IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 11, 2020, accepted August 28, 2020, date of publication September 7, 2020, date of current version September 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3022317

Analysis of Multiobjective Algorithms for the Classification of Multi-Label Video Datasets

GIZEM NUR KARAGOZ[1], ADNAN YAZICI[1,2], TANSEL DOKEROGLU[3], AND AHMET COSAR[4]

[1]Department of Computer Engineering, Middle East Technical University, 06800 Ankara, Turkey
[2]Department of Computer Science, Nazarbayev University, Nur-Sultan 010000, Kazakhstan
[3]Department of Computer Engineering, TED University, 06420 Ankara, Turkey
[4]Department of Computer Engineering, Ankara Science University, 06200 Ankara, Turkey

Corresponding author: Gizem Nur Karagoz (gizem.karagoz@metu.edu.tr)

This work received funding from NU Faculty-development competitive research grants program, Nazarbayev University, Grant number-110119FD4543.

ABSTRACT It is of great importance to extract and validate an optimal subset of non-dominated features for effective multi-label classification. However, deciding on the best subset of features is an NP-Hard problem and plays a key role in improving the prediction accuracy and the processing time of video datasets. In this study, we propose autoencoders for dimensionality reduction of video data sets and ensemble the features extracted by the multi-objective evolutionary Non-dominated Sorting Genetic Algorithm and the autoencoder. We explore the performance of well-known multi-label classification algorithms for video datasets in terms of prediction accuracy and the number of features used. More specifically, we evaluate Non-dominated Sorting Genetic Algorithm-II, autoencoders, ensemble learning algorithms, Principal Component Analysis, Information Gain, and Correlation Based Feature Selection. Some of these algorithms use feature selection techniques to improve the accuracy of the classification. Experiments are carried out with local feature descriptors extracted from two multi-label datasets, the MIR-Flickr dataset which consists of images and the Wireless Multimedia Sensor dataset that we have generated from our video recordings. Significant improvements in the accuracy performance of the algorithms are observed while the number of features is being reduced.

INDEX TERMS Feature selection, multi-label, multi-objective optimization, autoencoder, ensemble, classification.

I. INTRODUCTION

Multi-label classification has been applied to many problems in various fields of application, including the diagnosis of diseases based on many signs and symptoms [1] and also used in many tools developed for the classification of social media resources, images, bioinformatics [2], videos [3], patient classification [4], text [5], and audio that may need to be assigned with more than one label [6]. Images are the subject of research on multi-label classification problems in multimedia resources. If an image of the sea is to be labeled as a beach, a comprehensive analysis of the scene may be necessary to identify the image. An image of the sea containing sunbeds, parasols, people, bags, sand, and sailing provides more accurate clues for identifying the image. The absence of certain objects on the image can also be useful for the classification of the scene. The absence of a truck or a skyscraper reinforces the idea that this image is a beach. This concept is called *Semantic Scene Analysis/Classification* [7]. In general, the structure of a scene is first generated, and then the associated objects are detected for semantic analysis.

An important aspect of real data is that it usually has multiple scopes. An image taken by the camera can include many features, correlated or not. Tagging this rich data content with simple binary labels may not be possible in many cases. For this reason, multi-label classification is an important field of data classification. For binary classification, data is labeled as one of two classes, while for multi-class classification, there are more than two possible classes and each row of data is labeled with only one class. On the other hand, for multi-label classification, there are more than two possible labels and each row of the data can have more than one label.

Irrelevant and/or redundant data should be filtered before being transmitted to big data stores in order to speed up

The associate editor coordinating the review of this manuscript and approving it for publication was Kai Li.

data processing. Concentrating on relevant big data might also increase the accuracy of the classification and provide better data analysis models. A widely used filtering method is the selection of features, which is used in preprocessing the data to achieve these goals. The feature selection process searches for the most relevant and sufficient subset of features for data mining and classification. There are three main methods for performing feature selection: filtering, wrapper, and embedded methods. The filtering methods use computationally inexpensive evaluation functions over all available data features, providing a ranking of the features that can be used to select only a feasible portion of the data [8]. Wrapper methods use learning algorithms to determine the most relevant subsets of features used for training to maximize the performance of learning. The evaluation of wrapper algorithms is computationally very expensive, but they can determine the most valuable subset of features [9]. Embedded methods combine feature selection methods with a model construction process (wrapper), so that they have an ability to stop the attribute filtering process when the performance achieved by the classification/learning algorithm reaches a sufficient level [10].

In this study we first use autoencoders to implement the dimensionality reduction for video data. The number of layers of autoencoder is determined with a heuristic approach. Subsequently, the sets of reduced number of features extracted with the two regularized autoencoders, dropout and denoising autoencoders, are determined as a latent space representation of input data. Then, feature selection is applied to the same input data with NSGA-II. After both feature selection operations are done and reduced dimensional feature-sets are obtained, these feature-sets are combined. Thus, most descriptive features that are selected by two different methods are combined and ensemble feature selection results are achieved with NSGA-II and multi-label classification algorithms. Our ensemble feature selection approach provides better results than the previous results on the datasets used in this study. The Hamming score is increased while the number of features is being reduced during multi-objective optimization.

Second, we analyze the performance of multi-label classification algorithms, Non-dominated Sorting Genetic Algorithm (NSGA-II) [11], autoencoders, ensemble learning algorithms, Principal Component Analysis (PCA), Information Gain (IG), and Correlation Based Feature Selection (CBFS). Binary Relevance (BR), Classifier Chains (CC), Pruned Sets (PS) and Random k Label-sets (RAkEL) are the main multi-label classification algorithms. Support Vector Machines (SVM), J48-Decision Tree (J48) and Logistic Regression (LR) are used to evaluate the fitness values (prediction accuracy). Thanks to parallel computing (using lightweight multi-threading), the fitness value calculations of the chromosomes are sped-up in NSGA-II. To our knowledge, we have implemented for the first time dimensionality reduction algorithms using autoencoders for multi-label classification. The under-complete autoencoders are used in the form of denoising and drop-out regularization in different noise factors with well-tuned parameter settings.

Two different datasets are used in our experiments, MIR-Flickr dataset which consists of images and the Wireless Multimedia Sensor (WMS) dataset that we have generated from our own video recordings. For the WMS dataset, three minute-video and 1,000 frames of this video are provided in a multi-labeled format (three labels) with Scale-Invariant Feature Transform (SIFT) local feature descriptors (100 bags of visual words for each). To the best of our knowledge, the image/video datasets and the selection of features on local descriptors are not studied and evaluated before, for the first time in this study. Additionally, we review state-of-the-art feature selection algorithms and improvements to ensembled feature sets that are extracted by two different feature selection approaches, deep autoencoder and the scalable multi-objective evolutionary algorithm, with the second optimization step are carried out for the first time in our study. In this paper, we show that the quality of the results of our approach is improved with higher Hamming scores and fewer features. The contributions of our study can be listed as:

- Autoencoders are proposed to implement the dimensionality reduction of video data and a heuristic approach is developed to determine (tune) the number of layers of the proposed autoencoder.
- A parallel multi-objective NSGA-II algorithm is used to select the best subset of features and the resulting set is combined with the feature set of the autoencoder.
- The proposed algorithms are verified to be robust after comprehensive experiments. There are small deviations from the best solutions reported in literature.
- An efficient multi-objective ensemble method is introduced to extract the most descriptive features of video datasets. The Hamming-score is improved while the minimum number of features is being used.

Section 2 provides information on recent studies. Section 3 provides information on the related theoretical background to the problem. The models proposed for the selection of the features are described in detail and the validation algorithms are explained in Section 4. The experimental results of the proposed algorithms are evaluated and discussed in Section 5. Our final remarks and future studies are presented in the last section.

II. RELATED WORK

This section summarizes the algorithms in literature that have been used for multi-objective feature selection and multi-label classification. A feature selection method that consists of a heuristic checklist that provides a basic road-map by asking questions about features and labels is proposed in [12]. A feature selection research for multi-objective optimization algorithms that integrate genetic algorithms and machine learning techniques is presented in [13]. A new multi-label feature selection method in classification for

a multi-objective Particle Swarm Optimization PSO algorithm is presented in [14]. The NSGA-II and Evolutionary Non-Dominated Radial Slots based algorithm (ENORA) is reported in [15]. An algorithm to eliminate irrelevant, noisy and redundant features during face recognition is developed in [16]. A NSGA-II with Naive Bayes (NB) and SVM for feature selection is proposed in [17]. A feature selection approach based on the weighted relevancy is given in [18]. Three multi-objective feature selection methods for binary classification problems with machine learning are proposed in a recent study [19]. Proposed techniques consist of two phases; feature subset selection and applying machine learning techniques for better accuracy prediction. 1-NN algorithm as a classifier on NSGA-II algorithm for multi-objective feature selection are used in [20]. A multi-objective NSGA-II feature selection algorithm for multi-label data classification with Label Powerset (LP), Binary Relevance (BR), Classifier Chain (CC) and Calibrated Label Ranking (CLR) is used in [21].

A hybrid genetic algorithm with SVM on feature selection for hyper-spectral image classification in order to get better band combination means to find irrelevant band combinations with the minimal number of bands is developed in [22]. A feature selection algorithm, Reduced Pareto set Genetic Algorithm with elitism (RPSGAe), with SVM is proposed in [23]. A multi-objective PSO for feature selection with Linear Forward Selection (LFS) and Greedy Step-wise Backward Selection (GSBS) methods is proposed [24]. A PSO algorithm focused on performance metrics of multi-objective optimization algorithms is developed [25]. In this study, hyper-volume and two-set-coverage are investigated. A Teaching Learning Based Optimization (TLBO) algorithm for feature selection is proposed in [26]. In TLBO, the best learners are selected as teachers and the remaining individuals are called students. Pareto optimal results are reported as candidate features for feature subset selection operation. The selection of multi-label features using the ant-colony optimization is studied in [27]. The authors use the multi-label k-nearest neighborhood algorithm to evaluate the subsets of features and compare them with some other approaches. Additionally, an applied a multi-objective optimization algorithm based on decomposition which is the Tchebycheff method for the purpose of feature selection is developed in [28]. The authors use multi-label benchmark datasets for validation of the proposed feature selection approach. All results are compared with other well-known multi-objective optimization algorithms such as NSGA-II and PSO. A multi-objective feature selection Artificial Bee Colony (ABC) algorithm to maximize the classification performance and to minimize the number of selected features is proposed in [29], [30]. The ABC algorithm is reported to outperform other methods in terms of both the dimensionality reduction and the classification accuracy.

