Received: September 21, 2022. Revised: November 19, 2022.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Methods for Predicting the Rise of the New Labels from a High-Dimensional Data Stream

Komagal Yallini Shanmuga Sundaram Kanagaraj[1]* **Mahendiran Nallappan[1]**

[1]*Department of Computer Science, Sri Rama Krishna College of Arts and Science, Coimbatore, Tamil Nadu, India*
*Corresponding author's Email: komagayallini123@gmail.com

Abstract: In data engineering, multi-label learning (MLL) has emerged to classify the instances through a specific characteristic that associates with the set of class labels (CLs). Mostly, the learning design was adaptive and newer views may exist in a data stream (DS); so, MLL has to classify the features with newer CLs. To tackle this issue, the MLL with emerging new labels (MuENL) and handling high-dimensional DSs (MuENLHD) technique has been adopted which considers the CLs in the test data were similar to that in the learning data. But, it was not able to deal with the adaptive situation where multiple newer CLs exist since it can manage only a single newer CL in one iteration. Hence this article proposes an MLL with emerging multiple new labels (MuEMNL) and MuEMNLHD versions to combat the issues in a complex scenario wherein several newer CLs are found. The main idea of this technique is to split the newer CL group into many newer CLs independently for changing the complex scenario. In this technique, four different steps are executed: i) creates a linear classification model to adjust the pairwise CL sorting error and the categorization error on the given CLs, ii) develops a novel outlier identifier depending on the primary and test DS, iii) discovers the cluster for the MuEMNL and MuEMNLHD depending on the OPTICS clustering and iv) employs a classifier updating scheme to integrate newer CLs for designing a robust classifier. Finally, the experimental outcomes exhibit that these techniques on low-dimensional databases attain an overall mean precision of 67.08% and an overall F1-score of 64.02% compared to the classical MLL techniques. Similarly, these techniques on high-dimensional databases achieve mean precision of 64.6% and a mean F1-score of 63.1% compared to the classical MLL techniques.

Keywords: Multi-label learning, MuENL, MuENLHD, OPTICS clustering, Pairwise label ranking, Linear classifier, Outlier identifier.

1. Introduction

In machine learning, each element is referred to as a unique feature vector associated with a particular CL. Each aspect could contain several CLs in multiple applications—for example, a photograph could have many subjects and a musical track could fall in several parts of the human brain [1-3]. In MLL, all items in standard supervised learning are marked by a specific characteristic if associated with a set of CLs [4]. The approach maintains the relevant CL groups for undefined characteristics. MLL has been widely employed in multiple concerns including computerized efficacy in audiovisual data [5]. Over the past decades, ML text categorization was assessed in numerous practical systems that could include newer CLs with possibly the best CLs in an expected DS structure.

A learning methodology can reallocate and change a pre-trained structure to a different configuration in a complex world [6]. This methodology must be able to recreate a pre-trained structure in the ML design to newer features and new classifying systems for all newer CLs that were produced. There has been no ground truth (GT) for CLs in the adaptive MLL setup at every moment in the DS except for real learning data. Therefore, the main problems were to find and simulate newer CLs. In general, it was highly difficult to recognize the features of having a newer CL. Since newer CLs do not really exist in the prior data and mostly co-existed

with a few tremendous CLs, splitting functionality from those with well-known CLs with newer CLs was exceedingly hard. The failure was increased with an increase in newer CLs in a DS owing to an unsuitable classification. Thus, it was a challenging task to create suitable structures for increasing the classification ability in a DS. To address such a challenge, several MLL strategies have been reported with interesting methods to detect the associations between tagged and untagged features.

From these perspectives, MuENwas presented [7] to identify and categorize the features with ENLs. This MuENL technique has different major phases: 1) categorizing the features of recently identified CLs, 2) detecting the existence of a newer CL, and 3) creating a new classification framework for all newer CLs which work collaboratively with the classifier for the identified CLs. Additionally, this technique has been updated to MuENLHD to cope with the sparse high-dimensional DSs by minimizing the dimensionality through a streaming kernel principal component analysis (PCA). Conversely, this technique was able to handle only a single newer CL in a particular step. Alternatively, if the test collection has many newer CLs in one step, then this technique will consider the multiple newer CLs as a single newer CL. This results in efficiency degradation.

Therefore, in this paper, the MuEMNL and MuEMNLHD techniques are proposed to solve the challenges in an adaptive scenario where many newer CLs exist. In this technique, the newer CL group is split into multiple newer CLs independently for modifying the adaptive scenario. This technique executes four different phases: i) a linear classification model is created for optimizing the pairwise CL sorting error and the categorization error on the given CLs, ii) a novel outlier identifier is created depending on both primary and test DS, iii) the cluster for the MuEMNL and MuEMNLHD are discovered depending on the OPTICS algorithm and iv) a classifier updating scheme is used to combine newer CLs for designing a robust classifier. Thus, these techniques can lessen the classification loss for the future DS which has similar newer CL efficiently.

The remaining portions of this paper are structured as: section 2 discusses the works associated with the MLL for different purposes. Section 3 describes the MuEMNL and MuEMNLHD techniques. Section 4 shows their experimental findings. Section 5 concludes the entire study and provides the potential improvement.

2. Literature survey

A multi-class extreme learning machine (ELM) was developed [8] by label properties (LLP) and applied to categorize the depositors for improving client relationships. But, the accuracy was not efficient. A View-label-specific feature (VLSF) was developed [9] for multi-view MLL. However, it was time-consuming since its computation burden was high for large-scale datasets.

A novel technique was designed [10] to train label-specific characteristics for ML categorization (LSML) with inadequate tags. However, its efficiency was poor if the range of weighting parameters was very high. An incremental kernel (IK) ELM was suggested [11] for MLL with the use of new CLs. This model comprises a novelty identifier and an ML classifier. A novel incremental ML classification (i.e., MENL-IK) and its improved version (i.e., MENL-iIK) were designed for predicting a CL set for every instance, which includes output units incrementally and modify themselves in untagged instances. But, it has a high computational difficulty since the kernel size within this model structure was high. Also, the efficiency was sensitive to the buffer size.

