. Springer. Cited on page 10.

Harris, D. and Harris, S. (2010). *Digital design and computer architecture*. Morgan Kaufmann. Cited on page 37.

Higgins, I., Pal, A., Rusu, A., Matthey, L., Burgess, C., Pritzel, A., Botvinick, M., Blundell, C., and Lerchner, A. (2017). Darla: Improving zero-shot transfer in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, páginas 1480–1490. JMLR. org. Cited on page 5.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800. Cited on page 12.

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507. Cited on pages 4, 5, 8, 10, 11, 13, 16, and 53.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*. Cited on page 16.

Jia, B.-B. and Zhang, M.-L. (2020a). Maximum margin multi-dimensional classification. In *AAAI*, páginas 4312–4319. Cited on page 25.

Jia, B.-B. and Zhang, M.-L. (2020b). Md-knn: An instance-based approach for multi-dimensional classification. *dimension*, 10(12):29. Cited on pages 19 and 58.

Jia, B.-B. and Zhang, M.-L. (2020c). Multi-dimensional classification via knn feature augmentation. *Pattern Recognition*, página 107423. Cited on pages 4, 21, 40, 43, and 58.

Jolliffe, I. (2011). Principal component analysis. In *International encyclopedia of statistical science*, páginas 1094–1096. Springer. Cited on page 5.

Kanai, S., Fujiwara, Y., Yamanaka, Y., and Adachi, S. (2018). Sigsoftmax: Reanalysis of the softmax bottleneck. In *Advances in Neural Information Processing Systems*, páginas 286–296. Cited on page 37.

Keller, A., Gerkin, R. C., Guan, Y., Dhurandhar, A., Turu, G., Szalai, B., Mainland, J. D., Ihara, Y., Yu, C. W., Wolfinger, R., et al. (2017). Predicting human olfactory perception from chemical features of odor molecules. *Science*, página eaal2014. Cited on page 4.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. Cited on page 37.

Kocev, D., Vens, C., Struyf, J., and Džeroski, S. (2013). Tree ensembles for predicting structured outputs. *Pattern Recognition*, 46(3):817–833. Cited on page 4.

Lange, S. and Riedmiller, M. (2010). Deep auto-encoder neural networks in reinforcement learning. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, páginas 1–8. IEEE. Cited on pages 4, 5, and 11.

Last, M., Sinaiski, A., and Subramania, H. S. (2011). Condition-based maintenance with multi-target classification models. *New Generation Computing*, 29(3):245. Cited on pages 21, 39, and 57.

LeCun, Y. and Ranzato, M. (2013). Deep learning tutorial. In *Tutorials in International Conference on Machine Learning (ICML'13)*, página 35. Citeseer. Cited on page 16.

Luaces, O., Díez, J., Barranquero, J., del Coz, J. J., and Bahamonde, A. (2012). Binary relevance efficacy for multilabel classification. *Progress in Artificial Intelligence*, 1(4):303–313. Cited on page 22.

Nwankpa, C., Ijomah, W., Gachagan, A., and Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*. Cited on page 37.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830. Cited on pages 35, 37, and 45.

Read, J., Bielza, C., and Larrañaga, P. (2014). Multi-dimensional classification with super-classes. *IEEE Transactions on knowledge and data engineering*, 26(7):1720–1733. Cited on pages xiii, 17, 19, 20, 21, 22, 25, 31, 33, 40, 43, 44, 48, 57, and 58.

Read, J., Pfahringer, B., Holmes, G., and Frank, E. (2011). Classifier chains for multi-label classification. *Machine learning*, 85(3):333–359. Cited on pages 4, 25, and 43.

Read, J., Reutemann, P., Pfahringer, B., and Holmes, G. (2016). Meka: a multi-label/multi-target extension to weka. *The Journal of Machine Learning Research*, 17(1):667–671. Cited on pages 38 and 42.

Reddi, S. J., Kale, S., and Kumar, S. (2019). On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*. Cited on page 37.

Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory. Technical report, COLORADO UNIV AT BOULDER DEPT OF COMPUTER SCIENCE. Cited on pages 4, 5, and 11.

Sorower, M. S. (2010). A literature survey on algorithms for multi-label learning. *Oregon State University, Corvallis*, 18. Cited on pages 25, 39, 40, 41, and 58.

Spyromitros-Xioufis, E., Tsoumakas, G., Groves, W., and Vlahavas, I. (2012). Multi-label classification methods for multi-target regression. *arXiv preprint arXiv*, 1211. Cited on pages 4, 19, and 20.

Spyromitros-Xioufis, E., Tsoumakas, G., Groves, W., and Vlahavas, I. (2016). Multi-target regression via input space expansion: treating targets as inputs. *Machine Learning*, 104(1):55–98. Cited on page 21.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958. Cited on page 37.

Tsanas, A. and Xifara, A. (2012). Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and Buildings*, 49:560 – 567. Cited on page 4.

Tsoumakas, G., Dimou, A., Spyromitros, E., Mezaris, V., Kompatsiaris, I., and Vlahavas, I. (2009a). Correlation-based pruning of stacked binary relevance models for multi-label learning. In *Proceedings of the 1st international workshop on learning from multi-label data*, páginas 101–116. Cited on page 4.

Tsoumakas, G., Dimou, A., Spyromitros, E., Mezaris, V., Kompatsiaris, I., and Vlahavas, I. (2009b). Correlation-based pruning of stacked binary relevance models for multi-label learning. In *Proceedings of the 1st international workshop on learning from multi-label data*, páginas 101–116. Cited on pages 27, 28, and 43.

Tsoumakas, G., Katakis, I., and Vlahavas, I. (2009c). Mining multi-label data. In *Data mining and knowledge discovery handbook*, páginas 667–685. Springer. Cited on page 4.

Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., and Vlahavas, I. (2011). Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, 12(Jul):2411–2414. Cited on page 16.

Tsoumakas, G., Spyromitros-Xioufis, E., and Vlahavas, I. (2014a). Drawing parallels between multi-label classification and multi-target regression. In *ECML PKDD 2014 Workshop on Multi-Target Prediction*. Cited on pages xiii, 19, 23, 26, 29, and 30.

Tsoumakas, G., Spyromitros-Xioufis, E., Vrekou, A., and Vlahavas, I. (2014b). *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part III*, chapter Multi-target Regression via Random Linear Target Combinations, páginas 225–240. Springer Berlin Heidelberg, Berlin, Heidelberg. Cited on page 21.

Van Der Maaten, L., Postma, E., and Van den Herik, J. (2009). Dimensionality reduction: a comparative. *J Mach Learn Res*, 10:66–71. Cited on pages 7 and 8.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, páginas 1096–1103. Cited on page 45.

Wan, S., Mak, M.-W., and Kung, S.-Y. (2012). mgoasvm: Multi-label protein subcellular localization based on gene ontology and support vector machines. *BMC Bioinformatics*, 13(1):290. Cited on page 4.

Yan, Y., Li, S., Yang, Z., Zhang, X., Li, J., Wang, A., and Zhang, J. (2017). Multi-label learning with label-specific feature selection. In *International Conference on Neural Information Processing*, páginas 305–315. Springer. Cited on page 4.

Zabalza, J., Ren, J., Zheng, J., Zhao, H., Qing, C., Yang, Z., Du, P., and Marshall, S. (2016). Novel segmented stacked autoencoder for effective dimensionality reduction and feature extraction in hyperspectral imaging. *Neurocomputing*, 185:1–10. Cited on pages 4, 5, 11, and 53.

Zhang, M.-L. and Zhang, K. (2010). Multi-label learning by exploiting label dependency. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, páginas 999–1008. Cited on page 33.

Zhang, W., Liu, X., Ding, Y., and Shi, D. (2012). Multi-output ls-svr machine in extended feature space. In *Computational Intelligence for Measurement Systems and Applications (CIMSA), 2012 IEEE International Conference on*, páginas 130–134. IEEE. Cited on page 19.

Zhang, Z. (2016). Variable selection with stepwise and best subset approaches. *Annals of translational medicine*, 4(7). Cited on page 9.

Appendix A

Detailed results for Multi-Target Classification using the Extracted Features

This appendix presents all the results obtained during our final experiments, discussed in Section 6.4. The data presented here consist of the combination of all the multi-target classifiers used, the feature extraction methods, and the percentage of features extracted relative to the number of features present in the original dataset.

The results are organized as follows: each column represents the feature extraction method used, and each row represents one of the datasets used. The values present in the tables correspond to the mean obtained for HS and EMR across all the 10 folds from the cross-validation setup (as presented in Chapter 5), and the values in parentheses represent the standard deviation. In bold, we present the feature extraction methods that obtained the best results in each dataset. The last column represents the results obtained with the original dataset, without extracting features. We use an asterisk ($*$) to highlight when a result of a multi-target classifier with extracted features was the same or better than that obtained with the original dataset.

The rest of this appendix is organized as follows: Section A.1 shows the results obtained by Class Relevance (CR), Section A.2 shows the results obtained by Classifier Chains (CC), Section A.3 shows the results obtained by Ensemble of Classifier Chains (ECC), Section A.4 shows the results obtained by Class Relevance Stacking (CRS) and, finally, Section A.5 shows the results obtained by Super-Class Classifier (SCC). The predictive performances of the multi-target classifiers were measured with Hamming Score (HS) and Exact Match Ratio (EMR).

A.1 Class Relevance (CR) Results

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
Hamming Score (HS)								
Music	0.706 (0.021)	**0.712 (0.015)**	0.702 (0.017)	0.705 (0.014)	0.658 (0.010)	0.656 (0.007)	0.638 (0.017)	0.733
Scene	0.788 (0.011)	0.785 (0.014)	0.795 (0.006)	0.790 (0.008)	**0.803 (0.011)**	0.791 (0.015)	0.759 (0.007)	0.857
Yeast	0.768 (0.002)*	0.769 (0.002)*	0.769 (0.001)*	0.769 (0.001)*	**0.772 (0.003)***	0.768 (0.005)*	0.730 (0.003)	0.741
ENRON	0.924 (0.003)	0.924 (0.001)	0.924 (0.002)	0.926 (0.003)	0.928 (0.004)	**0.936 (0.002)***	0.879 (0.002)	0.936
Medical	0.963 (0.003)	0.963 (0.002)	0.963 (0.001)	0.964 (0.002)	0.963 (0.003)	**0.971 (0.002)**	—	0.988
Slashdot	**0.945 (0.001)**	0.944 (0.000)	**0.945 (0.001)**	0.944 (0.001)	0.934 (0.003)	0.943 (0.005)	—	0.952
LLOG	**0.983 (0.001)***	0.982 (0.001)*	**0.983 (0.000)***	**0.983 (0.000)***	**0.983 (0.001)***	**0.983 (0.001)***	—	0.979
Solar Flare	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	0.860 (0.014)	0.905
Bridges	—	—	—	—	—	—	—	0.622
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.959 (0.001)	0.986
Exact Match Ratio (EMR)								
Music	0.149 (0.050)	0.157 (0.030)	0.149 (0.022)	**0.163 (0.021)**	0.100 (0.023)	0.103 (0.022)	0.080 (0.027)	0.165
Scene	0.236 (0.046)	0.253 (0.016)	0.250 (0.011)	0.244 (0.010)	**0.300 (0.018)**	0.158 (0.042)	0.226 (0.022)	0.404
Yeast	0.018 (0.002)	0.018 (0.002)	0.018 (0.002)	0.018 (0.002)	**0.054 (0.008)**	0.046 (0.018)	0.035 (0.009)	0.068
ENRON	0.003 (0.003)	0.003 (0.002)	0.003 (0.002)	0.002 (0.002)	0.026 (0.005)*	**0.030 (0.011)***	0.015 (0.005)	0.023
Medical	0.026 (0.009)	0.020 (0.014)	0.030 (0.012)	0.022 (0.013)	0.164 (0.021)	**0.171 (0.030)**	—	0.615
Slashdot	0.002 (0.003)	0.004 (0.004)	0.002 (0.001)	0.003 (0.001)	**0.147 (0.012)**	0.091 (0.008)	—	0.239
LLOG	0.144 (0.006)	0.138 (0.011)	0.144 (0.007)	0.144 (0.009)	**0.167 (0.007)**	0.161 (0.013)	—	0.207
Solar Flare	**0.796 (0.013)***	**0.796 (0.013)***	**0.796 (0.013)***	**0.796 (0.013)***	**0.796 (0.013)***	**0.796 (0.013)***	0.783 (0.024)	0.785
Bridges	—	—	—	—	—	—	—	0.140
Thyroid	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.003)**	0.914

Table A.1: Results using CR with 10% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
Hamming Score (HS)								
Music	0.693 (0.017)	0.699 (0.024)	0.701 (0.013)	**0.703 (0.017)**	0.660 (0.006)	0.657 (0.010)	0.618 (0.023)	0.733
Scene	0.783 (0.015)	0.786 (0.011)	0.770 (0.010)	0.787 (0.010)	**0.817 (0.016)**	0.781 (0.010)	0.750 (0.007)	0.857
Yeast	0.768 (0.002)*	0.769 (0.001)*	0.769 (0.002)*	0.768 (0.002)*	**0.774 (0.002)***	0.771 (0.006)*	0.728 (0.003)	0.741
ENRON	0.916 (0.002)	0.916 (0.003)	0.916 (0.002)	0.920 (0.003)	0.922 (0.005)	**0.933 (0.002)**	0.875 (0.003)	0.936
Medical	0.955 (0.002)	0.958 (0.004)	0.955 (0.002)	0.956 (0.003)	0.950 (0.008)	**0.969 (0.003)**	—	0.988
Slashdot	**0.940 (0.003)**	0.938 (0.003)	0.938 (0.003)	0.939 (0.003)	0.917 (0.008)	0.939 (0.005)	—	0.952
LLOG	0.980 (0.001)*	0.980 (0.001)*	0.980 (0.001)*	0.980 (0.001)*	0.982 (0.002)*	**0.983 (0.001)***	—	0.979
Solar Flare	**0.911 (0.008)***	**0.911 (0.008)***	0.909 (0.007)*	**0.911 (0.008)***	0.909 (0.007)*	0.902 (0.024)	0.864 (0.007)	0.905
Bridges	0.544 (0.027)	0.544 (0.036)	0.529 (0.044)	0.532 (0.064)	0.470 (0.047)	**0.560 (0.037)**	0.545 (0.019)	0.622
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.959 (0.000)	0.986
Exact Match Ratio (EMR)								
Music	0.111 (0.035)	0.140 (0.013)	0.128 (0.024)	**0.146 (0.022)**	0.104 (0.013)	0.102 (0.021)	0.078 (0.021)	0.165
Scene	0.209 (0.038)	0.235 (0.017)	0.203 (0.016)	0.219 (0.021)	**0.304 (0.025)**	0.139 (0.035)	0.216 (0.014)	0.404
Yeast	0.017 (0.003)	0.018 (0.001)	0.018 (0.002)	0.018 (0.003)	0.056 (0.003)	**0.060 (0.010)**	0.027 (0.006)	0.068
ENRON	0.003 (0.002)	0.002 (0.002)	0.003 (0.002)	0.004 (0.004)	0.023 (0.004)*	**0.028 (0.008)***	0.008 (0.004)	0.023
Medical	0.031 (0.010)	0.041 (0.010)	0.032 (0.006)	0.034 (0.010)	**0.181 (0.021)**	0.166 (0.041)	—	0.615
Slashdot	0.011 (0.008)	0.014 (0.006)	0.012 (0.005)	0.013 (0.007)	**0.145 (0.013)**	0.085 (0.014)	—	0.239
LLOG	0.121 (0.010)	0.121 (0.015)	0.123 (0.017)	0.122 (0.020)	0.164 (0.013)	**0.167 (0.007)**	—	0.207
Solar Flare	**0.794 (0.015)***	**0.794 (0.015)***	0.790 (0.015)*	**0.794 (0.015)***	0.792 (0.013)*	0.770 (0.068)	0.791 (0.017)*	0.785
Bridges	0.040 (0.018)	0.028 (0.020)	0.028 (0.020)	0.038 (0.021)	0.016 (0.023)	0.035 (0.024)	**0.068 (0.022)**	0.140
Thyroid	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.003)**	0.914

Table A.2: Results using CR with 20% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
Hamming Score (HS)								
Music	0.688 (0.019)	0.690 (0.012)	0.682 (0.016)	**0.697 (0.018)**	0.661 (0.009)	0.657 (0.011)	0.620 (0.019)	0.733
Scene	0.777 (0.019)	0.785 (0.014)	0.780 (0.009)	0.778 (0.014)	**0.806 (0.011)**	0.797 (0.019)	0.743 (0.010)	0.857
Yeast	0.768 (0.001)*	0.768 (0.002)*	0.768 (0.001)*	0.769 (0.001)*	**0.773 (0.001)***	0.766 (0.006)*	0.726 (0.003)	0.741
ENRON	0.911 (0.002)	0.911 (0.002)	0.911 (0.002)	0.912 (0.003)	0.921 (0.002)	**0.933 (0.002)**	0.870 (0.003)	0.936
Medical	0.953 (0.001)	0.954 (0.002)	0.951 (0.002)	0.953 (0.002)	0.936 (0.006)	**0.965 (0.005)**	—	0.988
Slashdot	0.935 (0.004)	0.931 (0.005)	0.932 (0.003)	0.932 (0.005)	0.912 (0.013)	**0.942 (0.007)**	—	0.952
LLOG	0.978 (0.001)	0.979 (0.001)*	0.978 (0.001)	0.978 (0.001)	0.980 (0.001)*	**0.982 (0.001)***	—	0.979
Solar Flare	0.908 (0.009)*	**0.910 (0.006)***	0.909 (0.008)*	**0.910 (0.006)***	0.909 (0.006)*	**0.910 (0.006)***	0.863 (0.010)	0.905
Bridges	0.539 (0.054)	0.549 (0.031)	**0.557 (0.030)**	0.517 (0.057)	0.446 (0.024)	0.551 (0.028)	0.538 (0.014)	0.622
Thyroid	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	0.959 (0.000)	0.986
Exact Match Ratio (EMR)								
Music	0.109 (0.041)	**0.140 (0.027)**	0.109 (0.018)	0.136 (0.025)	0.112 (0.017)	0.102 (0.024)	0.066 (0.017)	0.165
Scene	0.195 (0.040)	0.223 (0.014)	0.209 (0.013)	0.218 (0.019)	**0.294 (0.022)**	0.145 (0.047)	0.200 (0.024)	0.404
Yeast	0.018 (0.002)	0.017 (0.002)	0.017 (0.003)	0.018 (0.002)	0.057 (0.004)	**0.061 (0.010)**	0.029 (0.005)	0.068
ENRON	0.001 (0.001)	0.003 (0.002)	0.002 (0.001)	0.002 (0.002)	0.019 (0.005)	**0.030 (0.011)***	0.004 (0.002)	0.023
Medical	0.032 (0.009)	0.033 (0.008)	0.029 (0.010)	0.033 (0.007)	**0.190 (0.013)**	0.156 (0.039)	—	0.615
Slashdot	0.018 (0.008)	0.022 (0.006)	0.018 (0.006)	0.020 (0.008)	**0.146 (0.012)**	0.087 (0.007)	—	0.239
LLOG	0.109 (0.010)	0.112 (0.010)	0.108 (0.013)	0.105 (0.011)	**0.165 (0.013)**	0.150 (0.023)	—	0.207
Solar Flare	0.788 (0.024)*	**0.794 (0.012)***	0.789 (0.022)*	**0.794 (0.012)***	0.792 (0.013)*	**0.794 (0.012)***	0.788 (0.026)*	0.785
Bridges	**0.054 (0.021)**	0.030 (0.024)	0.040 (0.018)	0.033 (0.028)	0.019 (0.018)	0.019 (0.018)	0.044 (0.022)	0.140
Thyroid	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	0.914

Table A.3: Results using CR with 30% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
Hamming Score (HS)								
Music	0.676 (0.023)	**0.694 (0.015)**	0.673 (0.009)	0.686 (0.015)	0.659 (0.014)	0.659 (0.013)	0.617 (0.021)	0.733
Scene	0.772 (0.016)	0.781 (0.012)	0.781 (0.015)	0.786 (0.015)	**0.813 (0.016)**	0.786 (0.007)	0.741 (0.009)	0.857
Yeast	0.767 (0.001)*	0.768 (0.002)*	0.768 (0.002)*	0.766 (0.003)*	**0.773 (0.002)***	0.766 (0.006)*	0.723 (0.003)	0.741
ENRON	0.909 (0.003)	0.908 (0.001)	0.908 (0.002)	0.909 (0.002)	0.921 (0.004)	**0.932 (0.003)**	—	0.936
Medical	0.951 (0.002)	0.953 (0.002)	0.951 (0.002)	0.952 (0.001)	0.962 (0.020)	**0.972 (0.000)**	—	0.988
Slashdot	0.924 (0.004)	0.920 (0.005)	0.920 (0.005)	0.924 (0.004)	0.911 (0.011)	**0.942 (0.006)**	—	0.952
LLOG	0.977 (0.001)	0.977 (0.001)	0.977 (0.001)	0.978 (0.001)	0.979 (0.001)*	**0.983 (0.001)***	—	0.979
Solar Flare	**0.910 (0.008)***	**0.910 (0.008)***	**0.910 (0.008)***	**0.910 (0.008)***	**0.910 (0.008)***	**0.910 (0.008)***	0.865 (0.007)	0.905
Bridges	0.520 (0.053)	0.528 (0.056)	**0.545 (0.053)**	0.529 (0.032)	0.444 (0.049)	0.536 (0.032)	0.529 (0.025)	0.622
Thyroid	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	0.959 (0.000)	0.986
Exact Match Ratio (EMR)								
Music	0.106 (0.039)	0.121 (0.020)	0.108 (0.018)	**0.128 (0.023)**	0.107 (0.020)	0.103 (0.022)	0.068 (0.015)	0.165
Scene	0.193 (0.032)	0.234 (0.017)	0.204 (0.016)	0.221 (0.016)	**0.294 (0.028)**	0.140 (0.023)	0.196 (0.019)	0.404
Yeast	0.017 (0.002)	0.018 (0.002)	0.017 (0.002)	0.016 (0.004)	0.058 (0.007)	**0.070 (0.011)***	0.026 (0.005)	0.068
ENRON	0.002 (0.002)	0.002 (0.002)	0.002 (0.001)	0.003 (0.002)	0.017 (0.005)	**0.025 (0.008)***	—	0.023
Medical	0.025 (0.007)	0.030 (0.008)	0.032 (0.007)	0.029 (0.007)	**0.206 (0.022)**	0.069 (0.077)	—	0.615
Slashdot	0.026 (0.009)	0.028 (0.006)	0.034 (0.006)	0.027 (0.007)	**0.143 (0.009)**	0.097 (0.013)	—	0.239
LLOG	0.099 (0.012)	0.096 (0.013)	0.101 (0.011)	0.104 (0.009)	0.165 (0.014)	**0.166 (0.007)**	—	0.207
Solar Flare	0.794 (0.018)*	0.794 (0.018)*	0.794 (0.018)*	0.794 (0.018)*	0.794 (0.018)*	0.794 (0.018)*	**0.795 (0.016)***	0.785
Bridges	0.049 (0.022)	0.047 (0.028)	**0.051 (0.027)**	0.044 (0.024)	0.009 (0.011)	0.037 (0.016)	0.040 (0.026)	0.140
Thyroid	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	0.914

Table A.4: Results using CR with 40% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.668 (0.016)	**0.691 (0.018)**	0.680 (0.022)	**0.691 (0.024)**	0.660 (0.010)	0.664 (0.010)	0.602 (0.024)	0.733
Scene	0.766 (0.019)	0.781 (0.016)	0.777 (0.015)	0.781 (0.009)	**0.814 (0.020)**	0.801 (0.013)	0.738 (0.007)	0.857
Yeast	0.766 (0.003)*	0.767 (0.002)*	0.765 (0.005)*	0.767 (0.002)*	**0.774 (0.001)***\	0.767 (0.008)*	0.723 (0.003)	0.741
ENRON	0.908 (0.001)	0.907 (0.002)	0.907 (0.002)	0.907 (0.002)	0.919 (0.003)	**0.932 (0.004)**	—	0.936
Medical	0.951 (0.002)	0.951 (0.001)	0.951 (0.002)	0.952 (0.001)	0.964 (0.014)	**0.970 (0.004)**	—	0.988
Slashdot	0.921 (0.007)	0.920 (0.004)	0.918 (0.008)	0.920 (0.006)	0.911 (0.010)	**0.938 (0.003)**	—	0.952
LLOG	0.975 (0.001)	0.976 (0.001)	0.975 (0.002)	0.976 (0.001)	0.980 (0.002)*	**0.983 (0.001)***\	—	0.979
Solar Flare	**0.911 (0.005)***\	**0.911 (0.005)***\	**0.911 (0.005)***\	**0.911 (0.005)***\	0.909 (0.007)*	**0.911 (0.005)***\	0.865 (0.007)	0.905
Bridges	**0.555 (0.034)**	0.520 (0.037)	0.517 (0.041)	0.518 (0.042)	0.463 (0.034)	0.550 (0.043)	0.519 (0.025)	0.622
Thyroid	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	0.959 (0.000)	0.986
				Exact Match Ratio (EMR)				
Music	0.094 (0.027)	0.124 (0.021)	0.111 (0.022)	**0.135 (0.026)**	0.106 (0.016)	0.092 (0.028)	0.062 (0.020)	0.165
Scene	0.189 (0.033)	0.219 (0.013)	0.203 (0.017)	0.216 (0.018)	**0.294 (0.029)**	0.152 (0.023)	0.195 (0.015)	0.404
Yeast	0.017 (0.002)	0.017 (0.002)	0.016 (0.003)	0.018 (0.002)	0.057 (0.004)	**0.060 (0.009)**	0.025 (0.006)	0.068
ENRON	0.002 (0.001)	0.002 (0.001)	0.002 (0.002)	0.002 (0.001)	0.019 (0.005)	**0.035 (0.012)***\	—	0.023
Medical	0.035 (0.007)	0.030 (0.007)	0.032 (0.010)	0.031 (0.006)	**0.206 (0.017)**	0.056 (0.067)	—	0.615
Slashdot	0.029 (0.007)	0.027 (0.005)	0.027 (0.006)	0.028 (0.007)	**0.150 (0.014)**	0.115 (0.010)	—	0.239
LLOG	0.088 (0.012)	0.093 (0.009)	0.089 (0.013)	0.094 (0.008)	**0.164 (0.018)**	**0.164 (0.013)**	—	0.207
Solar Flare	**0.795 (0.010)***\	**0.795 (0.010)***\	**0.795 (0.010)***\	**0.795 (0.010)***\	0.792 (0.013)*	**0.795 (0.010)***\	0.794 (0.015)*	0.785
Bridges	**0.049 (0.022)**	0.021 (0.016)	0.042 (0.020)	0.028 (0.018)	0.016 (0.015)	0.030 (0.024)	0.035 (0.019)	0.140
Thyroid	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	0.914

Table A.5: Results using CR with 50% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.671 (0.022)	0.687 (0.015)	0.679 (0.019)	**0.697 (0.013)**	0.662 (0.007)	0.679 (0.011)	0.609 (0.012)	0.733
Scene	0.770 (0.014)	0.781 (0.021)	0.777 (0.011)	0.779 (0.016)	**0.815 (0.019)**	0.803 (0.017)	0.738 (0.008)	0.857
Yeast	0.767 (0.003)*	0.765 (0.004)*	0.765 (0.003)*	0.766 (0.004)*	**0.773 (0.003)***\	0.769 (0.005)*	0.716 (0.005)	0.741
ENRON	0.907 (0.002)	0.905 (0.002)	0.906 (0.002)	0.906 (0.002)	0.919 (0.004)	**0.932 (0.003)**	—	0.936
Medical	0.951 (0.001)	0.950 (0.001)	0.950 (0.001)	0.951 (0.001)	0.957 (0.016)	**0.972 (0.001)**	—	0.988
Slashdot	0.915 (0.007)	0.914 (0.006)	0.911 (0.005)	0.919 (0.005)	0.903 (0.010)	**0.938 (0.003)**	—	0.952
LLOG	0.975 (0.001)	0.975 (0.001)	0.975 (0.001)	0.975 (0.001)	0.978 (0.002)	**0.983 (0.001)***\	—	0.979
Solar Flare	**0.911 (0.004)***\	**0.911 (0.004)***\	0.909 (0.007)*	**0.911 (0.004)***\	0.909 (0.003)*	**0.911 (0.004)***\	0.863 (0.010)	0.905
Bridges	**0.548 (0.024)**	0.539 (0.028)	0.534 (0.045)	0.545 (0.041)	0.481 (0.036)	0.544 (0.041)	0.508 (0.021)	0.622
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.959 (0.000)	0.986
				Exact Match Ratio (EMR)				
Music	0.096 (0.040)	0.137 (0.024)	0.110 (0.023)	**0.147 (0.020)**	0.112 (0.015)	0.097 (0.010)	0.069 (0.018)	0.165
Scene	0.194 (0.033)	0.228 (0.016)	0.218 (0.020)	0.221 (0.007)	**0.289 (0.026)**	0.144 (0.023)	0.193 (0.017)	0.404
Yeast	0.017 (0.003)	0.016 (0.004)	0.016 (0.003)	0.017 (0.002)	0.058 (0.003)	**0.063 (0.011)**	0.023 (0.006)	0.068
ENRON	0.001 (0.002)	0.002 (0.001)	0.002 (0.002)	0.002 (0.002)	0.016 (0.006)	**0.029 (0.009)***\	—	0.023
Medical	0.032 (0.010)	0.025 (0.010)	0.029 (0.010)	0.033 (0.007)	**0.201 (0.021)**	0.048 (0.068)	—	0.615
Slashdot	0.030 (0.005)	0.030 (0.004)	0.030 (0.004)	0.028 (0.005)	**0.136 (0.005)**	0.122 (0.009)	—	0.239
LLOG	0.090 (0.012)	0.089 (0.007)	0.090 (0.011)	0.087 (0.009)	0.155 (0.018)	**0.160 (0.015)**	—	0.207
Solar Flare	**0.795 (0.010)***\	**0.795 (0.010)***\	0.790 (0.015)*	**0.795 (0.010)***\	0.792 (0.009)*	**0.795 (0.010)***\	0.786 (0.018)*	0.785
Bridges	0.033 (0.022)	0.033 (0.024)	0.040 (0.024)	**0.042 (0.023)**	0.019 (0.018)	0.019 (0.023)	0.040 (0.030)	0.140
Thyroid	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	0.914

