IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 March 2024, accepted 5 April 2024, date of publication 9 April 2024, date of current version 17 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3386841

RESEARCH ARTICLE

Improving Neural Network-Based Multi-Label Classification With Pattern Loss Penalties

WORAWITH SANGKATIP[1], PHATTHANAPHONG CHOMPHUWISET[2], KAVEEPOJ BUNLUEWONG[2], SAKORN MEKRUKSAVANICH[3], (Member, IEEE), EMMANUEL OKAFOR[4], AND OLARIK SURINTA[5]

[1]Department of Information Technology, Faculty of Informatics, Mahasarakham University, Kham Riang, Maha Sarakham 44150, Thailand
[2]Polar Laboratory, Department of Computer Science, Faculty of Informatics, Mahasarakham University, Kham Riang, Maha Sarakham 44150, Thailand
[3]Department of Computer Engineering, School of Information and Communication Technology, University of Phayao, Mae Ka, Phayao 56000, Thailand
[4]SDAIA-KFUPM Joint Research Center for Artificial Intelligence, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia
[5]Multi-Agent Intelligent Simulation Laboratory (MISL) Research Unit, Department of Information Technology, Faculty of Informatics, Mahasarakham University, Kham Riang, Maha Sarakham 44150, Thailand

Corresponding author: Olarik Surinta (olarik.s@msu.ac.th)

This research project was financially supported by Mahasarakham University under Grant 6717011/2567.

ABSTRACT This research work introduces two novel loss functions, pattern-loss (POL) and label similarity-based instance modeling (LSIM), for improving the performance of multi-label classification using artificial neural network-based techniques. These loss functions incorporate additional optimization constraints based on the distribution of multi-label class patterns and the similarity of data instances. By integrating these patterns during the network training process, the trained model is tuned to align with the existing patterns in the training data. The proposed approach decomposes the loss function into two components: the cross entropy loss and the pattern loss derived from the distribution of class-label patterns. Experimental evaluations were conducted on eight standard datasets, comparing the proposed methods with three existing techniques.The results demonstrate the effectiveness of the proposed approach, with POL and LSIM consistently achieving superior accuracy performance compared to the benchmark methods.

INDEX TERMS Multi-label classification, label correlation, label-specific features, deep neural network, loss functions.

I. INTRODUCTION

Multi-label classification (MLC) is one of the supervised learning methods that explicitly classifies data instances into a set of mutual classes (or multiple labels) [1], [2]. MLC has been applied to problems domains, such as document classification [3], [4], medical diagnosis [5], recommendation systems [6], product review classification [7], categorising a video clips into several categories [8], [9], classifying the patient diseases [10], [11], [12] and classifying human emotions from audio [13], [14]. Boutell et al. [15] introduced a seminal work to classify multi-objects within individual image scenes by addressing the problem as an MLC task, and has inspired much further work. Prior, includes MLC applied to phenotypic data domains, including Clare and King [16].

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

The approaches to solve MLC problems can broadly be categorized into three groups [17]: adaptation methods, problem transformation methods, and ensemble methods. Adaptation methods (AM) inherently adapts the conventional machine learning algorithms that tackle multi-class classification problems to MLC. Quinlan's C4.5 decision tree algorithm was adapted to tackle multi-label classification tasks, called ML-C4.5 [16]. The K-Nearest Neighbors (KNN) technique was modified, known as ML-KNN [18]. Problem transformation methods (PTM) on the other hand, convert the MLC problems to one or more single-label learning problem(s), such that conventional classification algorithm can be applied to solve the problem directly. Binary relevance (BR) [19], classifier chains (CC) [20], label power-set (LP) are the examples. Ensemble methods (EM) were introduced by combining PTM methods aimed to deliberately improve the performance of the (class) classification problem

(in order to solve the MLC task). RAkEL [21], EPS [22], and ECC [23] are some of the examples from the ensemble family.

The current challenges of MLC have focused mainly on improving classification efficiency, where research has been carried out to cope with the specific issues underlying the generalization of the classification task.

Feature engineering techniques [24], [25] can generally be divided into feature selection (FS), feature transformation (FT), feature reconstruction (FR). FS preserves a subset of features for the data and dismisses the others. FT and FR constructs a set of features from the original feature sets [26]. FT can be exploited by implementing, for example, deep learning algorithms that can encode data features and produce corresponding latent feature sets used in the classification processes [27], [28]. Cheng et al. [28] identified a significant drawback of applying a conventional auto-encoder scheme for generating latent features and proposed an extreme learning-based method to generate and extract the correlation of label-to-label and feature-to-feature in data used in the classification process.

Label correlation (LC) technique has demonstrated efficacy to improve the performance of MLC [29], [30], [31]. The technique examines the basis correlation of the feature-to-labels in the data by quantifying a degree of their dependency. Zhang [32] proposed the LIFT method, which applied a K-means clustering algorithm to group the positive and negative instances of each label in the data. Then, the characteristics of the data were extracted through the distance measurement between the data instances and the cluster centers of each label. Subsequently, the relationship between the labels was derived by generating additional attributes of the data. The LIFT method achieved high performance on 17 benchmark datasets compared to other multi-label learning algorithms. Huang et al. [33] proposed a technique to learn the dispersion of label attributes, including common attributes. They applied double-label correlation to differentiate labels for each category. Luaces et al. [34] investigated conditional dependency between labels, using its measure as a basis for synthetic dataset generation and MLC model evaluations.

Several studies have shown that neural networks and deep neural networks can improve the MLC classification performance. Zhang et al. [35] proposed a method for using deep neural networks (DNNs) to classify multi-label data, namely GroupNet. The technique used the conventional PTM family to transform the problem (using BR and LP) to a multi-class classification task. Then, the classification was carried out using a convolutional neural network (CNN) [36]. Transforming a small number of label classes can make the problem more applicable in solving it as multi-class classification. However, it can become an intractable problem using the transformation method when datasets contain a large number of label classes. Recent research has demonstrated success in applying deep learning-based approaches for solving MLC problems [37], [38], [39]. Some of these techniques rely on the autoencoders which allow an unsupervised learning process to carry out for generating encoded features for the classification [28], [40]. Lipton et al. [41] implemented a recurrent neural network (RNN) to classify diagnostic data with 128 diagnostic labels. A sequence-to-sequence based approach was essentially applied to solve this classification problem. The input features were mapped into dense time series set and the output labels were encoded into multivariate binary classes [42], [43]. Then, these input sequences and their corresponding output labels were fed to the RNN to generate a model.

A. RESEARCH CONTRIBUTIONS

This research proposes a classification approach for neural networks with multi-label data. It is inspired by the LIFT method proposed by Zhang [32]. The two proposed methods classify and predict data labels by integrating label distribution and *label patterns* information from the origin dataset. Label patterns represent the relationships between groups of labels that appear in the data. These labels help identify shared patterns among different data sets. Typically, a group of labels exhibits relationships between individual labels. For example, when labels A and B co-occur, label C tends to appear as well, indicating consistent meaning. Conversely, some labels may be inversely related. If labels A and B are present, label C must not co-occur. Researchers have focused on identifying and utilizing label patterns, which play a crucial role in multi-label classification. Leveraging label patterns can enhance the accuracy and rationality of classification results. These label patterns are assembled during the training procedure, and used in the loss function to guide the optimization of network parameters in order to obtain a generalized model. Within the loss function, a *pattern-loss* penalty guides the reward (negative penalties) of class label predictions that are anticipated to predict the known *label patterns*. The training process's loss function is divided into two components, i.e., (i) cross entropy loss and (ii) associated loss (*pattern-loss*) obtained from the information of the class-label patterns.