In recent years, ensemble feature selection techniques have become popular. Among those, a comprehensive review of ensemble feature selection techniques is presented [31]. An ensemble feature selection on medical datasets is developed [32]. They combine three types of feature selection techniques (filter, wrapper, and embedded). As described in their study, these three methods are combined and they show that an average union and multi-intersection based ensemble feature selection approaches perform better than those of single feature selectors. They validate their methods with small scale and also high dimensional datasets. There are some other studies in the literature for ensemble feature selection methods [33], [34].

The dimensionality of hand-crafted image features by using deep autoencoders is reduced [35]. The authors use fusion and transfer learning and perform training with 10,000 images from Yahoo Flicker Creative Commons 100M dataset. They create four deep-autoencoder models with changing encoding-dimensions, starting from 32 up to 256. This work hyper-parameters are stated as, batch size 950 with 350 epochs, the loss function is L1 (mean absolute error) and the activation function is Rectified Linear Units (RELU). While some feature sets yield better results after features are used with autoencoder, some feature-sets are yield as not appropriate for autoencoder dimensionality reduction.

A distributed computation model to measure the quality of each feature with respect to multiple labels on Apache Spark is developed [36], [37]. A parallel algorithm with Graphics Processing Units (GPU) for computing the multi-label k-Nearest Neighbor classifier without any loss of accuracy is presented [38]. Experiments verify that it is able to achieve 200 times speed-up compared to a sequential execution with a single CPU.

Autoencoders with state-of-the-art dimensionality reduction algorithms on two different image datasets (Modified National Institute of Standards and Technology and Olivetti Face Datasets) are developed in [39]. PCA, linear discriminant analysis (LDA), locally linear embedding (LLE) and ISO map dimensionality reduction techniques are used during the experiments. The autoencoders are observed to provide results competitive with state-of-the-art algorithms. The autoencoders extract different structures than other methods. This property works well on the repetitive structures on simpler datasets. The number of hidden layer nodes should be equal to intrinsic dimensionality to get the best performance. Because this study is about images and pixel-wised reproducible representations, dimensionality reduction with autoencoder part and setting parameters (such as the number of nodes in the hidden layer) is studied in our research. Feature selection by shallow autoencoders on 7 benchmark datasets consisting of image and text data separately is developed in [40]. The authors report better solutions in most cases when the results are compared with Laplacian Score (LS), Multi-cluster Feature Selection (MCFS), Unsupervised Discriminative Feature Selection (UDFS) and regularized self-representation. Variational-autoencoder as an additional optimizer for the encoders is proposed in [41]. The model autoencoder consists of two stages. The first stage uses the

optimization of feature subsets and the second stage is used for shallow regularized autoencoder optimization concerning the weights and biases. The proposed method based on autoencoder is reported to give better results than other state-of-the-art algorithms.

There are many types of autoencoders and depending on the dataset, features, and correlations among labels, most of the time it is observed that autoencoders have good performance. Most models are created and evaluated on image data but to the best of our knowledge, our research is unique in that it uses image descriptors instead of pixel based images directly.

III. AUTOENCODERS AND DIMENSIONALITY REDUCTION

The aim of an autoencoder is to learn a representation (encoding) for input data, typically for dimensionality reduction, by training the artificial neural network in an unsupervised manner. During this process, the semantic structure of the data is learned with smaller representations; therefore, it is typically used for dimensionality reduction with synthetic features that are created by optimized weights and biases [42]. Information retrieval with dimensionality reduction was first implemented by Hinton and Salakhutdinov for semantic hashing in 2009 [43].

Autoencoders reduce the input dimension through the bottleneck (i.e. code) layer to have a smaller size representation of the actual data. They try to reproduce the input from that bottleneck layer as output. The simplest autoencoders are called vanilla autoencoders that contain only three layers (input, bottleneck/code, and output layers). It can be thought to be the skeleton of autoencoders in general. Mathematical representation of the autoencoder is stated in Equations 1 and 2. The function f refers to the encoder which takes input X as the parameter and creates the code h. The decoder function (g) takes as its an input the bottleneck layer (h) to reconstruct the output layer ($\hat{X}$) (as performed by a similar version of the input layer (X)). Finally, this similarity is measured by the loss function which is stated in Equation 3.

$$h = f(x) \tag{1}$$

$$\hat{X} = g(h) \tag{2}$$

$$L(x, g(f(x))) \tag{3}$$

Types of autoencoders are examined under two titles considering the output size of the encoders; under-complete autoencoders and over-complete autoencoders [44].

A. UNDER-COMPLETE AUTOENCODERS

When the number of nodes in the code layer (the output size for the encoder part of the autoencoders) is smaller than the input layer, these autoencoders are called under-complete autoencoders. As shown in the Figure 1, input, bottleneck, and the output dimensionalities are represented as $|X|$, $|h|$, and $|\hat{X}|$ respectively. The autoencoder tries to copy input to the output with learned coefficients and the size of the input dimension ($|X|$) is equal to the size of output dimension ($|\hat{X}|$). Also, the size of the input dimension ($|X|$) is greater than the size of the bottleneck layer ($|h|$) because the type of the autoencoder is an under-complete autoencoder. The main goal is to reduce the loss between the input and output, but it should not be zero to avoid memorizing or copying the input directly to the output. Therefore, the number of hidden layers is smaller than the input.

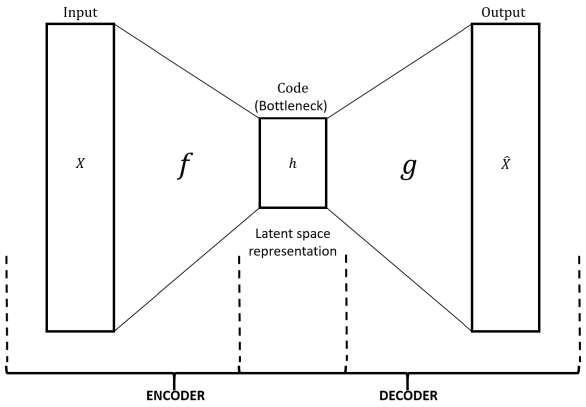

FIGURE 1. General structure of under-complete autoencoder.

B. OVER-COMPLETE AUTOENCODERS

For the over-complete autoencoders, the size of h might be equal or larger than the size of input X. Figure 2 demonstrates this structure. Over-complete autoencoders are preferred for classification purposes when the feature-set with higher dimensional representation is required. The technique of increasing the number of neurons to a higher dimensionality than the actual feature-set is used for extracting hidden structures in data with more significant features. However, the model can memorize directly without generating structural identifications of the input data. In order to avoid this situation, some regularization operations are used. The types of regularized autoencoders are sparse autoencoders, denoising autoencoders, contractive autoencoders, and regularized autoencoders with dropout. These are good for under-complete autoencoders to create better reduced dimensional representations.

C. REGULARIZATION FOR AUTOENCODERS

During the training of neural networks, it may be difficult to learn the key features to be able to perform prediction on previously unseen validation data using the existing dataset due to its unsuitability or small size. In addition to this, the model that has been learned might not be good enough. In such situations, some regularization techniques may be useful to tackle the problem. The first regularization mentioned and implemented in our study is the denoising autoencoder. The denoising autoencoders are created to prevent over-fitting and to extract better representations of the input data through the bottleneck. Another method is dropout regularization which is an extended version of the denoising autoencoder.

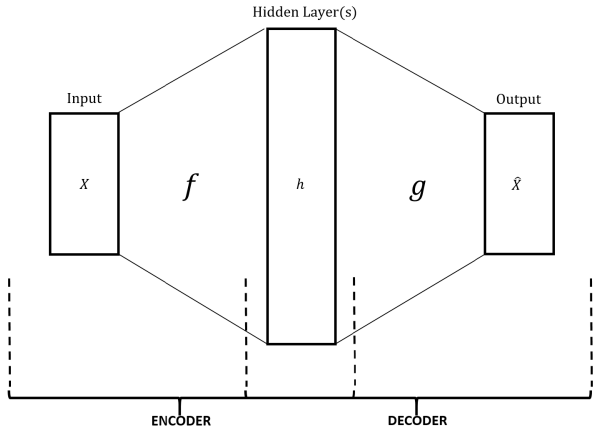

FIGURE 2. General structure of over-complete autoencoder.

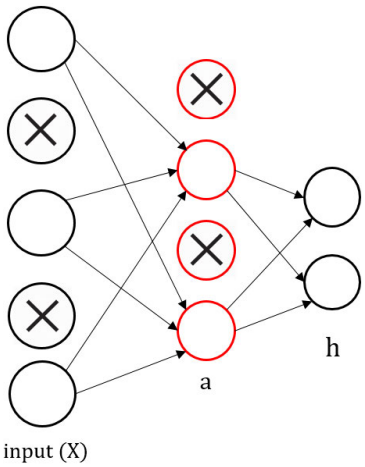

FIGURE 3. Dropout regularization for dimensionality reduction.

1) DROPOUT REGULARIZATION

In dropout regularization, some randomly selected nodes with all of its connections are dropped out with probability, p. Since this probability parameter is a hyper-parameter to apply dropout regularization, it should be tuned carefully. For most of the problems, p=0.5 gives successful results [45]. This regularization is applied to all hidden layers and the input layer within the encoder part of the autoencoder in contrast to the denoising autoencoder. In other words, while training, some nodes are discarded with their weights and biases. Therefore, efficient nodes from the high dimensional model are selected and this method is used to prevent over-fitting as denoising autoencoder. Figure 3 shows the structure of the dropout regularization (nodes marked with X represent dropped out nodes).

2) DENOISING AUTOENCODERS

are regularized forms of autoencoders to force the model for better learning [46]. As shown in Figure 4, a specific rate of noise is added to the input layer or randomly selected nodes are blanked-out from the input layer. Later, the model is trained through some noisy input so that the input is not the same as the output as in regular autoencoders. The model of the autoencoder is trained with this noisy input and forced to avoid this noise. This process extracts better representations for dimensionality reduction.