Table A.6: Results using CR with 60% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
Hamming Score (HS)								
Music	0.666 (0.019)	0.683 (0.018)	0.662 (0.017)	0.681 (0.023)	0.663 (0.005)	**0.688 (0.008)**	0.596 (0.022)	0.733
Scene	0.778 (0.016)	0.785 (0.013)	0.770 (0.014)	0.773 (0.015)	**0.809 (0.011)**	0.801 (0.020)	0.740 (0.007)	0.857
Yeast	0.764 (0.003)*	0.765 (0.003)*	0.764 (0.004)*	0.766 (0.002)*	**0.774 (0.001)***	0.769 (0.004)*	0.715 (0.006)	0.741
ENRON	0.904 (0.002)	0.906 (0.002)	0.907 (0.001)	0.905 (0.002)	0.921 (0.004)	**0.933 (0.001)**	—	0.936
Medical	0.950 (0.002)	0.950 (0.002)	0.950 (0.002)	0.951 (0.001)	0.956 (0.011)	**0.970 (0.004)**	—	0.988
Slashdot	0.906 (0.004)	0.908 (0.004)	0.907 (0.005)	0.914 (0.005)	0.902 (0.012)	**0.939 (0.000)**	—	0.952
LLOG	0.974 (0.001)	0.975 (0.001)	0.974 (0.001)	0.974 (0.001)	0.977 (0.003)	**0.983 (0.001)***	—	0.979
Solar Flare	**0.910 (0.005)***	0.908 (0.005)*	**0.910 (0.005)***	**0.910 (0.005)***	0.906 (0.012)*	**0.910 (0.005)***	0.863 (0.008)	0.905
Bridges	0.557 (0.025)	0.548 (0.020)	0.545 (0.035)	**0.560 (0.020)**	0.491 (0.035)	0.558 (0.032)	0.482 (0.019)	0.622
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.957 (0.010)	**0.961 (0.000)**	0.959 (0.000)	0.986
Exact Match Ratio (EMR)								
Music	0.102 (0.036)	0.126 (0.022)	0.092 (0.014)	**0.127 (0.026)**	0.110 (0.011)	0.094 (0.014)	0.054 (0.015)	0.165
Scene	0.201 (0.032)	0.239 (0.018)	0.207 (0.012)	0.217 (0.015)	**0.300 (0.015)**	0.160 (0.025)	0.187 (0.016)	0.404
Yeast	0.016 (0.003)	0.016 (0.003)	0.016 (0.002)	0.015 (0.002)	0.058 (0.005)	**0.059 (0.010)**	0.022 (0.006)	0.068
ENRON	0.002 (0.002)	0.002 (0.002)	0.002 (0.001)	0.001 (0.001)	0.016 (0.004)	**0.028 (0.010)***	—	0.023
Medical	0.034 (0.006)	0.031 (0.005)	0.030 (0.007)	0.031 (0.008)	**0.207 (0.018)**	0.058 (0.079)	—	0.615
Slashdot	0.029 (0.004)	0.029 (0.003)	0.028 (0.005)	0.034 (0.005)	**0.137 (0.008)**	0.120 (0.008)	—	0.239
LLOG	0.081 (0.013)	0.084 (0.009)	0.081 (0.009)	0.081 (0.011)	0.148 (0.018)	**0.161 (0.013)**	—	0.207
Solar Flare	**0.794 (0.008)***	0.788 (0.009)*	**0.794 (0.008)***	**0.794 (0.008)***	0.783 (0.032)	**0.794 (0.008)***	0.788 (0.018)*	0.785
Bridges	**0.056 (0.019)**	0.040 (0.024)	0.044 (0.013)	0.049 (0.022)	0.026 (0.016)	0.037 (0.016)	0.028 (0.018)	0.140
Thyroid	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	0.725 (0.056)	**0.743 (0.003)**	**0.743 (0.001)**	0.914

Table A.7: Results using CR with 70% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
Hamming Score (HS)								
Music	0.667 (0.024)	0.687 (0.018)	0.665 (0.016)	**0.689 (0.013)**	0.661 (0.014)	0.687 (0.007)	0.601 (0.017)	0.733
Scene	0.769 (0.018)	0.785 (0.013)	0.779 (0.020)	0.774 (0.014)	**0.820 (0.012)**	0.800 (0.027)	0.739 (0.007)	0.857
Yeast	0.762 (0.004)*	0.765 (0.004)*	0.763 (0.003)*	0.763 (0.002)*	**0.773 (0.002)***	0.769 (0.007)*	0.714 (0.004)	0.741
ENRON	0.906 (0.002)	0.904 (0.002)	0.906 (0.002)	0.905 (0.002)	0.918 (0.005)	**0.933 (0.002)**	—	0.936
Medical	0.950 (0.002)	0.949 (0.002)	0.950 (0.001)	0.950 (0.002)	0.949 (0.015)	**0.970 (0.003)**	—	0.988
Slashdot	0.907 (0.003)	0.908 (0.005)	0.906 (0.004)	0.911 (0.005)	0.904 (0.013)	**0.937 (0.004)**	—	0.952
LLOG	0.974 (0.001)	0.974 (0.001)	0.974 (0.001)	0.974 (0.001)	0.976 (0.002)	**0.983 (0.001)***	—	0.979
Solar Flare	0.910 (0.007)*	**0.911 (0.006)***	0.910 (0.007)*	0.909 (0.008)*	0.907 (0.008)*	**0.911 (0.006)***	0.856 (0.012)	0.905
Bridges	0.539 (0.022)	0.515 (0.037)	0.537 (0.025)	0.516 (0.052)	0.480 (0.031)	**0.552 (0.036)**	0.492 (0.042)	0.622
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.957 (0.010)	**0.961 (0.000)**	0.959 (0.000)	0.986
Exact Match Ratio (EMR)								
Music	0.091 (0.030)	0.125 (0.014)	0.094 (0.020)	**0.138 (0.018)**	0.112 (0.021)	0.080 (0.027)	0.067 (0.014)	0.165
Scene	0.198 (0.032)	0.224 (0.018)	0.210 (0.019)	0.219 (0.020)	**0.290 (0.019)**	0.143 (0.022)	0.188 (0.016)	0.404
Yeast	0.016 (0.004)	0.016 (0.002)	0.016 (0.003)	0.015 (0.003)	0.058 (0.004)	**0.059 (0.010)**	0.021 (0.005)	0.068
ENRON	0.002 (0.002)	0.002 (0.002)	0.002 (0.001)	0.002 (0.003)	0.016 (0.006)	**0.022 (0.007)**	—	0.023
Medical	0.030 (0.010)	0.030 (0.007)	0.028 (0.006)	0.030 (0.005)	**0.208 (0.019)**	0.104 (0.087)	—	0.615
Slashdot	0.030 (0.004)	0.029 (0.006)	0.029 (0.004)	0.034 (0.006)	**0.136 (0.008)**	0.114 (0.016)	—	0.239
LLOG	0.083 (0.010)	0.075 (0.005)	0.078 (0.011)	0.081 (0.007)	0.151 (0.018)	**0.164 (0.012)**	—	0.207
Solar Flare	0.792 (0.014)*	0.794 (0.012)*	0.791 (0.017)*	0.790 (0.018)*	0.784 (0.022)	**0.795 (0.012)***	0.773 (0.019)	0.785
Bridges	0.021 (0.016)	0.012 (0.016)	0.033 (0.026)	0.033 (0.030)	0.028 (0.014)	**0.047 (0.026)**	0.042 (0.021)	0.140
Thyroid	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	0.724 (0.056)	**0.743 (0.002)**	**0.743 (0.002)**	0.914

Table A.8: Results using CR with 80% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.656 (0.022)	0.682 (0.025)	0.648 (0.013)	**0.699 (0.018)**	0.687 (0.005)	0.653 (0.008)	0.590 (0.018)	0.733
Scene	0.768 (0.015)	0.767 (0.021)	0.776 (0.015)	0.775 (0.012)	**0.813 (0.013)**	0.787 (0.015)	0.735 (0.010)	0.857
Yeast	0.761 (0.005)*	0.764 (0.004)*	0.763 (0.003)*	0.762 (0.003)*	**0.772 (0.002)***	0.768 (0.005)*	0.708 (0.009)	0.741
ENRON	0.904 (0.002)	0.904 (0.002)	0.905 (0.003)	0.905 (0.001)	0.919 (0.006)	**0.930 (0.004)**	—	0.936
Medical	0.949 (0.001)	0.949 (0.002)	0.950 (0.001)	0.951 (0.001)	0.942 (0.012)	**0.970 (0.003)**	—	0.988
Slashdot	0.904 (0.004)	0.906 (0.004)	0.901 (0.003)	0.912 (0.007)	0.895 (0.012)	**0.939 (0.001)**	—	0.952
LLOG	0.973 (0.001)	0.974 (0.001)	0.974 (0.001)	0.974 (0.001)	0.976 (0.003)	**0.982 (0.002)***	—	0.979
Solar Flare	0.910 (0.005)*	**0.911 (0.004)***	**0.911 (0.004)***	**0.911 (0.004)***	0.901 (0.021)	**0.911 (0.004)***	0.856 (0.014)	0.905
Bridges	0.532 (0.033)	0.545 (0.023)	0.504 (0.043)	0.516 (0.037)	0.506 (0.027)	**0.553 (0.038)**	0.499 (0.055)	0.622
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.959 (0.001)	0.986
				Exact Match Ratio (EMR)				
Music	0.087 (0.023)	0.114 (0.027)	0.098 (0.020)	**0.133 (0.034)**	0.088 (0.020)	0.114 (0.016)	0.056 (0.014)	0.165
Scene	0.192 (0.027)	0.215 (0.017)	0.208 (0.015)	0.206 (0.011)	**0.291 (0.024)**	0.130 (0.025)	0.188 (0.022)	0.404
Yeast	0.015 (0.004)	0.016 (0.003)	0.015 (0.003)	0.013 (0.003)	0.057 (0.006)	**0.062 (0.008)**	0.020 (0.006)	0.068
ENRON	0.001 (0.001)	0.002 (0.003)	0.002 (0.002)	0.001 (0.001)	0.017 (0.004)	**0.027 (0.009)***	—	0.023
Medical	0.029 (0.006)	0.027 (0.008)	0.029 (0.005)	0.032 (0.006)	**0.211 (0.027)**	0.111 (0.075)	—	0.615
Slashdot	0.028 (0.004)	0.030 (0.005)	0.029 (0.002)	0.030 (0.004)	**0.131 (0.005)**	0.123 (0.005)	—	0.239
LLOG	0.074 (0.008)	0.082 (0.005)	0.077 (0.007)	0.081 (0.010)	0.145 (0.014)	**0.160 (0.017)**	—	0.207
Solar Flare	0.793 (0.007)*	**0.795 (0.007)***	**0.795 (0.007)***	**0.795 (0.007)***	0.768 (0.061)	**0.795 (0.007)***	0.772 (0.029)	0.785
Bridges	0.044 (0.028)	**0.049 (0.028)**	0.035 (0.024)	0.023 (0.028)	0.033 (0.019)	0.047 (0.023)	0.040 (0.031)	0.140
Thyroid	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	0.914

Table A.9: Results using CR with 90% of the original features.

A.2 Classifier Chains (CC) Results

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.687 (0.058)	**0.706 (0.012)**	0.699 (0.020)	**0.706 (0.012)**	0.601 (0.011)	0.599 (0.017)	0.612 (0.020)	0.709
Scene	0.785 (0.016)	0.788 (0.010)	0.792 (0.007)	0.787 (0.004)	**0.807 (0.005)**	0.789 (0.012)	0.753 (0.008)	0.853
Yeast	0.661 (0.017)	0.661 (0.022)	0.673 (0.019)	0.651 (0.021)	**0.678 (0.016)**	0.677 (0.005)	0.643 (0.008)	0.735
ENRON	0.915 (0.001)	0.915 (0.002)	0.917 (0.002)	0.917 (0.002)	0.926 (0.004)	**0.932 (0.002)**	0.878 (0.002)	0.935
Medical	0.961 (0.001)	0.962 (0.002)	0.961 (0.002)	0.962 (0.001)	0.962 (0.004)	**0.967 (0.003)**	—	0.988
Slashdot	0.908 (0.001)	0.907 (0.002)	0.907 (0.001)	0.907 (0.001)	**0.921 (0.004)**	0.920 (0.003)	—	0.938
LLOG	0.980 (0.002)*	0.981 (0.001)*	0.981 (0.001)*	**0.982 (0.002)***	0.980 (0.003)*	0.978 (0.003)	—	0.980
Solar Flare	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	0.864 (0.008)	0.905
Bridges	—	—	—	—	—	—	—	0.622
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.959 (0.001)	0.987
				Exact Match Ratio (EMR)				
Music	**0.212 (0.038)***	0.206 (0.022)*	0.197 (0.030)*	0.209 (0.020)*	0.149 (0.015)	0.144 (0.017)	0.151 (0.028)	0.169
Scene	0.350 (0.042)	0.360 (0.025)	0.366 (0.020)	0.355 (0.014)	**0.408 (0.016)**	0.357 (0.026)	0.366 (0.023)	0.523
Yeast	0.012 (0.007)	0.018 (0.011)	0.012 (0.006)	0.011 (0.004)	0.059 (0.015)	0.056 (0.005)	**0.060 (0.007)**	0.132
ENRON	0.013 (0.005)	0.016 (0.003)	0.019 (0.006)	0.014 (0.006)	0.036 (0.008)	**0.046 (0.012)**	0.035 (0.008)	0.048
Medical	0.046 (0.020)	0.040 (0.013)	0.053 (0.014)	0.053 (0.020)	0.198 (0.031)	**0.205 (0.031)**	—	0.630
Slashdot	0.061 (0.006)	0.059 (0.009)	0.057 (0.003)	0.058 (0.008)	**0.192 (0.017)**	0.163 (0.011)	—	0.309
LLOG	0.124 (0.018)	0.126 (0.014)	0.123 (0.016)	0.133 (0.018)	**0.186 (0.020)**	0.181 (0.016)	—	0.214
Solar Flare	**0.796 (0.013)***	**0.796 (0.013)***	**0.796 (0.013)***	**0.796 (0.013)***	**0.796 (0.013)***	**0.796 (0.013)***	0.791 (0.017)*	0.785
Bridges	—	—	—	—	—	—	—	0.140
Thyroid	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.003)**	0.920

Table A.10: Results using CC with 10% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.668 (0.052)	**0.698 (0.015)**	0.694 (0.013)	0.690 (0.018)	0.603 (0.012)	0.604 (0.013)	0.581 (0.019)	0.709
Scene	0.772 (0.017)	0.784 (0.006)	0.774 (0.009)	0.780 (0.008)	**0.810 (0.005)**	0.782 (0.010)	0.743 (0.008)	0.853
Yeast	0.653 (0.021)	0.662 (0.023)	0.654 (0.015)	0.658 (0.021)	0.668 (0.013)	**0.676 (0.008)**	0.646 (0.014)	0.735
ENRON	0.914 (0.001)	0.914 (0.002)	0.914 (0.002)	0.915 (0.001)	0.926 (0.005)	**0.930 (0.002)**	0.874 (0.002)	0.935
Medical	0.958 (0.001)	0.959 (0.001)	0.958 (0.002)	0.957 (0.002)	0.952 (0.006)	**0.968 (0.003)**	—	0.988
Slashdot	0.905 (0.001)	0.906 (0.001)	0.906 (0.001)	0.905 (0.001)	0.911 (0.008)	**0.920 (0.001)**	—	0.938
LLOG	0.978 (0.001)	**0.979 (0.001)**	**0.979 (0.001)**	**0.979 (0.001)**	0.978 (0.003)	0.978 (0.003)	—	0.980
Solar Flare	**0.911 (0.008)***	**0.911 (0.008)***	0.909 (0.007)*	**0.911 (0.008)***	**0.911 (0.008)***	**0.911 (0.008)***	0.866 (0.004)	0.905
Bridges	0.557 (0.037)	**0.570 (0.033)**	0.550 (0.045)	0.532 (0.059)	0.506 (0.061)	0.561 (0.035)	0.538 (0.019)	0.622
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.959 (0.000)	0.987
				Exact Match Ratio (EMR)				
Music	0.173 (0.025)*	**0.194 (0.019)***	0.178 (0.015)*	0.190 (0.018)*	0.151 (0.012)	0.149 (0.014)	0.118 (0.018)	0.169
Scene	0.318 (0.047)	0.349 (0.017)	0.319 (0.021)	0.337 (0.021)	**0.410 (0.012)**	0.343 (0.022)	0.337 (0.025)	0.523
Yeast	0.013 (0.007)	0.016 (0.010)	0.015 (0.005)	0.015 (0.008)	0.054 (0.005)	0.058 (0.011)	**0.059 (0.011)**	0.132
ENRON	0.013 (0.003)	0.015 (0.002)	0.014 (0.004)	0.015 (0.004)	**0.042 (0.010)**	0.040 (0.012)	0.034 (0.011)	0.048
Medical	0.058 (0.008)	0.062 (0.014)	0.059 (0.011)	0.064 (0.007)	**0.201 (0.022)**	0.200 (0.053)	—	0.630
Slashdot	0.061 (0.008)	0.064 (0.010)	0.062 (0.009)	0.057 (0.008)	**0.188 (0.018)**	0.160 (0.008)	—	0.309
LLOG	0.107 (0.014)	0.111 (0.013)	0.112 (0.017)	0.110 (0.010)	**0.195 (0.014)**	0.185 (0.014)	—	0.214
Solar Flare	0.794 (0.015)*	0.794 (0.015)*	0.790 (0.015)*	0.794 (0.015)*	0.794 (0.015)*	0.794 (0.015)*	**0.796 (0.010)***	0.785
Bridges	0.051 (0.025)	0.044 (0.016)	0.047 (0.015)	0.035 (0.024)	0.016 (0.021)	0.030 (0.026)	**0.063 (0.015)**	0.140
Thyroid	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.003)**	0.920

Table A.11: Results using CC with 20% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
Hamming Score (HS)								
Music	0.665 (0.044)	**0.694 (0.015)**	0.683 (0.024)	0.687 (0.012)	0.605 (0.015)	0.594 (0.013)	0.593 (0.016)	0.709
Scene	0.768 (0.016)	0.781 (0.004)	0.776 (0.009)	0.778 (0.007)	**0.805 (0.006)**	0.790 (0.014)	0.741 (0.007)	0.853
Yeast	0.665 (0.007)	0.657 (0.016)	0.658 (0.014)	0.656 (0.016)	0.667 (0.014)	**0.682 (0.012)**	0.647 (0.010)	0.735
ENRON	0.913 (0.002)	0.913 (0.002)	0.912 (0.002)	0.913 (0.001)	0.923 (0.003)	**0.930 (0.002)**	0.871 (0.003)	0.935
Medical	0.957 (0.002)	0.955 (0.002)	0.955 (0.002)	0.957 (0.001)	0.937 (0.008)	**0.964 (0.005)**	—	0.988
Slashdot	0.905 (0.001)	0.905 (0.001)	0.904 (0.001)	0.905 (0.001)	0.906 (0.010)	**0.919 (0.004)**	—	0.938
LLOG	**0.978 (0.001)**	0.977 (0.001)	0.977 (0.001)	**0.978 (0.001)**	0.976 (0.002)	**0.978 (0.004)**	—	0.980
Solar Flare	0.908 (0.009)*	**0.910 (0.006)***	0.909 (0.008)*	**0.910 (0.006)***	**0.910 (0.006)***	**0.910 (0.006)***	0.864 (0.007)	0.905
Bridges	0.527 (0.055)	0.537 (0.041)	**0.544 (0.029)**	0.520 (0.051)	0.441 (0.029)	**0.544 (0.046)**	0.534 (0.022)	0.622
Thyroid	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	0.959 (0.000)	0.987
Exact Match Ratio (EMR)								
Music	0.167 (0.020)	0.178 (0.028)*	0.170 (0.029)*	**0.194 (0.021)***	0.149 (0.021)	0.140 (0.015)	0.116 (0.025)	0.169
Scene	0.304 (0.044)	0.340 (0.013)	0.329 (0.026)	0.335 (0.018)	**0.400 (0.017)**	0.365 (0.033)	0.330 (0.018)	0.523
Yeast	0.021 (0.005)	0.017 (0.011)	0.016 (0.004)	0.019 (0.006)	0.053 (0.003)	**0.064 (0.012)**	0.059 (0.004)	0.132
ENRON	0.014 (0.006)	0.015 (0.003)	0.017 (0.006)	0.014 (0.004)	0.029 (0.005)	**0.045 (0.013)**	0.022 (0.008)	0.048
Medical	0.062 (0.016)	0.054 (0.009)	0.065 (0.013)	0.064 (0.014)	**0.207 (0.014)**	0.180 (0.050)	—	0.630
Slashdot	0.067 (0.008)	0.070 (0.009)	0.065 (0.007)	0.066 (0.006)	**0.185 (0.010)**	0.162 (0.011)	—	0.309
LLOG	0.096 (0.007)	0.091 (0.008)	0.098 (0.014)	0.097 (0.011)	**0.195 (0.026)**	0.178 (0.024)	—	0.214
Solar Flare	0.788 (0.024)*	**0.794 (0.012)***	0.789 (0.022)*	**0.794 (0.012)***	**0.794 (0.012)***	**0.794 (0.012)***	0.790 (0.022)*	0.785
Bridges	0.035 (0.024)	0.037 (0.016)	**0.049 (0.022)**	0.042 (0.025)	0.021 (0.016)	0.026 (0.016)	**0.049 (0.020)**	0.140
Thyroid	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	0.920

Table A.12: Results using CC with 30% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
Hamming Score (HS)								
Music	0.677 (0.035)	0.680 (0.020)	0.672 (0.016)	**0.691 (0.013)**	0.608 (0.021)	0.597 (0.016)	0.586 (0.026)	0.709
Scene	0.770 (0.016)	0.780 (0.008)	0.775 (0.007)	0.778 (0.005)	**0.806 (0.006)**	0.788 (0.011)	0.738 (0.007)	0.853
Yeast	0.661 (0.012)	0.658 (0.011)	0.668 (0.012)	0.660 (0.015)	0.678 (0.013)	**0.683 (0.007)**	0.649 (0.007)	0.735
ENRON	0.913 (0.002)	0.912 (0.002)	0.912 (0.002)	0.912 (0.002)	0.922 (0.003)	**0.930 (0.003)**	—	0.935
Medical	0.955 (0.002)	0.956 (0.002)	0.954 (0.002)	0.956 (0.002)	0.963 (0.017)	**0.970 (0.002)**	—	0.988
Slashdot	0.905 (0.001)	0.904 (0.001)	0.904 (0.001)	0.905 (0.001)	0.909 (0.006)	**0.920 (0.003)**	—	0.938
LLOG	0.977 (0.001)	0.977 (0.001)	0.977 (0.000)	0.977 (0.000)	0.976 (0.002)	**0.978 (0.003)**	—	0.980
Solar Flare	**0.910 (0.008)***	**0.910 (0.008)***	**0.910 (0.008)***	0.909 (0.009)*	**0.910 (0.008)***	**0.910 (0.008)***	0.865 (0.007)	0.905
Bridges	0.527 (0.058)	0.540 (0.051)	**0.549 (0.041)**	0.517 (0.032)	0.466 (0.066)	0.540 (0.034)	0.525 (0.042)	0.622
Thyroid	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	0.957 (0.011)	**0.961 (0.001)**	0.959 (0.000)	0.987
Exact Match Ratio (EMR)								
Music	0.155 (0.040)	0.164 (0.024)	0.162 (0.019)	**0.179 (0.015)***	0.150 (0.013)	0.141 (0.011)	0.115 (0.014)	0.169
Scene	0.310 (0.041)	0.340 (0.016)	0.321 (0.018)	0.331 (0.016)	**0.400 (0.015)**	0.360 (0.028)	0.322 (0.015)	0.523
Yeast	0.022 (0.006)	0.022 (0.009)	0.024 (0.011)	0.025 (0.009)	0.059 (0.010)	**0.067 (0.010)**	0.056 (0.011)	0.132
ENRON	0.014 (0.003)	0.016 (0.004)	0.013 (0.004)	0.016 (0.006)	0.028 (0.007)	**0.038 (0.010)**	—	0.048
Medical	0.058 (0.013)	0.060 (0.011)	0.063 (0.009)	0.066 (0.013)	**0.231 (0.017)**	0.092 (0.087)	—	0.630
Slashdot	0.071 (0.006)	0.069 (0.005)	0.067 (0.005)	0.070 (0.006)	**0.186 (0.007)**	0.167 (0.009)	—	0.309
LLOG	0.092 (0.013)	0.095 (0.007)	0.089 (0.007)	0.095 (0.009)	0.184 (0.019)	**0.191 (0.017)**	—	0.214
Solar Flare	0.794 (0.018)*	0.794 (0.018)*	0.794 (0.018)*	0.791 (0.024)*	0.794 (0.018)*	0.794 (0.018)*	**0.795 (0.016)***	0.785
Bridges	**0.058 (0.026)**	**0.058 (0.024)**	**0.058 (0.022)**	0.044 (0.022)	0.014 (0.011)	0.040 (0.015)	0.047 (0.021)	0.140
Thyroid	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	0.725 (0.054)	**0.743 (0.003)**	**0.743 (0.003)**	0.920

Table A.13: Results using CC with 40% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
Hamming Score (HS)								
Music	0.658 (0.032)	0.686 (0.013)	0.670 (0.017)	**0.694 (0.016)**	0.609 (0.024)	0.594 (0.012)	0.571 (0.020)	0.709
Scene	0.762 (0.013)	0.778 (0.007)	0.772 (0.010)	0.774 (0.008)	**0.808 (0.008)**	0.796 (0.007)	0.737 (0.008)	0.853
Yeast	0.660 (0.009)	0.657 (0.011)	0.655 (0.006)	0.655 (0.008)	0.669 (0.011)	**0.675 (0.012)**	0.646 (0.007)	0.735
ENRON	0.911 (0.001)	0.912 (0.002)	0.911 (0.001)	0.911 (0.002)	0.923 (0.003)	**0.930 (0.003)**	—	0.935
Medical	0.954 (0.001)	0.954 (0.001)	0.954 (0.002)	0.956 (0.001)	0.965 (0.013)	**0.969 (0.004)**	—	0.988
Slashdot	0.903 (0.001)	0.904 (0.001)	0.905 (0.001)	0.904 (0.001)	0.910 (0.009)	**0.921 (0.000)**	—	0.938
LLOG	0.977 (0.001)	0.976 (0.001)	0.976 (0.000)	0.977 (0.001)	0.977 (0.003)	**0.978 (0.003)**	—	0.980
Solar Flare	**0.911 (0.005)***	**0.911 (0.005)***	**0.911 (0.005)***	**0.911 (0.005)***	0.910 (0.006)*	**0.911 (0.005)***	0.866 (0.007)	0.905
Bridges	0.549 (0.036)	0.530 (0.042)	0.512 (0.046)	0.524 (0.057)	0.455 (0.041)	**0.554 (0.039)**	0.517 (0.021)	0.622
Thyroid	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	0.959 (0.000)	0.987
Exact Match Ratio (EMR)								
Music	0.142 (0.037)	**0.176 (0.011)***	0.159 (0.026)	0.169 (0.026)*	0.142 (0.010)	0.137 (0.009)	0.093 (0.022)	0.169
Scene	0.295 (0.037)	0.338 (0.011)	0.315 (0.027)	0.324 (0.015)	**0.409 (0.014)**	0.378 (0.017)	0.317 (0.023)	0.523
Yeast	0.027 (0.011)	0.020 (0.008)	0.022 (0.004)	0.021 (0.008)	0.055 (0.002)	**0.059 (0.008)**	0.057 (0.011)	0.132
ENRON	0.013 (0.003)	0.013 (0.003)	0.016 (0.004)	0.012 (0.003)	0.035 (0.010)	**0.040 (0.013)**	—	0.048
Medical	0.061 (0.008)	0.062 (0.012)	0.062 (0.012)	0.062 (0.011)	**0.239 (0.022)**	0.065 (0.080)	—	0.630
Slashdot	0.066 (0.009)	0.068 (0.006)	0.073 (0.010)	0.064 (0.007)	**0.184 (0.017)**	0.179 (0.006)	—	0.309
LLOG	0.091 (0.016)	0.089 (0.010)	0.086 (0.010)	0.091 (0.007)	**0.186 (0.015)**	0.182 (0.019)	—	0.214
Solar Flare	0.795 (0.010)*	0.795 (0.010)*	0.795 (0.010)*	0.795 (0.010)*	0.792 (0.013)*	0.795 (0.010)*	**0.797 (0.013)***	0.785
Bridges	**0.049 (0.027)**	0.040 (0.035)	0.040 (0.030)	0.037 (0.019)	0.014 (0.015)	0.037 (0.019)	0.035 (0.019)	0.140
Thyroid	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	0.920

Table A.14: Results using CC with 50% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
Hamming Score (HS)								
Music	0.663 (0.028)	0.680 (0.013)	0.674 (0.016)	**0.690 (0.014)**	0.615 (0.016)	0.609 (0.019)	0.589 (0.022)	0.709
Scene	0.767 (0.015)	0.778 (0.007)	0.777 (0.008)	0.775 (0.008)	**0.805 (0.006)**	0.794 (0.009)	0.738 (0.009)	0.853
Yeast	0.659 (0.010)	0.655 (0.010)	0.661 (0.009)	0.655 (0.014)	0.674 (0.010)	**0.684 (0.013)**	0.647 (0.008)	0.735
ENRON	0.911 (0.001)	0.910 (0.002)	0.911 (0.002)	0.912 (0.002)	0.923 (0.005)	**0.930 (0.003)**	—	0.935
Medical	0.954 (0.002)	0.954 (0.002)	0.953 (0.002)	0.954 (0.001)	0.959 (0.012)	**0.971 (0.002)**	—	0.988
Slashdot	0.903 (0.001)	0.903 (0.001)	0.903 (0.001)	0.904 (0.001)	0.904 (0.005)	**0.922 (0.001)**	—	0.938
LLOG	0.976 (0.001)	0.976 (0.001)	0.976 (0.001)	0.976 (0.001)	0.975 (0.002)	**0.980 (0.003)***	—	0.980
Solar Flare	**0.911 (0.004)***	**0.911 (0.004)***	0.909 (0.007)*	**0.911 (0.004)***	**0.911 (0.004)***	**0.911 (0.004)***	0.863 (0.010)	0.905
Bridges	0.546 (0.026)	**0.551 (0.028)**	0.520 (0.053)	0.540 (0.036)	0.472 (0.031)	0.542 (0.038)	0.490 (0.028)	0.622
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.959 (0.000)	0.987
Exact Match Ratio (EMR)								
Music	0.146 (0.036)	0.160 (0.024)	0.154 (0.025)	**0.164 (0.016)**	0.148 (0.016)	0.140 (0.012)	0.110 (0.021)	0.169
Scene	0.306 (0.043)	0.340 (0.021)	0.336 (0.020)	0.328 (0.015)	**0.397 (0.014)**	0.372 (0.020)	0.321 (0.026)	0.523
Yeast	0.023 (0.009)	0.023 (0.008)	0.027 (0.012)	0.025 (0.012)	0.059 (0.007)	**0.068 (0.012)**	0.056 (0.009)	0.132
ENRON	0.013 (0.004)	0.014 (0.004)	0.013 (0.005)	0.014 (0.005)	0.031 (0.007)	**0.038 (0.011)**	—	0.048
Medical	0.055 (0.010)	0.054 (0.012)	0.056 (0.006)	0.061 (0.010)	**0.223 (0.022)**	0.064 (0.092)	—	0.630
Slashdot	0.070 (0.007)	0.069 (0.005)	0.065 (0.009)	0.065 (0.007)	0.173 (0.006)	**0.184 (0.008)**	—	0.309
LLOG	0.088 (0.009)	0.092 (0.012)	0.088 (0.010)	0.081 (0.010)	**0.173 (0.014)**	**0.173 (0.023)**	—	0.214
Solar Flare	**0.795 (0.010)***	**0.795 (0.010)***	0.790 (0.015)*	**0.795 (0.010)***	**0.795 (0.011)***	**0.795 (0.010)***	0.788 (0.018)*	0.785
Bridges	0.037 (0.019)	0.040 (0.028)	0.037 (0.028)	**0.047 (0.029)**	0.021 (0.020)	0.019 (0.023)	0.037 (0.012)	0.140
Thyroid	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	0.920