In this article, we propose and report on two *pattern-losses*: (i) Patterns Of the Label (**POL**) is introduced as the pattern loss to constrain the predicted classes of the data instance with respect to the existing patterns of labels in the training data. (ii) Label SIM-ilarity (**LSIM**) is also implemented as pattern-loss deriving the predicted class of a data instance towards its similar data instances. The classification performance of POL and LSIM are compared with (state-of-the-art) benchmarks in neural network-based techniques.

The key contributions of this article are:

- Two novel loss functions – **POL** and **LSIM** – for neural network model training on multi-label classification (MLC) problems, as inspired by Zhang [32], see Section III.
- A thorough evaluation of classification performance of **POL**, **LSIM** and three state-of-art MLC neural network techniques (BP-MLL, ML-HARAM, ANN). Measured

over eight datasets from the MULAN dataset series in Section IV, as modelled after [44].

Outline. Section II) explains neural networks for multi-label classification. Section III also includes the MULAN dataset analyses. Section IV for experiments and results. Statistical significance testing of results in Section V, followed by discussion and conclusion.

II. NEURAL NETWORKS FOR MULTI-LABEL CLASSIFICATION

A. PRELIMINARIES OF NEURAL NETWORKS

Model learning from data using a neural network concept for multi-label classification can be formally expressed as follows: Let X be a space of data instances comprising n data instances x, i.e. $\forall x \in X, x = \{x_1, \ldots x_d\}$ (where d is the number of instance features) a set of d-dimensional features, and a set p a possible label space $Y = \{y_1, \ldots y_p\}$, i.e. $y = \{y_1, ..y_m\}$ where $y = \{0, 1\}$ and m denotes the dimension of the labels y associated with x. We assume that a feasible solution for the neural network is a set of training networks $N = n_1, \ldots, n_k$. The objective is to find an optimal network configuration (weights) that produces predicted outcomes Y' that closely match the true values of Y. To construct a generalized model for the task, the network takes the input x through a set of hidden layers $H = \{h_1(.), \ldots, h_L(.)\}$ where L is the number of the hidden layer in the network.

Each hidden layer contains a set of adjustable and tractable parameters w and b. An additional final layer of the network is augmented by an activation function(s) that transforms the layer outputs into (0/1 binary-bit label) prediction outcomes. Therefore, the prediction obtained from a network can be evaluated as follows:

$$[y_1, y_2, \ldots, y_m] = [h_1(x), h_2(x), \ldots, h_L(x)], \tag{1}$$

then we can simplify to

$$Y_x = g(H(x)), \tag{2}$$

where $g(\cdot)$ represents the activation function applied to the collective output of the hidden layers $H(x)$, yielding the final predicted labels Y_x.

B. NEURAL NETWORK FOR MULTI-LABEL CLASSIFICATION

For a classification problem, a network architecture can be constructed with d inputs and $|y|$ outputs (one for each label). The number of nodes for an input layer is typically the features or attributes of a dataset, and the connections of the input layer to the hidden layer can be different depending on how many nodes are selected for the hidden layer. The hidden layer (H) can be decomposed by multiple different layers stacked together, depending an application. The hidden layer is connected to the output layer. In a forward pass procedure, each node in the network convey the information from input to the output layer. At this point, the network attempts to learn the sample data that is passed in, and carries out a prediction of the data, where the nodes of the output layer are probabilities that the sample is of a certain multi-label. In the forward pass and back-propagation procedure (training), the process continues until a certain number of iterations are met, or the network converges.

During a network training task, each data instance is associated with a multi-class labels ($\mathbf{y}$). Thus, this allows the training process to carry out for generalizing a model using the information of the predicted label ($\hat{y}$) and the associated label. An error ϵ, cost E, or loss function l is a key during the network training process. An optimization procedure will modify the network's parameters for its next training cycle (epoch). The basis of that modification is derived from the loss function value. The optimization procedure aims to consistently minimize the value returned from the loss function, over all epochs.

In multi-label classification problems and in (binary/) multi-class classification problems, loss functions are similar. In multi-label classification, the loss function can additionally consider individual label predictions in a variety of manners to steer the rate of improvements. This issue underlies our research. Typically, Binary cross-entropy is a standard (commonly used) loss function for neural networks and DNNs, defined as follows:

$$E = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(\hat{y}^{\mathrm{i}}, y^{\mathrm{i}}) \tag{3}$$

where

$$\mathcal{L}(\hat{y}^{\mathrm{i}}, y^{\mathrm{i}}) = -\sum_{j=1}^{m} y_j log(\hat{y}) + (1 - y_j) log(1 - \hat{y}_j) \tag{4}$$

This E determines the distribution of predicted labels ($\hat{y}$) and the true labels (y) of data instances. It is intended to use with binary classification where the target value is 0 or 1. It will calculate a difference between the actual and predicted probability distributions for predicting class 1. The score is minimized and a perfect value is 0.

III. MATERIALS AND METHODS

The main purpose of this work is to develop a technique that can classify multi-label data efficiently. The key concept to solve MLC problems, in this work, is to apply a generic neural network-based technique that can perform the classification by integrating information about the *patterns of labels* in our evaluation datasets. Measurements derived from the label patterns are used to construct an additional component of a model's *loss* function, which we refer to as *Pattern-Loss* term. Constructed in a similar manner to Zhang and Zhou [45] minimizing penalty in Equation 3. The label patterns are determined by examining the *cardinality* and *frequency distribution* of the prediction label (Y) in the data. See detailed explanation of these proposed methods in subsection III-B. The overall process of the technique presented in this work is illustrated in Figure 1.

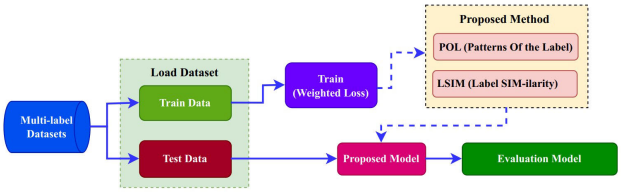


FIGURE 1. The overall process of the proposed method.

A. DATASET AND LABEL PATTERN ANALYSIS

This section describes the multi-label datasets (MLD), analysis of their label patterns and their numerical label pattern distributions. The proposed method was experimented on eight datasets. These datasets include seven from the *MULAN* [46] repository: birds, enron, emotions, medical, yeast, scene, and cal500. Additionally, one dataset was sourced from the foodtruck dataset [47]. There are eight datasets with different data topologies and domains, shown in Table 1. Each of the datasets has different characteristics, such as the number of data instances, the number of features (data dimensions), number of label bits (per instance), number of unique label-bit sets, *cardinality* and *density*.

In the multi-label context, the *cardinality* is the mean proportion of active labels (y) in the set of possible label bits (Y), defined as:

$$Card = \frac{1}{N}\sum_{i=1}^{N}|y_i| \tag{5}$$

and *density* is an equivalent normalized measure to compare all datasets. It is *cardinality*, divided by the number of label bits $|Y|$,

$$Dens = \frac{\frac{1}{N}\sum_{i=1}^{N}|y_i|}{|Y|} \tag{6}$$

Each data instance (x_i) is associated with a particular multi-label entity (or label bit set) y_i, (for example, $y_{x_i} = [0, 1, 1, 1, 0]_{x_i}$ is associated to x_i). Each dataset has a number of *unique label-bit sets* (see *Label-bit sets* column in Table 1). We refer to each unique label-bit set, as a label-bit – *pattern*. The number of unique label-bit *patterns* can be expressed as $\|P\|$, where a single *pattern* (p) and the full (unique) set of label-bit *patterns* is given by $p \in P$. Where $p = 2^{|Y|}$, i.e. is a binary label-bit set, and p is originates from Y. The frequency characteristics of these *patterns* will guide our proposed loss functions penalties, as we will describe in the following sections.