IV. MULTI-LABEL VIDEO DATA CLASSIFICATION ALGORITHMS

In this section, we explain multi-label image and video data classification algorithms, multi-objective NSGA-II, autoencoders, and ensemble algorithms. Binary Relevance (BR) [47], Classifier Chains (CC) [48], Pruned Sets (PS) [49], and Random k-Labelsets (RAkEL) are used as multi-label classification algorithms. Basic classifiers that are applied for multi-label classification algorithms are SVM, LR, and Decision Tree (J48).

To deal with multi-label classification problems, three main approaches are applied: data transformation, method adoption and ensemble-based classifiers [47]. For the data transformation approach, the multi-label data is transferred into multi-class or binary-class data, and then the problem is solved with base classifiers and the results are combined. The best known algorithms included in this study are BR, CC, Label Powerset (LP) and PS. In the adoption approach, the existing classification algorithms that solve multi-class or binary-class problems are modified as its multi-label version. Therefore, each algorithm has a different and unique solution in the method adoption approach. The third approach ensembles the algorithms used in this study and uses the advantage of assembling these algorithms. The well-known ensemble multi-label classifier is RAkEL that involves both BR and LP.

NSGA-II is a classical population-based multi-objective algorithm developed by Deb *et al.* [11]. We implement the parallel version of this algorithm for the experimental comparisons of our proposed algorithms (autoencoder and ensemble algorithms). The NSGA-II algorithm starts with a random initial population of chromosomes. Each chromosome has a selected set of features for a given dataset. Non-dominated sorting operation is performed by considering the Pareto-fronts. Individuals in smaller fronts have higher priorities. The binary tournament method is applied to generate a new population. At each crossover and bit-flip mutation operations, two new children are generated. Only the best half of the individuals is used to breed a new population. Individuals with worse fitness values are eliminated. When the maximum number of generations has been produced, the algorithm terminates [11]. The pseudocode of the NSGA-II algorithm is provided in Algorithm 1.

The evaluation metrics used for multi-class or binary classification cannot be used directly for multi-label classification. The accuracy of the labels must be taken into account in the label set. In this way, Hamming loss is a sample-based metric that is used primarily. The loss measure is calculated for each instance and an average value is calculated. The symmetric difference (Δ) is found between the prediction and the actual label sets for all labels per instance (Equation 4).

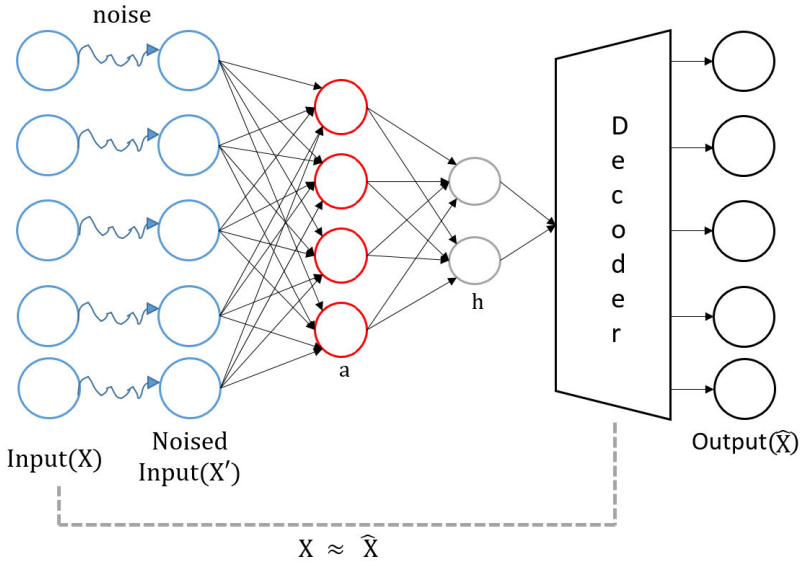

FIGURE 4. Denoising autoencoder structure (*h* represents bottleneck and *a* represents hidden layer(s) of autoencoder).

Algorithm 1: The Pseudocode of NSGA-II

```
#gen: number of generations;
P ← generate an initial population randomly;
S ← {} // set of evaluated chromosomes
for i ← 1 to #gen do
    foreach u in P do
        if u does not exist in S then
            u.objective_1 ← #selected features;
            u.objective_2 ← HammingScore(u, mca);
            S ← S ∪ {u};
        else
            u.objective ← S[u].objective;
        end
    end
    P ← NSGA-II(P) //generate a new population
end
return ParetoOptimalSolutions(P);
```

Then, it is normalized according to the number of instances and the number of labels [47].

$$HammingLoss = \frac{1}{n}\frac{1}{k}\sum_{i=1}^{n} \mid Y_i \Delta Z_i \mid \tag{4}$$

Deep denoising autoencoder: Autoencoders are well-known architectures due to their efficiency in dimensionality reduction. We use an under-complete deep autoencoder with 10 encoder layers and 10 decoder layers for the solution of our problem. All of the layers in encoder and decoder are fully-connected (dense) with additional dropout layers to avoid overfitting. Encoder and decoder layers are symmetric for both the number of nodes and layers. This structure is preferred based on the implementation of autoencoders with a similar purpose as in the study of Petscharnig *et al.* [35]. Since our aim is to learn latent space features while reconstructing the input, sharing weights in this way is more reasonable. In addition to this, we work on SIFT and Segmentation-based Fractal Texture Analysis (SFTA) local image descriptors, while the features are being extracted. Data is turned into 'flat images'. Because of this, autoencoder layers are selected as 'dense' layers.

After the training and testing processes are performed on our autoencoder model, latent space representation is extracted as reduced dimensional synthetic data. After being sure about the number of layers in our deep autoencoder, we create a 10-layered network that has symmetrical layers in encoder and decoder. Since the number of nodes is reduced in the code layer, all intermediary layers are candidates for being latent space representation. Through this perspective, all reduced dimensional representations are extracted to compare as shown in Figure 5. We have created ten different autoencoders with varying numbers of layers.

Figure 6 shows our autoencoder model that is configured for both datasets concerning the number of layers. Ten encoder layers as Dense with RELU activation and dropout layers with 0.5 probability. The number of nodes for each layer is changed. For the MirFlickr dataset that has 42 features originally, the number of nodes starts from 38 to 2 as the bottleneck. For WMS dataset with 100 features, the number of nodes starts from 90 to 10 up to the code layer in the bottleneck and it is selected as 5. These values (bottleneck values) are selected in this way like the nodes of the code layer are decided for the general number of selected features from the NSGA-II. Other layers are distributed concerning the code and the input sizes. Denoising autoencoder is implemented for the same structure.

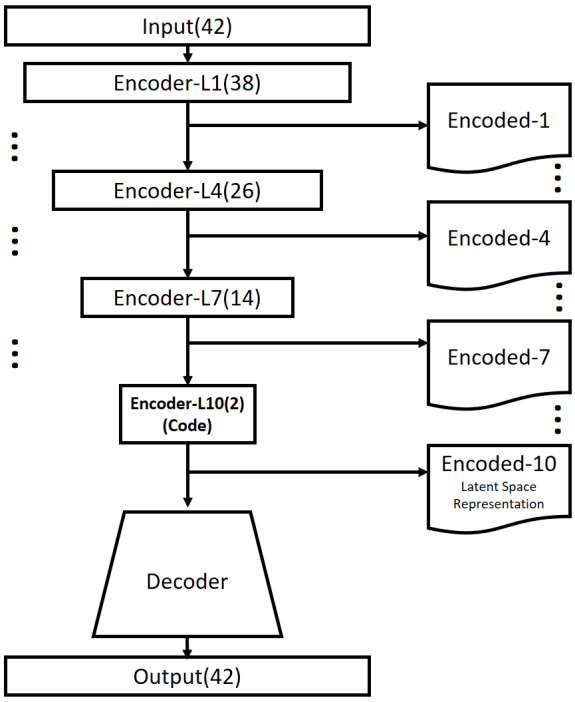

FIGURE 5. Reduced dimensional representation extraction on autoencoder.

Parallel multi-objective evolutionary algorithms are efficient tools for the optimization of NP-Hard problems [50], [51]. The performance of the optimization can be considerably improved by using a well-grained parallel calculation of chromosomes with intelligent operators (mutation and crossover). The fitness calculation of the chromosomes in this study requires a lot of time because of the applied machine learning techniques. This process prevents the exploration of more subset of features of selected elements. Therefore, we implement a Parallel-NSGA-II algorithm [52]. The proposed algorithm by Multi-Objective Evolutionary Algorithm (MOEA) framework keeps a population at the memory of the master processor and calculates the fitness values of the chromosomes at each slave processor. Since the calculation of the accuracy with a selected number of features is well-grained, it is observed that this parallelization technique of the conventional NSGA-II gets an almost linear speed-up during the experiments. It is possible to calculate a larger number of fitness values and obtain better results than the standard (serial) version of the NSGA-II algorithm.

V. PERFORMANCE EVALUATION OF THE ALGORITHMS

The experiments are performed on a computer with 8 core 64-bit CPU (I7-3632QM, 2.20GHz). The algorithms are developed with Java programming language and the MOEA framework [52]. Multi-label machine learning algorithms are implemented with MEKA (a multi-label extension of Waikato Environment for Knowledge Analysis (WEKA) machine learning toolkit) [53]. Deep autoencoder is implemented in Python programming language with Keras library that uses Tensorflow backend.

Multi-label machine learning algorithms are selected from a rich set of multi-label classification approaches. Data-transformation approaches based on multi-class and binary classification problems and ensemble-based approaches are applied. Additionally, some recent versions of the algorithms are applied on the datasets and by considering the results based on both the execution time and the success of the algorithms, BR, CC, PS, and RAkEL are selected as multi-label classification algorithms.

In our experiments, two multi-label video/image datasets are used to verify the algorithms. The first dataset is the most widely used and publicly available image dataset MIR-Flickr [54]. This dataset consists of 25,000 images. Important features of the dataset are extracted in a study by Costa *et al.* [55]. This feature set that is extracted with the Segmentation based Fractal Texture Analysis (SFTA) algorithm is used in our experiments. This extraction creates binary images with binary stack decomposition. Extracted features are transformed into vectors as feature sets [56]. There are 42 features in MIR-Flickr dataset and at most 23 labels for each image (Car, Bird, Lake, Night, Water, Sky, People, Baby, Clouds, Tree, Portrait, Dog, Animals, Female, Transport, Flower, Indoor, Male, Food, River, Structures, Sea, Sunset). The labels and the correlations between the labels are presented in Figure 7. With respect to the correlation chart, the most correlated labels are people and males, sky and clouds. Some interesting correlations are revealed such as baby and sky exist together in almost all samples of a baby. The night and male join have occurred in half of the night samples. Since this dataset has many aspects and repetitive structures are rare, the number of unique label-sets is 390 and the maximum occurrence for a label-set is 37 with labels; structures, sunset, transport, and indoor.