Table A.15: Results using CC with 60% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
Hamming Score (HS)								
Music	0.669 (0.028)	0.678 (0.010)	0.658 (0.017)	**0.684 (0.021)**	0.607 (0.021)	0.607 (0.010)	0.588 (0.017)	0.709
Scene	0.769 (0.012)	0.780 (0.011)	0.777 (0.007)	0.775 (0.005)	**0.805 (0.005)**	0.790 (0.010)	0.739 (0.006)	0.853
Yeast	0.659 (0.010)	0.656 (0.010)	0.660 (0.009)	0.661 (0.011)	0.669 (0.011)	**0.673 (0.015)**	0.643 (0.008)	0.735
ENRON	0.910 (0.001)	0.910 (0.001)	0.911 (0.002)	0.911 (0.001)	0.923 (0.005)	**0.931 (0.003)**	—	0.935
Medical	0.953 (0.001)	0.952 (0.001)	0.953 (0.002)	0.954 (0.002)	0.957 (0.010)	**0.970 (0.004)**	—	0.988
Slashdot	0.903 (0.001)	0.904 (0.001)	0.903 (0.001)	0.904 (0.001)	0.900 (0.012)	**0.921 (0.001)**	—	0.938
LLOG	0.975 (0.000)	0.976 (0.001)	0.975 (0.001)	0.976 (0.000)	0.974 (0.001)	**0.979 (0.004)**	—	0.980
Solar Flare	**0.910 (0.005)***	0.908 (0.005)*	**0.910 (0.005)***	**0.910 (0.005)***	**0.910 (0.005)***	**0.910 (0.005)***	0.864 (0.008)	0.905
Bridges	0.547 (0.029)	0.548 (0.033)	0.540 (0.040)	0.529 (0.048)	0.494 (0.023)	**0.561 (0.030)**	0.499 (0.013)	0.622
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.957 (0.010)	**0.961 (0.000)**	0.959 (0.000)	0.987
Exact Match Ratio (EMR)								
Music	0.152 (0.029)	0.160 (0.021)	0.136 (0.023)	**0.170 (0.033)***	0.146 (0.014)	0.147 (0.016)	0.102 (0.020)	0.169
Scene	0.313 (0.031)	0.343 (0.020)	0.329 (0.017)	0.326 (0.012)	**0.394 (0.016)**	0.360 (0.023)	0.326 (0.017)	0.523
Yeast	0.030 (0.008)	0.024 (0.008)	0.028 (0.008)	0.031 (0.011)	**0.059 (0.007)**	**0.059 (0.010)**	0.052 (0.009)	0.132
ENRON	0.015 (0.004)	0.014 (0.005)	0.012 (0.002)	0.010 (0.003)	0.032 (0.007)	**0.048 (0.010)***	—	0.048
Medical	0.059 (0.009)	0.052 (0.014)	0.056 (0.012)	0.059 (0.012)	**0.234 (0.022)**	0.060 (0.081)	—	0.630
Slashdot	0.068 (0.008)	0.070 (0.007)	0.068 (0.005)	0.071 (0.007)	**0.181 (0.019)**	0.179 (0.005)	—	0.309
LLOG	0.077 (0.013)	0.085 (0.012)	0.079 (0.009)	0.086 (0.007)	0.166 (0.010)	**0.175 (0.013)**	—	0.214
Solar Flare	**0.794 (0.008)***	0.788 (0.009)*	**0.794 (0.008)***	**0.794 (0.008)***	**0.794 (0.008)***	**0.794 (0.008)***	0.792 (0.017)*	0.785
Bridges	0.044 (0.019)	**0.047 (0.029)**	0.040 (0.028)	0.042 (0.025)	0.040 (0.018)	0.044 (0.016)	0.035 (0.019)	0.140
Thyroid	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	0.725 (0.056)	**0.743 (0.003)**	**0.743 (0.002)**	0.920

Table A.16: Results using CC with 70% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
Hamming Score (HS)								
Music	0.655 (0.026)	0.675 (0.017)	0.662 (0.016)	**0.685 (0.014)**	0.614 (0.023)	0.601 (0.020)	0.581 (0.016)	0.709
Scene	0.771 (0.015)	0.777 (0.006)	0.776 (0.011)	0.774 (0.005)	**0.807 (0.006)**	0.789 (0.014)	0.735 (0.008)	0.853
Yeast	0.654 (0.006)	0.650 (0.007)	0.661 (0.007)	0.658 (0.011)	**0.677 (0.023)**	0.672 (0.010)	0.650 (0.005)	0.735
ENRON	0.911 (0.001)	0.910 (0.001)	0.910 (0.002)	0.911 (0.002)	0.923 (0.004)	**0.930 (0.002)**	—	0.935
Medical	0.952 (0.002)	0.952 (0.001)	0.951 (0.001)	0.954 (0.002)	0.952 (0.013)	**0.969 (0.003)**	—	0.988
Slashdot	0.903 (0.001)	0.903 (0.001)	0.903 (0.001)	0.904 (0.001)	0.904 (0.011)	**0.921 (0.001)**	—	0.938
LLOG	0.975 (0.001)	0.975 (0.001)	0.975 (0.001)	0.975 (0.001)	0.974 (0.002)	**0.979 (0.004)**	—	0.980
Solar Flare	0.910 (0.007)*	**0.911 (0.006)***	0.910 (0.007)*	0.910 (0.007)*	**0.911 (0.006)***	**0.911 (0.006)***	0.864 (0.007)	0.905
Bridges	0.534 (0.018)	0.521 (0.047)	0.523 (0.037)	0.518 (0.074)	0.469 (0.037)	**0.559 (0.022)**	0.488 (0.071)	0.622
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.957 (0.010)	**0.961 (0.000)**	0.959 (0.000)	0.987
Exact Match Ratio (EMR)								
Music	0.141 (0.031)	0.161 (0.017)	0.142 (0.013)	**0.180 (0.023)***	0.149 (0.016)	0.134 (0.015)	0.107 (0.019)	0.169
Scene	0.320 (0.043)	0.337 (0.013)	0.329 (0.024)	0.324 (0.016)	**0.405 (0.019)**	0.355 (0.033)	0.316 (0.021)	0.523
Yeast	0.026 (0.005)	0.023 (0.007)	0.033 (0.011)	0.029 (0.010)	**0.067 (0.016)**	0.059 (0.011)	0.060 (0.007)	0.132
ENRON	0.013 (0.004)	0.013 (0.004)	0.013 (0.004)	0.012 (0.004)	0.034 (0.005)	**0.036 (0.011)**	—	0.048
Medical	0.064 (0.008)	0.053 (0.008)	0.053 (0.010)	0.056 (0.011)	**0.227 (0.023)**	0.121 (0.101)	—	0.630
Slashdot	0.068 (0.006)	0.067 (0.005)	0.068 (0.004)	0.067 (0.006)	0.175 (0.016)	**0.184 (0.005)**	—	0.309
LLOG	0.080 (0.009)	0.081 (0.012)	0.077 (0.009)	0.083 (0.010)	**0.180 (0.017)**	0.174 (0.019)	—	0.214
Solar Flare	0.792 (0.014)*	0.794 (0.012)*	0.791 (0.017)*	0.791 (0.016)*	**0.795 (0.012)***	**0.795 (0.012)***	0.788 (0.015)*	0.785
Bridges	0.028 (0.025)	0.033 (0.026)	0.051 (0.020)	0.047 (0.028)	0.030 (0.015)	**0.056 (0.019)**	0.040 (0.033)	0.140
Thyroid	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	0.725 (0.056)	**0.743 (0.002)**	**0.743 (0.002)**	0.920

Table A.17: Results using CC with 80% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.648 (0.033)	**0.686 (0.018)**	0.655 (0.012)	0.682 (0.023)	0.638 (0.024)	0.596 (0.012)	0.582 (0.017)	0.709
Scene	0.761 (0.013)	0.773 (0.019)	0.770 (0.006)	0.771 (0.006)	**0.800 (0.009)**	0.784 (0.012)	0.731 (0.008)	0.853
Yeast	0.658 (0.009)	0.661 (0.009)	0.662 (0.012)	0.654 (0.011)	**0.678 (0.012)**	0.674 (0.014)	0.644 (0.004)	0.735
ENRON	0.909 (0.001)	0.910 (0.001)	0.910 (0.002)	0.911 (0.001)	0.923 (0.003)	**0.931 (0.001)**	—	0.935
Medical	0.951 (0.002)	0.953 (0.001)	0.952 (0.002)	0.954 (0.002)	0.943 (0.014)	**0.968 (0.004)**	—	0.988
Slashdot	0.903 (0.001)	0.903 (0.001)	0.902 (0.001)	0.903 (0.001)	0.902 (0.009)	**0.921 (0.001)**	—	0.938
LLOG	0.975 (0.001)	0.975 (0.000)	0.975 (0.001)	0.975 (0.000)	0.974 (0.002)	**0.978 (0.003)**	—	0.980
Solar Flare	0.910 (0.005)*	**0.911 (0.004)***	**0.911 (0.004)***	**0.911 (0.004)***	**0.911 (0.004)***	**0.911 (0.004)***	0.861 (0.009)	0.905
Bridges	0.534 (0.040)	0.535 (0.031)	0.501 (0.045)	0.532 (0.044)	0.499 (0.023)	**0.553 (0.041)**	0.514 (0.043)	0.622
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.959 (0.001)	0.987
				Exact Match Ratio (EMR)				
Music	0.137 (0.030)	0.154 (0.021)	0.139 (0.014)	**0.161 (0.024)**	0.134 (0.010)	0.144 (0.019)	0.104 (0.013)	0.169
Scene	0.296 (0.034)	0.329 (0.023)	0.319 (0.014)	0.312 (0.010)	**0.384 (0.021)**	0.346 (0.036)	0.306 (0.022)	0.523
Yeast	0.028 (0.007)	0.027 (0.015)	0.032 (0.009)	0.026 (0.012)	0.063 (0.013)	**0.065 (0.012)**	0.053 (0.005)	0.132
ENRON	0.013 (0.003)	0.012 (0.004)	0.011 (0.003)	0.013 (0.004)	0.033 (0.007)	**0.045 (0.011)**	—	0.048
Medical	0.058 (0.006)	0.053 (0.009)	0.058 (0.012)	0.057 (0.012)	**0.240 (0.026)**	0.128 (0.083)	—	0.630
Slashdot	0.068 (0.007)	0.068 (0.008)	0.067 (0.006)	0.065 (0.005)	0.170 (0.008)	**0.179 (0.006)**	—	0.309
LLOG	0.083 (0.009)	0.077 (0.009)	0.078 (0.009)	0.082 (0.009)	0.170 (0.012)	**0.179 (0.022)**	—	0.214
Solar Flare	0.793 (0.007)*	**0.795 (0.007)***	**0.795 (0.007)***	**0.795 (0.007)***	**0.795 (0.007)***	**0.795 (0.007)***	0.783 (0.018)	0.785
Bridges	0.040 (0.024)	**0.051 (0.023)**	0.035 (0.019)	0.026 (0.028)	0.032 (0.024)	0.042 (0.025)	**0.051 (0.033)**	0.140
Thyroid	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	0.920

Table A.18: Results using CC with 90% of the original features.

A.3 Ensemble of Classifier Chains (ECC) Results

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
Hamming Score (HS)								
Music	0.715 (0.022)	0.723 (0.016)	0.717 (0.015)	**0.725 (0.018)**	0.635 (0.022)	0.597 (0.012)	0.623 (0.026)	0.771
Scene	0.822 (0.015)	0.825 (0.006)	**0.832 (0.009)**	0.826 (0.008)	0.822 (0.015)	0.776 (0.009)	0.779 (0.006)	0.897
Yeast	0.736 (0.002)	0.735 (0.003)	**0.738 (0.003)**	0.734 (0.001)	0.734 (0.007)	0.731 (0.007)	0.708 (0.004)	0.793
ENRON	0.933 (0.001)	0.933 (0.001)	0.933 (0.001)	0.933 (0.001)	0.940 (0.002)	**0.941 (0.002)**	0.892 (0.001)	0.942
Medical	0.972 (0.001)	0.972 (0.000)	0.971 (0.000)	0.971 (0.001)	0.970 (0.004)	**0.974 (0.002)**	—	0.989
Slashdot	0.946 (0.000)	0.946 (0.000)	0.946 (0.000)	0.946 (0.000)	0.941 (0.001)	**0.947 (0.003)**	—	0.951
LLOG	**0.984 (0.000)***	**0.984 (0.000)***	**0.984 (0.000)***	**0.984 (0.000)***	0.983 (0.001)*	0.983 (0.001)*	—	0.983
Solar Flare	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	0.866 (0.007)	0.905
Bridges	—	—	—	—	—	—	—	0.590
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.959 (0.001)	0.986
Exact Match Ratio (EMR)								
Music	0.192 (0.065)	**0.228 (0.024)**	0.207 (0.028)	0.218 (0.024)	0.152 (0.020)	0.122 (0.022)	0.124 (0.023)	0.245
Scene	0.339 (0.080)	0.368 (0.017)	0.349 (0.025)	0.338 (0.015)	**0.401 (0.027)**	0.275 (0.028)	0.338 (0.014)	0.573
Yeast	0.032 (0.006)	0.031 (0.004)	0.032 (0.005)	0.027 (0.004)	0.081 (0.007)	**0.086 (0.006)**	0.056 (0.007)	0.175
ENRON	0.002 (0.002)	0.003 (0.002)	0.002 (0.002)	0.003 (0.003)	0.034 (0.006)	**0.036 (0.011)**	0.026 (0.004)	0.069
Medical	0.005 (0.005)	0.004 (0.004)	0.009 (0.009)	0.008 (0.005)	**0.191 (0.035)**	0.182 (0.029)	—	0.640
Slashdot	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	**0.145 (0.008)**	0.090 (0.009)	—	0.230
LLOG	0.154 (0.005)	0.154 (0.004)	0.154 (0.005)	0.154 (0.005)	**0.168 (0.006)**	0.158 (0.016)	—	0.197
Solar Flare	0.796 (0.013)*	0.796 (0.013)*	0.796 (0.013)*	0.796 (0.013)*	0.796 (0.013)*	0.796 (0.013)*	**0.797 (0.012)***	0.785
Bridges	—	—	—	—	—	—	—	0.116
Thyroid	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.003)**	0.918

Table A.19: Results using ECC with 10% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
Hamming Score (HS)								
Music	0.714 (0.016)	0.722 (0.017)	0.715 (0.024)	**0.727 (0.015)**	0.638 (0.011)	0.596 (0.014)	0.609 (0.018)	0.771
Scene	0.823 (0.015)	**0.833 (0.009)**	0.825 (0.007)	0.826 (0.007)	0.824 (0.012)	0.776 (0.004)	0.775 (0.007)	0.897
Yeast	**0.739 (0.003)**	0.736 (0.002)	0.738 (0.003)	0.737 (0.003)	**0.739 (0.007)**	0.725 (0.006)	0.707 (0.004)	0.793
ENRON	0.932 (0.001)	0.932 (0.001)	0.932 (0.000)	0.932 (0.001)	0.938 (0.004)	**0.941 (0.001)**	0.892 (0.004)	0.942
Medical	0.971 (0.000)	0.971 (0.000)	0.970 (0.001)	0.971 (0.000)	0.963 (0.007)	**0.974 (0.002)**	—	0.989
Slashdot	**0.946 (0.000)**	**0.946 (0.000)**	**0.946 (0.000)**	**0.946 (0.000)**	0.933 (0.005)	0.944 (0.005)	—	0.951
LLOG	**0.984 (0.000)***	**0.984 (0.000)***	**0.984 (0.000)***	**0.984 (0.000)***	**0.984 (0.001)***	0.983 (0.001)*	—	0.983
Solar Flare	**0.911 (0.008)***	**0.911 (0.008)***	0.909 (0.007)*	**0.911 (0.008)***	**0.911 (0.008)***	**0.911 (0.008)***	0.866 (0.004)	0.905
Bridges	0.516 (0.035)	0.514 (0.053)	0.507 (0.053)	0.477 (0.053)	0.438 (0.027)	**0.540 (0.038)**	0.522 (0.030)	0.590
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.959 (0.000)	0.986
Exact Match Ratio (EMR)								
Music	0.181 (0.064)	0.214 (0.026)	0.207 (0.035)	**0.216 (0.018)**	0.158 (0.014)	0.120 (0.018)	0.121 (0.022)	0.245
Scene	0.291 (0.072)	0.333 (0.014)	0.307 (0.019)	0.309 (0.013)	**0.404 (0.016)**	0.267 (0.016)	0.317 (0.026)	0.573
Yeast	0.030 (0.005)	0.027 (0.005)	0.030 (0.007)	0.031 (0.006)	**0.085 (0.006)**	0.084 (0.009)	0.050 (0.010)	0.175
ENRON	0.003 (0.002)	0.004 (0.002)	0.004 (0.002)	0.003 (0.003)	0.031 (0.006)	**0.034 (0.008)**	0.019 (0.009)	0.069
Medical	0.018 (0.004)	0.020 (0.005)	0.016 (0.008)	0.018 (0.005)	**0.232 (0.020)**	0.173 (0.045)	—	0.640
Slashdot	0.000 (0.000)	0.000 (0.001)	0.000 (0.001)	0.001 (0.001)	**0.152 (0.009)**	0.086 (0.004)	—	0.230
LLOG	0.154 (0.005)	0.154 (0.005)	0.154 (0.005)	0.154 (0.005)	0.162 (0.011)	**0.166 (0.005)**	—	0.197
Solar Flare	**0.796 (0.016)***	**0.796 (0.016)***	0.791 (0.013)*	**0.796 (0.016)***	**0.796 (0.016)***	**0.796 (0.016)***	**0.796 (0.010)***	0.785
Bridges	0.035 (0.024)	0.030 (0.021)	0.033 (0.019)	0.030 (0.028)	0.021 (0.022)	0.032 (0.028)	**0.054 (0.018)**	0.116
Thyroid	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.003)**	**0.743 (0.002)**	**0.743 (0.003)**	0.918

Table A.20: Results using ECC with 20% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.719 (0.021)	**0.726 (0.010)**	0.720 (0.016)	0.721 (0.010)	0.647 (0.019)	0.601 (0.017)	0.610 (0.017)	0.771
Scene	0.817 (0.016)	0.829 (0.011)	0.824 (0.008)	**0.831 (0.006)**	0.821 (0.017)	0.782 (0.010)	0.769 (0.008)	0.897
Yeast	0.739 (0.003)	0.738 (0.003)	0.739 (0.002)	0.738 (0.002)	**0.743 (0.004)**	0.729 (0.006)	0.709 (0.004)	0.793
ENRON	0.931 (0.001)	0.931 (0.001)	0.931 (0.001)	0.931 (0.001)	0.938 (0.003)	**0.939 (0.003)**	0.889 (0.002)	0.942
Medical	0.970 (0.000)	0.970 (0.001)	0.971 (0.000)	0.971 (0.000)	0.955 (0.009)	**0.973 (0.003)**	—	0.989
Slashdot	**0.946 (0.000)**	**0.946 (0.000)**	**0.946 (0.000)**	**0.946 (0.000)**	0.932 (0.012)	0.944 (0.005)	—	0.951
LLOG	**0.984 (0.000)****	**0.984 (0.000)****	**0.984 (0.000)****	**0.984 (0.000)****	0.983 (0.001)*	0.983 (0.001)*	—	0.983
Solar Flare	**0.910 (0.006)****	**0.910 (0.006)****	**0.910 (0.006)****	**0.910 (0.006)****	**0.910 (0.006)****	**0.910 (0.006)****	0.865 (0.006)	0.905
Bridges	0.533 (0.058)	0.532 (0.033)	**0.540 (0.023)**	0.496 (0.058)	0.424 (0.021)	**0.540 (0.031)**	0.515 (0.025)	0.590
Thyroid	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	0.957 (0.010)	**0.961 (0.001)**	0.959 (0.000)	0.986
				Exact Match Ratio (EMR)				
Music	0.184 (0.064)	0.207 (0.030)	**0.208 (0.022)**	0.202 (0.030)	0.162 (0.018)	0.130 (0.022)	0.103 (0.018)	0.245
Scene	0.264 (0.072)	0.320 (0.018)	0.295 (0.030)	0.306 (0.013)	**0.404 (0.019)**	0.269 (0.019)	0.296 (0.025)	0.573
Yeast	0.033 (0.006)	0.031 (0.004)	0.030 (0.004)	0.031 (0.007)	0.087 (0.006)	**0.091 (0.007)**	0.051 (0.008)	0.175
ENRON	0.003 (0.002)	0.004 (0.003)	0.004 (0.002)	0.002 (0.001)	0.029 (0.006)	**0.037 (0.014)**	0.015 (0.009)	0.069
Medical	0.014 (0.007)	0.015 (0.006)	0.015 (0.005)	0.015 (0.006)	**0.252 (0.012)**	0.171 (0.043)	—	0.640
Slashdot	0.002 (0.001)	0.002 (0.001)	0.001 (0.000)	0.001 (0.001)	**0.160 (0.008)**	0.087 (0.008)	—	0.230
LLOG	0.153 (0.004)	0.153 (0.004)	0.153 (0.004)	0.153 (0.004)	0.164 (0.004)	**0.167 (0.004)**	—	0.197
Solar Flare	0.794 (0.012)*	0.794 (0.012)*	0.794 (0.012)*	0.794 (0.012)*	0.794 (0.012)*	0.794 (0.012)*	**0.795 (0.015)****	0.785
Bridges	0.037 (0.019)	0.037 (0.016)	0.032 (0.028)	0.044 (0.028)	0.021 (0.016)	0.016 (0.015)	**0.058 (0.026)**	0.116
Thyroid	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	0.725 (0.056)	**0.743 (0.003)**	**0.743 (0.003)**	0.918

Table A.21: Results using ECC with 30% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.706 (0.020)	0.721 (0.006)	0.707 (0.016)	**0.728 (0.013)**	0.644 (0.016)	0.601 (0.016)	0.612 (0.017)	0.771
Scene	0.821 (0.015)	0.823 (0.006)	0.822 (0.010)	**0.832 (0.008)**	0.825 (0.012)	0.779 (0.005)	0.772 (0.006)	0.897
Yeast	0.739 (0.003)	0.738 (0.002)	0.738 (0.004)	0.739 (0.004)	**0.741 (0.004)**	0.731 (0.006)	0.709 (0.003)	0.793
ENRON	0.931 (0.001)	0.932 (0.001)	0.932 (0.001)	0.931 (0.001)	0.938 (0.002)	**0.940 (0.001)**	—	0.942
Medical	0.971 (0.000)	0.970 (0.000)	0.971 (0.000)	0.970 (0.001)	**0.973 (0.009)**	0.972 (0.001)	—	0.989
Slashdot	**0.945 (0.001)**	**0.945 (0.000)**	**0.945 (0.000)**	**0.945 (0.001)**	0.933 (0.011)	0.941 (0.005)	—	0.951
LLOG	**0.984 (0.000)****	**0.984 (0.000)****	**0.984 (0.000)****	**0.984 (0.000)****	0.983 (0.000)*	0.983 (0.001)*	—	0.983
Solar Flare	**0.910 (0.008)****	**0.910 (0.008)****	**0.910 (0.008)****	**0.910 (0.008)****	**0.910 (0.008)****	**0.910 (0.008)****	0.865 (0.007)	0.905
Bridges	0.514 (0.051)	0.526 (0.055)	**0.538 (0.045)**	0.505 (0.038)	0.444 (0.050)	0.521 (0.048)	0.510 (0.046)	0.590
Thyroid	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	0.944 (0.017)	**0.961 (0.001)**	0.959 (0.000)	0.986
				Exact Match Ratio (EMR)				
Music	0.166 (0.058)	**0.214 (0.014)**	0.172 (0.016)	0.210 (0.021)	0.146 (0.018)	0.123 (0.019)	0.111 (0.016)	0.245
Scene	0.266 (0.070)	0.336 (0.018)	0.302 (0.019)	0.298 (0.012)	**0.414 (0.024)**	0.276 (0.012)	0.304 (0.018)	0.573
Yeast	0.028 (0.004)	0.030 (0.004)	0.029 (0.006)	0.030 (0.006)	0.087 (0.005)	**0.090 (0.014)**	0.046 (0.009)	0.175
ENRON	0.003 (0.002)	0.003 (0.002)	0.003 (0.002)	0.003 (0.002)	0.029 (0.005)	**0.036 (0.008)**	—	0.069
Medical	0.013 (0.006)	0.015 (0.003)	0.014 (0.006)	0.015 (0.005)	**0.264 (0.015)**	0.068 (0.069)	—	0.640
Slashdot	0.003 (0.001)	0.002 (0.001)	0.002 (0.001)	0.002 (0.001)	**0.169 (0.013)**	0.093 (0.010)	—	0.230
LLOG	0.153 (0.004)	0.154 (0.004)	0.153 (0.005)	0.153 (0.004)	0.167 (0.007)	**0.168 (0.006)**	—	0.197
Solar Flare	0.794 (0.018)*	0.794 (0.018)*	0.794 (0.018)*	0.794 (0.018)*	0.794 (0.018)*	0.794 (0.018)*	**0.795 (0.016)****	0.785
Bridges	**0.047 (0.018)**	0.042 (0.021)	0.042 (0.023)	0.040 (0.031)	0.012 (0.021)	0.033 (0.022)	0.042 (0.025)	0.116
Thyroid	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	0.652 (0.091)	**0.743 (0.003)**	**0.743 (0.003)**	0.918

Table A.22: Results using ECC with 40% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.710 (0.025)	0.731 (0.024)	0.711 (0.009)	**0.736 (0.011)**	0.642 (0.020)	0.598 (0.012)	0.613 (0.018)	0.771
Scene	0.815 (0.016)	0.828 (0.007)	0.820 (0.005)	**0.832 (0.007)**	0.828 (0.014)	0.782 (0.010)	0.769 (0.007)	0.897
Yeast	0.737 (0.003)	0.738 (0.003)	0.740 (0.003)	0.737 (0.003)	**0.742 (0.004)**	0.725 (0.009)	0.709 (0.003)	0.793
ENRON	0.932 (0.001)	0.931 (0.001)	0.931 (0.001)	0.932 (0.001)	0.938 (0.003)	**0.939 (0.002)**	—	0.942
Medical	0.971 (0.000)	0.970 (0.001)	0.970 (0.001)	0.971 (0.000)	**0.972 (0.011)**	0.971 (0.002)	—	0.989
Slashdot	**0.945 (0.000)**	**0.945 (0.000)**	**0.945 (0.000)**	**0.945 (0.000)**	0.935 (0.007)	0.940 (0.002)	—	0.951
LLOG	**0.984 (0.000)***	**0.984 (0.000)***	**0.984 (0.000)***	**0.984 (0.000)***	0.983 (0.000)*	0.983 (0.001)*	—	0.983
Solar Flare	**0.910 (0.004)***	**0.910 (0.004)***	**0.910 (0.004)***	**0.910 (0.004)***	**0.910 (0.004)***	**0.910 (0.004)***	0.865 (0.008)	0.905
Bridges	**0.534 (0.042)**	0.529 (0.026)	0.518 (0.036)	0.529 (0.032)	0.444 (0.030)	0.533 (0.047)	0.496 (0.032)	0.590
Thyroid	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	0.947 (0.017)	**0.961 (0.001)**	0.959 (0.000)	0.986
				Exact Match Ratio (EMR)				
Music	0.163 (0.062)	0.223 (0.032)	0.174 (0.016)	**0.227 (0.018)**	0.153 (0.022)	0.126 (0.016)	0.103 (0.015)	0.245
Scene	0.260 (0.066)	0.325 (0.016)	0.289 (0.023)	0.299 (0.016)	**0.414 (0.015)**	0.268 (0.025)	0.296 (0.021)	0.573
Yeast	0.028 (0.004)	0.029 (0.005)	0.031 (0.007)	0.028 (0.008)	**0.088 (0.005)**	0.080 (0.010)	0.046 (0.007)	0.175
ENRON	0.004 (0.004)	0.004 (0.003)	0.004 (0.004)	0.004 (0.004)	0.030 (0.003)	**0.040 (0.011)**	—	0.069
Medical	0.014 (0.005)	0.014 (0.007)	0.013 (0.007)	0.016 (0.005)	**0.267 (0.016)**	0.055 (0.067)	—	0.640
Slashdot	0.003 (0.002)	0.003 (0.002)	0.003 (0.002)	0.002 (0.001)	**0.171 (0.013)**	0.111 (0.011)	—	0.230
LLOG	0.153 (0.007)	0.153 (0.006)	0.154 (0.006)	0.153 (0.008)	**0.169 (0.009)**	0.162 (0.011)	—	0.197
Solar Flare	**0.795 (0.010)***	**0.795 (0.010)***	**0.795 (0.010)***	**0.795 (0.010)***	**0.795 (0.010)***	**0.795 (0.010)***	**0.795 (0.015)***	0.785
Bridges	**0.044 (0.019)**	0.040 (0.026)	0.030 (0.028)	0.040 (0.026)	0.005 (0.009)	0.018 (0.014)	0.028 (0.023)	0.116
Thyroid	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	0.670 (0.091)	**0.743 (0.003)**	**0.743 (0.003)**	0.918