Where a specific label pattern ($y_i \Leftrightarrow y_p$) is associated to a corresponding set of data instances (x_p), such that $y_p \mapsto x_p$, where $x_{pi} \in x_p \in X$.

For example, the *yeast* dataset contains 164 unique label patterns (i.e. y_i as a string literal), with which their frequency can be counted and probability calculated.

The *pattern loss* of the labeled dataset is informed by the number and frequency of patterns embedded patterns within the data. It can then display the number of labels occurring for each pattern, for example, the yeast dataset contains 164 pattern labels. Then analyze the number of each pattern to see how many there are. A histogram of data labels in each dataset are illustrated in Figure 2.

B. PROPOSED METHODS: INCORPORATING PATTERN-LOSS

This work proposes two customized loss function techniques for multi-label classification, i.e. (i) *patterns of the label* (***POL***) and (ii) *label similarity* (***LSIM***) - where both utilize *pattern-loss* to inform the penalty component of the model's loss function. See Algorithm 1 for POL and Algorithm 2 for LSIM.

POL begins by analyzing the patterns of binary labels denoted as $P = \{p_1, p_2, \ldots, p_n\}$, where n represents the number of binary label patterns present in the data. The weighted loss term in the algorithm is divided into two components: (i) the native term and (ii) the pattern term. These terms are combined using a weight factor α that ranges between 0 and 1, determining the relative importance of each loss term. The loss calculation incorporates both the native (l_{nv}) and pattern terms (l_{pn}), allowing the algorithm to appropriately balance their contributions based on the specified weight α. The computation of POL is demonstrated in Algorithm 1.

Algorithm 1 The Computational Algorithm of the POL Method

Input: y, $\hat{y}$, P, α
Output: $\mathcal{L}$
Initialisation : $\epsilon \leftarrow 0$
for $i \leftarrow 1$ to $|X|$ **do**
$l_{nv} \leftarrow -\frac{1}{m}\sum_{j=1}^{m}(y_j \log(\hat{y}_j) + (1 - y_j)\log(1 - \hat{y}_j))$
$l_{pt} \leftarrow \max(f_d(\hat{y}_i, P))$
$\epsilon \leftarrow \epsilon + (\alpha \times l_{nv}) + (1 - \alpha \times l_{pt})$
end for
$\mathcal{L} = \frac{1}{|X|} \times \epsilon$
return $\mathcal{L}$

where f_d denotes a distance function. The function calculates the Euclidean distance between the prediction and a pattern in the training instances, which is defined as follows:

$$f_d(\hat{y}, p) = \sqrt{\sum_{i=1}^{m}(\hat{y}_i - p_i)^2} \tag{7}$$

POL intuitively, aims to capture the variability of binary label patterns within a dataset by considering the predicted values and the actual values of the labels forming each pattern. The first loss in POL (denoted as l_1) utilizes the binary cross entropy (or log loss) function. This loss term is label-independent and accounts for the patterns exhibited by the data labels in the data. The second loss (l_2) serves to penalize the situation where the predicted labels deviate

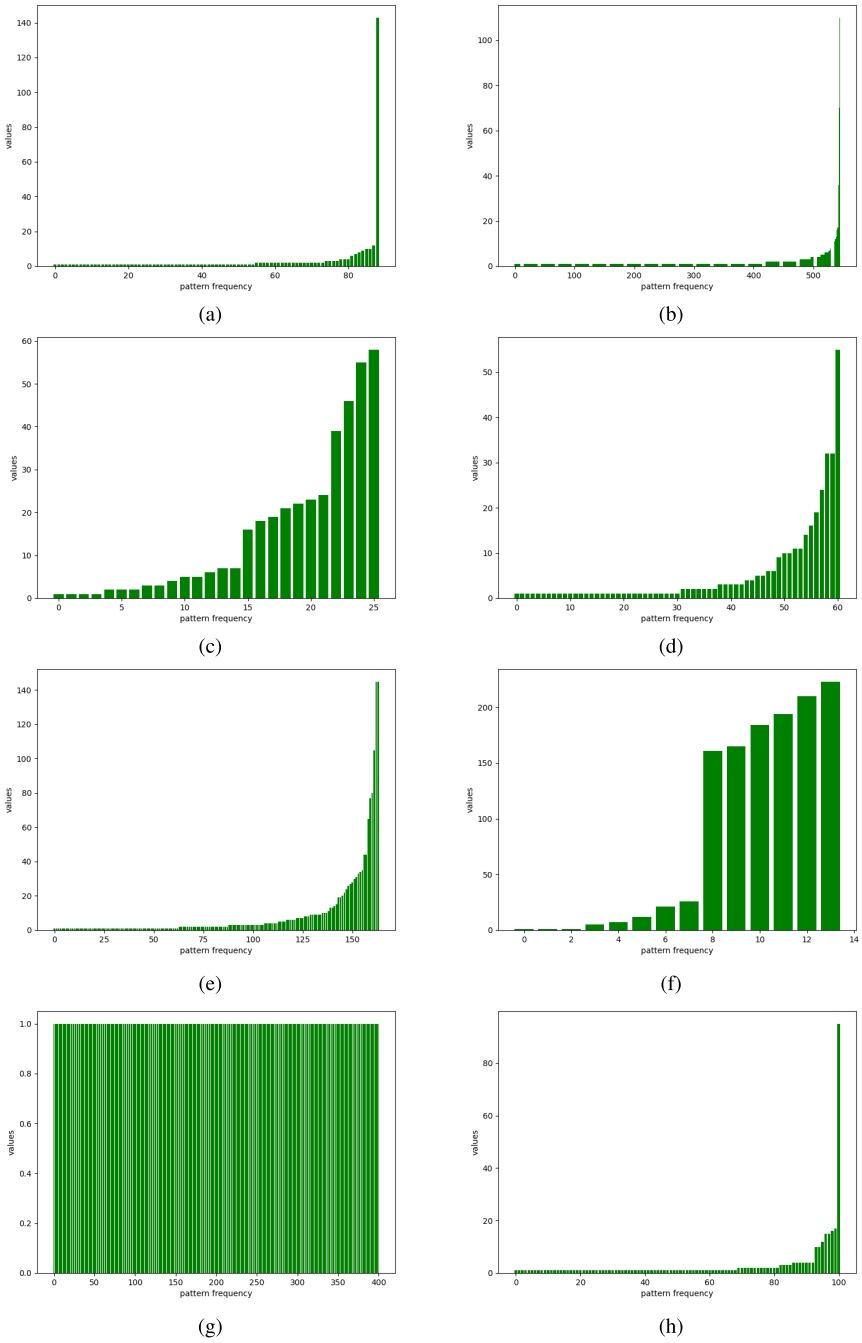


FIGURE 2. Visualized the frequency distributions of the label patterns in each dataset: (a) birds, (b) enron, (c) emotions, (d) medical, (e) yeast, (f) scene, (g) cal500, and (h) foodtruck.

significantly from the pattern represented by the actual training labels. This is achieved by penalizing the predicted labels that deviate the most in the pattern space. The objective is to encourage the predicted labels to stay close to the overall label pattern observed in the data. Then, both loss terms are regularized and weighted using the parameter α. The resulting loss value, denoted as ϵ, represents the combination of the two losses. By taking the mean of ϵ, the overall loss of the POL algorithm is obtained.

LSIM is introduced as the information that can be used to generalize the model. LSIM enables the identification of common feature characteristics among data instances classified under the same label pattern, while also providing insights into underlying features. During each training process, LSIM examines the similarity between data instances based on their features. To identify a set of similar instances $S = \{x_1,..x_k\}$ where k is the number of similar members (set by a predetermined value). S is generated by K-Nearest

TABLE 1. Characteristics of 8 multi-label datasets (*MULAN* repository and [47]) used in this research work.