The second dataset is created by recording videos using the WMS dataset that is designed for our earlier research. The recorded video files are split into five-second shots and all of the objects are identified by a human user and manually annotated as ground truth. There are three possible labels (person, group of people and vehicle). After the annotation process is completed, SIFT features are extracted based on key-point localization of objects [1]. The implementation is done using OpenCV library and the Python programming language [57]. Once the SIFT features are produced, the codebook is constructed to obtain a dictionary of visual words. During the construction of the codebook, the k-means clustering algorithm is applied to determine the centroids. Then, L1 normalization is applied to obtain the final version in the form of 100 bags of visual words for each frame. The data is extracted from 3-minute videos and 1000 video frames are used for feature extraction. Figure 8 shows the correlation between labels. Most correlated labels are person and groups of people. A person and a vehicle exist together for nearly 80% of vehicle object samples. Additionally, since

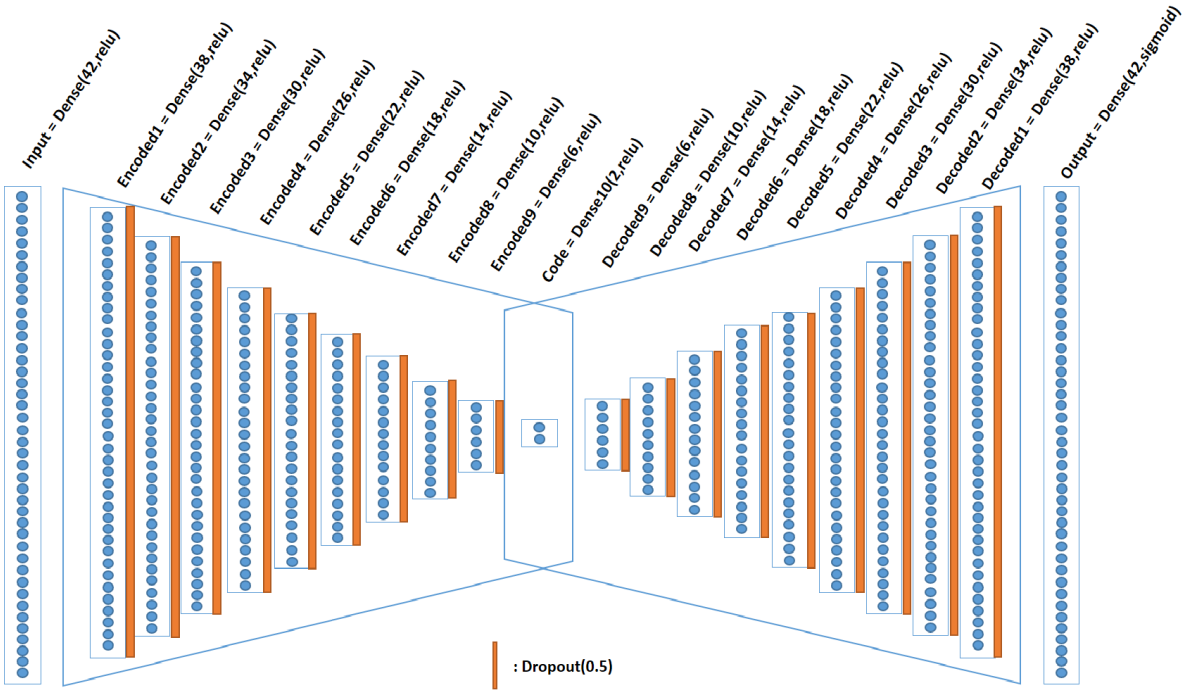

FIGURE 6. Autoencoder model for MirFlickr datasets.

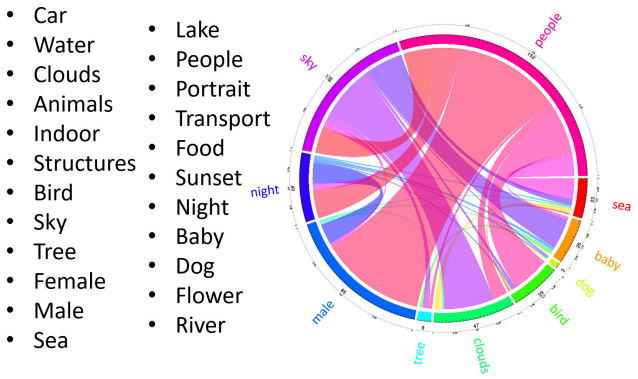

FIGURE 7. Labels and the chart of correlations for MIR-Flickr dataset.

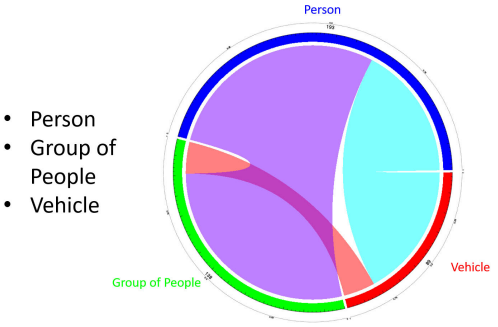

FIGURE 8. Labels and the chart of correlations for WMS dataset.

tree labels are available in the WMS dataset, 8 label-sets occur and the most correlated labels are person and vehicle with 70 samples.

The results reported here are the averages of five executions with five-fold cross-validation. This method is used to minimize the impact of random factors. The dataset is divided into five equal-size partitions and four of them are used for training. The remaining partition is used for testing. The average of these five executions is the final accuracy value. This is one of the most common techniques in the literature to evaluate the predictive accuracy of machine learning algorithms. The parameters used in the experiments for the NSGA-II algorithm are presented in Table 3. These parameters are decided after comprehensive experiments. The proposed method is sensitive to its parameters. We use the best parameter settings of NSGA-II that have been provided by previous studies.

Ensemble feature selection uses multi-label classification algorithms. All algorithms are selected considering different types of multi-label classification approaches. Methods BR and CC are used for binary data transformation and for multi-class data transformations PS is used. The Random k Labelset algorithm is implemented as an ensemble multi-label classifier, which is one of the state-of-the-art ensemble multi-label classifiers. In order to observe the performance difference between classic multi-label classification methods and state-of-the-art ensembles for multi-label classification, EnsembleML is implemented with widely used multi-label classification library, Meka. All other four implemented multi-label classification algorithms are implemented in ensemble version and results are represented in Tables 1 and 2 for WMS and MIR-Flickr dataset

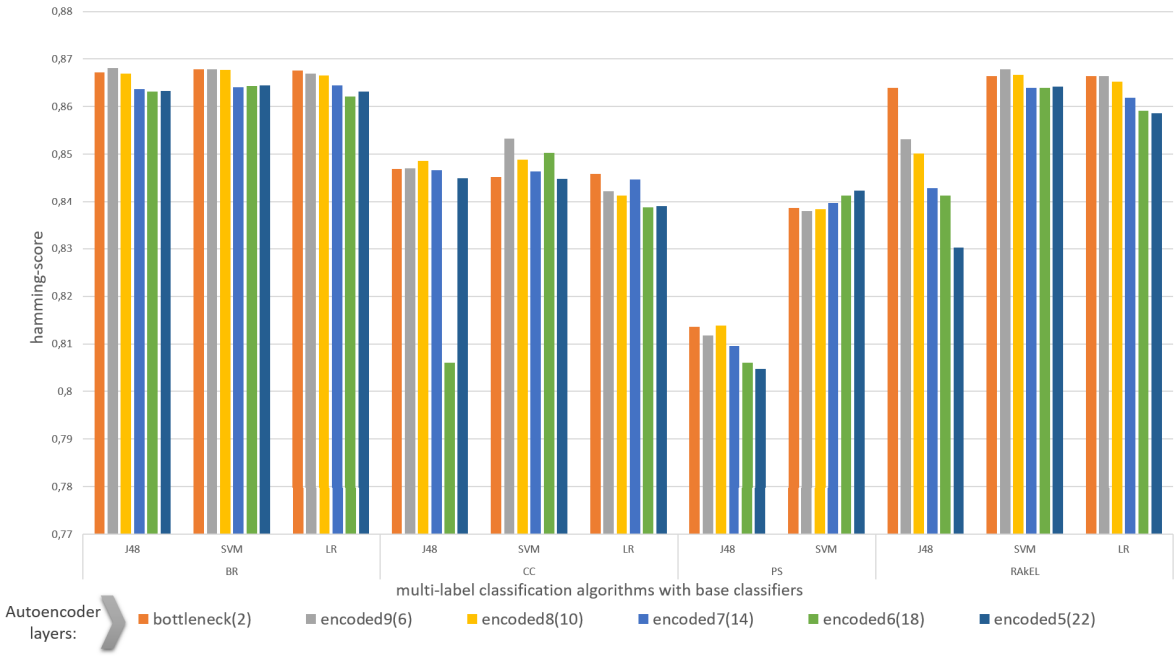

FIGURE 9. Setting the number of layers and the results of the algorithms for MirFlickr dataset (The results of the PS-LR are not added due to its long execution time).

TABLE 1. Ensemble Multi-label classification performances on WMS dataset considering both Hamming-score and execution time.

	Hamming-Score	Total-time
PS	0.674	0.123
EPS	0.775	0.754
BR	0.713	0.176
EBR	0.79	1.032
CC	0.711	0.182
ECC	0.781	1.043
RAkEL	0.674	1.174
ERAkEL	0.512	7.209

TABLE 2. Ensemble Multi-label classification performances on MIR-Flickr dataset considering both Hamming-score and execution time.

	Hamming-Score	Total-time
PS	0.805	0.633
EPS	0.852	3.285
BR	0.859	0.496
EBR	0.866	2.69
CC	0.84	0.753
ECC	0.861	5.561
RAkEL	0.802	0.844
ERAkEL	0.849	4.175

respectively. For all results, five-fold cross validation is applied and J48 decision tree algorithm is used as a base classifier.