Table A.23: Results using ECC with 50% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.722 (0.025)	**0.730 (0.013)**	0.717 (0.016)	0.729 (0.015)	0.648 (0.019)	0.602 (0.012)	0.623 (0.023)	0.771
Scene	0.816 (0.014)	0.828 (0.010)	0.824 (0.008)	**0.835 (0.008)**	0.826 (0.008)	0.783 (0.010)	0.769 (0.009)	0.897
Yeast	0.739 (0.003)	0.738 (0.003)	0.741 (0.004)	0.738 (0.004)	**0.745 (0.004)**	0.730 (0.012)	0.707 (0.003)	0.793
ENRON	0.932 (0.000)	0.931 (0.001)	0.932 (0.000)	0.931 (0.001)	0.937 (0.003)	**0.939 (0.001)**	—	0.942
Medical	0.971 (0.001)	0.970 (0.001)	0.971 (0.000)	0.970 (0.000)	**0.973 (0.006)**	0.972 (0.001)	—	0.989
Slashdot	**0.945 (0.000)**	**0.945 (0.000)**	**0.945 (0.000)**	**0.945 (0.000)**	0.931 (0.004)	0.940 (0.000)	—	0.951
LLOG	**0.984 (0.000)***	**0.984 (0.000)***	**0.984 (0.000)***	**0.984 (0.000)***	0.983 (0.001)*	0.983 (0.001)*	—	0.983
Solar Flare	**0.910 (0.004)***	**0.910 (0.004)***	**0.910 (0.004)***	**0.910 (0.004)***	**0.910 (0.004)***	**0.910 (0.004)***	0.864 (0.009)	0.905
Bridges	0.545 (0.014)	**0.558 (0.016)**	0.548 (0.024)	0.551 (0.049)	0.462 (0.038)	0.520 (0.046)	0.506 (0.033)	0.590
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.951 (0.015)	**0.961 (0.000)**	0.959 (0.000)	0.986
				Exact Match Ratio (EMR)				
Music	0.186 (0.059)	0.208 (0.020)	0.181 (0.024)	**0.214 (0.023)**	0.147 (0.019)	0.115 (0.021)	0.108 (0.012)	0.245
Scene	0.271 (0.065)	0.333 (0.017)	0.319 (0.015)	0.318 (0.010)	**0.412 (0.021)**	0.264 (0.018)	0.288 (0.026)	0.573
Yeast	0.028 (0.004)	0.028 (0.004)	0.030 (0.004)	0.030 (0.006)	**0.088 (0.007)**	0.081 (0.010)	0.043 (0.008)	0.175
ENRON	0.003 (0.003)	0.005 (0.004)	0.004 (0.002)	0.004 (0.002)	0.030 (0.003)	**0.039 (0.010)**	—	0.069
Medical	0.015 (0.004)	0.016 (0.006)	0.015 (0.003)	0.019 (0.008)	**0.279 (0.015)**	0.048 (0.061)	—	0.640
Slashdot	0.004 (0.001)	0.004 (0.001)	0.004 (0.001)	0.003 (0.001)	**0.167 (0.007)**	0.122 (0.005)	—	0.230
LLOG	0.153 (0.004)	0.153 (0.005)	0.154 (0.004)	0.154 (0.006)	0.166 (0.008)	**0.167 (0.005)**	—	0.197
Solar Flare	**0.794 (0.009)***	**0.794 (0.009)***	**0.794 (0.009)***	**0.794 (0.009)***	**0.794 (0.009)***	**0.794 (0.009)***	0.789 (0.023)*	0.785
Bridges	0.033 (0.021)	0.033 (0.026)	0.037 (0.012)	0.040 (0.028)	0.012 (0.019)	0.012 (0.016)	**0.044 (0.029)**	0.116
Thyroid	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	0.690 (0.083)	**0.743 (0.002)**	**0.743 (0.002)**	0.918

Table A.24: Results using ECC with 60% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
Hamming Score (HS)								
Music	0.714 (0.022)	0.728 (0.020)	0.716 (0.015)	**0.731 (0.013)**	0.651 (0.011)	0.618 (0.015)	0.617 (0.015)	0.771
Scene	0.824 (0.017)	**0.834 (0.008)**	0.821 (0.007)	0.833 (0.004)	0.822 (0.010)	0.792 (0.012)	0.771 (0.007)	0.897
Yeast	0.738 (0.003)	0.739 (0.003)	0.741 (0.003)	0.738 (0.003)	**0.748 (0.006)**	0.731 (0.007)	0.707 (0.002)	0.793
ENRON	0.932 (0.001)	0.931 (0.001)	0.931 (0.001)	0.932 (0.001)	0.939 (0.002)	**0.940 (0.001)**	—	0.942
Medical	0.970 (0.000)	0.970 (0.001)	**0.971 (0.001)**	0.970 (0.001)	0.970 (0.008)	**0.971 (0.002)**	—	0.989
Slashdot	**0.945 (0.000)**	**0.945 (0.000)**	**0.945 (0.001)**	**0.945 (0.000)**	0.935 (0.007)	0.940 (0.001)	—	0.951
LLOG	**0.984 (0.000)***	**0.984 (0.000)***	**0.984 (0.000)***	**0.984 (0.000)***	0.983 (0.001)*	0.983 (0.001)*	—	0.983
Solar Flare	**0.910 (0.005)***	**0.910 (0.005)***	**0.910 (0.005)***	**0.910 (0.005)***	**0.910 (0.005)***	**0.910 (0.005)***	0.866 (0.008)	0.905
Bridges	0.543 (0.046)	**0.549 (0.014)**	0.548 (0.022)	0.540 (0.046)	0.469 (0.022)	0.521 (0.034)	0.503 (0.044)	0.590
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.944 (0.017)	**0.961 (0.000)**	0.959 (0.000)	0.986
Exact Match Ratio (EMR)								
Music	0.168 (0.060)	**0.216 (0.032)**	0.174 (0.015)	0.215 (0.028)	0.165 (0.019)	0.122 (0.022)	0.104 (0.017)	0.245
Scene	0.275 (0.070)	0.330 (0.018)	0.314 (0.019)	0.310 (0.009)	**0.407 (0.019)**	0.265 (0.024)	0.299 (0.019)	0.573
Yeast	0.028 (0.005)	0.028 (0.004)	0.029 (0.004)	0.030 (0.006)	**0.091 (0.007)**	0.082 (0.009)	0.041 (0.006)	0.175
ENRON	0.003 (0.002)	0.003 (0.002)	0.004 (0.003)	0.003 (0.002)	0.030 (0.004)	**0.044 (0.007)**	—	0.069
Medical	0.015 (0.003)	0.013 (0.005)	0.015 (0.008)	0.018 (0.005)	**0.283 (0.015)**	0.046 (0.065)	—	0.640
Slashdot	0.003 (0.001)	0.003 (0.002)	0.004 (0.002)	0.004 (0.002)	**0.179 (0.015)**	0.122 (0.007)	—	0.230
LLOG	0.153 (0.006)	0.154 (0.006)	0.153 (0.006)	0.153 (0.005)	**0.170 (0.011)**	0.165 (0.006)	—	0.197
Solar Flare	0.793 (0.009)*	0.793 (0.009)*	0.794 (0.008)*	0.794 (0.008)*	0.794 (0.008)*	0.794 (0.008)*	**0.796 (0.015)***	0.785
Bridges	0.037 (0.019)	0.047 (0.018)	0.056 (0.021)	**0.058 (0.035)**	0.016 (0.015)	0.030 (0.015)	0.033 (0.019)	0.116
Thyroid	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	0.650 (0.090)	**0.743 (0.003)**	**0.743 (0.002)**	0.918

Table A.25: Results using ECC with 70% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
Hamming Score (HS)								
Music	0.715 (0.020)	**0.728 (0.019)**	0.719 (0.012)	0.727 (0.021)	0.656 (0.019)	0.623 (0.014)	0.612 (0.017)	0.771
Scene	0.816 (0.016)	0.830 (0.007)	0.828 (0.009)	**0.838 (0.007)**	0.828 (0.011)	0.783 (0.010)	0.768 (0.011)	0.897
Yeast	0.739 (0.003)	0.738 (0.003)	0.739 (0.003)	0.739 (0.003)	**0.743 (0.005)**	0.727 (0.008)	0.708 (0.003)	0.793
ENRON	0.932 (0.001)	0.931 (0.001)	0.931 (0.001)	0.931 (0.001)	0.938 (0.003)	**0.940 (0.001)**	—	0.942
Medical	0.970 (0.001)	0.971 (0.001)	0.971 (0.001)	0.970 (0.001)	0.965 (0.011)	**0.972 (0.002)**	—	0.989
Slashdot	**0.945 (0.000)**	**0.945 (0.000)**	**0.945 (0.000)**	**0.945 (0.000)**	0.933 (0.010)	0.939 (0.003)	—	0.951
LLOG	**0.984 (0.000)***	**0.984 (0.000)***	**0.984 (0.000)***	**0.984 (0.000)***	0.983 (0.001)*	**0.984 (0.001)***	—	0.983
Solar Flare	**0.911 (0.006)***	**0.911 (0.006)***	**0.911 (0.006)***	**0.911 (0.007)***	**0.911 (0.006)***	**0.911 (0.006)***	0.864 (0.009)	0.905
Bridges	0.541 (0.027)	0.527 (0.027)	0.518 (0.040)	0.518 (0.055)	0.458 (0.030)	**0.549 (0.040)**	0.517 (0.037)	0.590
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.951 (0.015)	**0.961 (0.000)**	0.959 (0.000)	0.986
Exact Match Ratio (EMR)								
Music	0.169 (0.059)	0.204 (0.021)	0.176 (0.017)	**0.206 (0.012)**	0.157 (0.026)	0.127 (0.020)	0.100 (0.016)	0.245
Scene	0.270 (0.069)	0.335 (0.015)	0.330 (0.025)	0.307 (0.012)	**0.409 (0.016)**	0.268 (0.025)	0.283 (0.028)	0.573
Yeast	0.028 (0.005)	0.029 (0.004)	0.028 (0.003)	0.030 (0.003)	**0.086 (0.008)**	0.080 (0.007)	0.038 (0.008)	0.175
ENRON	0.004 (0.001)	0.003 (0.002)	0.004 (0.002)	0.004 (0.002)	0.032 (0.005)	**0.035 (0.009)**	—	0.069
Medical	0.012 (0.005)	0.015 (0.005)	0.016 (0.008)	0.012 (0.006)	**0.286 (0.018)**	0.092 (0.081)	—	0.640
Slashdot	0.004 (0.001)	0.004 (0.002)	0.005 (0.002)	0.004 (0.001)	**0.173 (0.017)**	0.119 (0.011)	—	0.230
LLOG	0.153 (0.006)	0.153 (0.006)	0.152 (0.006)	0.152 (0.006)	0.162 (0.013)	**0.167 (0.006)**	—	0.197
Solar Flare	**0.795 (0.012)***	**0.795 (0.012)***	0.793 (0.013)*	0.794 (0.013)*	**0.795 (0.012)***	**0.795 (0.012)***	0.791 (0.021)*	0.785
Bridges	0.033 (0.019)	0.037 (0.024)	0.033 (0.028)	0.026 (0.027)	0.028 (0.023)	**0.049 (0.019)**	0.044 (0.030)	0.116
Thyroid	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	0.688 (0.083)	**0.743 (0.002)**	**0.743 (0.002)**	0.918

Table A.26: Results using ECC with 80% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.706 (0.018)	**0.721 (0.015)**	0.706 (0.014)	0.720 (0.017)	0.667 (0.014)	0.600 (0.011)	0.617 (0.022)	0.771
Scene	0.809 (0.016)	0.821 (0.009)	0.813 (0.008)	**0.835 (0.007)**	0.834 (0.010)	0.771 (0.007)	0.765 (0.009)	0.897
Yeast	0.738 (0.002)	0.740 (0.002)	0.739 (0.002)	0.739 (0.004)	**0.745 (0.005)**	0.729 (0.011)	0.705 (0.003)	0.793
ENRON	0.932 (0.001)	0.931 (0.001)	0.932 (0.000)	0.931 (0.001)	**0.940 (0.002)**	0.939 (0.001)	—	0.942
Medical	0.970 (0.001)	0.970 (0.001)	0.970 (0.001)	0.971 (0.001)	0.960 (0.010)	**0.972 (0.002)**	—	0.989
Slashdot	0.944 (0.000)	0.944 (0.001)	0.944 (0.001)	**0.945 (0.000)**	0.930 (0.011)	0.940 (0.000)	—	0.951
LLOG	**0.984 (0.000)***	**0.984 (0.000)***	**0.984 (0.000)***	**0.984 (0.000)***	0.983 (0.001)*	**0.984 (0.000)***	—	0.983
Solar Flare	**0.911 (0.004)***	**0.911 (0.004)***	**0.911 (0.004)***	**0.911 (0.004)***	**0.911 (0.004)***	**0.911 (0.004)***	0.864 (0.008)	0.905
Bridges	**0.554 (0.026)**	0.547 (0.028)	0.524 (0.043)	0.551 (0.035)	0.488 (0.034)	0.534 (0.041)	0.513 (0.046)	0.590
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.947 (0.017)	**0.961 (0.000)**	0.959 (0.001)	0.986
				Exact Match Ratio (EMR)				
Music	0.163 (0.050)	**0.212 (0.017)**	0.165 (0.014)	0.207 (0.026)	0.131 (0.027)	0.126 (0.014)	0.104 (0.018)	0.245
Scene	0.274 (0.073)	0.318 (0.008)	0.317 (0.017)	0.292 (0.017)	**0.398 (0.017)**	0.244 (0.026)	0.280 (0.034)	0.573
Yeast	0.029 (0.004)	0.030 (0.004)	0.030 (0.005)	0.030 (0.004)	**0.089 (0.007)**	0.081 (0.012)	0.039 (0.008)	0.175
ENRON	0.003 (0.002)	0.002 (0.002)	0.002 (0.002)	0.003 (0.001)	0.031 (0.004)	**0.039 (0.013)**	—	0.069
Medical	0.016 (0.004)	0.014 (0.006)	0.018 (0.005)	0.014 (0.005)	**0.280 (0.026)**	0.100 (0.068)	—	0.640
Slashdot	0.004 (0.002)	0.005 (0.002)	0.004 (0.001)	0.004 (0.001)	**0.168 (0.011)**	0.122 (0.005)	—	0.230
LLOG	0.153 (0.004)	0.154 (0.005)	0.153 (0.005)	0.153 (0.005)	0.159 (0.017)	**0.164 (0.005)**	—	0.197
Solar Flare	**0.795 (0.007)***	**0.795 (0.007)***	**0.795 (0.007)***	**0.795 (0.007)***	**0.795 (0.007)***	**0.795 (0.007)***	0.794 (0.018)*	0.785
Bridges	0.049 (0.024)	0.047 (0.021)	**0.058 (0.036)**	0.047 (0.018)	0.026 (0.019)	0.033 (0.024)	0.051 (0.031)	0.116
Thyroid	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	0.670 (0.089)	**0.743 (0.003)**	**0.743 (0.004)**	0.918

Table A.27: Results using ECC with 90% of the original features.

A.4 Class Relevance Stacking (CRS) Results

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
Hamming Score (HS)								
Music	0.711 (0.018)	**0.712 (0.015)**	0.702 (0.017)	0.705 (0.014)	0.658 (0.010)	0.656 (0.007)	0.638 (0.017)	0.785
Scene	0.789 (0.007)	0.785 (0.014)	0.795 (0.006)	0.790 (0.008)	**0.803 (0.011)**	0.791 (0.015)	0.759 (0.007)	0.898
Yeast	0.771 (0.005)	0.773 (0.004)	0.771 (0.004)	**0.776 (0.003)**	0.772 (0.004)	0.769 (0.003)	0.730 (0.003)	0.789
ENRON	0.924 (0.003)	0.924 (0.001)	0.924 (0.002)	0.926 (0.003)	0.928 (0.004)	**0.936 (0.002)**	0.879 (0.002)	0.941
Medical	0.963 (0.003)	0.963 (0.002)	0.963 (0.001)	0.964 (0.002)	0.963 (0.003)	**0.971 (0.002)**	—	0.989
Slashdot	**0.945 (0.001)**	0.944 (0.000)	**0.945 (0.001)**	0.944 (0.001)	0.934 (0.003)	0.943 (0.005)	—	0.953
LLOG	**0.983 (0.001)***	0.982 (0.001)	**0.983 (0.000)***	**0.983 (0.000)***	**0.983 (0.001)***	**0.983 (0.001)***	—	0.983
Solar Flare	**0.912 (0.004)***	**0.912 (0.004)***	0.907 (0.015)*	**0.912 (0.004)***	0.910 (0.002)*	**0.912 (0.004)***	0.860 (0.014)	0.905
Bridges	—	—	—	—	—	—	—	0.580
Thyroid	0.958 (0.010)	0.955 (0.012)	0.958 (0.010)	0.942 (0.016)	**0.961 (0.001)**	**0.961 (0.001)**	0.959 (0.001)	0.986
Exact Match Ratio (EMR)								
Music	**0.169 (0.018)**	0.157 (0.030)	0.149 (0.022)	0.163 (0.021)	0.100 (0.023)	0.103 (0.022)	0.080 (0.027)	0.249
Scene	0.252 (0.012)	0.253 (0.016)	0.250 (0.011)	0.244 (0.010)	**0.300 (0.018)**	0.158 (0.042)	0.226 (0.022)	0.519
Yeast	**0.081 (0.007)**	0.077 (0.011)	0.078 (0.009)	**0.081 (0.007)**	0.050 (0.009)	0.040 (0.019)	0.035 (0.009)	0.117
ENRON	0.003 (0.003)	0.003 (0.002)	0.003 (0.002)	0.002 (0.002)	0.026 (0.005)	**0.030 (0.011)**	0.015 (0.005)	0.050
Medical	0.026 (0.009)	0.020 (0.014)	0.030 (0.012)	0.022 (0.013)	0.164 (0.021)	**0.171 (0.030)**	—	0.625
Slashdot	0.002 (0.003)	0.004 (0.004)	0.002 (0.001)	0.003 (0.001)	**0.147 (0.012)**	0.091 (0.008)	—	0.247
LLOG	0.144 (0.006)	0.138 (0.011)	0.144 (0.007)	0.144 (0.009)	**0.167 (0.007)**	0.161 (0.013)	—	0.205
Solar Flare	**0.796 (0.006)***	**0.796 (0.006)***	0.783 (0.038)	**0.796 (0.006)***	0.791 (0.009)*	**0.796 (0.006)***	0.783 (0.024)	0.785
Bridges	—	—	—	—	—	—	—	0.116
Thyroid	0.726 (0.053)	0.708 (0.071)	0.729 (0.054)	0.637 (0.089)	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	0.916

Table A.28: Results using CRS with 10% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
Hamming Score (HS)								
Music	0.696 (0.016)	0.699 (0.024)	0.701 (0.013)	**0.703 (0.017)**	0.660 (0.006)	0.657 (0.010)	0.618 (0.023)	0.785
Scene	0.786 (0.009)	0.786 (0.011)	0.770 (0.010)	0.787 (0.010)	**0.817 (0.016)**	0.781 (0.010)	0.750 (0.007)	0.898
Yeast	0.767 (0.003)	0.765 (0.005)	0.764 (0.005)	0.767 (0.004)	**0.773 (0.001)**	**0.773 (0.003)**	0.728 (0.003)	0.789
ENRON	0.916 (0.002)	0.916 (0.003)	0.916 (0.002)	0.920 (0.003)	0.922 (0.005)	**0.933 (0.002)**	0.875 (0.003)	0.941
Medical	0.955 (0.002)	0.958 (0.004)	0.955 (0.002)	0.956 (0.003)	0.950 (0.008)	**0.969 (0.003)**	—	0.989
Slashdot	**0.940 (0.003)**	0.938 (0.003)	0.938 (0.003)	0.939 (0.003)	0.917 (0.008)	0.939 (0.005)	—	0.953
LLOG	0.980 (0.001)	0.980 (0.001)	0.980 (0.001)	0.980 (0.001)	0.982 (0.002)	**0.983 (0.001)***	—	0.983
Solar Flare	**0.910 (0.009)***	**0.910 (0.009)***	**0.910 (0.009)***	**0.910 (0.009)***	**0.910 (0.009)***	**0.910 (0.009)***	0.864 (0.007)	0.905
Bridges	0.529 (0.052)	0.551 (0.042)	0.516 (0.071)	0.512 (0.097)	0.457 (0.036)	**0.565 (0.030)**	0.545 (0.019)	0.580
Thyroid	0.938 (0.034)	0.947 (0.022)	0.945 (0.021)	0.948 (0.016)	**0.961 (0.000)**	**0.961 (0.000)**	0.959 (0.000)	0.986
Exact Match Ratio (EMR)								
Music	0.121 (0.022)	0.140 (0.013)	0.128 (0.024)	**0.146 (0.022)**	0.104 (0.013)	0.102 (0.021)	0.078 (0.021)	0.249
Scene	0.219 (0.017)	0.235 (0.017)	0.203 (0.016)	0.219 (0.021)	**0.304 (0.025)**	0.139 (0.035)	0.216 (0.014)	0.519
Yeast	0.071 (0.008)	0.070 (0.007)	0.070 (0.011)	**0.075 (0.009)**	0.057 (0.007)	0.056 (0.006)	0.027 (0.006)	0.117
ENRON	0.003 (0.002)	0.002 (0.002)	0.003 (0.002)	0.004 (0.004)	0.023 (0.004)	**0.028 (0.008)**	0.008 (0.004)	0.050
Medical	0.031 (0.010)	0.041 (0.010)	0.032 (0.006)	0.034 (0.010)	**0.181 (0.021)**	0.166 (0.041)	—	0.625
Slashdot	0.011 (0.008)	0.014 (0.006)	0.012 (0.005)	0.013 (0.007)	**0.145 (0.013)**	0.085 (0.014)	—	0.247
LLOG	0.121 (0.010)	0.121 (0.015)	0.123 (0.017)	0.122 (0.020)	0.164 (0.013)	**0.167 (0.007)**	—	0.205
Solar Flare	**0.794 (0.021)***	0.792 (0.021)*	**0.794 (0.021)***	**0.794 (0.021)***	0.792 (0.020)*	**0.794 (0.021)***	0.791 (0.017)*	0.785
Bridges	0.047 (0.028)	0.040 (0.030)	0.044 (0.035)	0.054 (0.030)	0.016 (0.015)	0.054 (0.015)	**0.068 (0.022)**	0.116
Thyroid	0.673 (0.091)	0.689 (0.084)	0.673 (0.090)	0.673 (0.087)	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.003)**	0.916

Table A.29: Results using CRS with 20% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
Hamming Score (HS)								
Music	0.688 (0.019)	0.690 (0.012)	0.682 (0.016)	**0.697 (0.018)**	0.661 (0.009)	0.657 (0.011)	0.620 (0.019)	0.785
Scene	0.783 (0.009)	0.785 (0.014)	0.780 (0.009)	0.778 (0.014)	**0.806 (0.011)**	0.797 (0.019)	0.743 (0.010)	0.898
Yeast	0.755 (0.004)	0.758 (0.007)	0.752 (0.005)	0.761 (0.005)	**0.773 (0.002)**	0.766 (0.005)	0.726 (0.003)	0.789
ENRON	0.911 (0.002)	0.911 (0.002)	0.911 (0.002)	0.912 (0.003)	0.921 (0.002)	**0.933 (0.002)**	0.870 (0.003)	0.941
Medical	0.953 (0.001)	0.954 (0.002)	0.951 (0.002)	0.953 (0.002)	0.936 (0.006)	**0.965 (0.005)**	—	0.989
Slashdot	0.935 (0.004)	0.931 (0.005)	0.932 (0.003)	0.932 (0.005)	0.912 (0.013)	**0.942 (0.007)**	—	0.953
LLOG	0.978 (0.001)	0.979 (0.001)	0.978 (0.001)	0.978 (0.001)	0.980 (0.001)	**0.982 (0.001)**	—	0.983
Solar Flare	**0.911 (0.007)***	0.910 (0.006)*	**0.911 (0.007)***	0.908 (0.009)*	0.910 (0.006)*	**0.911 (0.007)***	0.863 (0.010)	0.905
Bridges	0.533 (0.040)	0.505 (0.027)	0.514 (0.068)	0.515 (0.040)	0.467 (0.031)	**0.559 (0.030)**	0.538 (0.014)	0.580
Thyroid	0.942 (0.016)	0.948 (0.017)	0.947 (0.029)	0.945 (0.017)	**0.961 (0.000)**	**0.961 (0.000)**	0.959 (0.000)	0.986
Exact Match Ratio (EMR)								
Music	0.119 (0.021)	**0.140 (0.027)**	0.109 (0.018)	0.136 (0.025)	0.112 (0.017)	0.102 (0.024)	0.066 (0.017)	0.249
Scene	0.208 (0.018)	0.223 (0.014)	0.209 (0.013)	0.218 (0.019)	**0.294 (0.022)**	0.145 (0.047)	0.200 (0.024)	0.519
Yeast	0.061 (0.007)	0.067 (0.007)	0.054 (0.007)	**0.070 (0.008)**	0.059 (0.005)	0.063 (0.016)	0.029 (0.005)	0.117
ENRON	0.001 (0.001)	0.003 (0.002)	0.002 (0.001)	0.002 (0.002)	0.019 (0.005)	**0.030 (0.011)**	0.004 (0.002)	0.050
Medical	0.032 (0.009)	0.033 (0.008)	0.029 (0.010)	0.033 (0.007)	**0.190 (0.013)**	0.156 (0.039)	—	0.625
Slashdot	0.018 (0.008)	0.022 (0.006)	0.018 (0.006)	0.020 (0.008)	**0.146 (0.012)**	0.087 (0.007)	—	0.247
LLOG	0.109 (0.010)	0.112 (0.010)	0.108 (0.013)	0.105 (0.011)	**0.165 (0.013)**	0.150 (0.023)	—	0.205
Solar Flare	**0.795 (0.013)***	0.793 (0.014)*	**0.795 (0.013)***	0.788 (0.023)*	0.792 (0.014)*	**0.795 (0.013)***	0.788 (0.026)*	0.785
Bridges	0.040 (0.026)	0.016 (0.015)	**0.047 (0.031)**	0.026 (0.013)	0.009 (0.015)	0.037 (0.024)	0.044 (0.022)	0.116
Thyroid	0.641 (0.087)	0.677 (0.089)	0.710 (0.074)	0.656 (0.091)	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	0.916

Table A.30: Results using CRS with 30% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
Hamming Score (HS)								
Music	0.678 (0.022)	**0.694 (0.015)**	0.673 (0.009)	0.686 (0.015)	0.659 (0.014)	0.659 (0.013)	0.617 (0.021)	0.785
Scene	0.777 (0.009)	0.781 (0.012)	0.781 (0.015)	0.786 (0.015)	**0.813 (0.016)**	0.786 (0.007)	0.741 (0.009)	0.898
Yeast	0.743 (0.005)	0.753 (0.003)	0.742 (0.004)	0.750 (0.004)	**0.773 (0.001)**	0.767 (0.006)	0.723 (0.003)	0.789
ENRON	0.909 (0.003)	0.908 (0.001)	0.908 (0.002)	0.909 (0.002)	0.921 (0.004)	**0.932 (0.003)**	—	0.941
Medical	0.951 (0.002)	0.953 (0.002)	0.951 (0.002)	0.952 (0.001)	0.962 (0.020)	**0.972 (0.000)**	—	0.989
Slashdot	0.924 (0.004)	0.920 (0.005)	0.920 (0.005)	0.924 (0.004)	0.911 (0.011)	**0.942 (0.006)**	—	0.953
LLOG	0.977 (0.001)	0.977 (0.001)	0.977 (0.001)	0.978 (0.001)	0.979 (0.001)	**0.983 (0.001)***	—	0.983
Solar Flare	0.903 (0.024)	**0.910 (0.007)***	**0.910 (0.007)***	0.904 (0.015)	**0.910 (0.007)***	**0.910 (0.007)***	0.865 (0.007)	0.905
Bridges	0.502 (0.082)	0.474 (0.055)	0.518 (0.028)	0.518 (0.032)	0.426 (0.022)	**0.558 (0.034)**	0.529 (0.025)	0.580
Thyroid	0.938 (0.021)	0.948 (0.022)	0.944 (0.021)	0.935 (0.025)	**0.961 (0.000)**	**0.961 (0.000)**	0.959 (0.000)	0.986
Exact Match Ratio (EMR)								
Music	0.117 (0.029)	0.121 (0.020)	0.108 (0.018)	**0.128 (0.023)**	0.107 (0.020)	0.103 (0.022)	0.068 (0.015)	0.249
Scene	0.202 (0.014)	0.234 (0.017)	0.204 (0.016)	0.221 (0.016)	**0.294 (0.028)**	0.140 (0.023)	0.196 (0.019)	0.519
Yeast	0.052 (0.005)	0.056 (0.006)	0.051 (0.008)	0.058 (0.004)	0.059 (0.004)	**0.064 (0.011)**	0.026 (0.005)	0.117
ENRON	0.002 (0.002)	0.002 (0.002)	0.002 (0.001)	0.003 (0.002)	0.017 (0.005)	**0.025 (0.008)**	—	0.050
Medical	0.025 (0.007)	0.030 (0.008)	0.032 (0.007)	0.029 (0.007)	**0.206 (0.022)**	0.069 (0.077)	—	0.625
Slashdot	0.026 (0.009)	0.028 (0.006)	0.034 (0.006)	0.027 (0.007)	**0.143 (0.009)**	0.097 (0.013)	—	0.247
LLOG	0.099 (0.012)	0.096 (0.013)	0.101 (0.011)	0.104 (0.009)	0.165 (0.014)	**0.166 (0.007)**	—	0.205
Solar Flare	0.776 (0.056)	**0.795 (0.015)***	**0.795 (0.015)***	0.781 (0.035)	0.793 (0.015)*	**0.795 (0.015)***	**0.795 (0.016)***	0.785
Bridges	0.035 (0.036)	0.026 (0.022)	**0.047 (0.029)**	0.030 (0.015)	0.012 (0.019)	0.042 (0.039)	0.040 (0.026)	0.116
Thyroid	0.645 (0.087)	0.695 (0.084)	0.672 (0.086)	0.638 (0.089)	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.003)**	0.916