A new ML feature selection (MLFS) model was developed [12] for learning and exploiting local CL correlations. But, the accuracy of this model was very less. A discriminant MLL (DMLL) technique was developed [13] which imposes low-order patterns on each prediction of data belonging to similar CLs and highly independent patterns on the estimations of data belonging to various CLs. But, it needs to find a regional low-order CL pattern in ELM where the number of CLs was very huge.

A novel model was designed [14] called MLL based on feature and label space dimension reduction (MLL-FLSDR). However, the linear regression framework was sensitive to outliers. A multi-label learning technique was designed [15] called LF-LELC which considers the significance of CL vectors and builds the classification framework by the CL correlations. But, it does not handle the class-imbalance problem which may affect learning efficiency.

3. Proposed methodology

In this section, the design of a robust classifier is described for MLs having newer and previous CLs. Additionally, OPTICS-based clustering is integrated for solving the issues while multiple newer CLs are appearing simultaneously in one iteration. Fig. 1 shows the schematic representation of MuEMNL and MuEMNLHD techniques.

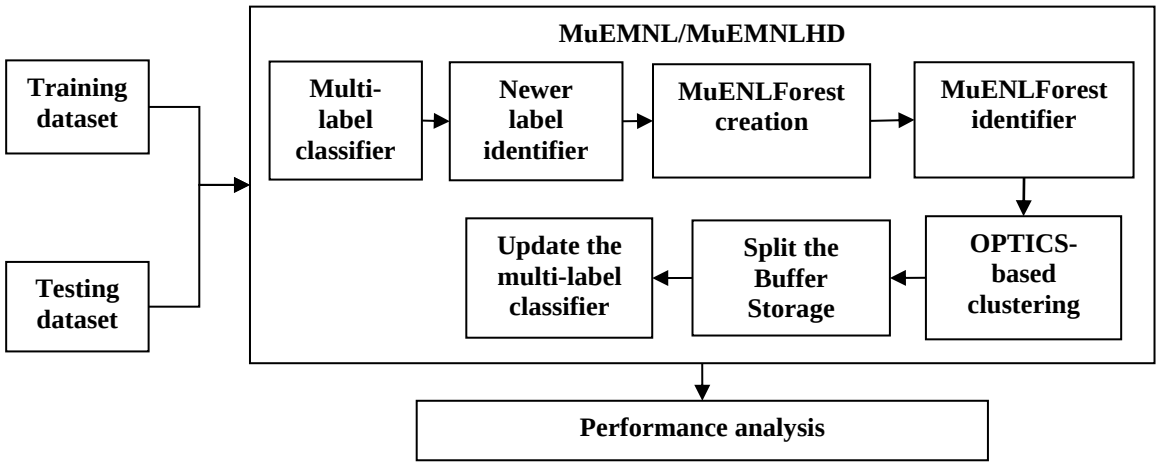

Figure. 1 Schematic overview of proposed multi-label learning technique

Table 1. Lists of Notations

Notations	Description
$\Re$	Input feature space
$\mathcal{R}_0$	Observed primary dataset in the learning task
r_t	Instances that contain a new CL at a period t
$\mathcal{R}_t$	Available data trunk at t
c_0	Previously known CLs in the primary training dataset
c	Index of the CLs collection
l	Maximum number of CLs collection
n	Number of clusters
Y_0	Primary CL matrix of $\mathcal{R}_0$
y_t	CL vector of the test instance r_t
$\mathcal{D}_t(r_t)$	Outlier identifier
$\mathcal{H}_t$	Robust classifier
$h_i(r)$	Linear classifier on CL i for r
ω_i	Weight value of linear classifier on CL i for r
b_i	Bias value of linear classifier on CL i for r
C_1, C_2	Variables
ψ	Dimension of random subset
g	Number of MuENLTree
s_t	Sampling weight
BS_{Max}	Maximum buffer storage
$MinIns$	Input variable
ε	Distance
S_z	Partition index for z^{th} cluster
$RP(r_x), RP(r_y)$	Ranges of the accessibility curve
$R_{B,i}$	Set of instances having i^{th} newer CLs
R_U	Set of instances having expected CLs
p	Unknown allocation of the newer CL of $R_{t,i}$
m	Number of instances in $[R_{B,i}; R_U]$

Table 1 lists the notations used in this study.

3.1 Problem definition

This technique is depending on an open adaptive multi-learning configuration, it contains a primary tagged training dataset, and then the untagged test instance is obtained sequentially in a DS manner. Consider $\Re$ is the input feature space, and $\mathcal{R}_0 = [r_{-n+1}, \dots, r_{-1}, r_0]^{\mathsf{T}} \subseteq \Re$ is denoted as the observed primary dataset in the learning task. After, the untagged test DS comprises an example r_t that possibly contains a newer CL at a period t. Consider $\mathcal{R}_t, t \in \{1,2, \dots, T\}$ is the available data trunk at t.

Also, $c_0 = \{1,2, \dots, l\}$ is denoted as the previously known CLs in the primary training dataset and c is the index of the CLs collection. During the period t, the actual c_0 can be upgraded to c_t and primary l can be upgraded to l', l' is the highest amount of CLs collection. Specifically, during t, assuming there are $n(n \geq 0)$ identified newer CLs, the CL set is augmented with n newer CLs, $l' = l + n; c_t = c_{t-1} \cup \{l'\}$. Consider $Y_0 = [y_{-n+1}, \dots, y_{-1}, y_0] \in \{-1,1\}^{l \times n}$ is the primary CL matrix of R_0 and $y_t = [y_{t,1}, \dots, y_{t,l}]$ is the CL vector of the test example r_t at t. The CL vector $y_{t,j}$ contains 2 opposite ranges:$\{-1,1\}$. If r_t has j^{th} CL, then $y_{t,j}$ must be 1; or else, $y_{t,j}$ must be -1.