Tuning the parameters of an autoencoder has an important effect on its performance. Since an autoencoder is a neural network, the number of layers and the number of nodes in each layer should be set properly for the model. The method

TABLE 3. Parameter settings for the NSGA-II algorithm (*N* is the number of individuals in the population).

Parameter	Value
Population size	50
Crossover rate	1.0
Mutation rate	1/*N*
Distribution index for mutation	20.0
Distribution index for crossover	15.0

explained in Section IV with Figure 5 is applied to select the number of layers and the termination condition. As a result of this operation, the bottleneck layer with 2 nodes and previous 5 layers with 6, 10, 14, 18, and 22 nodes are trained and reduced dimensional features are extracted from these layers for the MirFlickr dataset. Other layers are not tested because accuracy is decreased with earlier layers. With respect to the results displayed in Figure 9, layer-10 is selected. This layer which is also a bottleneck (the least number of nodes) of the model with 2 nodes provides better results. Additionally, a similar methodology is applied on WMS dataset and again 10 layers are used to detect the required number of layers. The results are given in Figures 10, 11, 12, and 13 for the BR, CC, PS, and RAkEL algorithms respectively. Concerning these results, Layer-6 with 40 nodes and Layer-5 with 50 nodes are selected. Layer-2 has good results but the number of features is too high for a dimensionality reduction.

Since implemented autoencoder is under-complete and aims dimensionality reduction, the number of nodes is started at a value smaller than the number of input features and it is reduced through bottleneck which is determined in the

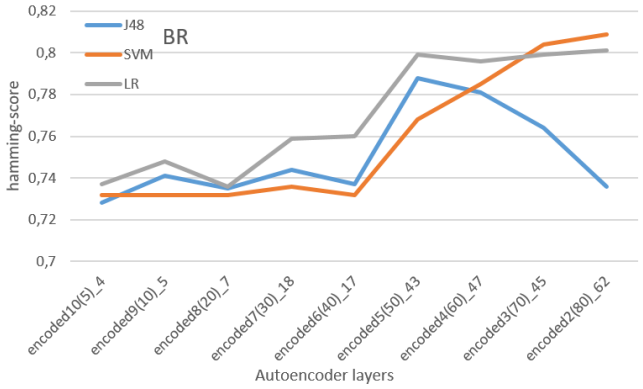

FIGURE 10. Setting the number of layers, the results of the BR algorithms for WMS dataset.

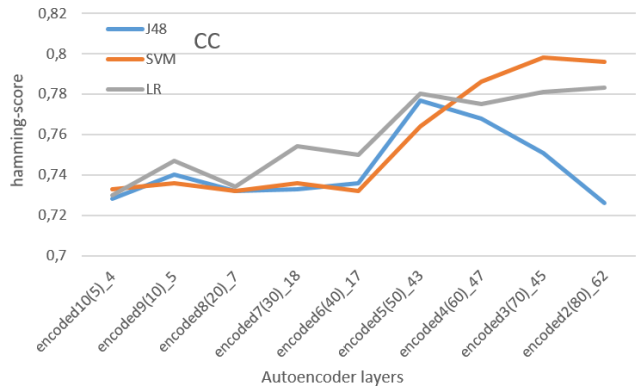

FIGURE 11. Setting the number of layers, the results of the CC algorithm for WMS dataset.

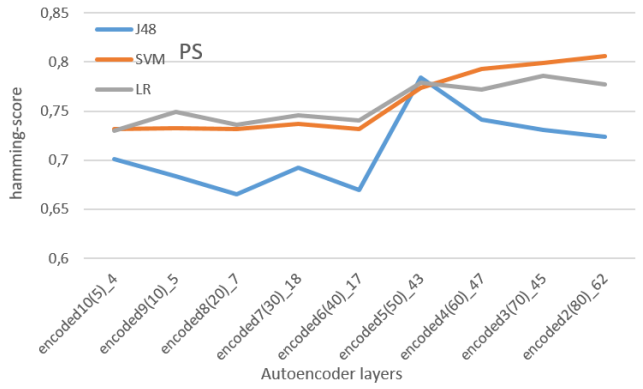

FIGURE 12. Setting the number of layers, the results of PS algorithm for WMS dataset.

previous step. Since MirFlickr dataset has 42 features, the number of nodes of the first layer is started from 38 and reduced by 4 at a time until it becomes 2. Similarly, WMS dataset has 100 features. The number of nodes in the first layer is started from 90 and reduced by ten until the bottleneck. The bottleneck has 5 nodes for this model.

By tuning the layers and number of nodes for MirFlickr dataset, 10 layers are selected with encoding-dimension 2. For WMS dataset, 5 layers are selected with encoding-dimension 50.

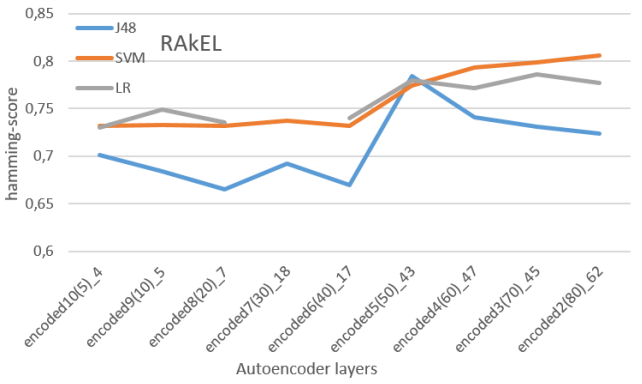

FIGURE 13. Setting the number of layers, the results of the RAkEL algorithms for WMS dataset.

TABLE 4. Selected parameters for the MirFlickr and WMS datasets.

	MirFlickr Dataset	WMS Dataset
Activation Function	relu, last layer sigmoid	relu, last layer sigmoid
Optimizer	Adam	Adam
Learning Rate	0.001	0.001
Loss Function	Mean Absolute Error	Mean Squared Error
Batch Size	256	128
Number of Epochs	100	300

The activation function, optimizer, learning rate, loss function, batch size and the number of epochs are also tuned experimentally. Adam [58] and Stochastic Gradient Descent (SGD) are used optimizers with 0.1, 0.01, 0.001 learning rates (See Figure 14).

Sigmoid and RELU are applied as loss functions, mean absolute error and mean squared error respectively to select best-fitted parameters as activation functions. 32, 128 and 256 batch sizes are tried on both datasets. For MirFlickr and WMS datasets, 100 and 300 epochs are applied respectively. Parameters used for both datasets are given in Table 4.

A. THE RESULTS OF THE DENOISING AUTOENCODER

In order to analyze the effect of another type of regularization than the dropout, denoising autoencoder is examined. All parameters remain the same as in the dropout regularized autoencoder. Dropout layers are omitted and noise is added to the input layer in different noise levels.

For WMS dataset denoising autoencoder is applied with varying noise factors 0.3, 0.5, 0.8 and results are given in Table 6. When average execution results are considered, 0.5 is selected as the best noise factor. However, varying all different noise factors, denoising autoencoder results are not better than those of dropout regularized autoencoder.

The same experiments are applied on the MIR-Flickr dataset and results are given in Table 7. When the noise factor is selected as 0.5, six (6) algorithms have better performance than the noise factor selection of 0.8 and the results of five (5)

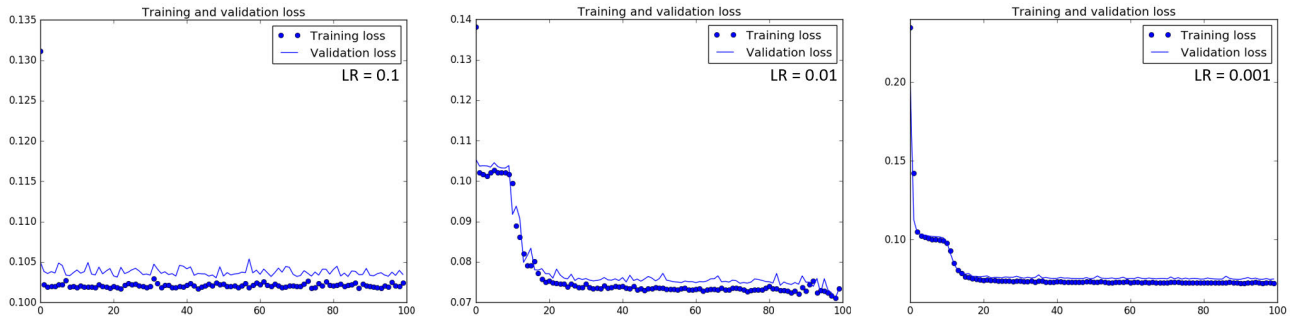

FIGURE 14. Applied different learning rates and the results of the parameter tuning experiments.

TABLE 5. The execution times of denoising and dropout regularized autoencoders in seconds.

	Run	Denoising AE	Dropout regularized AE
WMS	1	30.0	27.2
	2	26.3	24.7
	3	25.7	28.6
	4	26.1	26.9
	5	25.8	30.4
Mean Average		26.8	27.6
MIR-Flickr	1	76.5	84.9
	2	79.3	78.7
	3	82.9	99.6
	4	83.4	77.7
	5	74.2	82.2
Mean Average		79.2	84.6

TABLE 6. The results of the denoising autoencoder with noise factors 0.3, 0.5, 0.8 and comparison with dropout regularized autoencoder with WMS dataset. (NF is Noise Factor).

MLC	Base Classifier	NF: 0.3	NF: 0.5	NF: 0.8	AE with Dropout
BR	J48	0.69555	0.72622	0.71333	0.788
	SVM	0.73155	0.73155	0.73155	**0.809**
	LR	0.72488	0.72311	0.72266	0.799
CC	J48	0.71198	0.71822	0.69333	0.777
	SVM	0.72533	0.73199	0.73155	0.776
	LR	0.72088	0.71333	0.71777	**0.778**
PS	J48	0.66399	0.64044	0.64355	0.761
	SVM	0.71688	0.72755	0.73155	0.778
	LR	0.71155	0.70977	0.71822	**0.779**
RAkEL	J48	0.66399	0.64044	0.64355	0.761
	SVM	0.71688	0.72755	0.73155	0.773
	LR	0.71155	0.70977	0.71822	**0.779**
Average		0.70792	0.70833	0.70807	**0.77983**

algorithms are better than those with 0.3. If the mean averages of different noise factors are compared, the best selection does not change. For this dataset, the dropout regularized autoencoder performs better than denoising autoencoder. The difference between results on denoising and dropout regularized autoencoders is less than that is observed on the WMS dataset. As a result, since the dropout regularized autoencoder performance is better than denoising autoencoder for both datasets, dropout regularized autoencoder is used in the other experiments using ensembling.