Table A.31: Results using CRS with 40% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.670 (0.015)	**0.691 (0.018)**	0.680 (0.022)	**0.691 (0.024)**	0.660 (0.010)	0.664 (0.010)	0.602 (0.024)	0.785
Scene	0.771 (0.014)	0.781 (0.016)	0.777 (0.015)	0.781 (0.009)	**0.814 (0.020)**	0.801 (0.013)	0.738 (0.007)	0.898
Yeast	0.742 (0.003)	0.749 (0.007)	0.735 (0.008)	0.746 (0.005)	**0.773 (0.002)**	0.770 (0.006)	0.723 (0.003)	0.789
ENRON	0.908 (0.001)	0.907 (0.002)	0.907 (0.002)	0.907 (0.002)	0.919 (0.003)	**0.932 (0.004)**	—	0.941
Medical	0.951 (0.002)	0.951 (0.001)	0.951 (0.002)	0.952 (0.001)	0.964 (0.014)	**0.970 (0.004)**	—	0.989
Slashdot	0.921 (0.007)	0.920 (0.004)	0.918 (0.008)	0.920 (0.006)	0.911 (0.010)	**0.938 (0.003)**	—	0.953
LLOG	0.975 (0.001)	0.976 (0.001)	0.975 (0.002)	0.976 (0.001)	0.980 (0.002)	**0.983 (0.001)***	—	0.983
Solar Flare	0.908 (0.007)*	0.904 (0.021)	0.910 (0.005)*	0.906 (0.014)*	0.910 (0.005)*	**0.911 (0.005)***	0.865 (0.007)	0.905
Bridges	0.525 (0.054)	0.501 (0.039)	0.505 (0.035)	0.508 (0.054)	0.492 (0.034)	**0.550 (0.027)**	0.519 (0.025)	0.580
Thyroid	0.944 (0.030)	0.926 (0.031)	0.942 (0.022)	0.942 (0.022)	0.958 (0.010)	**0.961 (0.001)**	0.959 (0.000)	0.986
				Exact Match Ratio (EMR)				
Music	0.098 (0.019)	0.124 (0.021)	0.111 (0.022)	**0.135 (0.026)**	0.106 (0.016)	0.092 (0.028)	0.062 (0.020)	0.249
Scene	0.200 (0.013)	0.219 (0.013)	0.203 (0.017)	0.216 (0.018)	**0.294 (0.029)**	0.152 (0.023)	0.195 (0.015)	0.519
Yeast	0.048 (0.007)	0.054 (0.004)	0.045 (0.008)	0.055 (0.006)	0.056 (0.006)	**0.061 (0.008)**	0.025 (0.006)	0.117
ENRON	0.002 (0.001)	0.002 (0.001)	0.002 (0.002)	0.002 (0.001)	0.019 (0.005)	**0.035 (0.012)**	—	0.050
Medical	0.035 (0.007)	0.030 (0.007)	0.032 (0.010)	0.031 (0.006)	**0.206 (0.017)**	0.056 (0.067)	—	0.625
Slashdot	0.029 (0.007)	0.027 (0.005)	0.027 (0.006)	0.028 (0.007)	**0.150 (0.014)**	0.115 (0.010)	—	0.247
LLOG	0.088 (0.012)	0.093 (0.009)	0.089 (0.013)	0.094 (0.008)	**0.164 (0.018)**	**0.164 (0.013)**	—	0.205
Solar Flare	0.789 (0.016)*	0.775 (0.059)	0.793 (0.009)*	0.782 (0.041)	0.792 (0.009)*	**0.795 (0.008)***	0.794 (0.015)*	0.785
Bridges	**0.035 (0.026)**	0.018 (0.014)	0.030 (0.024)	0.023 (0.018)	0.019 (0.018)	0.021 (0.016)	**0.035 (0.019)**	0.116
Thyroid	0.688 (0.082)	0.623 (0.080)	0.660 (0.089)	0.657 (0.092)	0.726 (0.055)	**0.743 (0.004)**	**0.743 (0.003)**	0.916

Table A.32: Results using CRS with 50% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.675 (0.016)	0.687 (0.015)	0.679 (0.019)	**0.697 (0.013)**	0.662 (0.007)	0.679 (0.011)	0.609 (0.012)	0.785
Scene	0.772 (0.011)	0.781 (0.021)	0.777 (0.011)	0.779 (0.016)	**0.815 (0.019)**	0.803 (0.017)	0.738 (0.008)	0.898
Yeast	0.735 (0.005)	0.743 (0.006)	0.732 (0.006)	0.746 (0.006)	**0.773 (0.002)**	0.765 (0.008)	0.716 (0.005)	0.789
ENRON	0.907 (0.002)	0.905 (0.002)	0.906 (0.002)	0.906 (0.002)	0.919 (0.004)	**0.932 (0.003)**	—	0.941
Medical	0.951 (0.001)	0.950 (0.001)	0.950 (0.001)	0.951 (0.001)	0.957 (0.016)	**0.972 (0.001)**	—	0.989
Slashdot	0.915 (0.007)	0.914 (0.006)	0.911 (0.005)	0.919 (0.005)	0.903 (0.010)	**0.938 (0.003)**	—	0.953
LLOG	0.975 (0.001)	0.975 (0.001)	0.975 (0.001)	0.975 (0.001)	0.978 (0.002)	**0.983 (0.001)***	—	0.983
Solar Flare	0.906 (0.020)*	0.911 (0.008)*	0.909 (0.013)*	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.009)***	0.863 (0.010)	0.905
Bridges	0.495 (0.030)	0.524 (0.027)	0.512 (0.042)	0.520 (0.038)	0.490 (0.024)	**0.562 (0.022)**	0.508 (0.021)	0.580
Thyroid	0.914 (0.032)	0.948 (0.016)	0.939 (0.027)	0.945 (0.022)	**0.961 (0.001)**	**0.961 (0.000)**	0.959 (0.000)	0.986
				Exact Match Ratio (EMR)				
Music	0.104 (0.027)	0.137 (0.024)	0.110 (0.023)	**0.147 (0.020)**	0.112 (0.015)	0.097 (0.010)	0.069 (0.018)	0.249
Scene	0.202 (0.014)	0.228 (0.016)	0.218 (0.020)	0.221 (0.007)	**0.289 (0.026)**	0.144 (0.023)	0.193 (0.017)	0.519
Yeast	0.042 (0.008)	0.054 (0.005)	0.041 (0.006)	0.054 (0.004)	0.060 (0.004)	**0.068 (0.012)**	0.023 (0.006)	0.117
ENRON	0.001 (0.002)	0.002 (0.001)	0.002 (0.002)	0.002 (0.002)	0.016 (0.006)	**0.029 (0.009)**	—	0.050
Medical	0.032 (0.010)	0.025 (0.010)	0.029 (0.010)	0.033 (0.007)	**0.201 (0.021)**	0.048 (0.068)	—	0.625
Slashdot	0.030 (0.005)	0.030 (0.004)	0.030 (0.004)	0.028 (0.005)	**0.136 (0.005)**	0.122 (0.009)	—	0.247
LLOG	0.090 (0.012)	0.089 (0.007)	0.090 (0.011)	0.087 (0.009)	0.155 (0.018)	**0.160 (0.015)**	—	0.205
Solar Flare	0.782 (0.046)	0.794 (0.016)*	0.789 (0.028)*	0.796 (0.017)*	0.796 (0.017)*	**0.797 (0.017)***	0.786 (0.018)*	0.785
Bridges	0.028 (0.020)	0.042 (0.020)	0.037 (0.030)	0.037 (0.035)	0.023 (0.021)	**0.047 (0.026)**	0.040 (0.030)	0.116
Thyroid	0.582 (0.028)	0.678 (0.089)	0.680 (0.091)	0.674 (0.088)	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.002)**	0.916

Table A.33: Results using CRS with 60% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.667 (0.019)	0.683 (0.018)	0.662 (0.017)	0.681 (0.023)	0.663 (0.005)	**0.688 (0.008)**	0.596 (0.022)	0.785
Scene	0.782 (0.008)	0.785 (0.013)	0.770 (0.014)	0.773 (0.015)	**0.809 (0.011)**	0.801 (0.020)	0.740 (0.007)	0.898
Yeast	0.726 (0.003)	0.738 (0.005)	0.724 (0.008)	0.739 (0.008)	**0.773 (0.001)**	0.769 (0.006)	0.715 (0.006)	0.789
ENRON	0.904 (0.002)	0.906 (0.002)	0.907 (0.001)	0.905 (0.002)	0.921 (0.004)	**0.933 (0.001)**	—	0.941
Medical	0.950 (0.002)	0.950 (0.002)	0.950 (0.002)	0.951 (0.001)	0.956 (0.011)	**0.970 (0.004)**	—	0.989
Slashdot	0.906 (0.004)	0.908 (0.004)	0.907 (0.005)	0.914 (0.005)	0.902 (0.012)	**0.939 (0.000)**	—	0.953
LLOG	0.974 (0.001)	0.975 (0.001)	0.974 (0.001)	0.974 (0.001)	0.977 (0.003)	**0.983 (0.001)***	—	0.983
Solar Flare	**0.910 (0.006)***	0.908 (0.007)*	0.907 (0.010)*	0.909 (0.006)*	0.898 (0.026)	**0.910 (0.006)***	0.863 (0.008)	0.905
Bridges	0.511 (0.050)	0.525 (0.028)	0.539 (0.033)	0.527 (0.033)	0.487 (0.021)	**0.552 (0.045)**	0.482 (0.019)	0.580
Thyroid	0.946 (0.018)	0.946 (0.020)	0.925 (0.025)	0.934 (0.033)	0.958 (0.010)	**0.961 (0.001)**	0.959 (0.000)	0.986
				Exact Match Ratio (EMR)				
Music	0.109 (0.021)	0.126 (0.022)	0.092 (0.014)	**0.127 (0.026)**	0.110 (0.011)	0.094 (0.014)	0.054 (0.015)	0.249
Scene	0.211 (0.012)	0.239 (0.018)	0.207 (0.012)	0.217 (0.015)	**0.300 (0.015)**	0.160 (0.025)	0.187 (0.016)	0.519
Yeast	0.038 (0.006)	0.050 (0.007)	0.039 (0.007)	0.054 (0.010)	0.057 (0.004)	**0.068 (0.010)**	0.022 (0.006)	0.117
ENRON	0.002 (0.002)	0.002 (0.002)	0.002 (0.001)	0.001 (0.001)	0.016 (0.004)	**0.028 (0.010)**	—	0.050
Medical	0.034 (0.006)	0.031 (0.005)	0.030 (0.007)	0.031 (0.008)	**0.207 (0.018)**	0.058 (0.079)	—	0.625
Slashdot	0.029 (0.004)	0.029 (0.003)	0.028 (0.005)	0.034 (0.005)	**0.137 (0.008)**	0.120 (0.008)	—	0.247
LLOG	0.081 (0.013)	0.084 (0.009)	0.081 (0.009)	0.081 (0.011)	0.148 (0.018)	**0.161 (0.013)**	—	0.205
Solar Flare	**0.795 (0.012)***	0.790 (0.015)*	0.788 (0.025)*	0.792 (0.012)*	0.765 (0.067)	**0.795 (0.012)***	0.788 (0.018)*	0.785
Bridges	0.028 (0.025)	0.040 (0.033)	0.028 (0.020)	0.021 (0.024)	0.021 (0.016)	**0.044 (0.020)**	0.028 (0.018)	0.116
Thyroid	0.677 (0.086)	0.678 (0.088)	0.616 (0.072)	0.656 (0.093)	0.724 (0.054)	**0.743 (0.004)**	**0.743 (0.001)**	0.916

Table A.34: Results using CRS with 70% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.672 (0.019)	0.687 (0.018)	0.665 (0.016)	**0.689 (0.013)**	0.661 (0.014)	0.687 (0.007)	0.601 (0.017)	0.785
Scene	0.774 (0.014)	0.785 (0.013)	0.779 (0.020)	0.774 (0.014)	**0.820 (0.012)**	0.800 (0.027)	0.739 (0.007)	0.898
Yeast	0.725 (0.008)	0.732 (0.005)	0.717 (0.006)	0.739 (0.004)	**0.774 (0.002)**	0.766 (0.006)	0.714 (0.004)	0.789
ENRON	0.906 (0.002)	0.904 (0.002)	0.906 (0.002)	0.905 (0.002)	0.918 (0.005)	**0.933 (0.002)**	—	0.941
Medical	0.950 (0.002)	0.949 (0.002)	0.950 (0.001)	0.950 (0.002)	0.949 (0.015)	**0.970 (0.003)**	—	0.989
Slashdot	0.907 (0.003)	0.908 (0.005)	0.906 (0.004)	0.911 (0.005)	0.904 (0.013)	**0.937 (0.004)**	—	0.953
LLOG	0.974 (0.001)	0.974 (0.001)	0.974 (0.001)	0.974 (0.001)	0.976 (0.002)	**0.983 (0.001)***	—	0.983
Solar Flare	0.903 (0.018)	0.899 (0.023)	0.909 (0.007)*	0.892 (0.025)	0.909 (0.011)*	**0.912 (0.007)***	0.856 (0.012)	0.905
Bridges	0.517 (0.039)	0.521 (0.045)	0.509 (0.057)	0.495 (0.071)	0.493 (0.023)	**0.552 (0.037)**	0.492 (0.042)	0.580
Thyroid	0.928 (0.027)	0.923 (0.019)	0.946 (0.021)	0.945 (0.022)	0.958 (0.010)	**0.961 (0.001)**	0.959 (0.000)	0.986
				Exact Match Ratio (EMR)				
Music	0.100 (0.020)	0.125 (0.014)	0.094 (0.020)	**0.138 (0.018)**	0.112 (0.021)	0.080 (0.027)	0.067 (0.014)	0.249
Scene	0.208 (0.013)	0.224 (0.018)	0.210 (0.019)	0.219 (0.020)	**0.290 (0.019)**	0.143 (0.022)	0.188 (0.016)	0.519
Yeast	0.038 (0.008)	0.048 (0.005)	0.033 (0.005)	0.052 (0.004)	0.058 (0.004)	**0.064 (0.010)**	0.021 (0.005)	0.117
ENRON	0.002 (0.002)	0.002 (0.002)	0.002 (0.001)	0.002 (0.003)	0.016 (0.006)	**0.022 (0.007)**	—	0.050
Medical	0.030 (0.010)	0.030 (0.007)	0.028 (0.006)	0.030 (0.005)	**0.208 (0.019)**	0.104 (0.087)	—	0.625
Slashdot	0.030 (0.004)	0.029 (0.006)	0.029 (0.004)	0.034 (0.006)	**0.136 (0.008)**	0.114 (0.016)	—	0.247
LLOG	0.083 (0.010)	0.075 (0.005)	0.078 (0.011)	0.081 (0.007)	0.151 (0.018)	**0.164 (0.012)**	—	0.205
Solar Flare	0.774 (0.049)	0.766 (0.055)	0.791 (0.016)*	0.746 (0.067)	0.791 (0.026)*	**0.796 (0.017)***	0.773 (0.019)	0.785
Bridges	0.035 (0.032)	0.037 (0.019)	0.037 (0.022)	0.021 (0.030)	0.018 (0.020)	0.030 (0.021)	**0.042 (0.021)**	0.116
Thyroid	0.628 (0.078)	0.587 (0.046)	0.689 (0.080)	0.676 (0.088)	0.725 (0.055)	**0.743 (0.003)**	**0.743 (0.002)**	0.916

Table A.35: Results using CRS with 80% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.660 (0.022)	0.682 (0.025)	0.648 (0.013)	**0.699 (0.018)**	0.687 (0.005)	0.653 (0.008)	0.590 (0.018)	0.785
Scene	0.770 (0.012)	0.767 (0.021)	0.776 (0.015)	0.775 (0.012)	**0.813 (0.013)**	0.787 (0.015)	0.735 (0.010)	0.898
Yeast	0.724 (0.006)	0.728 (0.005)	0.720 (0.007)	0.730 (0.004)	**0.774 (0.002)**	0.772 (0.005)	0.708 (0.009)	0.789
ENRON	0.904 (0.002)	0.904 (0.002)	0.905 (0.003)	0.905 (0.001)	0.919 (0.006)	**0.930 (0.004)**	—	0.941
Medical	0.949 (0.001)	0.949 (0.002)	0.950 (0.001)	0.951 (0.001)	0.942 (0.012)	**0.970 (0.003)**	—	0.989
Slashdot	0.904 (0.004)	0.906 (0.004)	0.901 (0.003)	0.912 (0.007)	0.895 (0.012)	**0.939 (0.001)**	—	0.953
LLOG	0.973 (0.001)	0.974 (0.001)	0.974 (0.001)	0.974 (0.001)	0.976 (0.003)	**0.982 (0.002)**	—	0.983
Solar Flare	**0.911 (0.006)***	0.900 (0.027)	0.906 (0.008)*	0.906 (0.013)*	0.907 (0.008)*	**0.911 (0.006)***	0.856 (0.014)	0.905
Bridges	0.536 (0.030)	0.472 (0.066)	0.499 (0.039)	0.493 (0.040)	0.462 (0.038)	**0.542 (0.030)**	0.499 (0.055)	0.580
Thyroid	0.930 (0.022)	0.931 (0.031)	0.909 (0.032)	0.928 (0.015)	0.958 (0.010)	**0.961 (0.001)**	0.959 (0.001)	0.986
				Exact Match Ratio (EMR)				
Music	0.094 (0.012)	0.114 (0.027)	0.098 (0.020)	**0.133 (0.034)**	0.088 (0.020)	0.114 (0.016)	0.056 (0.014)	0.249
Scene	0.199 (0.010)	0.215 (0.017)	0.208 (0.015)	0.206 (0.011)	**0.291 (0.024)**	0.130 (0.025)	0.188 (0.022)	0.519
Yeast	0.039 (0.006)	0.039 (0.007)	0.039 (0.008)	0.049 (0.008)	0.056 (0.003)	**0.058 (0.008)**	0.020 (0.006)	0.117
ENRON	0.001 (0.001)	0.002 (0.003)	0.002 (0.002)	0.001 (0.001)	0.017 (0.004)	**0.027 (0.009)**	—	0.050
Medical	0.029 (0.006)	0.027 (0.008)	0.029 (0.005)	0.032 (0.006)	**0.211 (0.027)**	0.111 (0.075)	—	0.625
Slashdot	0.028 (0.004)	0.030 (0.005)	0.029 (0.002)	0.030 (0.004)	**0.131 (0.005)**	0.123 (0.005)	—	0.247
LLOG	0.074 (0.008)	0.082 (0.005)	0.077 (0.007)	0.081 (0.010)	0.145 (0.014)	**0.160 (0.017)**	—	0.205
Solar Flare	**0.796 (0.013)***	0.778 (0.044)	0.782 (0.032)	0.786 (0.027)*	0.784 (0.021)	**0.796 (0.013)***	0.772 (0.029)	0.785
Bridges	0.026 (0.024)	0.028 (0.025)	0.032 (0.028)	0.021 (0.022)	0.012 (0.012)	0.019 (0.018)	**0.040 (0.031)**	0.116
Thyroid	0.631 (0.078)	0.655 (0.090)	0.602 (0.054)	0.584 (0.055)	0.726 (0.052)	**0.743 (0.003)**	**0.743 (0.003)**	0.916

Table A.36: Results using CRS with 90% of the original features.

A.5 Super-Class Classifier (SCC) Results

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
Hamming Score (HS)								
Music	0.665 (0.038)	0.681 (0.017)	0.672 (0.017)	**0.694 (0.017)**	0.623 (0.013)	0.592 (0.016)	0.589 (0.015)	0.712
Scene	0.781 (0.017)	0.786 (0.006)	0.786 (0.011)	0.782 (0.007)	**0.803 (0.008)**	0.763 (0.010)	0.744 (0.006)	0.845
Yeast	0.680 (0.004)	0.679 (0.004)	0.680 (0.005)	0.679 (0.004)	0.697 (0.008)	**0.708 (0.009)**	0.652 (0.003)	0.724
ENRON	0.909 (0.002)	0.908 (0.001)	0.909 (0.001)	0.908 (0.002)	0.917 (0.003)	**0.920 (0.002)**	0.871 (0.002)	0.923
Medical	0.951 (0.001)	0.952 (0.001)	0.951 (0.001)	0.952 (0.002)	0.963 (0.003)	**0.965 (0.004)**	—	0.989
Slashdot	—	—	—	—	—	—	—	—
LLOG	—	—	—	—	—	—	—	—
Solar Flare	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	**0.912 (0.008)***	0.904 (0.023)	0.865 (0.008)	0.905
Bridges	—	—	—	—	—	—	—	0.570
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.959 (0.001)	0.986
Exact Match Ratio (EMR)								
Music	0.164 (0.039)	0.175 (0.017)	0.168 (0.019)	**0.201 (0.030)***	0.154 (0.020)	0.126 (0.022)	0.106 (0.018)	0.186
Scene	0.340 (0.047)	0.351 (0.016)	0.352 (0.027)	0.336 (0.019)	**0.395 (0.018)**	0.297 (0.025)	0.342 (0.015)	0.522
Yeast	0.038 (0.004)	0.039 (0.006)	0.042 (0.008)	0.042 (0.007)	0.083 (0.010)	**0.088 (0.004)**	0.053 (0.006)	0.134
ENRON	0.012 (0.004)	0.011 (0.004)	0.009 (0.003)	0.008 (0.003)	0.030 (0.005)*	**0.044 (0.013)***	0.026 (0.004)*	0.026
Medical	0.067 (0.009)	0.067 (0.018)	0.060 (0.012)	0.069 (0.016)	0.186 (0.028)	**0.236 (0.024)**	—	0.645
Slashdot	—	—	—	—	—	—	—	—
LLOG	—	—	—	—	—	—	—	—
Solar Flare	**0.796 (0.013)***	**0.796 (0.013)***	**0.796 (0.013)***	**0.796 (0.013)***	**0.796 (0.013)***	0.775 (0.062)	0.794 (0.012)*	0.785
Bridges	—	—	—	—	—	—	—	0.163
Thyroid	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.003)**	0.924

Table A.37: Results using SCC with 10% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
Hamming Score (HS)								
Music	0.673 (0.026)	0.674 (0.019)	0.676 (0.017)	**0.685 (0.012)**	0.620 (0.014)	0.592 (0.019)	0.581 (0.020)	0.712
Scene	0.773 (0.015)	0.783 (0.008)	0.772 (0.006)	0.774 (0.003)	**0.800 (0.009)**	0.760 (0.004)	0.737 (0.010)	0.845
Yeast	0.680 (0.003)	0.678 (0.005)	0.681 (0.005)	0.681 (0.007)	0.700 (0.006)	**0.701 (0.010)**	0.648 (0.004)	0.724
ENRON	0.909 (0.003)	0.909 (0.002)	0.909 (0.002)	0.909 (0.001)	**0.919 (0.002)**	**0.919 (0.003)**	0.870 (0.003)	0.923
Medical	0.951 (0.001)	0.952 (0.001)	0.951 (0.001)	0.952 (0.001)	0.956 (0.007)	**0.964 (0.005)**	—	0.989
Slashdot	—	—	—	—	—	—	—	—
LLOG	—	—	—	—	—	—	—	—
Solar Flare	0.910 (0.008)*	**0.911 (0.008)***	0.908 (0.009)*	**0.911 (0.008)***	0.910 (0.009)*	**0.911 (0.008)***	0.866 (0.004)	0.905
Bridges	0.492 (0.031)	0.462 (0.052)	0.482 (0.042)	0.450 (0.060)	0.406 (0.027)	**0.511 (0.037)**	0.495 (0.041)	0.570
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.959 (0.000)	0.986
Exact Match Ratio (EMR)								
Music	0.159 (0.029)	**0.175 (0.019)**	0.161 (0.023)	0.174 (0.019)	0.145 (0.015)	0.124 (0.017)	0.110 (0.020)	0.186
Scene	0.318 (0.041)	0.346 (0.018)	0.315 (0.016)	0.322 (0.008)	**0.391 (0.022)**	0.292 (0.013)	0.323 (0.027)	0.522
Yeast	0.040 (0.008)	0.036 (0.007)	0.039 (0.006)	0.038 (0.008)	0.086 (0.006)	**0.087 (0.009)**	0.049 (0.009)	0.134
ENRON	0.012 (0.005)	0.010 (0.005)	0.013 (0.005)	0.013 (0.004)	0.027 (0.005)*	**0.037 (0.013)***	0.029 (0.011)*	0.026
Medical	0.066 (0.009)	0.063 (0.011)	0.073 (0.016)	0.068 (0.012)	0.202 (0.015)	**0.222 (0.027)**	—	0.645
Slashdot	—	—	—	—	—	—	—	—
LLOG	—	—	—	—	—	—	—	—
Solar Flare	0.792 (0.015)*	0.794 (0.015)*	0.785 (0.024)*	0.794 (0.015)*	0.792 (0.017)*	0.794 (0.015)*	**0.796 (0.010)***	0.785
Bridges	0.030 (0.015)	0.021 (0.022)	0.030 (0.015)	0.026 (0.032)	0.007 (0.011)	0.016 (0.015)	**0.044 (0.020)**	0.163
Thyroid	**0.743 (0.003)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.003)**	0.924

Table A.38: Results using SCC with 20% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.665 (0.030)	**0.686 (0.014)**	0.669 (0.019)	0.671 (0.023)	0.625 (0.020)	0.599 (0.014)	0.588 (0.018)	0.712
Scene	0.764 (0.016)	0.776 (0.007)	0.768 (0.010)	0.773 (0.007)	**0.802 (0.004)**	0.762 (0.008)	0.737 (0.009)	0.845
Yeast	0.679 (0.003)	0.677 (0.004)	0.679 (0.003)	0.677 (0.006)	0.700 (0.006)	**0.706 (0.007)**	0.649 (0.005)	0.724
ENRON	0.909 (0.001)	0.909 (0.001)	0.910 (0.001)	0.908 (0.002)	0.915 (0.003)	**0.920 (0.003)**	0.868 (0.003)	0.923
Medical	0.951 (0.001)	0.950 (0.001)	0.951 (0.001)	0.951 (0.001)	0.949 (0.007)	**0.963 (0.004)**	—	0.989
Slashdot	—	—	—	—	—	—	—	—
LLOG	—	—	—	—	—	—	—	—
Solar Flare	**0.910 (0.006)***	**0.910 (0.006)***	**0.910 (0.006)***	**0.910 (0.006)***	**0.910 (0.006)***	**0.910 (0.006)***	0.865 (0.006)	0.905
Bridges	0.498 (0.048)	0.517 (0.027)	0.478 (0.063)	0.479 (0.082)	0.408 (0.021)	**0.519 (0.039)**	0.488 (0.031)	0.570
Thyroid	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	0.951 (0.016)	**0.961 (0.001)**	0.959 (0.000)	0.986
				Exact Match Ratio (EMR)				
Music	0.159 (0.025)	**0.185 (0.021)**	0.165 (0.024)	0.162 (0.024)	0.153 (0.019)	0.123 (0.019)	0.118 (0.021)	0.186
Scene	0.293 (0.046)	0.327 (0.016)	0.307 (0.026)	0.314 (0.017)	**0.395 (0.014)**	0.292 (0.024)	0.320 (0.024)	0.522
Yeast	0.040 (0.005)	0.038 (0.007)	0.039 (0.005)	0.038 (0.003)	0.087 (0.008)	**0.088 (0.010)**	0.052 (0.005)	0.134
ENRON	0.012 (0.005)	0.014 (0.005)	0.012 (0.005)	0.011 (0.005)	0.027 (0.005)*	**0.040 (0.011)***	0.019 (0.010)	0.026
Medical	0.061 (0.009)	0.062 (0.014)	0.067 (0.013)	0.067 (0.024)	**0.216 (0.016)**	0.214 (0.030)	—	0.645
Slashdot	—	—	—	—	—	—	—	—
LLOG	—	—	—	—	—	—	—	—
Solar Flare	0.794 (0.012)*	0.794 (0.012)*	0.794 (0.012)*	0.794 (0.012)*	0.794 (0.012)*	0.794 (0.012)*	**0.795 (0.015)***	0.785
Bridges	0.023 (0.023)	0.023 (0.018)	0.035 (0.032)	**0.044 (0.028)**	0.026 (0.024)	0.018 (0.014)	0.042 (0.027)	0.163
Thyroid	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	0.688 (0.085)	**0.743 (0.003)**	**0.743 (0.003)**	0.924

Table A.39: Results using SCC with 30% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.652 (0.028)	0.678 (0.022)	0.660 (0.013)	**0.680 (0.016)**	0.624 (0.015)	0.598 (0.015)	0.578 (0.019)	0.712
Scene	0.765 (0.015)	0.780 (0.009)	0.774 (0.007)	0.774 (0.005)	**0.806 (0.006)**	0.760 (0.005)	0.733 (0.008)	0.845
Yeast	0.681 (0.003)	0.679 (0.004)	0.677 (0.006)	0.680 (0.006)	0.699 (0.008)	**0.706 (0.010)**	0.649 (0.003)	0.724
ENRON	0.909 (0.003)	0.909 (0.001)	0.909 (0.002)	0.908 (0.001)	0.914 (0.003)	**0.920 (0.002)**	—	0.923
Medical	0.951 (0.001)	0.951 (0.001)	0.950 (0.001)	0.951 (0.001)	**0.967 (0.006)**	0.966 (0.006)	—	0.989
Slashdot	—	—	—	—	—	—	—	—
LLOG	—	—	—	—	—	—	—	—
Solar Flare	**0.910 (0.008)***	0.909 (0.008)*	**0.910 (0.008)***	0.909 (0.009)*	**0.910 (0.008)***	**0.910 (0.008)***	0.865 (0.007)	0.905
Bridges	0.490 (0.053)	0.467 (0.042)	**0.497 (0.044)**	0.465 (0.025)	0.407 (0.039)	0.491 (0.055)	0.460 (0.066)	0.570
Thyroid	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	0.951 (0.016)	**0.961 (0.001)**	0.959 (0.000)	0.986
				Exact Match Ratio (EMR)				
Music	0.142 (0.029)	**0.172 (0.027)**	0.149 (0.018)	0.166 (0.019)	0.145 (0.019)	0.125 (0.016)	0.100 (0.019)	0.186
Scene	0.297 (0.042)	0.337 (0.023)	0.319 (0.018)	0.321 (0.016)	**0.404 (0.013)**	0.290 (0.012)	0.312 (0.019)	0.522
Yeast	0.041 (0.005)	0.040 (0.004)	0.039 (0.004)	0.038 (0.004)	0.083 (0.007)	**0.091 (0.013)**	0.050 (0.004)	0.134
ENRON	0.011 (0.004)	0.013 (0.003)	0.012 (0.004)	0.013 (0.002)	0.022 (0.005)	**0.034 (0.008)***	—	0.026
Medical	0.066 (0.013)	0.060 (0.008)	0.057 (0.016)	0.066 (0.006)	**0.217 (0.020)**	0.121 (0.099)	—	0.645
Slashdot	—	—	—	—	—	—	—	—
LLOG	—	—	—	—	—	—	—	—
Solar Flare	0.793 (0.019)*	0.793 (0.017)*	0.794 (0.018)*	0.791 (0.024)*	0.794 (0.018)*	0.794 (0.018)*	**0.795 (0.016)***	0.785
Bridges	**0.047 (0.018)**	0.009 (0.011)	0.044 (0.020)	0.042 (0.020)	0.016 (0.015)	0.033 (0.022)	**0.047 (0.031)**	0.163
Thyroid	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	0.689 (0.084)	**0.743 (0.003)**	**0.743 (0.003)**	0.924