3.2 Algorithm

Consider that the GT is not accessible in the whole test DS. The challenges required to be resolved in newer MLL majorly occur in 2 phases: (i) the primary phase is to build an identifier for detecting the newer CLs and (ii) the auxiliary phase is to obtain several newer CLs separately when they eject in a single step, i.e., while it may essential to adjust the

Algorithm 1: MuEMNL

Input: Primary training data: R_0, Y_0, C_0
Output: Function set $\mathcal{H}_t$ for every r_t
Begin
Obtain a primary $\mathcal{H}_0$ by learning R_0, Y_0;
Build a primary newer CL identifier $\mathcal{D}_0$ depending on R_0;
Initialize sampling eight vector $s_0 = 1_{|r_0|}$;
$\mathcal{H}_1 = [\mathcal{H}_0, \mathcal{D}_0]$; $\mathcal{D}_1 = \mathcal{D}_0$;
Repeat
Get a newer instance $r_t, R_t = [R_{t-1}; r_t^\top]$;
Expand the sampling weight vector $s_t = [s_{t-1}; 1]$ concurrently;
$\boldsymbol{if}(\mathcal{D}_t(r_t) \geq 1)$
Include r_t to buffer;
$\boldsymbol{if}(|Buffer| \geq BS_{Max})$
Apply *Algorithm 2*;
Perform OPTICS to split buffer storage;
Obtain n clusters for n newer CLs;
$\boldsymbol{while}(i > n)$
Create $\mathcal{D}_{t+i}$ and $\mathcal{H}_{t+i}$ from $i = 0$ and every $\mathcal{D}_{t+i}$ relies on $\mathcal{D}_{t+i-1}$ iteratively;
$\boldsymbol{end\ while}$
Empty buffer;
$l \leftarrow l + n$; $c_t = c_{t-1} \cup \{l\}$;
Update $s_t \leftarrow 0.8 s_t$;
$\boldsymbol{end\ if}$
$\boldsymbol{end\ if}$
$c_t = c_{t-n}$; $\mathcal{D}_t = \mathcal{D}_{t-n}$; $\mathcal{H}_t = \mathcal{H}_{t-n}$;
Until
Obtain $\mathcal{H}_t$;
End

framework for several CLs simultaneously. Or else, a high opportunity exists that it may erroneously learn a single weight vector for several newer CLs.

These challenges are solved by utilizing the 2 major techniques: 1). First, the MuENL/MuENLHD is extended, which will create an outlier identifier $\mathcal{D}_t(r_t)$ depending on the prior learning instances and CL characteristics. When the result of $\mathcal{D}_t(r_t) = 1$, then it denotes that the current r_t has a newer CL. Or else, if the output is -1, then it denotes that there is no newer CL in r_t. 2). Then, the density-based spatial clustering called OPTICS scheme is adapted to the buffer storage while it achieves the specific limit. Specifically, for the instance with newer CLs in the storage, such instances are clustered into many clusters and every cluster transforms into specific newer CL. Moreover, while this process is completed, the classifier framework updating task is performed stage-by-stage, and every stage is executed based on the prior stage for constructing a more robust classifier: $\mathcal{H}_t = [h_{t,1}, \dots, h_{t,l}] \rightarrow \mathcal{H}_t' = [h_{t,1}, \dots, h_{t,l}, \mathcal{D}_t]$.

Algorithm 1 describes the MuEMNL technique. It comprises 4 units: the multi-label classifier for $\mathcal{H}_0$, the outlier identifier $\mathcal{D}_t$, OPTICS splitting several newer CLs residing in the buffer storage into n newer CLs, and updating the framework $\mathcal{H}_t \rightarrow \mathcal{H}_{t+n}$. The point that necessitates state is that the weighted sampling vector is utilized for minimizing the chance of prior instances being chosen during the creation of the extended MuENL technique and offers a preference for newer instances. So, consider s is multiplied by a decay value of 0.8.

3.2.1. Linear multi-label classifier

For an instance r, the linear classifier is defined on CL i as Eq. (1):

$$h_i(r) = sign\left(\omega_i^\top r + b_i\right) \tag{1}$$

When the misclassification error and the pairwise CL ranking error are reduced for obtaining the total efficiency, the convex optimization for every ω_i is written as:

$$\min_{\omega_i b_i, \xi, \zeta} \frac{1}{2}\|\omega_i\|^2 + C_1 \sum_{k=1}^{n} \xi_k + C_2 \sum_{j=1}^{l} \sum_{k=1}^{n} \zeta_{j,k} \tag{2}$$

Subject to $y_{i,k} f_{i,k} \geq 1 - \xi_k$
$\Delta_{j,k}\left(f_{i,k} - f_{j,k}\right) \geq 1 - \zeta_{j,k}$
$\xi_k \geq 0, \zeta_{j,k} \geq 0$
$j \in \{1, \dots, l\}, k \in \{1, \dots, n\}$

In Eq. (2), $\Delta_{j,k} = y_{i,k} - y_{j,k}, f_{i,k} = \omega_i^\top r_k + b_i$, and C_1, C_2 define 2 variables to balance. To make it simpler, the computation task b_i is replaced by inserting a quality range 1 in the last part of r_k, then $f_{i,k} = \omega_i^\top [r_k : 1]$. So, Eq. (2) is rewritten as Eq. (3):

$$\min_{\omega_i} \sum_{j=1}^{l} \sum_{k=1}^{n} \left[1 - \left(y_{i,k} - y_{j,k}\right)\left(f_{i,k} - f_{j,k}\right)\right]_+ + \lambda_1 \sum_{k=1}^{n} \left[1 - y_{i,k} f_{i,k}\right]_+ + \frac{\lambda_2}{2}\|\omega_i\|^2 \tag{3}$$

3.2.2. Newer label identification

Assume that a data set contains a CL or attribute that is a newer version of one that was previously there. So, the MuENLForest [7] is adopted as the newer CL identifier which finds the attributes in the attribute space and the CLs. It has g MuENLTree; and every MuENLTree is constructed by the random subset of $\left(\mathcal{R}_t, \mathcal{H}_t(\mathcal{R}_t)\right)$ of dimension ψ using a

sampling weight s_t . Also, a new unit of the MuENLForest is an enclosing sphere at every leaf node. So, examples with identical characteristics will be situated on the identical leaf node. Once the MuENLForest is built i.e., $\mathcal{D}_t(\cdot)$, the newer CLs are predicted. If $\mathcal{D}_t(r_t) = 1$, then it means r_t has a newer CL. Or else, if $\mathcal{D}_t(r_t) = -1$, then it doesn't contain a newer CL. Particularly, when the example occurs on a similar leaf node yet outer of the enclosing sphere, it recommends that the example contain few characteristics. So, the example maintains a newer CL with a significantly higher chance.