TABLE 7. The results of the denoising autoencoder with noise factors 0.3, 0.5, 0.8 and comparison with dropout regularized autoencoder for MIR-Flickr dataset. (NF is Noise Factor).

MLC	Base Classifier	NF: 0.3	NF: 0.5	NF: 0.8	AE with Dropout
BR	J48	**0.86778**	0.86769	**0.86778**	0.8669
	SVM	**0.86778**	**0.86778**	**0.86778**	0.8683
	LR	**0.86778**	0.86773	0.86704	0.8675
CC	J48	0.83326	0.83360	0.83334	0.8486
	SVM	0.84647	0.84647	0.84647	**0.8532**
	LR	0.83773	0.83378	0.83313	0.8433
PS	J48	0.81469	0.83273	0.81847	0.814
	SVM	0.83830	0.83847	0.83569	**0.8423**
	LR	-	0.83547	-	0.8397
RAkEL	J48	0.86204	0.86713	0.86291	0.859
	SVM	**0.86778**	**0.86778**	**0.86778**	0.86647
	LR	0.86686	0.86760	0.86699	0.86647
Average		0.85158	0.85371	0.85219	**0.85419**

The execution times of denoising autoencoder and dropout regularized autoencoder are presented in Table 5. Mainly, dropout regularized autoencoder takes slightly more execution time than denoising autoencoder since dropout regularization is applied on each hidden layer but denoising autoencoder uses only the input of the model. Although the execution times of both autoencoders are not much different, when compared with the NSGA-II, autoencoder takes much less execution time. While autoencoder takes 0.80 seconds on average for MIR-Flickr dataset, NSGA-II takes a minimum of 20 minutes for the execution of a generation in the average with parallel implementation. The execution time of our ensemble approaches used in this study changes depending on the number of ensembled features.

The computational complexity of the algorithms mainly depends on the performance of the machine learning algorithms. The techniques used in this study are polynomial time processes. The cross validation is another reason of the long execution times. Training period of the datasets with many features takes longer times than the execution time of smaller feature sets. NSGA-II has a termination condition that depends on the number of the generations and autoencoders work with the number of epochs. These are the main parameters for the cost effectiveness of the proposed methods in our study.

TABLE 8. Overall results for WMS dataset with autoencoder and ensembled NSGA-II and autoencoder. (HS is Hamming-score, #Feat is Number of Features). Algo-I is NSGA-II-BRJ48 & AE-40, Algo-II is one more step optimization of Algo-I results with NSGA-II, Algo-III is NSGA-II-BRSVM & AE-50, Algo-IV is one more step optimization of Algo-III results with NSGA-II (NA stands for Not Available).

		Before Feat.Sel.		NSGA-II		Autoencoder		Algo-I		Algo-II		Algo-III		Algo-IV	
MLC Algorithm	Base Classifier	HS	#Feat.	HS	#Feat.	HS	#Feat.	HS	#Feat.	HS	#Feat.	HS	#Feat	HS	#Feat.
BR	J48	0.6447		0.7848	14	0.778	40	NA		NA	NA	NA		NA	NA
	SVM	0.7164		0.8493	42	0.809	80	NA		NA	NA	NA		NA	NA
	LR	0.6940		**0.8455**	32	0,799	40	NA		NA	NA	NA		NA	NA
CC	J48	0.6455		0.7813	14	0,777	40	NA		NA	NA	0.7364		0.7983	22
	SVM	0.7181		0.7658	8	0,776	11	0.7793		0.7834	4	NA		NA	NA
	LR	0.6877	100	0.7776	11	0,778	10	0.7817	54	0.7871	8	**0.8043**	92	0.7876	12
PS	J48	0.6206		0.7557	11	0.761	10	NA		NA	NA	0.7451		0.7907	34
	SVM	0.7037		0.7772	9	0.778	11	**0.7920**		0.7822	12	NA		NA	NA
	LR	0.6624		0.7671	45	0.779	10	0.7713		0.7835	8	0.7652		0.7872	18
RAkEL	J48	0.6206		0.7523	10	0.761	10	NA		NA	NA	0.7450		0.7787	5
	SVM	0.7037		0.7649	47	0.773	10	0.7921		0.7782	7	NA		NA	NA
	LR	0.6624		0.7713	21	0.779	10	0.7802		0.7862	10	0.7745		**0.8127**	38

B. HETEROGENEOUS ENSEMBLE APPROACH FOR FEATURE SELECTION

There are two types of feature selection ensembles, homogeneous and heterogeneous [31]. In the homogeneous approach, the same feature selection algorithm is applied on different subsets of data and the results of all the subsets of features obtained using the same algorithm are aggregated. In the heterogeneous approach, there are multiple feature selection algorithms applied on the same dataset. After all of the feature selection algorithms are applied and reduced dimensional feature-sets are recorded, all of these results are ensembled.

We implement heterogeneous approach for the feature selection (See Figure 15). We use two different feature selection algorithms in our proposed approach. Since autoencoders are used for dimensionality reduction, latent space representation of the implemented autoencoder model is saved as the produced synthetic features from actual input. On the other hand, features discovered by the genetic algorithm are saved for a result of heterogeneous ensemble feature selection scheme. The collection of this new feature-set is given as input to the genetic algorithm for the next step of optimization.

C. THE COMPARISON OF AUTOENCODER RESULTS AND ENSEMBLE WITH NON-DOMINATED SORTING GENETIC ALGORITHM-II

After having the results of the NSGA-II algorithm, one of the most popular dimensionality reduction algorithms from the aspect of deep neural networks, autoencoder is tested. Table 8 presents the overall results of WMS dataset based on the autoencoder and ensemble of NSGA-II and autoencoder. For all steps, the number of features and hamming-score values are stated. BR algorithm does not perform well with reduced dimensional autoencoder features as image descriptor features used with the NSGA-II algorithm. Almost all algorithms have better results considering both objectives which are the number of features and Hamming-score.

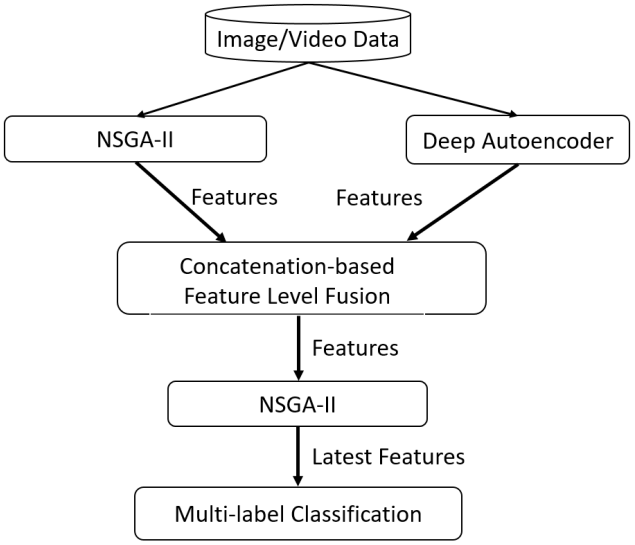

FIGURE 15. Proposed Heterogeneous Ensemble Feature Selection Schema.

We concatenate reduced dimensional datasets that are generated from autoencoder and NSGA-II algorithms. Two different merged datasets are revealed. By selecting the subset of datasets generated by the NSGA-II algorithm to merge, different aspects are applied. The first one selects the subset considering the Pareto-optimal results which are selected as BR-J48 algorithm with 14 features, and the second one selects subset considering the highest Hamming-score which is BR-SVM with 42 features. Then the selection of reduced dimensional representations by autoencoder is performed in the selection of the number of layers. These combinations are merged and MLC machine learning algorithms are applied to evaluate newly merged datasets. Since features are generated with results of BR and J48 algorithms for the first merged dataset, BR and J48 algorithms are not applied again. These rows are represented as NA. Similarly, BR and SVM algorithms are used when the dataset is created for Algo-III. Because of this reason, these algorithms are not

TABLE 9. Overall results for MirFlickr dataset with autoencoder and ensembled NSGA-II and autoencoder. (HS is Hamming-score, #Feat is Number of Features). Algo-V is NSGA-II-CCJ48 & AE-2, Algo-VI is NSGA-II-BRJ48 & AE-2, Algo-VII is one more step optimization of Algo-VI (NA stands for Not Available).

		Without Feat. Sel.		NSGA-II		Autoencoder		Algo-V		Algo-VI		Algo-VII	
MLC Algorithm	Base Classifier	HS	#Feat.	HS	#Feat.	HS	#Feat.	HS	#Feat.	HS	#Feat.	HS	#Feat.
BR	J48	0.8618		**0.8886**	6	0.8669	10	NA		NA		NA	NA
	SVM	0.8657		0.8796	25	0.8683	2	0.8658		NA		NA	NA
	LR	0.8611		0.8667	4	0,8675	2	0.8648		NA		NA	NA
CC	J48	0.8383		**0.8860**	5	0,8486	10	NA		NA		NA	NA
	SVM	0.8511		**0.8860**	5	0,8532	6	NA		0.8492		0.8521	1
	LR	0.8431	42	0.8653	6	0,8433	2	NA	8	0.8456	8	0.8475	2
PS	J48	0.8079		0.8504	4	0.8140	2	NA		NA		NA	NA
	SVM	0.8429		**0.8601**	9	0.8423	22	0.8436		0.8430		0.8439	2
	LR	0.8020		0.8214	15	0.8397	2	-		-		-	-
RAkEL	J48	0.8023		0.8225	8	0.8590	2	NA		NA		NA	NA
	SVM	0.8654		0.8189	9	0.8664	2	0.8654		**0.8678**		**0.8678**	1
	LR	0.8449		0.8121	12	0.8665	2	0.8641		0.8633		0.8674	1

used for validation of the Algo-III and Algo-IV and represented as NA.

For the final stage, the results of these merged algorithms (Algo-I and Algo-III) are applied on NSGA-II again for one more optimization step. The results of both algorithms (Algo-II and Algo-IV) are improved in terms of the number of features and Hamming-score when compared to the previous step considering all objectives on 6 MLC machine learning algorithm combinations.