Table A.40: Results using SCC with 40% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
Hamming Score (HS)								
Music	0.646 (0.030)	0.675 (0.014)	0.653 (0.019)	**0.686 (0.018)**	0.628 (0.013)	0.600 (0.015)	0.575 (0.020)	0.712
Scene	0.766 (0.014)	0.782 (0.010)	0.769 (0.007)	0.778 (0.006)	**0.804 (0.004)**	0.764 (0.008)	0.735 (0.005)	0.845
Yeast	0.677 (0.005)	0.677 (0.004)	0.678 (0.005)	0.678 (0.006)	**0.699 (0.008)**	0.694 (0.012)	0.650 (0.004)	0.724
ENRON	0.909 (0.002)	0.909 (0.002)	0.909 (0.001)	0.909 (0.002)	0.916 (0.004)	**0.922 (0.002)**	—	0.923
Medical	0.950 (0.001)	0.951 (0.001)	0.950 (0.001)	0.951 (0.001)	0.966 (0.009)	**0.967 (0.006)**	—	0.989
Slashdot	—	—	—	—	—	—	—	—
LLOG	—	—	—	—	—	—	—	—
Solar Flare	0.910 (0.005)*	**0.911 (0.005)***	**0.911 (0.005)***	**0.911 (0.005)***	**0.911 (0.005)***	0.894 (0.034)	0.863 (0.009)	0.905
Bridges	0.501 (0.044)	0.495 (0.046)	0.508 (0.052)	0.497 (0.037)	0.434 (0.034)	**0.527 (0.037)**	0.474 (0.032)	0.570
Thyroid	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	**0.961 (0.001)**	0.947 (0.017)	**0.961 (0.001)**	0.959 (0.000)	0.986
Exact Match Ratio (EMR)								
Music	0.139 (0.036)	0.165 (0.025)	0.148 (0.027)	**0.180 (0.024)**	0.148 (0.029)	0.125 (0.021)	0.110 (0.025)	0.186
Scene	0.300 (0.038)	0.339 (0.023)	0.312 (0.016)	0.331 (0.014)	**0.401 (0.012)**	0.302 (0.019)	0.317 (0.019)	0.522
Yeast	0.040 (0.007)	0.038 (0.003)	0.037 (0.005)	0.040 (0.005)	**0.085 (0.006)**	0.084 (0.007)	0.051 (0.006)	0.134
ENRON	0.012 (0.003)	0.012 (0.004)	0.010 (0.003)	0.012 (0.003)	0.026 (0.003)*	**0.048 (0.012)***	—	0.026
Medical	0.055 (0.005)	0.060 (0.012)	0.056 (0.012)	0.065 (0.012)	**0.230 (0.023)**	0.093 (0.106)	—	0.645
Slashdot	—	—	—	—	—	—	—	—
LLOG	—	—	—	—	—	—	—	—
Solar Flare	0.793 (0.009)*	**0.795 (0.010)***	**0.795 (0.010)***	**0.795 (0.010)***	**0.795 (0.010)***	0.748 (0.094)	0.788 (0.019)*	0.785
Bridges	0.035 (0.026)	0.040 (0.026)	**0.047 (0.021)**	0.042 (0.023)	0.019 (0.029)	0.026 (0.013)	0.019 (0.025)	0.163
Thyroid	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	0.670 (0.091)	**0.743 (0.003)**	0.742 (0.003)	0.924

Table A.41: Results using SCC with 50% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
Hamming Score (HS)								
Music	0.656 (0.029)	**0.686 (0.020)**	0.657 (0.018)	0.668 (0.013)	0.638 (0.009)	0.603 (0.012)	0.586 (0.010)	0.712
Scene	0.770 (0.016)	0.778 (0.009)	0.777 (0.006)	0.777 (0.008)	**0.805 (0.006)**	0.761 (0.006)	0.733 (0.010)	0.845
Yeast	0.677 (0.005)	0.678 (0.004)	0.678 (0.004)	0.677 (0.003)	**0.702 (0.006)**	0.698 (0.015)	0.647 (0.005)	0.724
ENRON	0.909 (0.001)	0.908 (0.002)	0.909 (0.002)	0.909 (0.001)	0.914 (0.004)	**0.920 (0.003)**	—	0.923
Medical	0.950 (0.001)	0.950 (0.001)	0.950 (0.002)	0.950 (0.001)	0.958 (0.008)	**0.967 (0.006)**	—	0.989
Slashdot	—	—	—	—	—	—	—	—
LLOG	—	—	—	—	—	—	—	—
Solar Flare	**0.911 (0.004)***	**0.911 (0.004)***	**0.911 (0.004)***	0.910 (0.005)*	0.909 (0.003)*	**0.911 (0.004)***	0.859 (0.008)	0.905
Bridges	0.508 (0.029)	**0.532 (0.017)**	0.509 (0.051)	0.528 (0.032)	0.450 (0.029)	0.506 (0.049)	0.470 (0.046)	0.570
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.951 (0.016)	**0.961 (0.000)**	0.958 (0.002)	0.986
Exact Match Ratio (EMR)								
Music	0.152 (0.033)	**0.178 (0.016)**	0.159 (0.020)	0.158 (0.021)	0.153 (0.017)	0.118 (0.012)	0.106 (0.019)	0.186
Scene	0.308 (0.041)	0.332 (0.022)	0.327 (0.015)	0.328 (0.020)	**0.402 (0.016)**	0.292 (0.018)	0.309 (0.026)	0.522
Yeast	0.039 (0.007)	0.040 (0.007)	0.037 (0.003)	0.039 (0.004)	0.089 (0.006)	**0.090 (0.012)**	0.047 (0.004)	0.134
ENRON	0.011 (0.005)	0.011 (0.004)	0.012 (0.006)	0.010 (0.003)	0.026 (0.004)*	**0.044 (0.013)***	—	0.026
Medical	0.061 (0.006)	0.066 (0.010)	0.062 (0.015)	0.056 (0.014)	**0.218 (0.017)**	0.084 (0.095)	—	0.645
Slashdot	—	—	—	—	—	—	—	—
LLOG	—	—	—	—	—	—	—	—
Solar Flare	**0.795 (0.010)***	**0.795 (0.010)***	**0.795 (0.010)***	0.794 (0.011)*	0.792 (0.010)*	**0.795 (0.010)***	0.780 (0.020)	0.785
Bridges	0.030 (0.015)	0.032 (0.026)	0.033 (0.026)	0.037 (0.019)	0.014 (0.019)	0.021 (0.020)	**0.042 (0.027)**	0.163
Thyroid	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	0.689 (0.084)	**0.743 (0.002)**	0.736 (0.010)	0.924

Table A.42: Results using SCC with 60% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.651 (0.025)	0.672 (0.020)	0.655 (0.015)	**0.683 (0.015)**	0.630 (0.016)	0.606 (0.014)	0.576 (0.023)	0.712
Scene	0.769 (0.014)	0.782 (0.003)	0.773 (0.005)	0.773 (0.009)	**0.803 (0.005)**	0.765 (0.006)	0.731 (0.007)	0.845
Yeast	0.678 (0.003)	0.675 (0.004)	0.680 (0.003)	0.675 (0.005)	**0.701 (0.007)**	0.698 (0.008)	0.644 (0.006)	0.724
ENRON	0.908 (0.001)	0.908 (0.001)	0.909 (0.001)	0.909 (0.001)	0.914 (0.003)	**0.921 (0.002)**	—	0.923
Medical	0.949 (0.002)	0.950 (0.001)	0.950 (0.001)	0.951 (0.001)	0.957 (0.009)	**0.964 (0.008)**	—	0.989
Slashdot	—	—	—	—	—	—	—	—
LLOG	—	—	—	—	—	—	—	—
Solar Flare	**0.910 (0.005)***	0.908 (0.007)*	**0.910 (0.005)***	**0.910 (0.005)***	**0.910 (0.005)***	**0.910 (0.005)***	0.864 (0.009)	0.905
Bridges	0.506 (0.076)	0.495 (0.047)	0.492 (0.041)	0.504 (0.045)	0.455 (0.036)	**0.516 (0.051)**	0.471 (0.056)	0.570
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.947 (0.017)	**0.961 (0.000)**	0.948 (0.009)	0.986
				Exact Match Ratio (EMR)				
Music	0.134 (0.029)	**0.174 (0.028)**	0.142 (0.020)	0.165 (0.012)	0.149 (0.023)	0.115 (0.025)	0.103 (0.023)	0.186
Scene	0.304 (0.035)	0.340 (0.008)	0.317 (0.014)	0.320 (0.024)	**0.400 (0.012)**	0.298 (0.012)	0.303 (0.019)	0.522
Yeast	0.044 (0.007)	0.039 (0.004)	0.039 (0.008)	0.038 (0.006)	**0.090 (0.008)**	0.083 (0.012)	0.047 (0.008)	0.134
ENRON	0.010 (0.004)	0.010 (0.004)	0.010 (0.003)	0.010 (0.004)	0.025 (0.004)	**0.046 (0.012)***	—	0.026
Medical	0.052 (0.009)	0.058 (0.011)	0.056 (0.008)	0.065 (0.019)	**0.226 (0.020)**	0.094 (0.093)	—	0.645
Slashdot	—	—	—	—	—	—	—	—
LLOG	—	—	—	—	—	—	—	—
Solar Flare	0.793 (0.009)*	0.789 (0.017)*	**0.794 (0.008)***	**0.794 (0.008)***	**0.794 (0.008)***	**0.794 (0.008)***	0.792 (0.018)*	0.785
Bridges	0.037 (0.030)	0.042 (0.029)	0.033 (0.019)	**0.049 (0.027)**	0.016 (0.021)	0.026 (0.020)	0.030 (0.024)	0.163
Thyroid	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	0.670 (0.091)	**0.743 (0.003)**	0.688 (0.048)	0.924

Table A.43: Results using SCC with 70% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
				Hamming Score (HS)				
Music	0.653 (0.030)	0.664 (0.028)	0.650 (0.015)	**0.684 (0.013)**	0.636 (0.017)	0.605 (0.022)	0.573 (0.020)	0.712
Scene	0.763 (0.013)	0.780 (0.007)	0.782 (0.009)	0.775 (0.006)	**0.805 (0.006)**	0.764 (0.006)	0.733 (0.011)	0.845
Yeast	0.679 (0.004)	0.677 (0.005)	0.680 (0.007)	0.678 (0.004)	**0.697 (0.006)**	0.694 (0.006)	0.648 (0.005)	0.724
ENRON	0.909 (0.002)	0.909 (0.002)	0.909 (0.002)	0.909 (0.002)	0.915 (0.003)	**0.918 (0.003)**	—	0.923
Medical	0.950 (0.001)	0.950 (0.001)	0.950 (0.001)	0.950 (0.001)	0.959 (0.009)	**0.966 (0.006)**	—	0.989
Slashdot	—	—	—	—	—	—	—	—
LLOG	—	—	—	—	—	—	—	—
Solar Flare	**0.911 (0.007)***	0.910 (0.007)*	0.910 (0.007)*	**0.911 (0.007)***	0.905 (0.021)*	**0.911 (0.006)***	0.861 (0.015)	0.905
Bridges	**0.521 (0.032)**	0.493 (0.023)	0.491 (0.024)	0.488 (0.054)	0.431 (0.023)	0.520 (0.057)	0.441 (0.058)	0.570
Thyroid	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	**0.961 (0.000)**	0.951 (0.015)	**0.961 (0.000)**	0.936 (0.004)	0.986
				Exact Match Ratio (EMR)				
Music	0.156 (0.030)	0.158 (0.020)	0.136 (0.024)	**0.174 (0.024)**	0.152 (0.023)	0.125 (0.027)	0.095 (0.018)	0.186
Scene	0.290 (0.032)	0.336 (0.016)	0.342 (0.021)	0.326 (0.015)	**0.400 (0.014)**	0.296 (0.012)	0.309 (0.030)	0.522
Yeast	0.039 (0.006)	0.042 (0.010)	0.041 (0.006)	0.040 (0.007)	**0.084 (0.007)**	0.078 (0.006)	0.049 (0.005)	0.134
ENRON	0.009 (0.004)	0.010 (0.005)	0.011 (0.003)	0.013 (0.006)	0.023 (0.006)	**0.034 (0.007)***	—	0.026
Medical	0.056 (0.009)	0.055 (0.013)	0.058 (0.009)	0.062 (0.014)	**0.231 (0.015)**	0.147 (0.113)	—	0.645
Slashdot	—	—	—	—	—	—	—	—
LLOG	—	—	—	—	—	—	—	—
Solar Flare	0.794 (0.013)*	0.793 (0.014)*	0.793 (0.015)*	0.794 (0.013)*	0.776 (0.060)	**0.795 (0.012)***	0.786 (0.023)*	0.785
Bridges	0.042 (0.027)	0.026 (0.020)	0.040 (0.028)	0.023 (0.018)	0.014 (0.015)	0.040 (0.028)	**0.047 (0.023)**	0.163
Thyroid	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	**0.743 (0.002)**	0.692 (0.080)	**0.743 (0.002)**	0.625 (0.024)	0.924

Table A.44: Results using SCC with 80% of the original features.

	AE	dAE	D-AE	D-dAE	RBM	DBN	PCA	Original
Hamming Score (HS)								
Music	0.636 (0.029)	**0.674 (0.021)**	0.637 (0.014)	0.670 (0.018)	0.636 (0.020)	0.589 (0.007)	0.577 (0.021)	0.712
Scene	0.765 (0.010)	0.772 (0.006)	0.773 (0.008)	0.772 (0.005)	**0.796 (0.011)**	0.755 (0.006)	0.730 (0.009)	0.845
Yeast	0.679 (0.003)	0.679 (0.003)	0.677 (0.005)	0.675 (0.004)	**0.706 (0.008)**	0.699 (0.014)	0.645 (0.004)	0.724
ENRON	0.908 (0.001)	0.909 (0.001)	0.909 (0.001)	0.909 (0.001)	0.914 (0.004)	**0.918 (0.002)**	—	0.923
Medical	0.949 (0.002)	0.950 (0.001)	0.950 (0.001)	0.950 (0.002)	0.952 (0.009)	**0.966 (0.004)**	—	0.989
Slashdot	—	—	—	—	—	—	—	—
LLOG	—	—	—	—	—	—	—	—
Solar Flare	0.904 (0.014)	0.910 (0.005)*	0.910 (0.006)*	0.910 (0.005)*	0.909 (0.007)*	**0.911 (0.004)***	0.859 (0.015)	0.905
Bridges	0.502 (0.037)	0.517 (0.044)	0.485 (0.068)	0.513 (0.046)	0.456 (0.030)	**0.527 (0.044)**	0.489 (0.034)	0.570
Thyroid	**0.961 (0.000)**	**0.961 (0.001)**	**0.961 (0.000)**	**0.961 (0.000)**	0.951 (0.016)	**0.961 (0.000)**	0.934 (0.002)	0.986
Exact Match Ratio (EMR)								
Music	0.128 (0.029)	**0.156 (0.028)**	0.129 (0.018)	0.152 (0.031)	0.137 (0.022)	0.120 (0.013)	0.111 (0.024)	0.186
Scene	0.295 (0.027)	0.315 (0.014)	0.316 (0.019)	0.316 (0.014)	**0.383 (0.025)**	0.274 (0.018)	0.297 (0.023)	0.522
Yeast	0.041 (0.005)	0.038 (0.005)	0.037 (0.008)	0.038 (0.004)	**0.090 (0.007)**	0.084 (0.013)	0.044 (0.009)	0.134
ENRON	0.012 (0.005)	0.013 (0.004)	0.010 (0.003)	0.013 (0.004)	0.026 (0.006)*	**0.041 (0.008)***	—	0.026
Medical	0.050 (0.012)	0.059 (0.007)	0.054 (0.007)	0.057 (0.013)	**0.237 (0.022)**	0.162 (0.088)	—	0.645
Slashdot	—	—	—	—	—	—	—	—
LLOG	—	—	—	—	—	—	—	—
Solar Flare	0.780 (0.029)	0.794 (0.007)*	0.793 (0.010)*	0.793 (0.008)*	0.790 (0.016)*	**0.795 (0.007)***	0.785 (0.019)*	0.785
Bridges	0.035 (0.028)	0.032 (0.028)	**0.040 (0.031)**	0.030 (0.018)	0.016 (0.011)	0.030 (0.024)	0.037 (0.032)	0.163
Thyroid	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	**0.743 (0.003)**	0.689 (0.084)	**0.743 (0.003)**	0.610 (0.011)	0.924

Table A.45: Results using SCC with 90% of the original features.

APPENDIX B

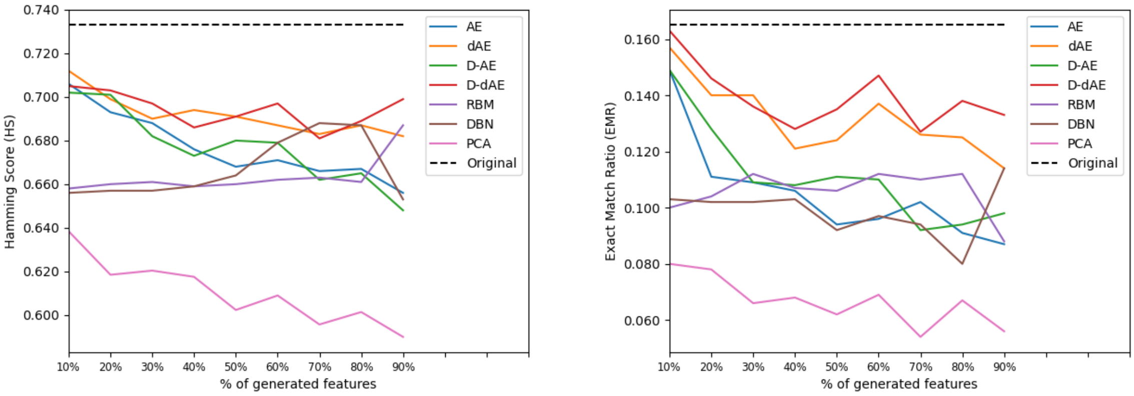
How Multi-Target Classifiers were Affected by the Number of Features Generated by the Feature-Extractors

This appendix presents all the results obtained during our final experiments, discussed in Sections 6.4 and 6.5. The data presented here is used to show how the number of features generated by the feature extractors selected for this research, influenced the performances of the multi-target classifiers.

For the graphs presented in this appendix, each line represents a feature extraction method, with the dashed black line representing the results obtained with the original dataset, i.e, without feature extraction.

The rest of this appendix is organized as follows: Section A.1 shows the results obtained by Class Relevance (CR), Section A.2 shows the results obtained by Classifier Chains (CC), Section A.3 shows the results obtained by Ensemble of Classifier Chains (ECC), Section A.4 shows the results obtained by Class Relevance Stacking (CRS) and, finally, Section A.5 shows the results obtained by Super-Class Classifier (SCC). The predictive performances of the multi-target classifiers were measured with Hamming Score (HS) and Exact Match Ratio (EMR).

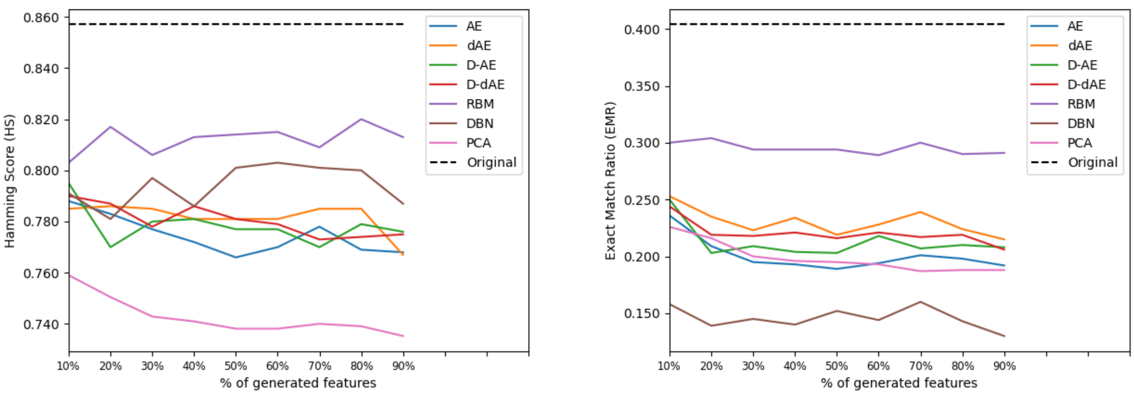
B.1 Class Relevance (CR) Results



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

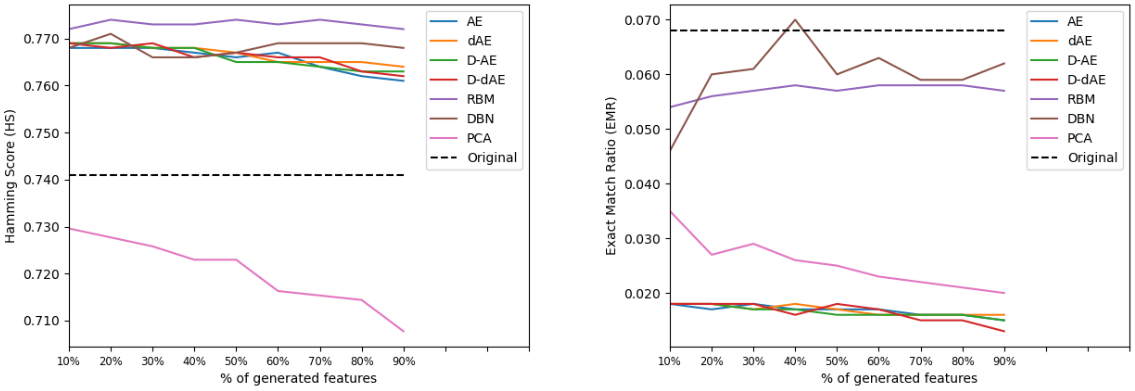
Figure B.1: How the CR classification performance is affected by the number of extracted features on Music dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

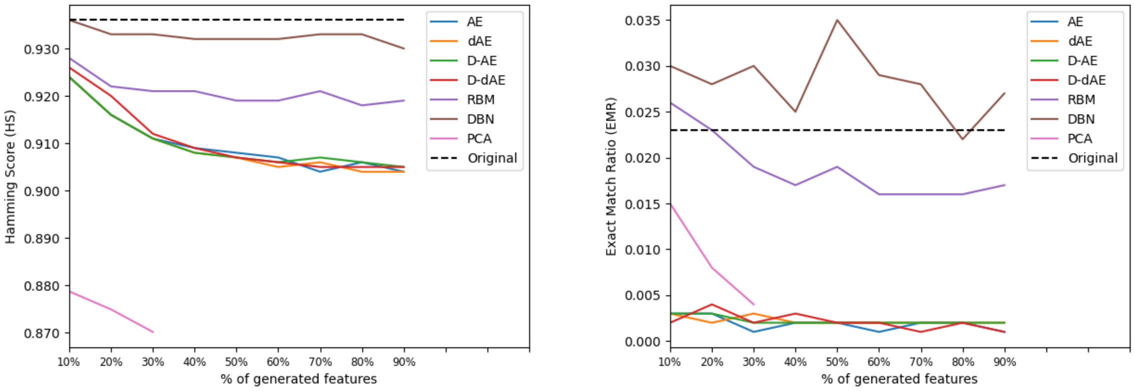
Figure B.2: How the CR classification performance is affected by the number of extracted features on Scene dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

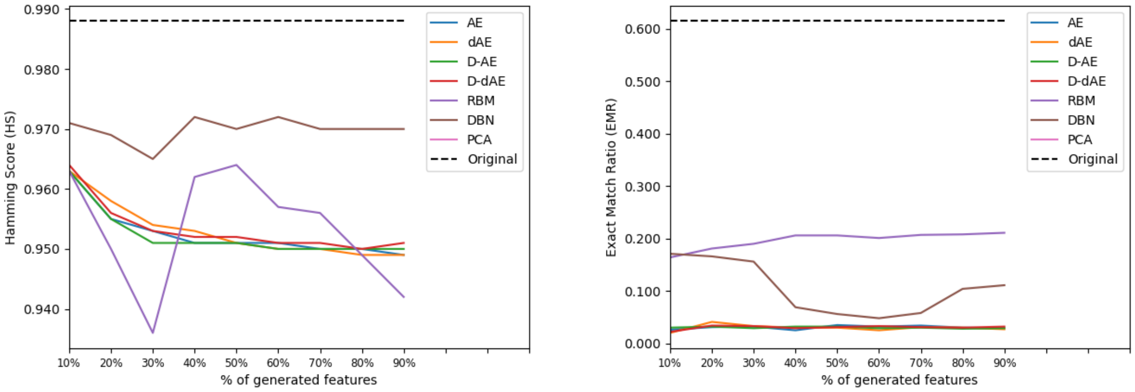
Figure B.3: How the CR classification performance is affected by the number of extracted features on Yeast dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

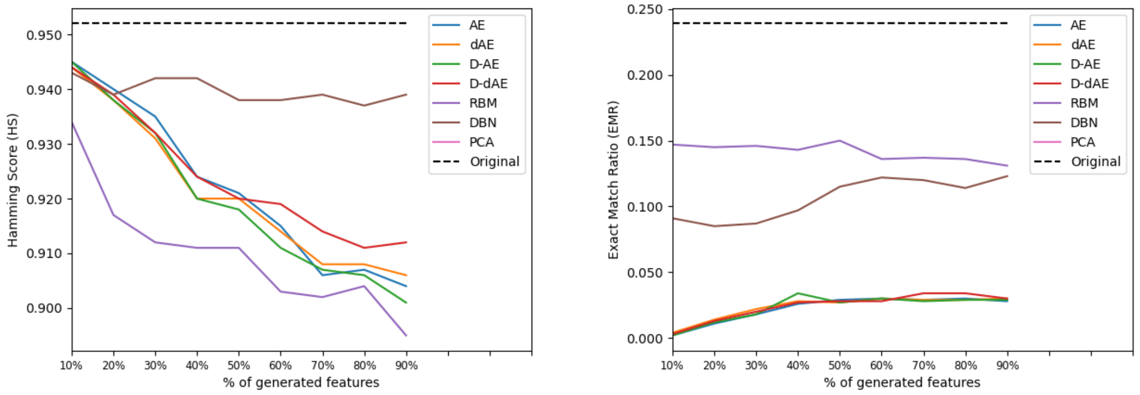
Figure B.4: How the CR classification performance is affected by the number of extracted features on ENRON dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

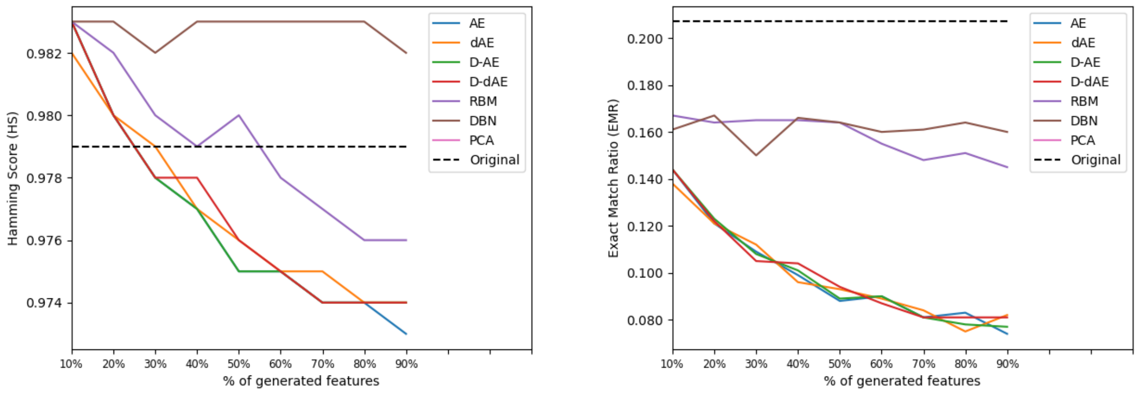
Figure B.5: How the CR classification performance is affected by the number of extracted features on MEDICAL dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

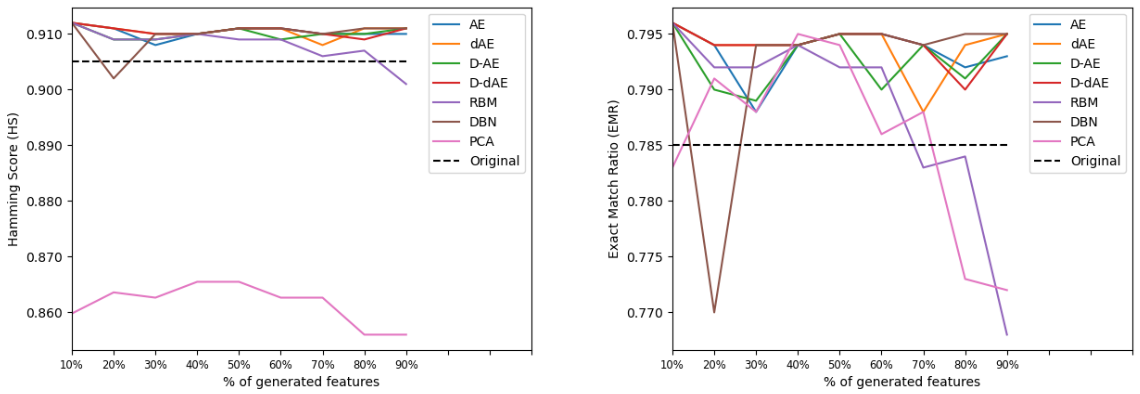
Figure B.6: How the CR classification performance is affected by the number of extracted features on SLASHDOT dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

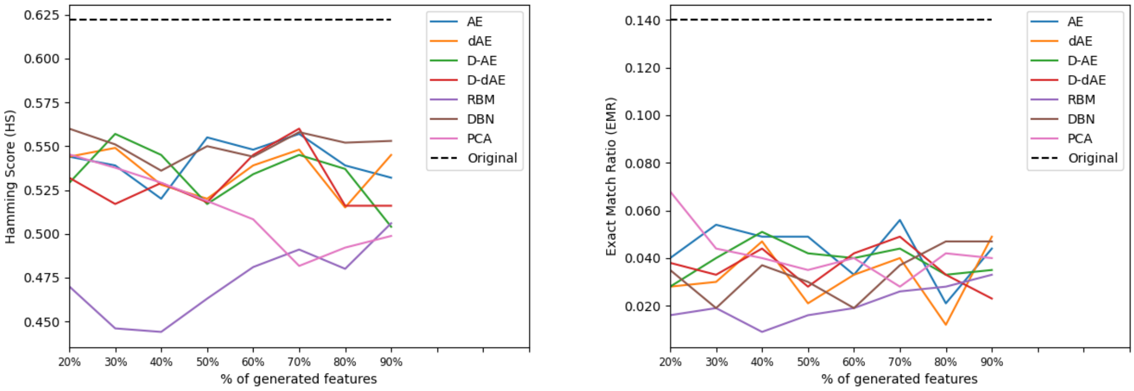
Figure B.7: How the CR classification performance is affected by the number of extracted features on LLOG dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

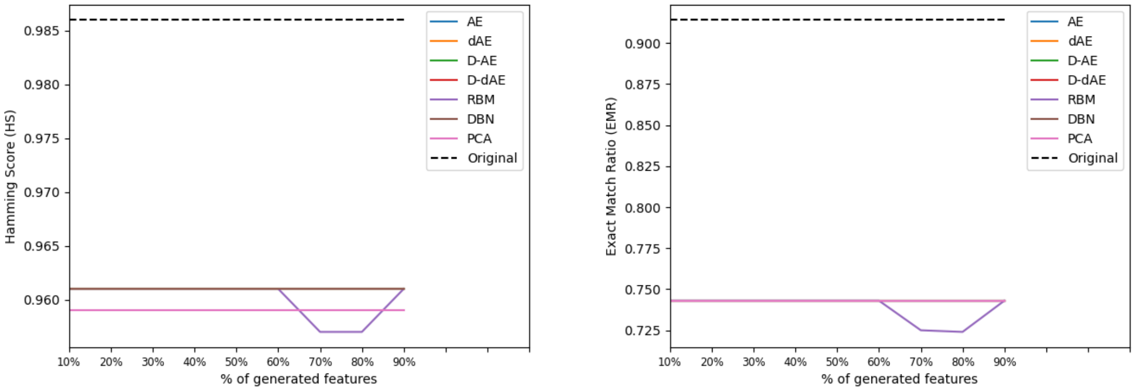
Figure B.8: How the CR classification performance is affected by the number of extracted features on SolarFlare dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

Figure B.9: How the CR classification performance is affected by the number of extracted features on Bridges dataset.