3.2.3. OPTICS for splitting buffer storage

Each period, while the newer CLs buffer storage attains maximum, i.e., BS_{Max}, the standard schemes consider each instance in the buffer contains a frequent newer CL. In the real-time scenario, it may satisfy the condition that the examples in a single buffer can have several CLs in a single step. For case, buffer storage has BS_{Max} examples applying with $n(n > 1)$ newer CLs. These natures of incorrectly learning one framework for multiple newer CLs are solved by introducing the OPTICS algorithm.

Typically, the OPTICS splits the buffer storage into n essential groups that may enhance the classification robustness for several newer CLs. The below section presents the information that how the OPTICS operations are combined with the splitting task:

The OPTICS is an extended version of the DBSCAN algorithm which utilizes a value of ε instead of a particular total threshold for detecting the groups of various regional densities i.e., for defining the closeness of sample distribution.

It utilizes the input variables $MinIns$ and created distance ε; but, ε is the highest threshold range. Also, ε' is used for defining the radius utilized by OPTICS, where $0 < \varepsilon' < \varepsilon$. The basic terms used in this OPTICS clustering algorithm are defined as follows:

- Major gap: The major gap of an instance r_t defines the gap ε' between r_t and its $MinIns$'s adjacent such that r_t refers to the major instance relating to ε'. When ε' range is superior to ε, the major gap is not defined.
- Accessibility gap (of r_x from r_y): When r_y is not a major data relating to ε , then the accessibility gap is not specified. Or else, the accessibility gap of a data r_x from a data r_y is the highest of r_y 's major gap and the minimum gap such that r_x is density-accessible from y.

A) Partition index to assess strength of accessibility curve in OPTICS

To evaluate the cluster formation in OPTICS, an accessibility curve is considered. The accessibility gap is projected for every instance ranked based on the ranked file. Because acquiring the accessibility curve is a transitional process in OPTICS, the hit of this algorithm essentially relies on the accessibility curve. After that, evaluating the curve strength is an evaluation of the result of OPTICS.

By neglecting the initial and final instances of the accessibility curve, the partition index is defined as the fraction of the average peak heights to the average range of instances among the peaks in the valleys. As, if there are peaks at r_x and r_y in the accessibility curve, with the group-c represented by the valley enclosed by such peaks, then the partition index for the group-c is

$$S_z = \frac{\frac{1}{2}\left(RP(r_x)+RP(r_y)\right)}{\frac{1}{r_y-r_x-1}\sum_{m=r_x+1}^{r_y-1} RP(m)} \tag{4}$$

In Eq. (4), S_z is the partition index for z^{th} cluster, and $RP(r_x), RP(r_y)$ are the ranges of the accessibility curve. The last partition index is a mean of the partition indices acquired for every group. It assesses the fraction of the peak heights to the mean of the benchmark heights. A greater partition index denotes that the peaks are comparatively very greater than the benchmark; so, creating groups by the threshold on the accessibility curve may be less responsive to outliers. This is a proximate indicator of how reliable OPTICS clustering is.

B) Evaluating sturdiness by additive noise

When dealing with noisy data, a clustering algorithm's sturdiness is evaluated. If clustering can continue to function well in the existence of greater interference ranges than the noise-free instances for a particular data set, it is said to be resilient. To analyze the sturdiness of OPTICS in correlation with multiple characteristics, a recursive clustering using additive noise (RCAN) scheme is applied. In this paper, white Gaussian noise is sequentially inserted into the multiple characteristics such that the signal-to-noise ratio (SNR) of the succeeding contaminated characteristics is reduced by 1dB per iteration, with the SNR range initiating from 100dB in the initial iteration.

In this case, the standard description of SNR is utilized i.e. $SNR = 10 \times \log\left(\frac{S}{N}\right)$, where S is the

Algorithm 2: OPTICS clustering
Input: New instances r_t **Output:** n clusters **Begin** Include white Gaussian noise to r_t in decrements of 1dB SNR from 100 to 0; Determine core and reachability distances; Compute the partition index using Eq. (4); Acquire n clusters for r_t multiple novel CLs; $\boldsymbol{if}(clustering\ updated)$ SNR value observed at this data instance; $\boldsymbol{else}$ Continue the clustering process; $\boldsymbol{end\ if}$ **End**

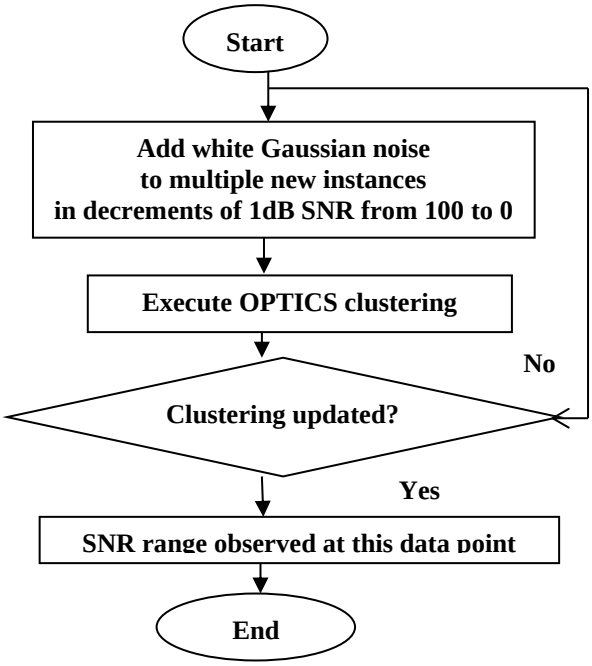

Figure. 2 Flowchart of evaluating robustness using additive noise

signal power (average squared range of each feature value), *N* is the noise power (average squared range of additive Gaussian noise), and the logarithm is considered to the base of 10. An SNR range of 100dB denotes a maximum quantity of signal associated with the outlier, while a range of 0dB denotes a similar quantity of outliers in the instance, creating it hard to differentiate a signal from an outlier.