Dataset	Domain	Instances	Features	Label Bits	Label-bit sets	Cardinality	Density
birds	Audio	645	260	19	133	1.014	0.053
enron	Text	1702	1001	53	753	3.378	0.064
emotions	Music	593	72	6	27	1.869	0.311
medical	Text	978	1449	45	94	1.245	0.028
yeast	Biology	2417	103	14	198	4.237	0.303
scene	Image	2407	294	6	15	1.074	0.179
cal500	Music	502	68	174	34	26.044	0.150
foodtruck [47]	Recommend	407	21	12	116	2.290	0.191

Neighbour (KNN) algorithm. Given S, there exists a label set $G = \{y_1,..y_k\}$ associated with S. Then, the loss term is calculated over the average loss in G and the predicted label, which is demonstrated in Algorithm 2.

Algorithm 2 The Computational Algorithm of the LSIM Method

Input: y, y', G
Output: L
Initialisation: $\epsilon \leftarrow 0$
1: **for** $i \leftarrow 1$ to N **do**
$s \leftarrow 0$
2: **for** $j \leftarrow 1$ to G **do**
$s \leftarrow -\frac{1}{|y_{ij}|} \sum_i (y_{ij} \log(y_i j') + (1 - y_i j) \log(1 - y_i j'))$
3: **end for**
$\epsilon \leftarrow (\frac{s}{|G|})$
4: **end for**
$L = \frac{1}{N} \times \epsilon$
5: **return** L

LSIM designates the loss function counting the class label of the similar data instance to the consideration, as described in Algorithm 2. The standard binary cross-entropy computation with the true label values and the labels of the (KNN, k) nearest data instances were used to calculate the (KNN cluster) localized model loss. In this work, we set $k = 3$ as the number of similar data instances of KNN. The final loss value was calculated as the mean average of all loss values from the (KNN cluster) localized data instances.

C. EXISTING METHODS

Comparative benchmarks of our proposed methods are provided by three existing and state-of-the-art multi-label classification (MLC) techniques, as follows.

Backpropagation for multi-label learning (BP-MLL) [45], [44]. This feed-forward neural network uses an error function to capture the correlation among the (MLC) labels. Its error function penalizes the predictions that include labels that are not truly relevant to the processed instance. BP-MLL network parameters used in the experiment are: input layer set to the number of the input features (attributes), two hidden layers, and output layer equal to the number of labels. *ReLU* as the activation function of the input and hidden layers and the *Sigmoid* function is used at the output layer. BP-MLL used the *cross-entropy* loss function scheme.

Multi-label hierarchical adaptive resonance associative map neural network (ML-HARAM) [48], [49]. This neural system was initially developed for text datasets with high dimensionality. Overall, it aims to increase the classification speed by adding an extra adaptive resonance theory (ART) layer to the network in order to group the learned prototypes into large clusters. ML-HARAM's parameters used in the experiment are: set *vigilance* to 0.95 as parameters for adaptive resonance theory networks. Define the *threshold* value as 0.05, which controls how many prototypes participate in the prediction.

In the experiment, the parameters of the ANN were set as follows: the input layer size was determined by the number of input features (attributes), two hidden layers were employed, and the output layer size matched the number of labels. The activation function used for the input and hidden layers was *ReLU*, while the *Sigmoid* function was applied to the output layer. Additionally, the ANN utilized the MLC *binary cross-entropy* loss function and the Adam gradient descent optimizer.

IV. EXPERIMENTS AND RESULTS

A. EXPERIMENT SETUP

The five techniques were evaluated on multi-label classification (MLC) problems. POL and LSIM are the two proposed (test) methods, see (subsection III-B). ANN has an identical network architecture and configuration, thus is a control for the POL and LSIM evaluation. BP-MLL and ML-HARAM are existing state-of-the-art methods, see (subsection III-C). Performance is evaluated over eight datasets with input features ranging from 21 and 1449 dimensions, and label quantities between 6 and 174, as described in section (subsection III-A).

All experiments were executed on a machine with an Intel Core i7-8565U (1.99 GHz) processor, 20.0 GB of memory and with a Windows 10 operating system. The experiments were conducted using the SciKit-multilearn open source Python 3.x library [50].

B. EVALUATION METRICS

Ten common example-based and label-based evaluation metrics for MLC [51] were selected to quantify performance, including precision, recall, F1 and Hamming loss with

TABLE 2. Loss function experiment methods.

Methods	Loss-Function
POL	See Algorithm 1
LSIM	See Algorithm 2
ANN (Control)	Binary Cross-Entropy
BP-MLL [45]	Binary Cross-Entropy, see Equation 3
ML-HARAM [48], [49]	Binary Cross-Entropy

TABLE 3. Experiment parameters.

Experiment constants	Value
Datasets	8
Epochs	10
Train : Validation Ratio	0.8 : 0.2
Cross Validation Folds	$k = 5$ (non-stratified)

specialized micro and macro variants for MLC. Intuitively, precision measures the model's ability not to label a negative sample as positive, recall is a score to find all positive samples, F1 is their ratio, where macro and micro variants are unweighted averages and global totals. Hamming loss is the fraction of labels incorrectly predicted.

Each evaluation metric relies on the collection and calculation of true positives (tp_j), true negatives (tn_j), false positives (fp_j), and false negatives (fn_j) obtained for each label $y : j = [1, .., m]$. Macro F_1 is the harmonic mean obtained from Precision and Recall, based on an average of each label y_j and an average over all labels. Note that, Macro variants calculate metrics for each label and find their unweighted mean; which does not take label imbalance into account. Micro F_1 is the harmonic mean derived from Micro Precision and Micro Recall, as given:

$$\text{Precision (P)} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Z_i|} \tag{8}$$

$$\text{Recall (R)} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i|} \tag{9}$$

$$\text{F1} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i| + |Z_i|} \tag{10}$$

$$\text{Macro Precision (Macro P)} = \frac{1}{m}\sum_{j=1}^{m}\frac{tp_j}{tp_j + fp_j} \tag{11}$$

$$\text{Macro Recall (Macro R)} = \frac{1}{m}\sum_{j=1}^{m}\frac{tp_j}{tp_j + fn_j} \tag{12}$$

$$\text{Macro F1} = \frac{1}{m}\sum_{j=1}^{m}\frac{2\times R_j \times P_j}{R_j + P_j} \tag{13}$$

and the micro- evaluation metric variants:

$$\text{Micro Precision (Micro P/MiP)} = \frac{\sum_{j=1}^{m} tp_j}{\sum_{j=1}^{m} tp_j + \sum_{j=1}^{m} fp_j} \tag{14}$$

$$\text{Micro Recall (Micro R/MiR)} = \frac{\sum_{j=1}^{m} tp_j}{\sum_{j=1}^{m} tp_j + \sum_{j=1}^{m} fn_j} \tag{15}$$

$$\text{Micro F1} = \frac{2\times MiR \times MiP}{MiR + MiP} \tag{16}$$

and finally,

$$\text{Hamming Loss} = \frac{1}{|N| \cdot |L|}\sum_{i=1}^{|N|}\sum_{j=1}^{|L|}\text{xor}(y_{ij}, \hat{y}_{ij}) \tag{17}$$

C. EXPERIMENT RESULTS

Multi-label classification (MLC) performance was evaluated over each proposed technique (POL and LSIM) and the comparative benchmark existing techniques (BP-MLL, ML-HARAM, ANN), across the eight datasets. Figure 3 and Tables 4, 5, 6 and 7 show metric results of the methods on each dataset; as means, standard deviations (± over the cross validation folds), rank (# as mean and standard deviation ±) where lower is preferred, and percentage of wins (Win %).