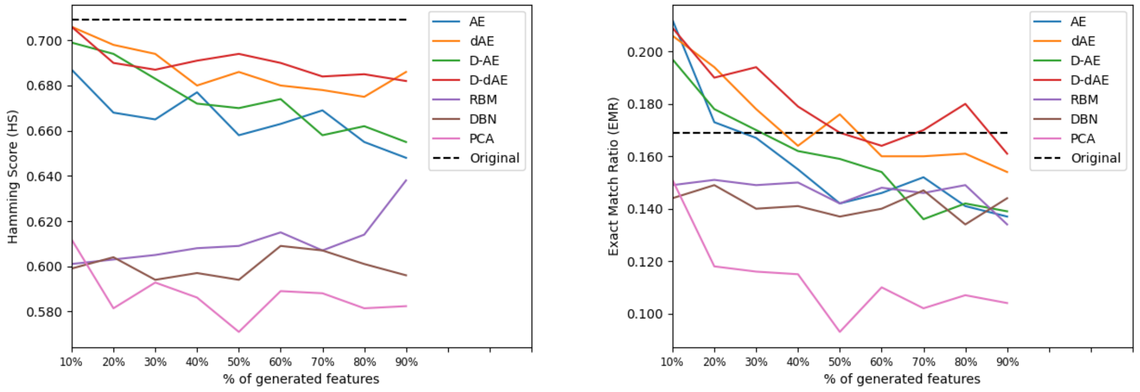


(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

Figure B.10: How the CR classification performance is affected by the number of extracted features on Thyroid dataset.

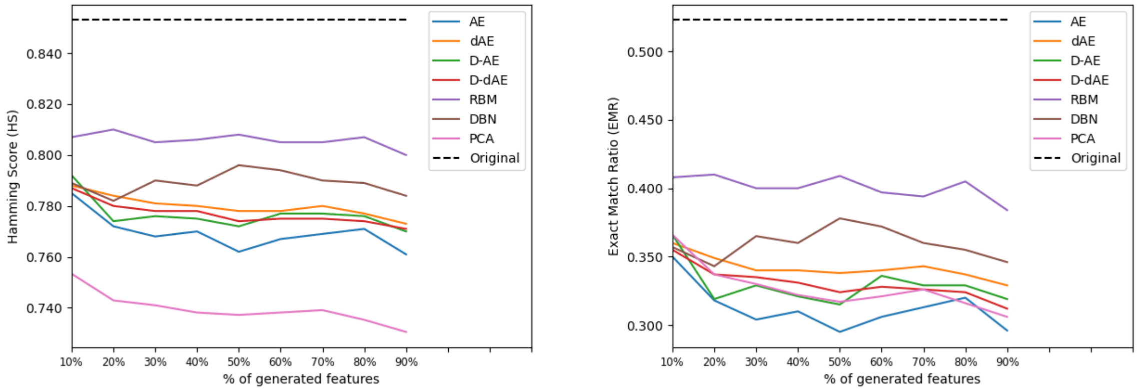
B.2 Classifier Chains (CC) Results



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

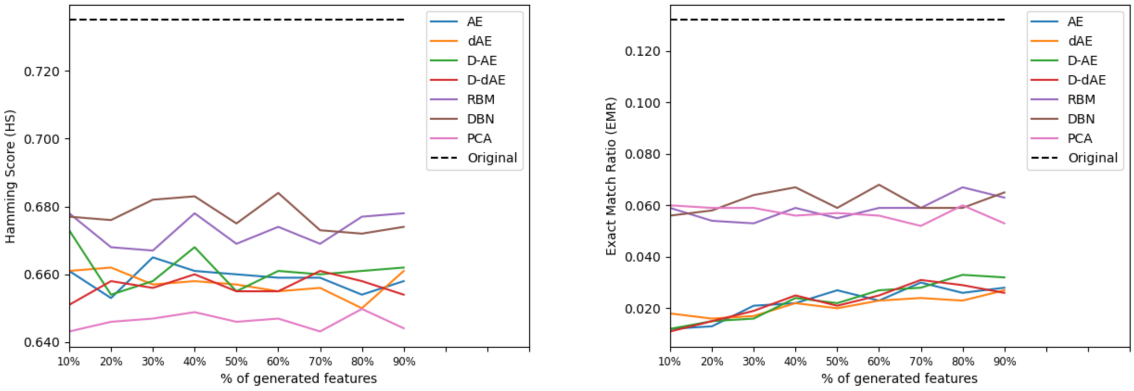
Figure B.11: How the CC classification performance is affected by the number of extracted features on Music dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

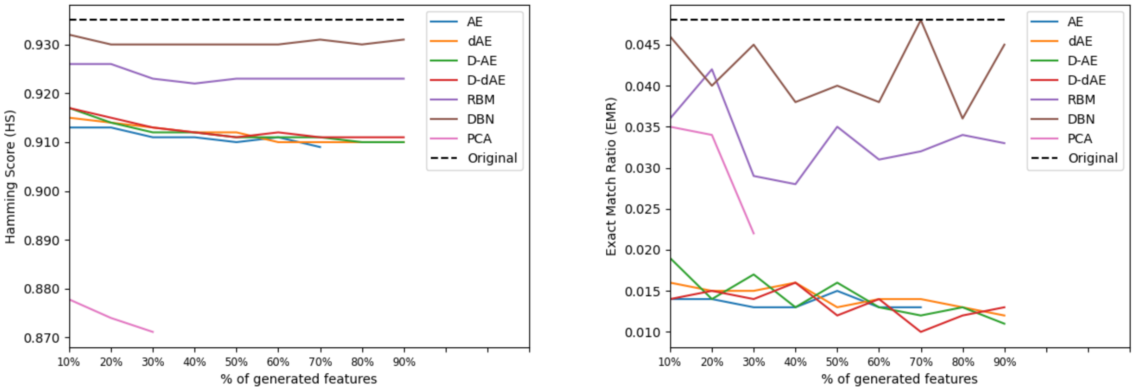
Figure B.12: How the CC classification performance is affected by the number of extracted features on Scene dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

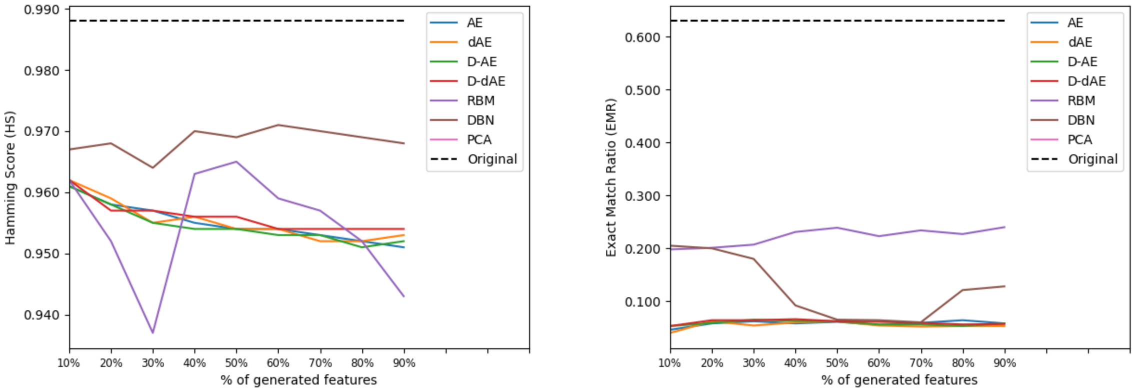
Figure B.13: How the CC classification performance is affected by the number of extracted features on Yeast dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

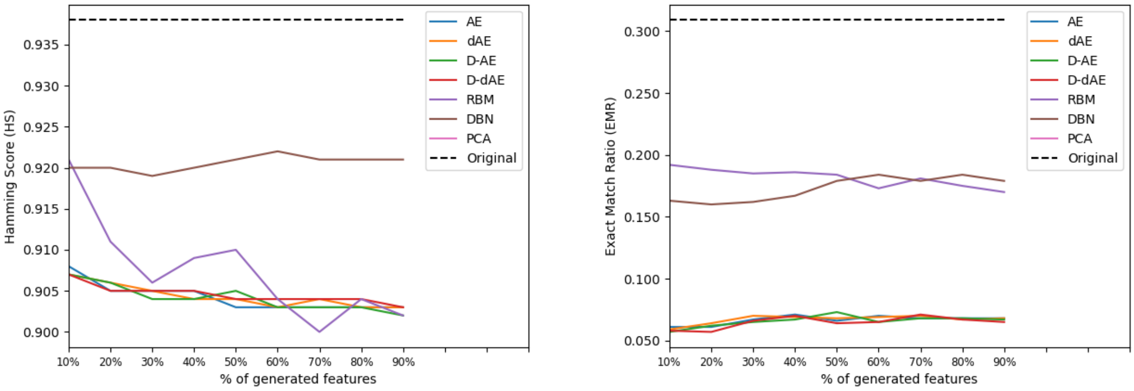
Figure B.14: How the CC classification performance is affected by the number of extracted features on ENRON dataset.



(a) Hamming Score (HS)

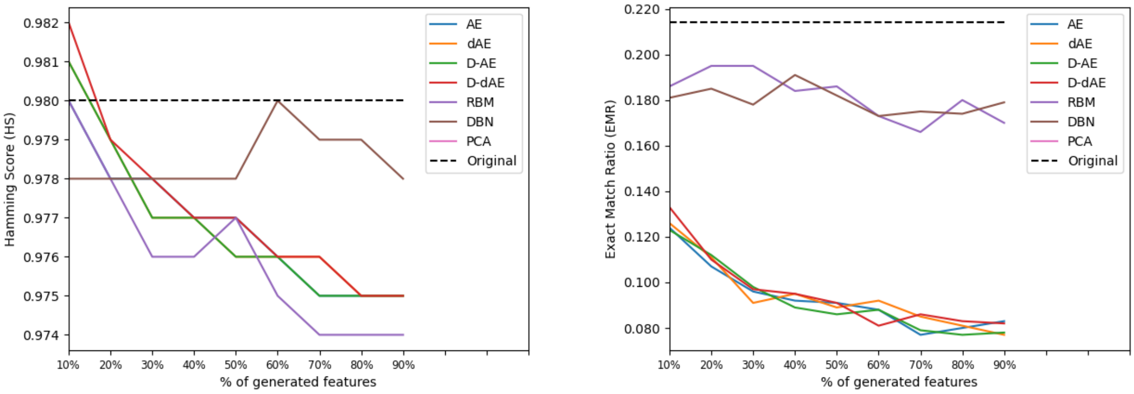
(b) Exact Match Ratio (EMR)

Figure B.15: How the CC classification performance is affected by the number of extracted features on MEDICAL dataset.



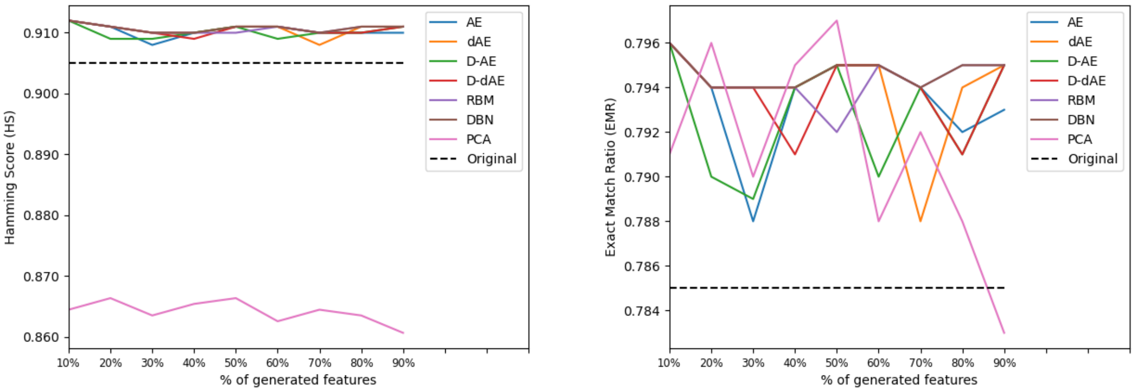
(a) Hamming Score (HS) (b) Exact Match Ratio (EMR)

Figure B.16: How the CC classification performance is affected by the number of extracted features on SLASHDOT dataset.



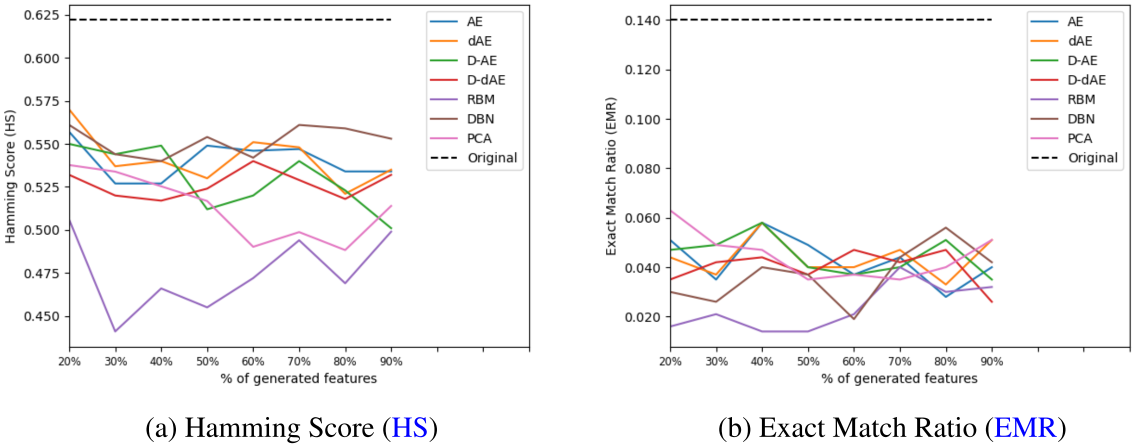
(a) Hamming Score (HS) (b) Exact Match Ratio (EMR)

Figure B.17: How the CC classification performance is affected by the number of extracted features on LLOG dataset.



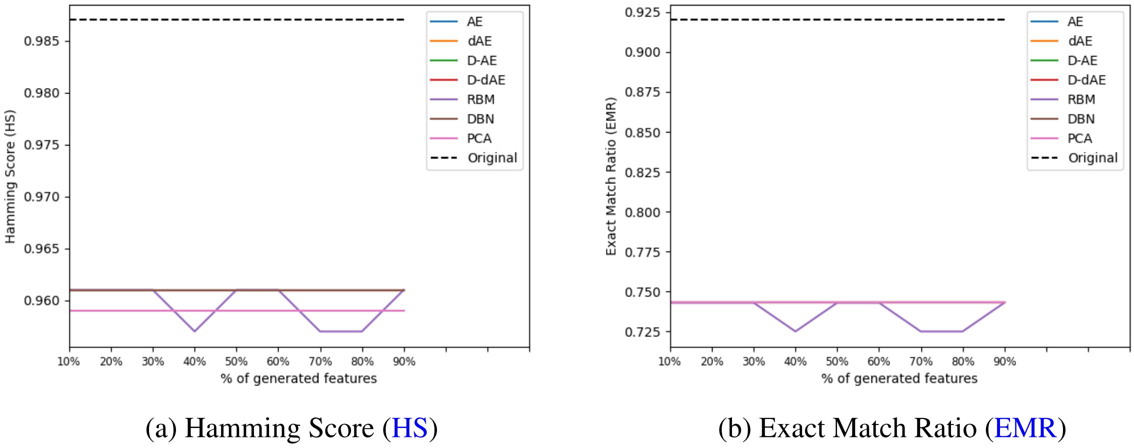
(a) Hamming Score (HS) (b) Exact Match Ratio (EMR)

Figure B.18: How the CC classification performance is affected by the number of extracted features on SolarFlare dataset.



(a) Hamming Score (HS) (b) Exact Match Ratio (EMR)

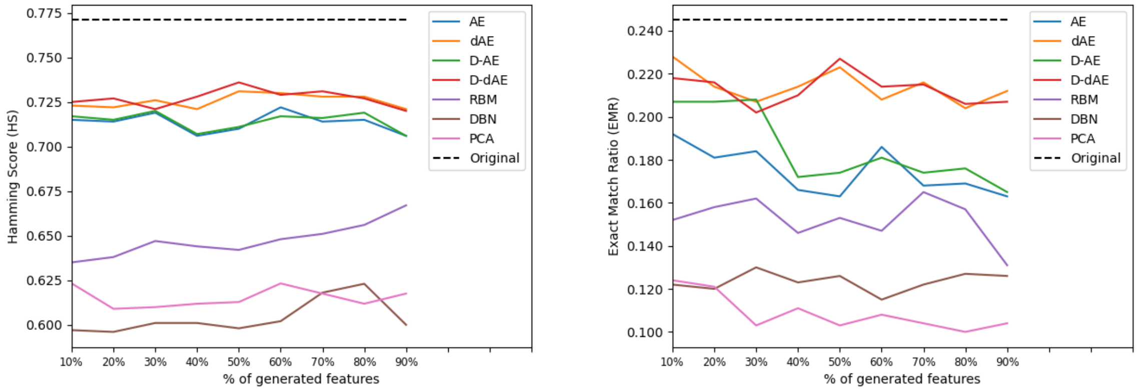
Figure B.19: How the CC classification performance is affected by the number of extracted features on Bridges dataset.



(a) Hamming Score (HS) (b) Exact Match Ratio (EMR)

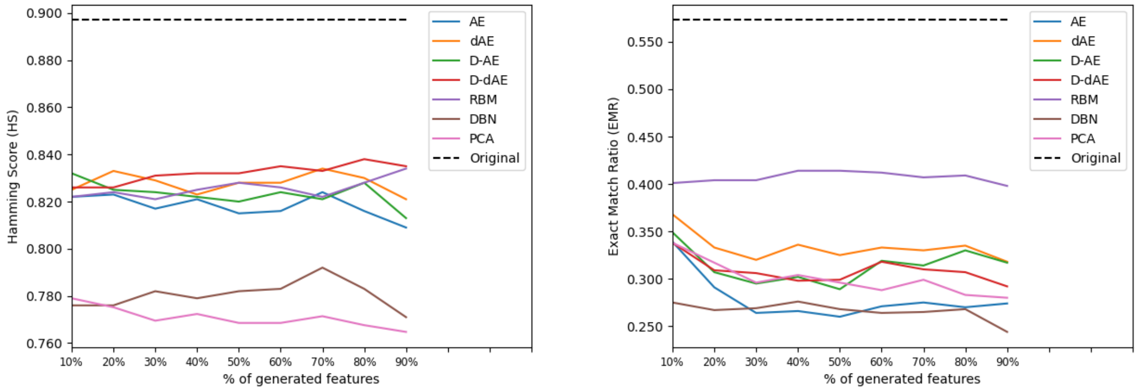
Figure B.20: How the CC classification performance is affected by the number of extracted features on Thyroid dataset.

B.3 Ensemble of Classifier Chains (ECC) Results



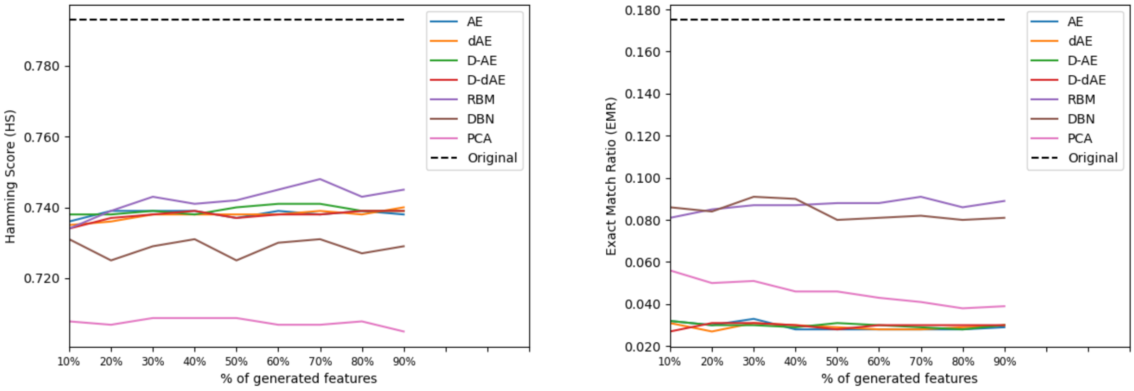
(a) Hamming Score (HS) (b) Exact Match Ratio (EMR)

Figure B.21: How the ECC classification performance is affected by the number of extracted features on Music dataset.



(a) Hamming Score (HS) (b) Exact Match Ratio (EMR)

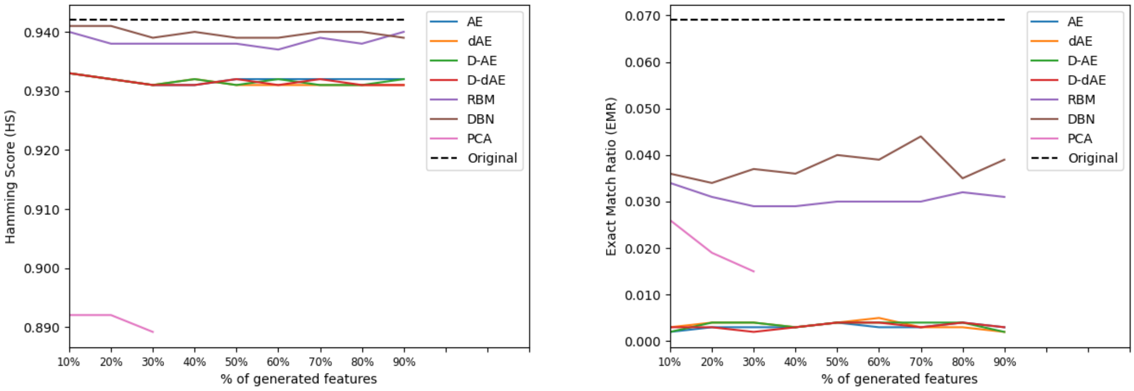
Figure B.22: How the ECC classification performance is affected by the number of extracted features on Scene dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

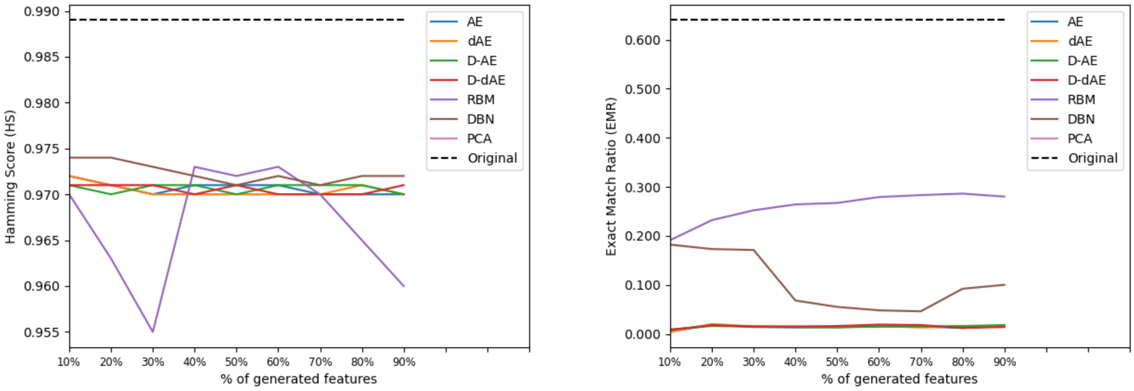
Figure B.23: How the ECC classification performance is affected by the number of extracted features on Yeast dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

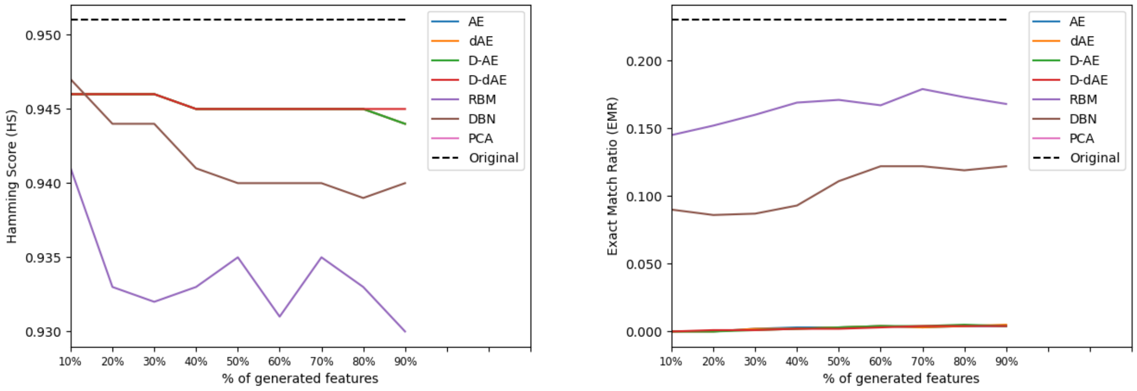
Figure B.24: How the ECC classification performance is affected by the number of extracted features on ENRON dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

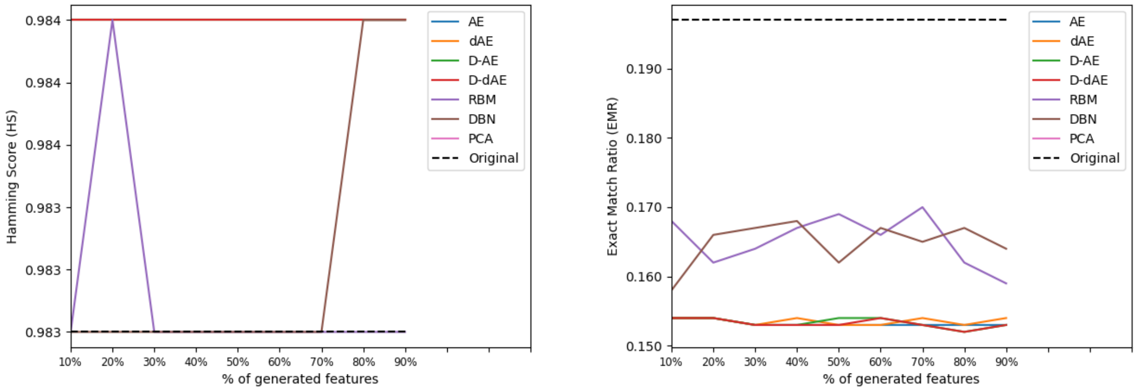
Figure B.25: How the ECC classification performance is affected by the number of extracted features on MEDICAL dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

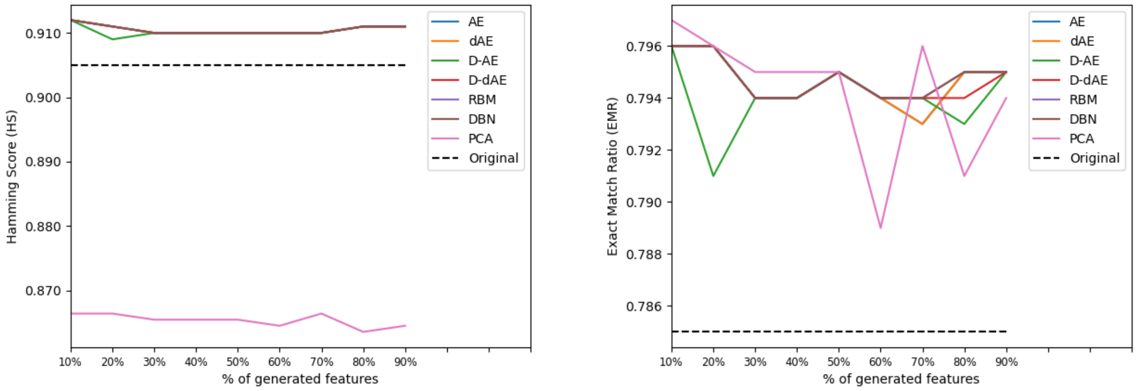
Figure B.26: How the ECC classification performance is affected by the number of extracted features on SLASHDOT dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

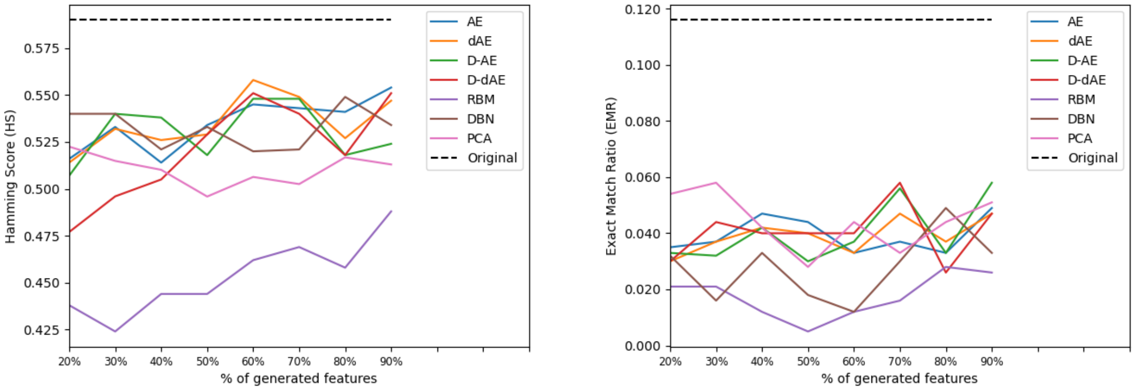
Figure B.27: How the ECC classification performance is affected by the number of extracted features on LLOG dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

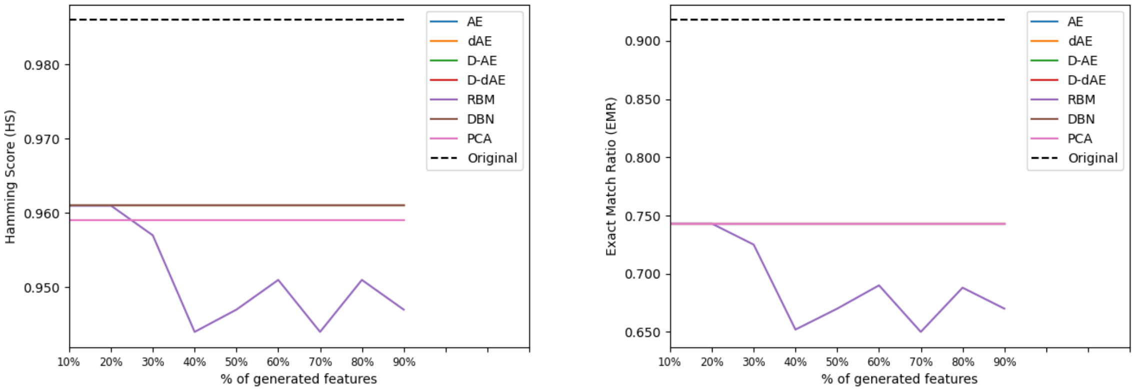
Figure B.28: How the ECC classification performance is affected by the number of extracted features on SolarFlare dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

Figure B.29: How the ECC classification performance is affected by the number of extracted features on Bridges dataset.