Table 9 presents the overall results of MirFlickr dataset based on autoencoder and ensemble version with NSGA-II. Similar operations are performed for MirFlickr dataset. The results of 5 algorithms are improved when compared with NSGA-II. CC-J48 with 6 features are concatenated with autoencoder model that has 2 dimension (Algo-V) and similarly, BR-J48 with 6 features are concatenated with autoencoder with the encoded dimension-2 (Algo-VI). Since Algo-VI results are better than Algo-V, one more step optimization is applied on Algo-VI with NSGA-II. The best results are recorded by RAkEL algorithm. All of applied MLC machine learning algorithm results are improved with autoencoder, Algo-V, Algor-VI and Algo-VII when compared to before feature selection and previous steps.

In conclusion, autoencoder based dimensionality reduction performs better on a dataset that is simple and includes repetitive structures and fewer labels, as in WMS dataset [39].

D. THE RESULTS OF CORRELATION BASED FEATURE SELECTION, INFORMATION GAIN, PRINCIPAL COMPONENT ANALYSIS ALGORITHMS

Our algorithms are also compared with PCA, IG, and CBFS. PCA is a linear dimensionality reduction technique that uses linear mapping via covariance or correlation relationship between features. Though variance of the low dimensional data is maximized and by using eigenvectors, most related features arise. This algorithm is based on a study by Pearson [59]. This supervised dimensionality reduction technique is revised in a book by Jolliffe [60]. The other implemented algorithm is IG, which is used for splitting decision trees but it is also a popular feature selection technique. The difference between the entropy of dataset D and the weighted sum of selected subset entropies is calculated as the information gain and the highest is selected as the strongest feature. For this purpose, searching is performed via ranking all attributes. Multi-label classification techniques are used while applying IG on multi-label data. Binary relevance based IG results are evaluated on other multi-label classification algorithms. The last feature selection algorithm we use is CBFS. It is a filter-based feature selection algorithm and ranks features by a heuristic evaluation function given in Equation 5. The average class-feature correlation is represented as r_{cf}. r_{ff} represents the average feature-feature correlation where k represents the number of features. The subsets are evaluated considering feature-feature and feature-class correlations of all features. Termination is performed by the 'best-fit' search method. If five consecutive subsets are not improved over the current best subset, then searching is terminated.

$$\mu_s = \frac{kr_{cf}}{\sqrt{k + k(k-1)r_{ff}}} \tag{5}$$

The algorithms are evaluated on both datasets. Tables 10 and 11 present the results on WMS and MIR-Flickr datasets, respectively. For both datasets, BR, CC, PS, and RAkEL multi-label classification algorithms are applied with base classifiers J48 decision tree, SVM and LR on new reduced subsets.

For the WMS dataset, similar results are recorded. Our proposed feature selection approach has further optimized ensemble feature selection. With NSGA-II (Algo-IV), the results are 0.7983 Hamming score with 22 features on the CC-J48 algorithm combination. CBFS can reach

TABLE 10. The results of CBFS, IG, and PCA algorithms on WMS Dataset.

Algorithm	Base Classifier	CBFS	IG	PCA
BR	LR	0.75167	**0.79200**	0.71810
	SVM	0.75200	0.79033	0.72238
	J48	0.71700	0.72767	0.70143
CC	LR	0.74033	**0.78833**	0.70238
	SVM	0.75167	0.78383	0.72286
	J48	0.71900	0.71883	0.67286
PS	LR	0.74700	0.78400	0.71095
	SVM	0.74700	**0.78650**	0.72810
	J48	0.68700	0.71400	0.66000
RAkEL	LR	0.74700	0.78400	0.71095
	SVM	0.74700	**0.78650**	0.72810
	J48	0.68700	0.71400	0.66000

TABLE 11. The results of CBFS, IG, and PCA algorithms on MIR-Flickr Dataset.

Algorithm	Base Classifier	CBFS	IG	PCA
BR	LR	0.86452	0.86428	0.86583
	SVM	0.86570	0.86570	**0.86778**
	J48	0.86335	0.86265	0.86409
CC	LR	0.84726	0.84596	0.84543
	SVM	**0.85787**	0.85726	0.84561
	J48	0.84674	0.84574	0.84435
PS	LR	-	-	-
	SVM	0.84139	**0.84165**	0.84048
	J48	0.81104	0.80615	0.80904
RAkEL	LR	0.85783	0.85833	0.86313
	SVM	0.86543	0.86496	**0.86739**
	J48	0.81100	0.81178	0.81709

0.7520 Hamming-score and IG has 0.79200 Hamming-score. PCA has the worst results (see Table 10).

With MIR-Flickr dataset, our proposed feature selection approach, which is further optimized ensemble feature selection, NSGA-II (BR-J48 & AE) (Algo-VII) has reached 0.8678 Hamming-score value with one feature. With the same algorithm combination, CBFS reports 0.86335 Hamming-score value with 17 features, IG reports 0.86265 with the same number of features and PCA has better results than both CBFS and IG (see Table 11).

The results obtained by our final approach include further optimization of the ensemble feature selection algorithms (Algo-II, Algo-IV for the WMS dataset, Algo-VII for the MIR-Flickr dataset). All the results are better for three algorithms out of five compared to the results of the autoencoder. However, the genetic algorithm (NSGA-II) gives better results than the autoencoder for the MIR-Flickr dataset. However, the autoencoder results on the WMS dataset are better than NSGA-II in eight out of twelve algorithms, since the WMS dataset includes many repetitive structures and the number of possible labels or the number of objects that should be found in the frames is not high, unlike the MIR-Flickr dataset. As far as the selection approaches of ensemble feature selection (Algo-I and Algo-III) are concerned, they work much better than the genetic algorithm (NSGA-II) and autoencoder. Finally, by applying further optimization on the selection of ensembled feature sets, we obtain the best results from these optimized algorithms (Algo-II and Algo-IV). In addition, Algo-IV improves the results for five out of six algorithms, as shown in Tables 6 and 7.

In order to show the significance of our algorithms, we run the algorithms many times and we obtain small deviation thorough these tests. Algo-III has 0.9% deviation with CC-J48, 0.5% deviation with CC-LR, 0.7% with PS-J48, 0.6% deviation with RAkEl-LR on WMS dataset and has 0.05% deviation with CC-LR, 0.01% deviation with PS-SVM, 0.02% with RAkEl-LR MIR-Flickr Dataset. The deviations are less than 1% in the average.

Additionally, t-tests are performed to verify the significance level of the proposed algorithms. α is selected as 0.05 (5%) for both datasets and the resulting value is obtained as 0.000092 after the experiments. All values are better than 0.05 so the results are decided to be statistically significant.

VI. CONCLUSION

We analyze the performance of multi-label video data classification algorithms through feature selection techniques. The multi-objective evolutionary NSGA-II is used for the feature selection process and autoencoders with regularizations as denoising autoencoder and drop-out regularization are implemented. An under-complete autoencoder is implemented for dimensionality reduction with two different techniques. The first one is denoising autoencoder, which is based on adding in a certain amount of noise to the input image for better learning through the output and the second one is dropout regularization before every hidden layer in encoder part for similar purposes with the denoising autoencoder. The number of layers is determined with a heuristic approach. A dimensionally reduced sub-set of data is extracted after all ten hidden layers. For MIR-Flickr dataset 10^{th} layer has better performance than others. However, for WMS dataset 5^{th} layer gives better results. Other parameters such as optimizer, activation function, learning rate are also set.

The reduced dimensional feature sets that are extracted with both dropout regularized autoencoders and NSGA-II are ensembled. The ensemble of these combined feature-sets is evaluated. Our proposed method which is based on ensemble feature selection using deep autoencoder and NSGA-II with two-step optimization is performed for the first time in the literature to the best of our knowledge. Additionally, for ensemble feature selection, deep autoencoders are not used before. When the results are discussed, our proposed method has competitive results compared to state-of-the-art algorithms and feature selection algorithms without applying ensembling. Our proposed method provides better results on the WMS dataset that has repetitive structures and a limited number of labels. The Hamming score is increased while the number of features is reduced during the multi-objective optimization. The algorithms succeeded in obtaining the sets of optimal Pareto solutions.

Algorithms CBFS, IG, and PCA can provide good results but NSGA-II has the best results with the longest execution times. Pareto optimal solutions of autoencoders have almost the same results with NSGA-II. But with less execution time and fewer number of features in the average. The proposed method provides (near)-optimal solutions. The exact solution (finding the optimal subset of features) is NP-Hard and its algorithm is not reasonable.

The proposed algorithms can be used for the classification of images, social media resources, videos, patients, texts, and audio files. In the future, new algorithms can be executed on more powerful parallel computing machines. Increasing the number of generations and exploring with diverse populations can yield better results. Other multi-label classification problems with diverse image feature descriptors can be used. Different types of autoencoders and multi-objective feature selection algorithms can also be developed with other possible ensemble feature selection techniques.

REFERENCES

[1] A. K. McCallum, "Multi-label text classification with a mixture model trained by EM," in *Proc. AAAI Workshop Text Learn.*, 1999, pp. 1–7.

[2] Z. Barutcuoglu, R. E. Schapire, and O. G. Troyanskaya, "Hierarchical multi-label prediction of gene function," *Bioinformatics*, vol. 22, no. 7, pp. 830–836, Apr. 2006.

[3] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, T. Mei, and H.-J. Zhang, "Correlative multi-label video annotation," in *Proc. 15th Int. Conf. Multimedia (MULTIMEDIA)*, 2007, pp. 17–26.

[4] G.-Z. Li, Z. He, F.-F. Shao, A.-H. Ou, and X.-Z. Lin, "Patient classification of hypertension in traditional Chinese medicine using multi-label learning techniques," *BMC Med. Genomics*, vol. 8, no. 3, p. 4, Dec. 2015.

[5] I. Katakis, G. Tsoumakas, and I. Vlahavas, "Multilabel text classification for automated tag suggestion," in *Proc. ECML/PKDD*, vol. 18, 2008, pp. 1–5.

[6] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. P. Vlahavas, "Multi-label classification of music into emotions," in *Proc. ISMIR*, vol. 8, 2008, pp. 325–330.

[7] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognit.*, vol. 37, no. 9, pp. 1757–1771, Sep. 2004.

[8] J. Miao and L. Niu, "A survey on feature selection," *Procedia Comput. Sci.*, vol. 91, pp. 919–926, Jan. 2016.