D. CLASSIFICATION METRIC PERFORMANCE OF PROPOSED METHODS

In this work all 10 metrics were calculated. Figure 3 and Tables 4, 5, 6 and 7 illustrate F1, Macro-F1, Micro-F1, and Hamming Loss results as a summarization of their MLC performance.

V. RESULTS ANALYSIS

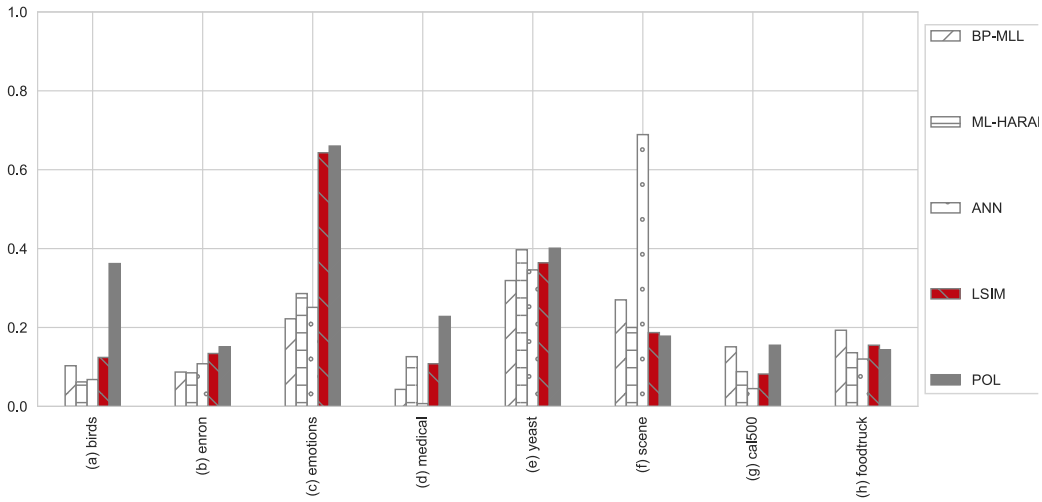
A. PERFORMANCE SIGNIFICANCE OF PROPOSED METHODS

Finally, we are interested in investigating the comparative significance of the existing MLC and proposed techniques in the experiments. The Bonferroni-Dunn test [52] is employed as a statistical method to serve the above purpose. Here, the difference between the average ranks of a proposed algorithm (control) and one comparable algorithm (test) can be compared with the following critical difference (CD):

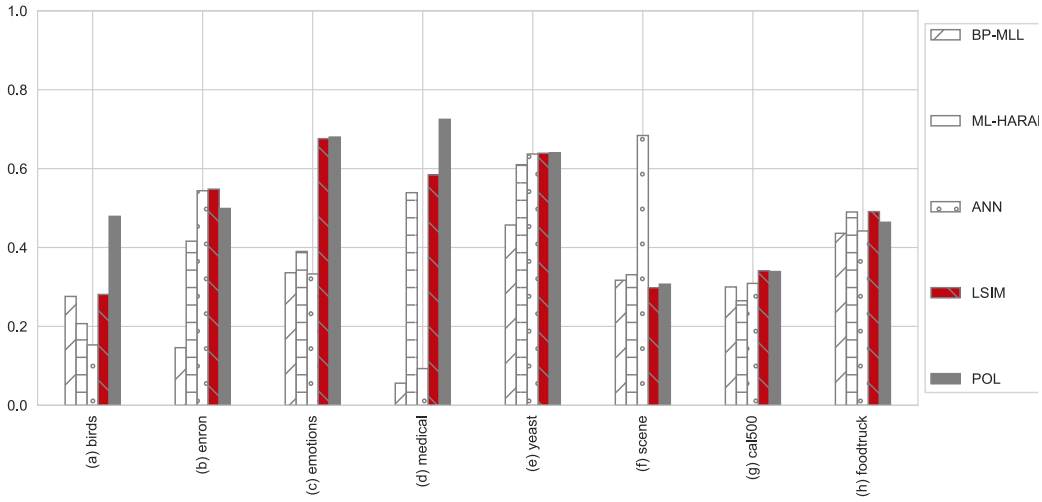
$$CD = q\alpha\sqrt{\frac{k(k+1)}{6N}} \tag{18}$$

For Bonferroni-Dunn test, we have $q\alpha = 2.498$ at significance level $\alpha = 0.05$ and thus CD = 1.974 ($k = 5$, $N = 8$). Accordingly, the performance between a proposed algorithm (control) and one comparable method (test) is deemed to be significantly different if their average ranks over all datasets differ by at least one CD.

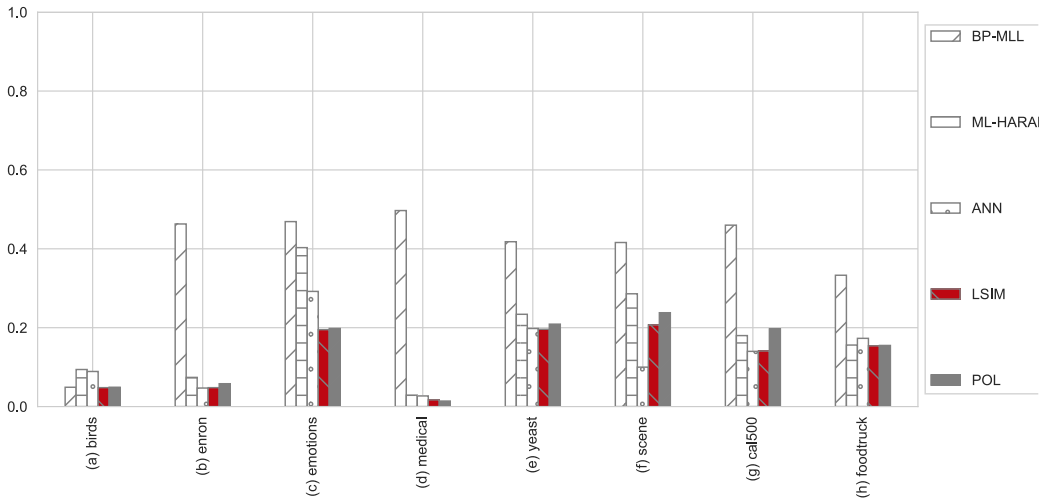
Figure 4 is a critical distance diagram of each technique's rank, which illustrates the results of the Bonferroni-Dunn test. The top line in the diagram is the axis along which the average rank of each multi-label classifier is plotted, from the lowest ranks (best performance) on the left to the highest ranks (worst performance) on the right. Groups of algorithms that are connected from one another are not statistically different (i.e. their average rank is within one CD).



(a) Macro F1 performance - for Precision/Recall balance, with equal weight per label. *(1.0 preferred.)*



(b) Micro F1 performance - for Precision/Recall balance, as a multi-class task. *(1.0 preferred.)*



(c) Hamming Loss performance - indicating accuracy, via the number of misclassified labels *(0.0 preferred.)*

FIGURE 3. Summary of performance per data. Highlighting proposed methods: *LSIM* (red) and *POL* (gray): (a) Macro-F1, (b) Micro-F1, and (c) Hamming Loss.

TABLE 4. *F*1 performance - precision/recall balance. (1.0 preferred).