Initiating with an instance consisting of an SNR of 100dB, OPTICS-based clustering is executed at every SNR range, reducing the SNR by 1dB in all iterations and concluding merely while the grouping creation has been updated. The lesser the SNR range, the greater outlier that exists in the instances, so highly vigorous the scheme could be in the mixture with the multiple input characteristics because the less SNR denotes a greater acceptance for the outlier. The flow of this process is shown in Fig. 2 and Algorithm 2 gives the pseudo code for OPTICS clustering.

In Algorithm 2, r_t new samples with white Gaussian noise to determine core distance and reachability distance values. After that, the partition index is calculated for clusters and n clusters are obtained. When the clustering is updated, the SNR value is observed at the measured distance; or else, the clustering procedure is continued.

Consider that examples having similar newer CL can situate in the identical group. Hence, various groups can indicate various CLs. Since it is shown that amongst the MuENLForest leaf nodes, such examples having newer CLs are estimated using their gap far from the regular examples situated in every group reliantly. Specifically, if the classifier is considered as learned for i^{th} CL, then the successive step is to learn the classifier for $i + 1^{th}$ CL depending on i^{th} CL classifier.

3.2.4. Multi-label classification update

After the OPTICS-based clustering process has finished the splitting task for multiple newer CLs, the multi-CL classification is updated. The trained categorizer is highly resilient to the failures of the identifier. The result is modeled as:

Consider that k groups gathered in the buffer storage and $R_{B,i}$ is the set of examples having i^{th} newer CLs, R_U refers to the collection of examples having expected CLs simply, where $R_U = {R_t}/{R_B}$ and $R_B = [R_{B,1}, \dots, R_{B,k}]$. Assume $p = [p_1, \dots, p_m]^\top$ is the unknown allocation of the newer CL of $R_{t,i} = [R_{B,i}; R_U]$ where m denotes the number of instances in $[R_{B,i}; R_U]$, and $p_k = 1$ if $r_k \in [R_{B,i}; R_U]$; or else, $p_k = 0$.

Different from Eq. (3), $y_{i,k}$ is replaced with $2p_k - 1$. So, the optimization dilemma of constructing a classifier ω_a and training p for the newer CL l is directed as:

$$\omega_a, p \leftarrow \min_{\omega_a, p} \sum_{j=1}^{l} \sum_{k=1}^{m} \left[1 - \left(2p_k - 1 - y_{j,k}\right)\left(f_{l,k} - f_{j,k}\right)\right]_+ + \lambda_1 \sum_{k=1}^{n} \left[1 - (2p_k - 1) f_{l,k}\right]_+ + \frac{\lambda_2}{2} \|\omega_a\|^2 + \frac{\lambda_3}{2} \|p\|^2 \quad (5)$$

$$\text{Subject to } p_k \in \{0,1\}, k = \{1, \dots, m\}$$

Because Eq. (5) is an NP-hard dilemma, the condition from $p_k \in \{0,1\}$ is changed to $p_k \in [0,1]$ to optimize p and ω_a alternatively, i.e. the

Algorithm 3: MuEMNLHD

Input: Primary training data: R_0, Y_0, C_0
Output: Function set $\mathcal{H}_t$ for every r_t
Begin
Create the random feature map φ;
Obtain a primary $\mathcal{H}_0$ by learning R_0, Y_0;
Build a primary newer CL identifier $\mathcal{D}_0$ depending on R_0;
Initialize sampling weight vector $s_0 = 1_{|r_0|}$;
Initialize $Buffer = \varphi$;
$\mathcal{H}_1 = [\mathcal{H}_0, \mathcal{D}_0]$; $\mathcal{D}_1 = \mathcal{D}_0$;
Repeat
Get a newer instance $r_t, R_t = [R_{t-1}; r_t^\top]$;
Expand the sampling weight vector $s_t = [s_{t-1}; 1]$ concurrently;
if $(\mathcal{D}_t(r_t) \geq 1)$
Include r_t to buffer;
if $(|Buffer| \geq BS_{Max})$
Apply Algorithm *1* and perform OPTICS to split buffer storage;
Obtain n clusters for n newer CLs;
while $(i > n)$
Create $\mathcal{D}_{t+i}$ and $\mathcal{H}_{t+i}$ from $i = 0$ and every $\mathcal{D}_{t+i}$ relies on $\mathcal{D}_{t+i-1}$ iteratively;
end while
Empty buffer;
$l \leftarrow l + n$; $c_t = c_{t-1} \cup \{l\}$;
Update $s_t \leftarrow 0.8 s_t$;
end if
end if
$c_t = c_{t-n}$; $\mathcal{D}_t = \mathcal{D}_{t-n}$; $\mathcal{H}_t = \mathcal{H}_{t-n}$;
Until
Obtain $\mathcal{H}_t$;
End

optimization in Eqs. (6) and (7) are carried out.

$$p \leftarrow \min_{p} \sum_{j=1}^{l} \sum_{k=1}^{m} \left[1 - \left(2p_k - 1 - y_{j,k}\right)\left(f_{l,k} - f_{j,k}\right)\right]_+ + \lambda_1 \sum_{k=1}^{n} \left[1 - (2p_k - 1) f_{l,k}\right]_+ + \frac{\lambda_3}{2} \|p\|^2 \quad (6)$$