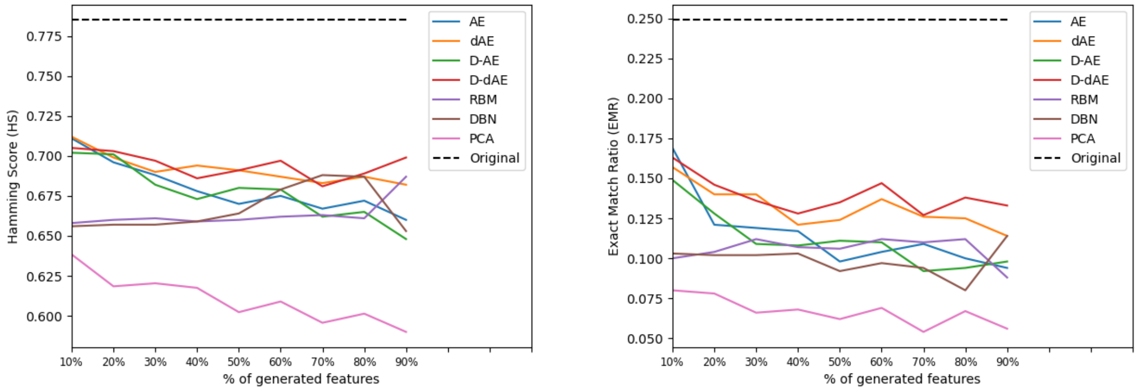


(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

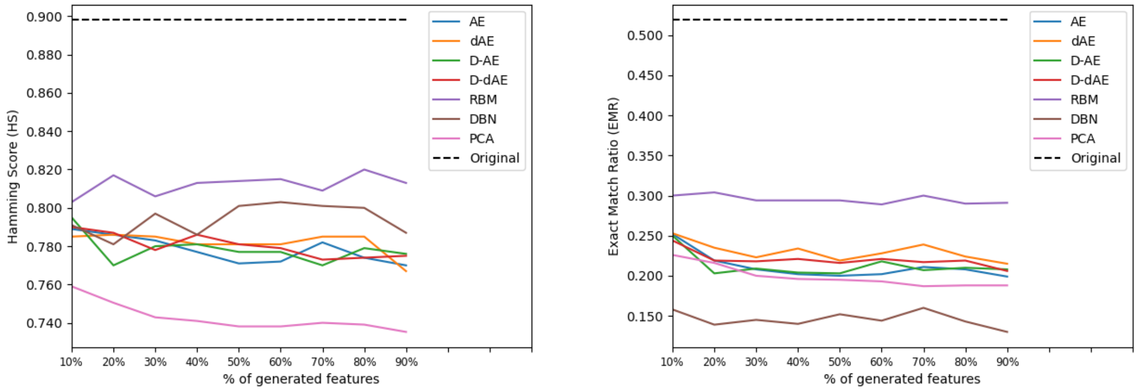
Figure B.30: How the ECC classification performance is affected by the number of extracted features on Thyroid dataset.

B.4 Class Relevance Stacking (CRS) Results



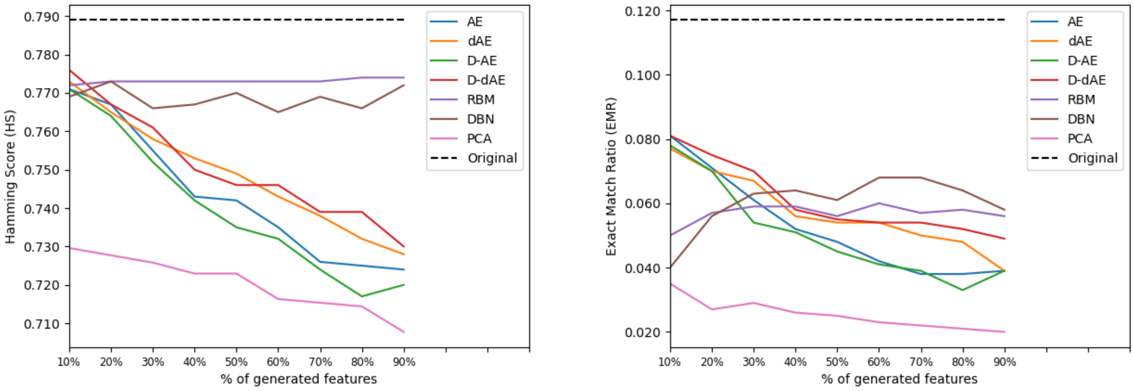
(a) Hamming Score (HS) (b) Exact Match Ratio (EMR)

Figure B.31: How the CRS classification performance is affected by the number of extracted features on Music dataset.



(a) Hamming Score (HS) (b) Exact Match Ratio (EMR)

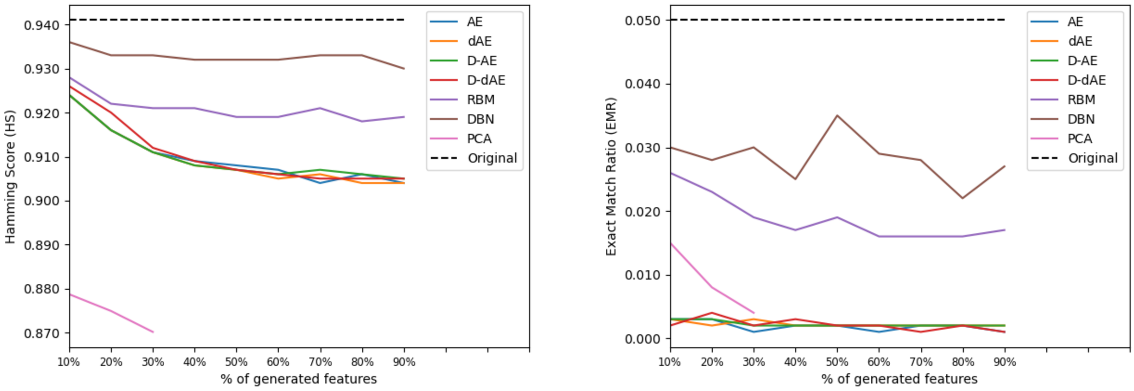
Figure B.32: How the CRS classification performance is affected by the number of extracted features on Scene dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

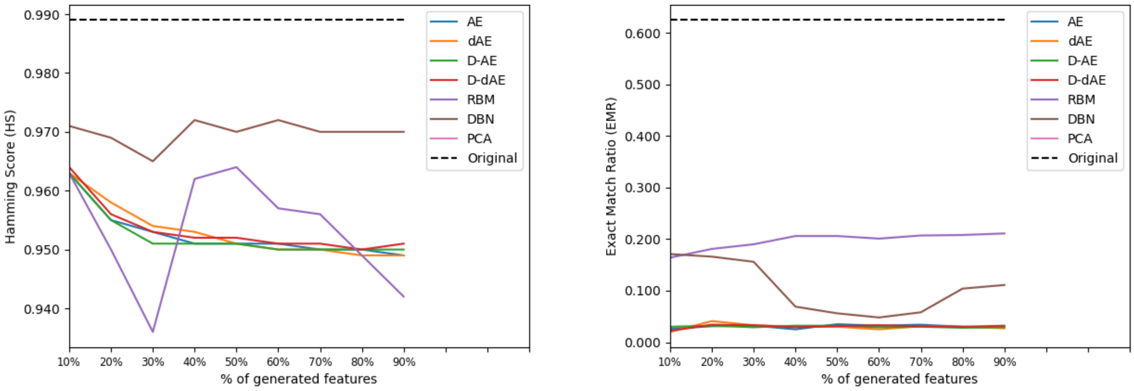
Figure B.33: How the CRS classification performance is affected by the number of extracted features on Yeast dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

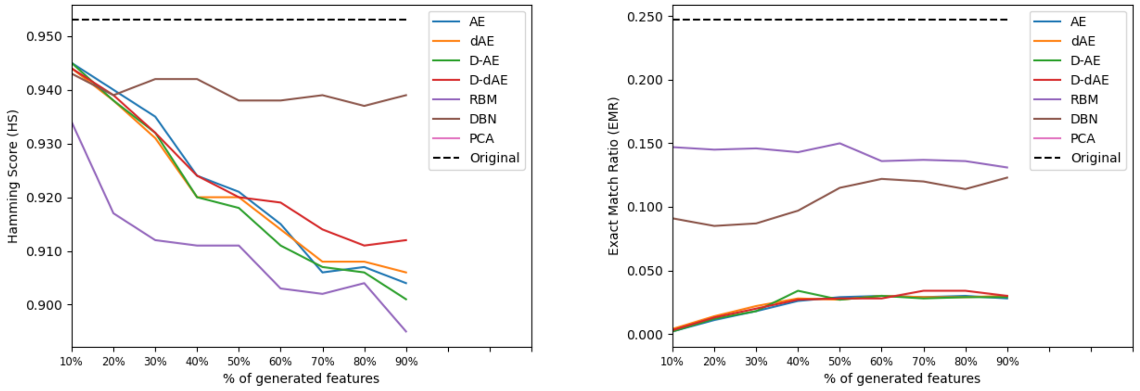
Figure B.34: How the CRS classification performance is affected by the number of extracted features on ENRON dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

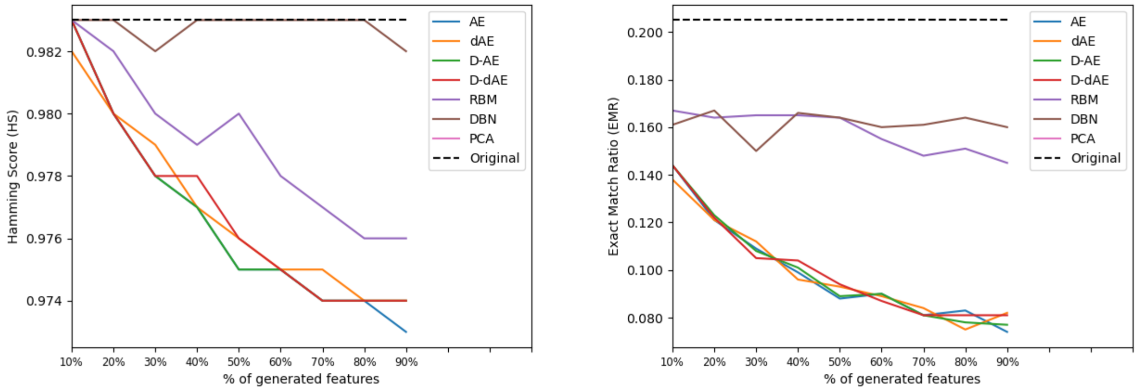
Figure B.35: How the CRS classification performance is affected by the number of extracted features on MEDICAL dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

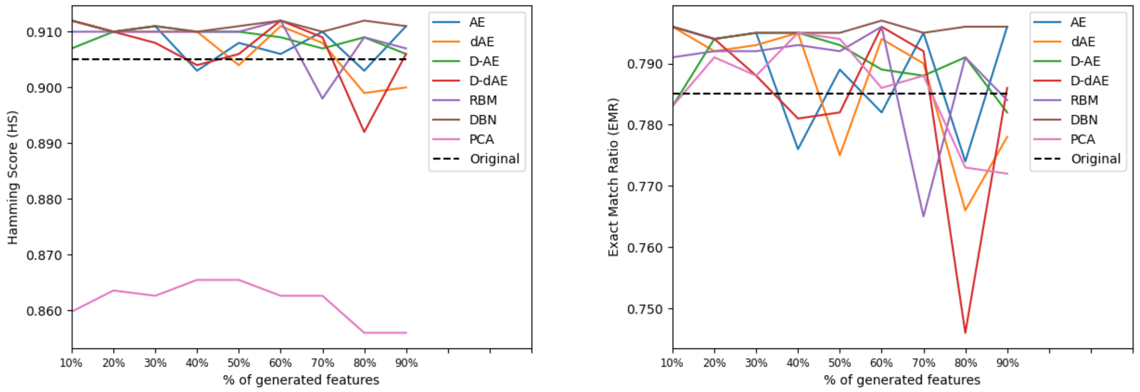
Figure B.36: How the CRS classification performance is affected by the number of extracted features on SLASHDOT dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

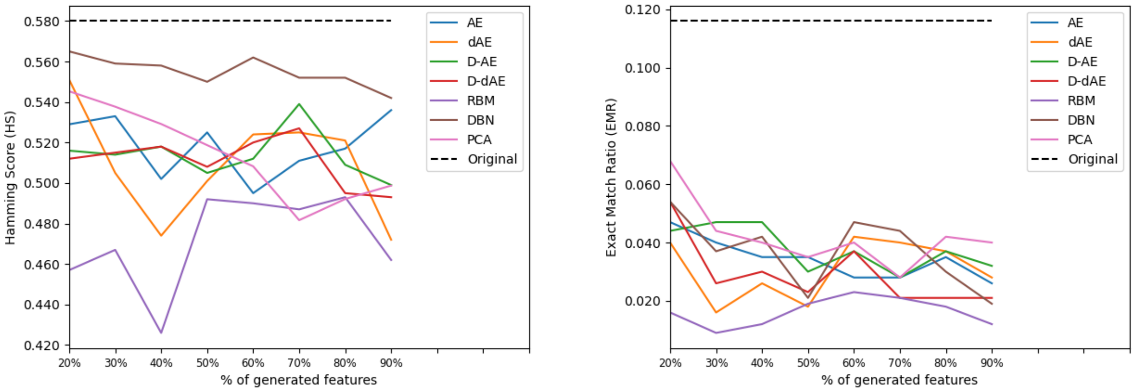
Figure B.37: How the CRS classification performance is affected by the number of extracted features on LLOG dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

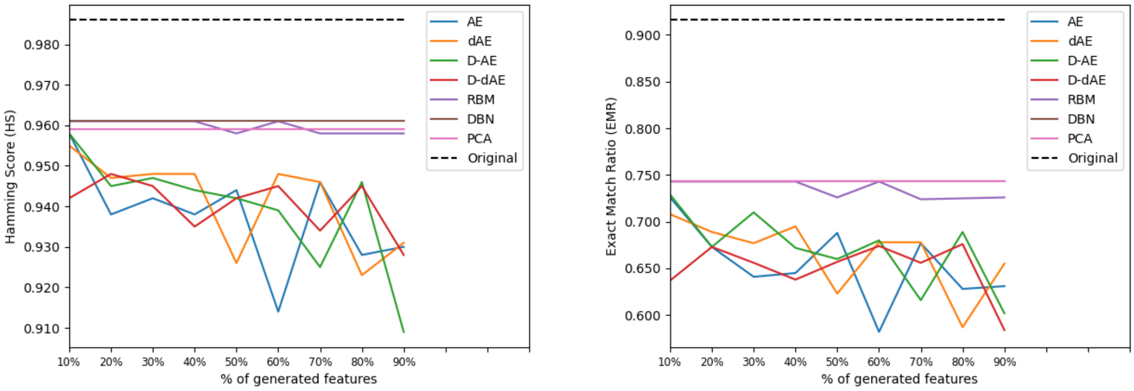
Figure B.38: How the CRS classification performance is affected by the number of extracted features on SolarFlare dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

Figure B.39: How the CRS classification performance is affected by the number of extracted features on Bridges dataset.

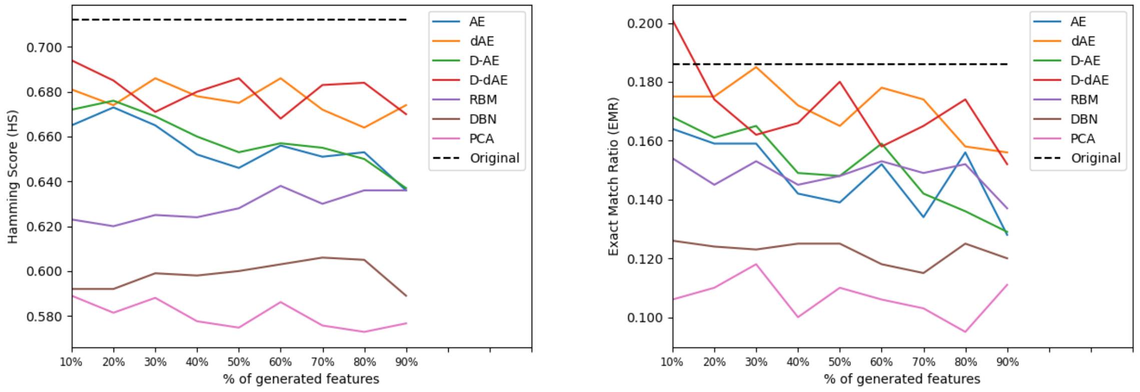


(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

Figure B.40: How the CRS classification performance is affected by the number of extracted features on Thyroid dataset.

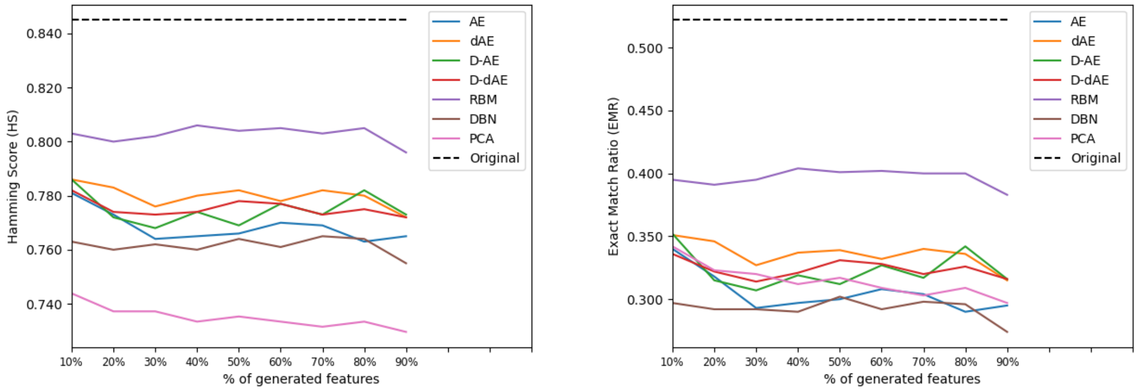
B.5 Super-Class Classifier (SCC) Results



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

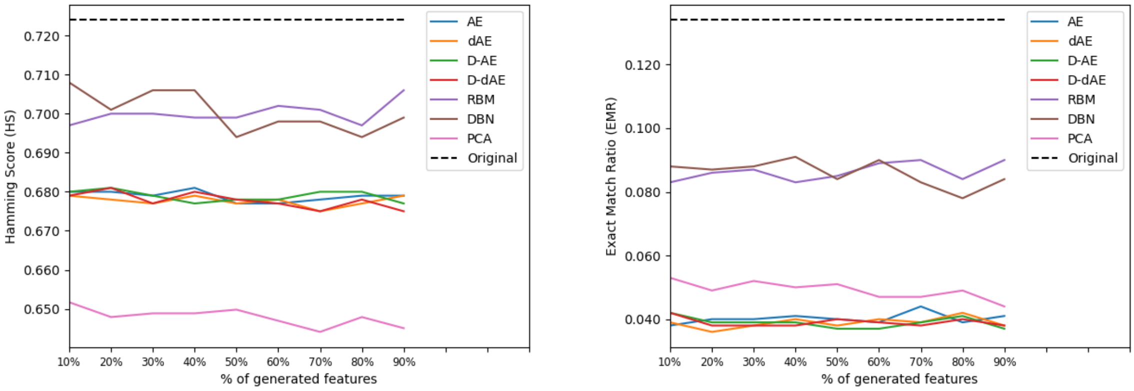
Figure B.41: How the SCC classification performance is affected by the number of extracted features on Music dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

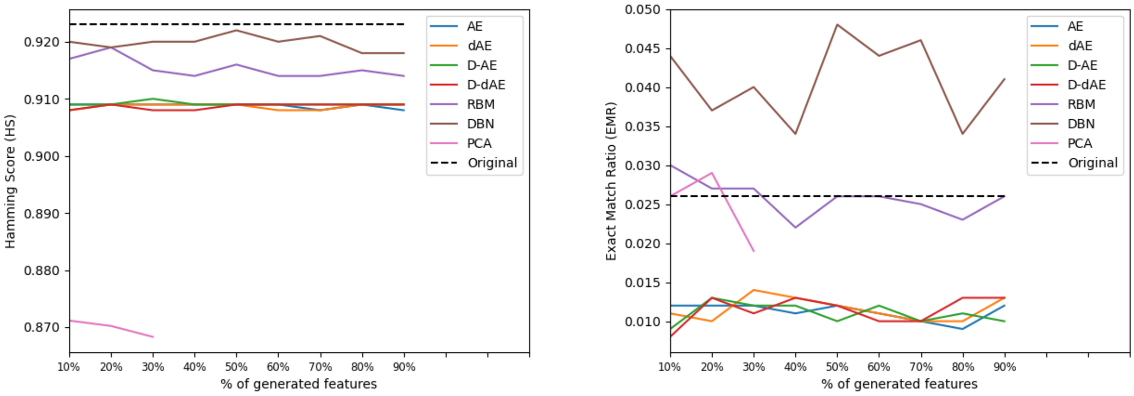
Figure B.42: How the SCC classification performance is affected by the number of extracted features on Scene dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

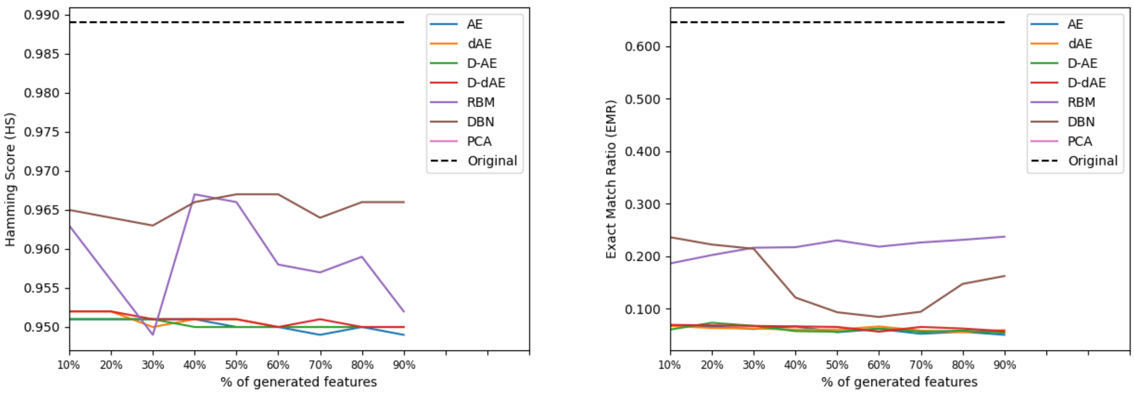
Figure B.43: How the SCC classification performance is affected by the number of extracted features on Yeast dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

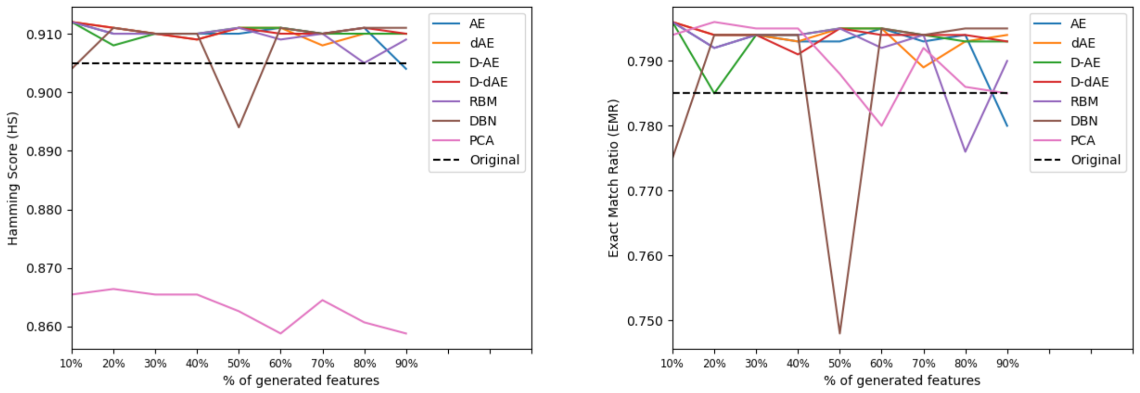
Figure B.44: How the SCC classification performance is affected by the number of extracted features on ENRON dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

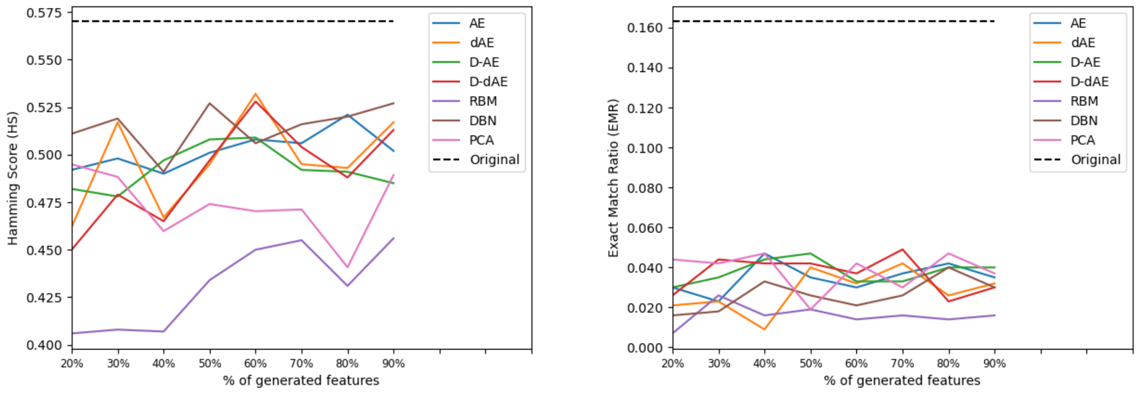
Figure B.45: How the SCC classification performance is affected by the number of extracted features on MEDICAL dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

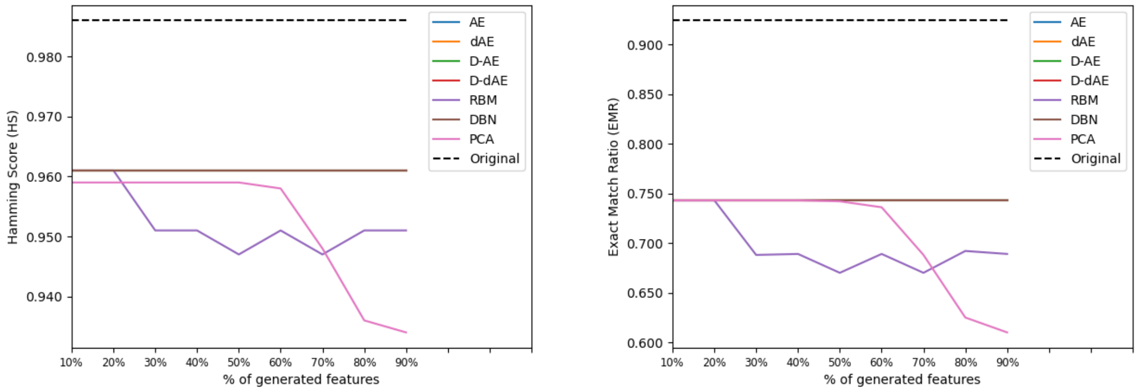
Figure B.46: How the SCC classification performance is affected by the number of extracted features on SolarFlare dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

Figure B.47: How the SCC classification performance is affected by the number of extracted features on Bridges dataset.



(a) Hamming Score (HS)

(b) Exact Match Ratio (EMR)

Figure B.48: How the SCC classification performance is affected by the number of extracted features on Thyroid dataset.