Datasets	*BP-MLL*	*ML-HARAM*	*ANN*	LSIM	POL
birds	0.105 ± 0.027(3)	0.131 ± 0.021(2)	0.063 ± 0.025(5)	0.100 ± 0.033(4)	0.200 ± 0.052(1)
enron	0.143 ± 0.016(5)	0.404 ± 0.016(4)	0.520 ± 0.015(2)	0.529 ± 0.014(1)	0.502 ± 0.010(3)
emotions	0.320 ± 0.114(4)	0.374 ± 0.058(3)	0.270 ± 0.064(5)	0.640 ± 0.043(2)	0.658 ± 0.059(1)
medical	0.055 ± 0.016(5)	0.568 ± 0.050(2)	0.056 ± 0.048(4)	0.446 ± 0.064(3)	0.729 ± 0.046(1)
yeast	0.440 ± 0.050(5)	0.589 ± 0.017(4)	0.613 ± 0.016(2)	0.610 ± 0.019(3)	0.619 ± 0.018(1)
scene	0.306 ± 0.043(3)	0.334 ± 0.239(2)	0.614 ± 0.030(1)	0.247 ± 0.191(5)	0.302 ± 0.231(4)
cal500	0.297 ± 0.021(4)	0.235 ± 0.066(5)	0.314 ± 0.014(3)	0.341 ± 0.004(1)	0.334 ± 0.017(2)
foodtruck	0.413 ± 0.054(5)	0.511 ± 0.042(1)	0.459 ± 0.047(4)	0.503 ± 0.028(2)	0.500 ± 0.038(3)
Avg.rank	4.25 ± 0.83	2.88 ± 1.27	3.25 ± 1.39	2.62 ± 1.32	**2.0 ± 1.12**
Win.%	0.0	0.125	0.125	0.25	**0.5**

TABLE 5. *Macro F*1 performance - precision/recall balance, with equal weight per label. (1.0 preferred).

Datasets	*BP-MLL*	*ML-HARAM*	*ANN*	LSIM	POL
birds	0.103 ± 0.019(3)	0.062 ± 0.025(5)	0.068 ± 0.030(4)	0.124 ± 0.043(2)	0.362 ± 0.035(1)
enron	0.087 ± 0.005(4)	0.085 ± 0.009(5)	0.108 ± 0.007(3)	0.134 ± 0.010(2)	0.151 ± 0.005(1)
emotions	0.222 ± 0.119(5)	0.286 ± 0.057(3)	0.251 ± 0.053(4)	0.643 ± 0.044(2)	0.660 ± 0.066(1)
medical	0.043 ± 0.009(4)	0.126 ± 0.022(2)	0.007 ± 0.005(5)	0.108 ± 0.026(3)	0.228 ± 0.023(1)
yeast	0.319 ± 0.035(5)	0.397 ± 0.016(2)	0.346 ± 0.008(4)	0.364 ± 0.008(3)	0.401 ± 0.013(1)
scene	0.270 ± 0.060(2)	0.200 ± 0.093(3)	0.689 ± 0.023(1)	0.187 ± 0.095(4)	0.178 ± 0.090(5)
cal500	0.151 ± 0.011(2)	0.088 ± 0.011(3)	0.045 ± 0.003(5)	0.082 ± 0.004(4)	0.155 ± 0.014(1)
foodtruck	0.193 ± 0.041(1)	0.136 ± 0.030(4)	0.120 ± 0.017(5)	0.155 ± 0.024(2)	0.143 ± 0.020(3)
Avg.rank	3.25 ± 1.39	3.38 ± 1.11	3.88 ± 1.27	2.75 ± 0.83	**1.75 ± 1.39**
Win.%	0.125	0.0	0.125	0.0	**0.75**

TABLE 6. *Micro F*1 performance - precision/recall balance, more weight to frequent labels. (1.0 preferred).

Datasets	*BP-MLL*	*ML-HARAM*	*ANN*	LSIM	POL
birds	0.276 ± 0.041(3)	0.207 ± 0.022(4)	0.153 ± 0.061(5)	0.281 ± 0.070(2)	0.479 ± 0.051(1)
enron	0.146 ± 0.016(5)	0.416 ± 0.024(4)	0.544 ± 0.017(2)	0.548 ± 0.013(1)	0.499 ± 0.013(3)
emotions	0.336 ± 0.109(4)	0.390 ± 0.053(3)	0.333 ± 0.054(5)	0.676 ± 0.038(2)	0.680 ± 0.056(1)
medical	0.056 ± 0.016(3)	0.539 ± 0.038(4)	0.093 ± 0.075(5)	0.584 ± 0.054(2)	0.725 ± 0.045(1)
yeast	0.457 ± 0.050(5)	0.610 ± 0.013(4)	0.637 ± 0.014(3)	0.639 ± 0.013(2)	0.640 ± 0.014(1)
scene	0.317 ± 0.038(3)	0.331 ± 0.222(2)	0.684 ± 0.024(1)	0.298 ± 0.219(5)	0.307 ± 0.227(4)
cal500	0.300 ± 0.021(4)	0.265 ± 0.061(5)	0.309 ± 0.015(3)	0.341 ± 0.003(1)	0.339 ± 0.017(2)
foodtruck	0.436 ± 0.058(5)	0.490 ± 0.043(2)	0.442 ± 0.034(4)	0.491 ± 0.028(1)	0.464 ± 0.026(3)
Avg.rank	4.0 ± 0.87	3.5 ± 1.0	3.5 ± 1.41	2.0 ± 1.22	**2.0 ± 1.12**
Win.%	0.0	0.0	0.125	0.375	**0.5**

TABLE 7. *Hamming Loss* performance - indicating accuracy, via the number of misclassified labels (0.0 preferred).

Datasets	*BP-MLL*	*ML-HARAM*	*ANN*	LSIM	POL
birds	0.049 ± 0.005(3)	0.094 ± 0.011(5)	0.089 ± 0.014(4)	0.048 ± 0.004(1)	0.049 ± 0.003(2)
enron	0.463 ± 0.038(5)	0.074 ± 0.003(4)	0.047 ± 0.002(1)	0.048 ± 0.002(2)	0.058 ± 0.001(3)
emotions	0.469 ± 0.117(5)	0.403 ± 0.036(4)	0.292 ± 0.025(3)	0.195 ± 0.021(1)	0.198 ± 0.026(2)
medical	0.497 ± 0.038(5)	0.029 ± 0.003(4)	0.027 ± 0.002(3)	0.017 ± 0.001(2)	0.014 ± 0.002(1)
yeast	0.418 ± 0.043(5)	0.234 ± 0.007(4)	0.198 ± 0.007(2)	0.196 ± 0.005(1)	0.209 ± 0.008(3)
scene	0.416 ± 0.134(5)	0.286 ± 0.105(4)	0.100 ± 0.007(1)	0.207 ± 0.066(2)	0.238 ± 0.079(3)
cal500	0.460 ± 0.033(5)	0.180 ± 0.009(3)	0.140 ± 0.002(1)	0.141 ± 0.002(2)	0.197 ± 0.004(4)
foodtruck	0.333 ± 0.092(5)	0.156 ± 0.012(3)	0.173 ± 0.015(4)	0.154 ± 0.009(1)	0.155 ± 0.010(2)
Avg.rank	4.75 ± 0.66	3.88 ± 0.6	2.38 ± 1.22	**1.5 ± 0.5**	2.5 ± 0.87
Win.%	0.0	0.0	0.375	**0.5**	0.125

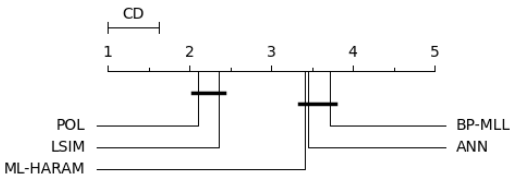


FIGURE 4. Summary comparison of method rank performances using posthoc analysis with Bonferroni-Dunn statistical test (CD = 0.6245, α = 0.05, k=5, N=80).