Subject to $p_k \in \{0,1\}, k = \{1, \ldots, m\}$

Then, the Eqns. (5) and (6) are solved by using the sub-gradient of the objective function. After, p is projected to $[0,1]$: $p \leftarrow \min(1, [p]_+)$ to assure the box condition in Eq. (7):

$$\omega_a \leftarrow \min_{\omega_a} \sum_{j=1}^{l} \sum_{k=1}^{m} \left[1 - \left(2p_k - 1 - y_{j,k}\right)\left(f_{l,k} - f_{j,k}\right)\right]_+ + \lambda_1 \sum_{k=1}^{n} \left[1 - (2p_k - 1) f_{l,k}\right]_+ + \frac{\lambda_2}{2} \|\omega_a\|^2 \quad (7)$$

Table 2. Statistics of various databases

Database	No. of instances	Size	Total No. of CLs	Mean No. of CLs for every instance
Birds	645	260	19	1014
CAL500	502	68	174	26044
Emotions	593	72	6	3378
Enron	1702	1001	53	3378
Yeast	2417	103	14	4237
20Newsgroup	19300	1006	20	1029

Table 3. Parameters utilized in MuEMNL/MuEMNLHD technique

Parameter	Technique	Definition
$\|Buffer\| = 127$		BS_{Max}
$\lambda_1, \lambda_2 \in \{0.001, 0.01, 0.1, 1\}$	Multi-label categorizer	Trade-off variable
$\|q\| \in \{1,3,5\}$ $g = 100$ $\psi = 256$	MuENLForest	Novel CL identification Forest creation
$MinIns = 5$ $\varepsilon = 1.325$	OPTICS	2 fixed parameters
$\lambda_3 = 1$	Multi-label classifier update	Trade-off parameter

Similarly, the MuEMNL technique is extended to high-dimensional data as MuEMNLHD. The pseudocode of this technique is given in Algorithm 3, which also involves similar processes as in Algorithm 1, but for high-dimensional data.

4. Results and discussion

The efficiency of MuEMNL and MuEMNLHD techniques is assessed by implementing them in MATLAB 2019b. In this experiment, 5 multi-label standard databases [16] such as birds, CAL500, emotions, Enron, and yeast are considered for evaluating the prediction efficiency of the MuEMNL technique. Also, the 20Newsgroup dataset [16] is used for evaluating the prediction efficiency of the MuEMNLHD technique. Table 2 presents the statistics of these databases.

For analysing both MuEMNL and MuEMNLHD, different metrics are considered such as average precision, F1-score, micro F1, . For a test set (r_n, Y_n), $h(r_n)$ defines the group of expected CLs for n^{th} example, $f(r_n, y)$ is the belief value that r_n fits the CL y.

- **Mean precision:** It defines the mean ratio of positive CLs sorted greater than the certain positive CL in Eq. (8):

$$Average\ precision = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|Y_i|}\sum_{y\in Y}\frac{|l_p|}{sort_{f(r_n,y)}} \quad (8)$$

Here, Y_i is the set of positive CLs, n is the number of test instances, l_p is the set of positive predicted CLs, which are sorted lower than CL y for r_n.

- **F1-score:** It refers to the harmonic average of precision and recall for every instance in Eq. (9):

$$F1-score = \frac{1}{n}\sum_{i=1}^{n}\frac{2\times Precision_i\times Recall_i}{Precision_i+Recall_i} \quad (9)$$

- **Micro-F1:** It is computed as Eq. (10):

$$Micro\ F1 = \sum_{i=1}^{n}\frac{2\times Precision_i\times Recall_i}{Precision_i+Recall_i} \quad (10)$$

- **Accuracy:** It measures the algorithm's ability to accurately predict CL of unknown instances. It is calculated as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (11)$$

In Eq. (11), TP is the True Positive, TN is the True Negative, FP is the False Positive, and FN is the False Negative.

- **Hamming loss:** It measures the ratio of instance-CL pairs, which have been misclassified, i.e. a relevant CL is missed or an irrelevant CL is predicted.
- **One-error:** It measures the ratio of instances whose top-ranked predicted CL is not in the ground-truth relevant CL set.
- **Coverage:** It measures how many steps are required, on average, to move down the ranked CL set of an instance thus covering all its relevant CLs. It is normalized by the number of possible CLs.
- **Ranking loss:** It measures the mean ratio of misordered CL pairs, i.e. an irrelevant CL of an instance is ranked higher than its relevant one.

4.1 Analysis of MuEMNL performance

The efficiency of MuEMNL is compared with the LLP-ELM [8], VLSF [9], MENL-IK [11], and LF-LELC [15] using 5 different low-dimensional datasets.

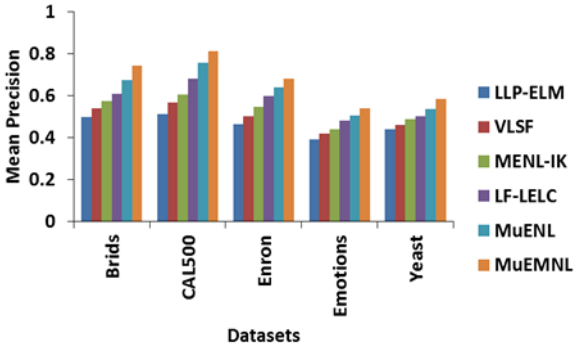

Figure. 3 Comparison of mean precision for low-dimensional dataset

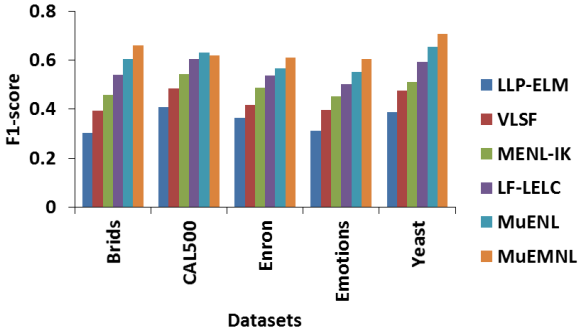

Figure. 4 Comparison of F1-score for low-dimensional datasets

Fig. 3 portrays the mean precision results for different MLL techniques executed on low-dimensional datasets. It observes that the MuEMNL technique realizes better mean precision compared to all other existing MLL techniques. For example, if the CAL500 dataset is considered, then the mean precision of MuEMNL is 59.02% higher than the LLP-ELM, 43.54% higher than the VLSF, 34.49% higher than the MENL-IK, 18.91% higher than the LF-LELC and 7.28% higher than the MuENL techniques.