[9] M. S. Srivastava, M. S. Joshi, and G. Madhvi, "A review paper on feature selection methodologies and their applications," *IJCSNS*, vol. 14, no. 5, p. 78, 2014.

[10] F. R. Bach, "Bolasso: Model consistent lasso estimation through the bootstrap," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 33–40.

[11] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[12] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Jan. 2003.

[13] J. Yin, T. Tao, and J. Xu, "A multi-label feature selection algorithm based on multi-objective optimization," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2015, pp. 1–7.

[14] Y. Zhang, D.-W. Gong, X.-Y. Sun, and Y.-N. Guo, "A PSO-based multi-objective multi-label feature selection method in classification," *Sci. Rep.*, vol. 7, no. 1, pp. 1–12, Dec. 2017.

[15] R. Vaishali, R. Sasikala, S. Ramasubbareddy, S. Remya, and S. Nalluri, "Genetic algorithm based feature selection and MOE fuzzy classification algorithm on Pima Indians diabetes dataset," in *Proc. Int. Conf. Comput. Netw. Informat. (ICCNI)*, Oct. 2017, pp. 1–5.

[16] L. D. Vignolo, D. H. Milone, and J. Scharcanski, "Feature selection for face recognition based on multi-objective evolutionary wrappers," *Expert Syst. Appl.*, vol. 40, no. 13, pp. 5077–5084, Oct. 2013.

[17] M. Labani, P. Moradi, M. Jalili, and X. Yu, "An evolutionary based multi-objective filter approach for feature selection," in *Proc. World Congr. Comput. Commun. Technol. (WCCCT)*, Feb. 2017, pp. 151–154.

[18] P. Zhang, W. Gao, and G. Liu, "Feature selection considering weighted relevancy," *Int. J. Speech Technol.*, vol. 48, no. 12, pp. 4615–4625, Dec. 2018.

[19] A. Deniz, H. E. Kiziloz, T. Dokeroglu, and A. Cosar, "Robust multiobjective evolutionary feature subset selection algorithm for binary classification using machine learning techniques," *Neurocomputing*, vol. 241, pp. 128–146, Jun. 2017.

[20] T. M. Hamdani, J.-M. Won, A. M. Alimi, and F. Karray, "Multi-objective feature selection with NSGA II," in *Proc. Int. Conf. Adapt. Natural Comput. Algorithms*. Berlin, Germany: Springer, 2007, pp. 240–247.

[21] M. A. Khan, A. Ekbal, E. L. Mencía, and J. Fürnkranz, "Multi-objective optimisation-based feature selection for multi-label classification," in *Proc. Int. Conf. Appl. Natural Lang. Inf. Syst.* Cham, Switzerland: Springer, 2017, pp. 38–41.

[22] S. Li, H. Wu, D. Wan, and J. Zhu, "An effective feature selection method for hyperspectral image classification based on genetic algorithm and support vector machine," *Knowl.-Based Syst.*, vol. 24, no. 1, pp. 40–48, Feb. 2011.

[23] A. Gaspar-Cunha, "Feature selection using multi-objective evolutionary algorithms: Application to cardiac SPECT diagnosis," in *Advances in Bioinformatics*. Berlin, Germany: Springer, 2010, pp. 85–92.

[24] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1656–1671, Dec. 2013.

[25] Y. Zhang, D.-W. Gong, and J. Cheng, "Multi-objective particle swarm optimization approach for cost-based feature selection in classification," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 14, no. 1, pp. 64–75, Jan. 2017.

[26] H. E. Kiziloz, A. Deniz, T. Dokeroglu, and A. Cosar, "Novel multiobjective TLBO algorithms for the feature subset selection problem," *Neurocomputing*, vol. 306, pp. 94–107, Sep. 2018.

[27] M. Paniri, M. B. Dowlatshahi, and H. Nezamabadi-pour, "MLACO: A multi-label feature selection algorithm based on ant colony optimization," *Knowl.-Based Syst.*, vol. 192, Mar. 2020, Art. no. 105285.

[28] A. Asilian Bidgoli, H. Ebrahimpour-Komleh, and S. Rahnamayan, "An evolutionary decomposition-based multi-objective feature selection for multi-label classification," *PeerJ Comput. Sci.*, vol. 6, p. e261, Mar. 2020.

[29] E. Hancer, B. Xue, M. Zhang, D. Karaboga, and B. Akay, "Pareto front feature selection based on artificial bee colony optimization," *Inf. Sci.*, vol. 422, pp. 462–479, Jan. 2018.

[30] T. Dokeroglu, E. Sevinc, T. Kucukyilmaz, and A. Cosar, "A survey on new generation Metaheuristic algorithms," *Comput. Ind. Eng.*, vol. 137, Nov. 2019, Art. no. 106040.

[31] V. Bolón-Canedo and A. Alonso-Betanzos, "Ensembles for feature selection: A review and future trends," *Inf. Fusion*, vol. 52, pp. 1–12, Dec. 2019.

[32] C.-W. Chen, Y.-H. Tsai, F.-R. Chang, and W.-C. Lin, "Ensemble feature selection in medical datasets: Combining filter, wrapper, and embedded feature selection results," *Expert Syst.*, Apr. 2020, Art. no. e12553. [Online]. Available: https://doi.org/10.1111/exsy.12553

[33] D. S. Guru, M. Suhil, S. K. Pavithra, and G. R. Priya, "Ensemble of feature selection methods for text classification: An analytical study," in *Proc. Int. Conf. Intell. Syst. Design Appl.* Cham, Switzerland: Springer, 2017, pp. 337–348.

[34] B. Seijo-Pardo, V. Bolón-Canedo, and A. Alonso-Betanzos, "Testing different ensemble configurations for feature selection," *Neural Process. Lett.*, vol. 46, no. 3, pp. 857–880, Dec. 2017.

[35] S. Petscharnig, M. Lux, and S. Chatzichristofis, "Dimensionality reduction for image features using deep learning and autoencoders," in *Proc. 15th Int. Workshop Content-Based Multimedia Indexing (CBMI)*, 2017, pp. 1–6.

[36] J. Gonzalez-Lopez, S. Ventura, and A. Cano, "Distributed multi-label feature selection using individual mutual information measures," *Knowl.-Based Syst.*, vol. 188, Jan. 2020, Art. no. 105052.

[37] J. Gonzalez-Lopez, S. Ventura, and A. Cano, "Distributed selection of continuous features in multilabel classification using mutual information," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 7, pp. 2280–2293, Jul. 2020.

[38] P. Skryjomski, B. Krawczyk, and A. Cano, "Speeding up k-nearest neighbors classifier for large-scale multi-label learning on GPUs," *Neurocomputing*, vol. 354, pp. 10–19, Aug. 2019.

[39] Y. Wang, H. Yao, and S. Zhao, "Auto-encoder based dimensionality reduction," *Neurocomputing*, vol. 184, pp. 232–242, Apr. 2016.
[40] K. Han, Y. Wang, C. Zhang, C. Li, and C. Xu, "Autoencoder inspired unsupervised feature selection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 2941–2945.
[41] S. Wang, Z. Ding, and F. Yun, "Feature selection guided auto-encoder," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1–7.
[42] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[43] R. Salakhutdinov and G. Hinton, "Semantic hashing," *Int. J. Approx. Reasoning*, vol. 50, no. 7, pp. 969–978, Jul. 2009.
[44] M. Antonelli Ponti, L. Sampaio Ferraz Ribeiro, T. Santana Nazare, T. Bui, and J. Collomosse, "Everything you wanted to know about deep learning for computer vision but were afraid to ask," in *Proc. 30th SIBGRAPI Conf. Graph., Patterns Images Tuts. (SIBGRAPI-T)*, Oct. 2017, pp. 17–41.
[45] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[46] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 1096–1103.
[47] F. Charte, M. J. del Jesus, and J. R. Antonio, *Multilabel Classification: Problem Analysis, Metrics and Techniques*. Cham, Switzerland: Springer, 2016.
[48] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Mach. Learn.*, vol. 85, no. 3, p. 333, 2011.
[49] J. Read, B. Pfahringer, and G. Holmes, "Multi-label classification using ensembles of pruned sets," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 995–1000.
[50] T. Dokeroglu and E. Sevinc, "Evolutionary parallel extreme learning machines for the data classification problem," *Comput. Ind. Eng.*, vol. 130, pp. 237–249, Apr. 2019.
[51] E. Cantú-Paz, "A survey of parallel genetic algorithms," *Calculateurs Paralleles, Reseaux Syst. Repartis*, vol. 10, no. 2, pp. 141–171, 1998.
[52] D. Hadka. (2020). Documentation for the MOEA Framework. Moeaframework. Accessed: Apr. 22, 2020. [Online]. Available: http://moeaframework.org/documentation.html
[53] J. Read, P. Reutemann, B. Pfahringer, and G. Holmes, "Meka: A multi-label/multi-target extension to Weka," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 667–671, Jan. 2016.
[54] M. J. Huiskes and M. S. Lew, "The MIR flickr retrieval evaluation," in *Proc. 1st ACM Int. Conf. Multimedia Inf. Retr. (MIR)*, 2008, pp. 39–43.
[55] A. F. Costa, A. J. M. Traina, and C. Traina, "MFS-map: Efficient context and content combination to annotate images," in *Proc. 29th Annu. ACM Symp. Appl. Comput. (SAC)*, 2014, pp. 945–950.
[56] A. F. Costa, G. Humpire-Mamani, and A. J. M. Traina, "An efficient algorithm for fractal analysis of textures," in *Proc. 25th SIBGRAPI Conf. Graph., Patterns Images*, Aug. 2012, pp. 39–46.
[57] B. Gary and A. Kaehler, *Learning OpenCV: Computer Vision With the OpenCV Library*. Newton, MA, USA: O'Reilly Media, Inc., 2008.
[58] K. P. Diederik and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980
[59] K. Pearson, "On lines and planes of closest fit to systems of points in space," *London, Edinburgh, Dublin Philos. Mag. J. Sci.*, vol. 2, pp. 559–572, Jun. 2010, doi: 10.1080/14786440109462720.
[60] I. T. Jolliffe, "Principal component analysis," in *Encyclopedia of Statistics in Behavioral Science*, B. Everitt and D. Howell, Eds. New York, NY, USA: Wiley, 2005.