Aggregating the algorithms' average ranks over all metrics (10) and all datasets (8), gives the *average ranks* as [3.714, 3.416, 3.453, 2.364, 2.102]. Tables 4, 5, 6 and 7 show four out of the ten metric result sets. Similarly to [44], these *averaged ranks* enables a summary significance analysis of the algorithms' ranked performance. Therefore, the Bonferroni-Dunn test with $q\alpha$ = 2.498 at significance level α = 0.05 and thus CD = 0.6245 (k=5, N=80), is shown in Figure 4.

VI. DISCUSSION

Pattern-Loss was introduced as a replacement loss function to assist neural network training to obtain a generalized model. Our *pattern-loss* essentially interpolates the optimal network parameters (w) to converge toward label-patterns exhibited in the data. The conducted experiments used five different ANN-based techniques, including the two proposed methods, and benchmark methods over eight datasets with different data topologies to examine the robustness of the methods.

The overall classification performance of positive and negative label predictions – i.e. the lowest type I/II errors as determined by balanced precision/recall metric results are given by the F1 metric variants (see Tables 4,5,6). Over all datasets, *POL* has consistently highest average rank (Avg. Rank) and highest proportion of Win% (rank #1) in these categories compared to the other techniques. *LSIM* has highest average rank and Win% as measured by the least number of misclassified labels (Hamming Loss, see Table 7), followed by ANN (with binary cross entropy loss) and then *POL*.

The proposed pair of techniques (*POL* and *LSIM*) reported significant critical difference in the average rank summarizing Bonferroni-Dunn (BD) tests, over all metrics and datasets. Purely in terms of the classification performance measurements, the BD test reports significant preference for *POL* and *LSIM* loss functions when compared to the other state-of-art multi-label neural network techniques (BP-MLL, ML-HARAM), and when compared to the experiment control (ANN) *binary cross-entropy* loss function.

VII. CONCLUSION

This work proposed and evaluated two loss functions, named *POL* and *LSIM*, for classifying multi-label data using Artificial Neural Networks. The loss functions guide the learning optimization at the end of each training epoch by deriving *label pattern* measurements from the binary multi-label data. These force predictions towards existing patterns in the training data. POL loss is decomposed by two loss terms, Binary Cross Entropy loss (independent) and a regularized-weighted pattern-based loss. LSIM depends on cluster-based similarities of the binary *label patterns* and their corresponding instance data.

The paper reports on initial stage empirical trials with statistical benchmark analysis that indicate POL and LSIM (as a pair) rank significantly higher than three state-of-the-art methods across eight MLC datasets.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] S. A. Chandran and J. R. Panicker, "An efficient multi-label classification system using ensemble of classifiers," in *Proc. Int. Conf. Intell. Comput., Instrum. Control Technol. (ICICICT)*, Jul. 2017, pp. 1133–1136.
[2] P. Prajapati and A. Thakkar, "Performance improvement of extreme multi-label classification using K-way tree construction with parallel clustering algorithm," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 8, pp. 6354–6364, Sep. 2022.
[3] K. Poczeta, M. Płaza, T. Michno, M. Krechowicz, and M. Zawadzki, "A multi-label text message classification method designed for applications in call/contact centre systems," *Appl. Soft Comput.*, vol. 145, Sep. 2023, Art. no. 110562.
[4] D. Zeng, E. Zha, J. Kuang, and Y. Shen, "Multi-label text classification based on semantic-sensitive graph convolutional network," *Knowl.-Based Syst.*, vol. 284, Jan. 2024, Art. no. 111303.
[5] J. Xie, X. Li, Y. Yuan, Y. Guan, J. Jiang, X. Guo, and X. Peng, "Knowledge-based dynamic prompt learning for multi-label disease diagnosis," *Knowl.-Based Syst.*, vol. 286, Feb. 2024, Art. no. 111395.
[6] M. Sun, J. Niu, X. Yang, Y. Gu, and W. Zhang, "CEHMR: Curriculum learning enhanced hierarchical multi-label classification for medication recommendation," *Artif. Intell. Med.*, vol. 143, Sep. 2023, Art. no. 102613.
[7] E. Deniz, H. Erbay, and M. Coşar, "Multi-label classification of e-commerce customer reviews via machine learning," *Axioms*, vol. 11, no. 9, p. 436, Aug. 2022.
[8] X. Li, H. Wu, M. Li, and H. Liu, "Multi-label video classification via coupling attentional multiple instance learning with label relation graph," *Pattern Recognit. Lett.*, vol. 156, pp. 53–59, Apr. 2022.
[9] K. Zhang, W. Liang, P. Cao, X. Liu, J. Yang, and O. Zaiane, "Label correlation guided discriminative label feature learning for multi-label chest image classification," *Comput. Methods Programs Biomed.*, vol. 245, Mar. 2024, Art. no. 108032.
[10] W. Sangkatip and J. Phuboon-Ob, "Non-communicable diseases classification using multi-label learning techniques," in *Proc. 5th Int. Conf. Inf. Technol. (InCIT)*, Oct. 2020, pp. 17–21.
[11] M. E. Sánchez-Gutiérrez and P. P. González-Pérez, "Multi-class classification of medical data based on neural network pruning and information-entropy measures," *Entropy*, vol. 24, no. 2, p. 196, Jan. 2022.
[12] W. Wen, H. Zhang, Z. Wang, X. Gao, P. Wu, J. Lin, and N. Zeng, "Enhanced multi-label cardiology diagnosis with channel-wise recurrent fusion," *Comput. Biol. Med.*, vol. 171, Mar. 2024, Art. no. 108210.
[13] B. Swaminathan, M. Jagadeesh, and S. Vairavasundaram, "Multi-label classification for acoustic bird species detection using transfer learning approach," *Ecol. Informat.*, vol. 80, May 2024, Art. no. 102471.
[14] X. Li, J. Liu, Y. Xie, P. Gong, X. Zhang, and H. He, "MAGDRA: A multi-modal attention graph network with dynamic routing-by-agreement for multi-label emotion recognition," *Knowl.-Based Syst.*, vol. 283, Jan. 2024, Art. no. 111126.
[15] 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.
[16] A. Clare and R. D. King, "Knowledge discovery in multi-label phenotype data," in *Proc. Eur. Conf. Princ. Data Mining Knowl. Discovery*. Cham, Switzerland: Springer, 2001, pp. 42–53.

[17] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *Int. J. Data Warehousing Mining*, vol. 3, pp. 1–13, Jul. 2007.

[18] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern Recognit.*, vol. 40, no. 7, pp. 2038–2048, Jul. 2007.

[19] S. Godbole and S. Sarawagi, "Discriminative methods for multi-labeled classification," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, in Lecture Notes in Computer Science: Including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 3056, May 2004, pp. 22–30.

[20] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," in *Machine Learning and Knowledge Discovery in Databases* (Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 5782. Cham, Switzerland: Springer, Sep. 2009, pp. 254–269.

[21] G. Tsoumakas and I. Vlahavas, "Random k-labelsets: An ensemble method for multilabel classification," in *Machine Learning: ECML 2007*, J. N. Kok, J. Koronacki, R. L. de Mantaras, S. Matwin, D. Mladenič, and A. Skowron, Eds. Berlin, Germany: Springer, 2007, pp. 406–417.

[22] 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.

[23] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Mach. Learn.*, vol. 85, no. 3, pp. 333–359, Dec. 2011.

[24] D. Guozhu and L. Huan, *Feature Engineering for Machine Learning and Data Analytics*, 1st ed. Boca Raton, FL, USA: CRC Press, Jun. 2018.

[25] G. Hafeez, I. Khan, S. Jan, I. A. Shah, F. A. Khan, and A. Derhab, "A novel hybrid load forecasting framework with intelligent feature engineering and optimization algorithm in smart grid," *Appl. Energy*, vol. 299, Oct. 2021, Art. no. 117178.