Fig. 4 shows the F1-score results for different MLL techniques executed on low-dimensional datasets. It observes that the MuEMNL technique realizes a better F1-score compared to all other existing MLL techniques. For example, if the yeast dataset is considered, then the F1-score of MuEMNL is 81.96% greater than the LLP-ELM, 48.32% greater than the VLSF, 38.16% greater than the MENL-IK, 19.06% greater than the LF-LELC and 7.95% greater than the MuENL techniques.

Fig. 5 shows the **Δ**Micro F1 results for different MLL techniques executed on low-dimensional datasets. It observes that the MuEMNL technique

Table 4. Performance of on low-dimensional datasets

Technique	Birds	CAL500	Enron	Emotions	Yeast
	Hamming loss				
LLP-ELM	0.126	0.148	0.073	0.131	0.235
VLSF	0.119	0.140	0.066	0.125	0.227
MENL-IK	0.111	0.133	0.059	0.117	0.220
LF-LELC	0.104	0.128	0.051	0.110	0.212
MuENL	0.096	0.121	0.045	0.102	0.208
MuEMNL	0.088	0.114	0.039	0.093	0.196
Technique	**One-error**				
LLP-ELM	0.253	0.150	0.281	0.235	0.252
VLSF	0.248	0.143	0.274	0.229	0.246
MENL-IK	0.241	0.137	0.266	0.221	0.240
LF-LELC	0.236	0.131	0.258	0.214	0.233
MuENL	0.230	0.125	0.251	0.209	0.225
MuEMNL	0.222	0.118	0.245	0.203	0.219
Technique	**Coverage**				
LLP-ELM	0.961	0.799	0.278	0.195	0.502
VLSF	0.948	0.783	0.265	0.188	0.489
MENL-IK	0.932	0.775	0.251	0.174	0.474
LF-LELC	0.924	0.768	0.239	0.161	0.458
MuENL	0.910	0.756	0.225	0.150	0.443
MuEMNL	0.907	0.744	0.218	0.142	0.435
Technique	**Ranking loss**				
LLP-ELM	0.172	0.230	0.121	0.193	0.208
VLSF	0.157	0.219	0.109	0.176	0.195
MENL-IK	0.134	0.205	0.094	0.168	0.181
LF-LELC	0.121	0.193	0.087	0.155	0.170
MuENL	0.108	0.180	0.075	0.141	0.163
MuEMNL	0.095	0.172	0.068	0.134	0.156
Technique	**Accuracy**				
LLP-ELM	0.751	0.524	0.632	0.799	0.723
VLSF	0.765	0.537	0.649	0.810	0.735
MENL-IK	0.779	0.552	0.664	0.821	0.749
LF-LELC	0.792	0.569	0.676	0.835	0.762
MuENL	0.805	0.584	0.691	0.847	0.774
MuEMNL	0.813	0.597	0.705	0.854	0.788

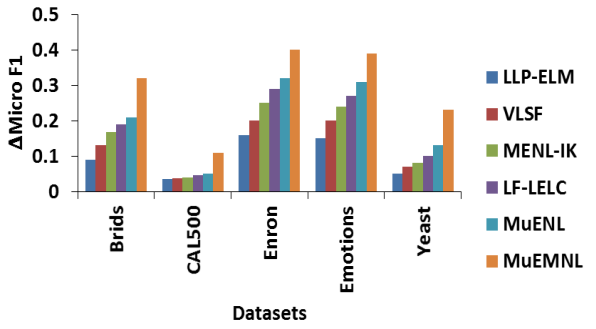

Figure. 5 Comparison of **Δ**Micro F1 for low-dimensional datasets

realizes better **Δ**Micro F1 compared to all other existing MLL techniques executed on birds, CAL500, Enron, emotions, and yeast datasets.

Table 4 provides accuracy, hamming loss, one-error, coverage, and ranking loss for different MLL techniques on low-dimensional datasets.

From this analysis, it is indicated that the MuEMNL technique achieves better performance in terms of hamming loss, one-error, coverage, ranking loss, and accuracy compared to the other MLL techniques applied to the low-dimensional datasets.

4.2 Analysis of MuEMNLHD performance

The efficiency of MuEMNLHD is compared with the MENL-iIK [11], DM2L [13], and MLL-FLSDR [14] using the 20Newsgroup dataset.

Fig. 6 shows the mean precision, F1-score, and **Δ**Micro F1 results for different MLL techniques executed on the 20Newsgroup dataset. It observes that the MuEMNLHD technique realizes better mean

Table 5. Performance on 20Newsgroup dataset

Metrics	MENL-iIK	DM2L	MLL-FLSDR	MuEMNLHD
Hamming loss	0.138	0.123	0.110	0.994
One-error	0.624	0.611	0.599	0.585
Coverage	0.289	0.275	0.263	0.258
Ranking loss	0.135	0.129	0.114	0.102
Accuracy	0.682	0.696	0.711	0.729

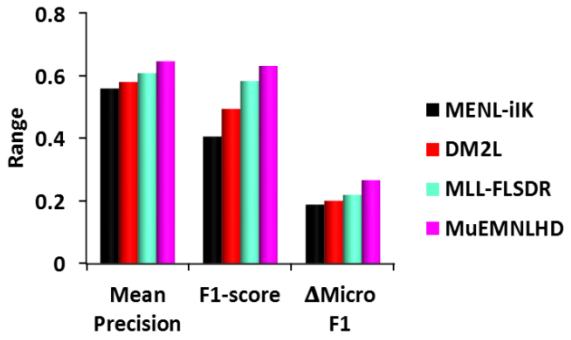

Figure. 6 Comparison of Different MLL techniques on 20Newsgroup dataset

precision, F1-score, and **Δ**Micro F1 compared to all other existing MLL techniques. That is, the mean precision of MuEMNLHD is 16.1% higher than the MENL-iIK, 11.9% higher than the DM2L, and 6.74% higher than the MLL-FLSDR techniques. Also, the F1-score of MuEMNLHD is 55.42% greater than the MENL-iIK, 27.73% greater than the DM2L, and 8.23% greater than the MLL-FLSDR techniques.

Table 5 provides accuracy, hamming loss, one-error, coverage, and ranking loss for different MLL techniques on the 20Newsgroup dataset (i.e., high-dimensional dataset).

From this scrutiny, it is addressed that MuEMNLHD
technique realizes good efficiency in terms of hamming loss, one-error, coverage, ranking loss, and accuracy compared to the other MLL techniques applied to the high-dimensional datasets.

5. Conclusion

In this article, two different versions of MuEMNL and MuEMNLHD techniques were presented to solve the MLL in complex scenarios. In this technique, different modules were performed. The findings revealed that the MuEMNL technique on low-dimensional databases attains 67.08% average mean precision, 75.14% mean accuracy, 64.02% mean F1-score, 0.125 mean ranking loss, 0.489 mean coverage, 0.201 mean one-error, and 0.106 mean hamming loss compared to the classical MLL techniques. Similarly, the MuEMNLHD technique on high-dimensional database achieve 64.6% mean precision, 63.1% F1-score, 72.9% accuracy, 0.102 ranking loss, 0.258 coverage, 0.994 hamming loss, and 0.585 one-errors compared to the classical MLL techniques. In future work, an ensemble technique is proposed to handle concept drifts when higher quantity of data arriving at high speeds.

Conflict of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, methodology, software, validation, Komagal Yallini; formal analysis, investigation, Mahendiran; resources, data curation, writing—original draft preparation, Komagal Yallini; writing—review and editing, Komagal Yallini; visualization; supervision, Mahendiran;

References

[1] T. Cour, B. Sapp, and B. Taskar, "Learning from Partial Labels", *The Journal of Machine Learning Research*, Vol. 12, pp. 1501-1536, 2011.

[2] S. Sun, and D. Zong, "LCBM: A Multi-View Probabilistic Model for Multi-Label Classification", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 43, No. 8, pp. 1-15, 2020.

[3] U. Sandouk, and K. Chen, "Learning Contextualized Music Semantics from Tags via a Siamese Neural Network", *ACM Transactions on Intelligent Systems and Technology*, Vol. 8, No. 2, pp. 1-21, 2017.

[4] D. Zufferey, T. Hofer, J. Hennebert, M. Schumacher, R. Ingold, and S. Bromuri, "Performance Comparison of Multi-Label Learning Algorithms on Clinical Data for Chronic Diseases", *Computers in Biology and Medicine*, Vol. 65, pp. 34-43, 2015.

[5] E. Gibaja, and S. Ventura, "A Tutorial on Multilabel Learning", *ACM Computing Surveys*, Vol. 47, No. 3, pp. 1-38, 2015.

[6] Z. Y. Zhang, P. Zhao, Y. Jiang, and Z. H Zhou, "Learning with Feature and Distribution Evolvable Streams", In: *Proc. of International*

Conference on Machine Learning, Vol. 119, pp. 11317-11327, 2020.

[7] Y. Zhu, K. M. Ting, and Z. H. Zhou, "Multi-Label Learning with Emerging New Labels", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 30, No. 10, pp. 1901-1914, 2018.

[8] Y. Qian, Q. Tong, and B. Wang, "Multi-Class Learning from Label Proportions for Bank Customer Classification", *Procedia Computer Science*, Vol. 162, pp. 421-428, 2019.

[9] J. Huang, X. Qu, G. Li, F. Qin, X. Zheng, and Q. Huang, "Multi-View Multi-Label Learning with View-Label-Specific Features", *IEEE Access*, Vol. 7, pp. 100979-100992, 2019.

[10] J. Huang, F. Qin, X. Zheng, Z. Cheng, Z. Yuan, W. Zhang, and Q Huang, "Improving Multi-Label Classification with Missing Labels by Learning Label-Specific Features", *Information Sciences*, Vol. 492, pp. 124-146, 2019.

[11] Y. Kongsorot, P. Horata, and P. Musikawan, "An Incremental Kernel Extreme Learning Machine for Multi-Label Learning with Emerging New Labels", *IEEE Access*, Vol. 8, pp. 46055-46070, 2020.

[12] Y. Ling, X. Wang, and Y. Ling, "Exploring Common and Label-Specific Features for Multi-Label Learning with Local Label Correlations", *IEEE Access*, Vol. 8, pp. 50969-50982, 2020.

[13] Z. Ma and S. Chen, "Expand Globally, Shrink Locally: Discriminant Multi-Label Learning with Missing Labels", *Pattern Recognition*, Vol. 111, pp. 1-33, 2021.

[14] J. Huang, P. Zhang, H. Zhang, G. Li, and H. Rui, "Multi-Label Learning via Feature and Label Space Dimension Reduction", *IEEE Access*, Vol. 8, pp. 20289-20303, 2020.

[15] C. Zhang and Z. Li, "Multi-Label Learning with Label-Specific Features via Weighting and Label Entropy Guided Clustering Ensemble", *Neurocomputing*, Vol. 419, pp. 59-69, 2021.

[16] https://www.uco.es/kdis/mllresources/