[26] Z. Deng, S. Wang, and F.-L. Chung, "A minimax probabilistic approach to feature transformation for multi-class data," *Appl. Soft Comput.*, vol. 13, no. 1, pp. 116–127, Jan. 2013.

[27] W. Sangkatip and P. Chomphuwiset, "Improving multi-label classification using feature reconstruction methods," *Current Appl. Sci. Technol.*, vol. 23, no. 1, pp. 1–10, Apr. 2022.

[28] Y. Cheng, D. Zhao, Y. Wang, and G. Pei, "Multi-label learning with kernel extreme learning machine autoencoder," *Knowl.-Based Syst.*, vol. 178, pp. 1–10, Aug. 2019.

[29] J. Li, P. Li, X. Hu, and K. Yu, "Learning common and label-specific features for multi-label classification with correlation information," *Pattern Recognit.*, vol. 121, Jan. 2022, Art. no. 108259.

[30] S. Nazmi, X. Yan, A. Homaifar, and M. Anwar, "Multi-label classification with local pairwise and high-order label correlations using graph partitioning," *Knowl.-Based Syst.*, vol. 233, Dec. 2021, Art. no. 107414.

[31] Y. Xiao, Y. Li, J. Yuan, S. Guo, Y. Xiao, and Z. Li, "History-based attention in Seq2Seq model for multi-label text classification," *Knowl.-Based Syst.*, vol. 224, Jul. 2021, Art. no. 107094.

[32] M.-L. Zhang and L. Wu, "Lift: Multi-label learning with label-specific features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 1, pp. 107–120, Jan. 2015.

[33] J. Huang, G. Li, Q. Huang, and X. Wu, "Joint feature selection and classification for multilabel learning," *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 876–889, Mar. 2018.

[34] O. Luaces, J. Díez, J. Barranquero, J. J. del Coz, and A. Bahamonde, "Binary relevance efficacy for multilabel classification," *Prog. Artif. Intell.*, vol. 1, no. 4, pp. 303–313, Dec. 2012.

[35] X. Zhang, H. Zhao, S. Zhang, and R. Li, "A novel deep neural network model for multi-label chronic disease prediction," *Frontiers Genet.*, vol. 10, Apr. 2019, Art. no. 351.

[36] M. V. Valueva, N. N. Nagornov, P. A. Lyakhov, G. V. Valuev, and N. I. Chervyakov, "Application of the residue number system to reduce hardware costs of the convolutional neural network implementation," *Math. Comput. Simul.*, vol. 177, pp. 232–243, Nov. 2020.

[37] A. Maxwell, R. Li, B. Yang, H. Weng, A. Ou, H. Hong, Z. Zhou, P. Gong, and C. Zhang, "Deep learning architectures for multi-label classification of intelligent health risk prediction," *BMC Bioinf.*, vol. 18, no. S14, Dec. 2017, Art. no. 523.

[38] S.-M. Lian, J.-W. Liu, R.-K. Lu, and X.-L. Luo, "Captured multi-label relations via joint deep supervised autoencoder," *Appl. Soft Comput.*, vol. 74, pp. 709–728, Jan. 2019.

[39] C.-K. Yeh, W.-C. Wu, W.-J. Ko, and Y.-C. F. Wang, "Learning deep latent spaces for multi-label classification," in *Proc. 31st AAAI Conf. Artif. Intell.*, Feb. 2017, pp. 2838–2844.

[40] A. Law and A. Ghosh, "Multi-label classification using a cascade of stacked autoencoder and extreme learning machines," *Neurocomputing*, vol. 358, pp. 222–234, Sep. 2019.

[41] Z. C. Lipton, D. C. Kale, C. Elkan, and R. Wetzel, "Learning to diagnose with LSTM recurrent neural networks," in *Proc. 4th Int. Conf. Learn. Represent.*, Nov. 2016, Paper 1511.03677.

[42] S.-F. Chen, Y.-C. Chen, C.-K. Yeh, and Y.-C. F. Wang, "Order-free RNN with visual attention for multi-label classification," in *Proc. 32nd AAAI Conf. Artif. Intell.*, Dec. 2018, Paper 1707.05495.

[43] G. Nápoles, M. Bello, and Y. Salgueiro, "Long-term cognitive network-based architecture for multi-label classification," *Neural Netw.*, vol. 140, pp. 39–48, Aug. 2021.

[44] J. Mandziuk and A. Zychowski, "Dimensionality reduction in multilabel classification with neural networks," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2019, pp. 1–8.

[45] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 10, pp. 1338–1351, Oct. 2006.

[46] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas, "MULAN: A Java library for multi-label learning," *J. Mach. Learn. Res.*, vol. 12, pp. 2411–2414, Jul. 2011.

[47] A. Rivolli, L. C. Parker, and A. C. P. L. F. de Carvalho, "Food truck recommendation using multi-label classification," in *Progress in Artificial Intelligence*, E. Oliveira, J. Gama, Z. Vale, and H. L. Cardoso, Eds. Cham, Switzerland: Springer, 2017, pp. 585–596.

[48] F. Benites and E. Sapozhnikova, "HARAM: A hierarchical ARAM neural network for large-scale text classification," in *Proc. IEEE Int. Conf. Data Mining Workshop (ICDMW)*, Nov. 2015, pp. 847–854.

[49] F. Benites and E. Sapozhnikova, "Improving scalability of ART neural networks," *Neurocomputing*, vol. 230, pp. 219–229, Mar. 2017.

[50] P. Szymański and T. Kajdanowicz, "A scikit-based Python environment for performing multi-label classification," 2017, *arXiv:1702.01460*.

[51] X.-Z. Wu and Z.-H. Zhou, "A unified view of multi-label performance measures," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, Aug. 2017, pp. 3780–3788.

[52] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.

WORAWITH SANGKATIP received the B.Sc. degree in information communication and technology, the M.Sc. degree in information technology, and the Ph.D. degree in information technology from Mahasarakham University, Mahasarakham, Thailand, in 2010, 2013, and 2023, respectively. His current research interests include data mining, machine learning, deep learning, data analytics, and pattern classification.

PHATTHANAPHONG CHOMPHUWISET received the B.Sc. degree in computer science from Mahasarakham University, Thailand, in 2004, the M.Sc. degree in computer science from Asian Institute of Technology, Bangkok, Thailand, in 2008, and the Ph.D. degree in computing from the University of Leeds, U.K., in 2012. His research interests include computer science, including computer vision, pattern recognition, language understanding, medical image processing, and data mining.

KAVEEPOJ BUNLUEWONG received the bachelor's degree in computer science from Udon Thani Rajabhat Institute, Thailand, and the master's degree in computer science from Khon Kaen University, Thailand. Currently, he is a Lecturer with the Department of Computer Science, Faculty of Informatics, Mahasarakham University, Thailand. His research interests include artificial intelligence, machine learning, and big data.

SAKORN MEKRUKSAVANICH (Member, IEEE) received the B.Eng. degree in computer engineering from Chiang Mai University, in 1999, the M.S. degree in computer science from the King Mongkut's Institute of Technology Ladkrabang, in 2004, and the Ph.D. degree in computer engineering from Chulalongkorn University, in 2012. He is currently a Faculty Member of the Department of Computer Engineering, School of Information and Communication Technology, University of Phayao, Phayao, Thailand. His current research interests include deep and machine learning, applying machine learning techniques in the software engineering field, human activity recognition, and wearable sensors.

EMMANUEL OKAFOR received the B.Eng. degree in electrical engineering and the M.Sc. degree in control engineering from Ahmadu Bello University (ABU), Zaria, Nigeria, in 2010 and 2014, respectively, and the Ph.D. degree in artificial intelligence from the University of Groningen, The Netherlands, in 2019. He was a Beneficiary of the MIT-ETT Fellowship with the Massachusetts Institute of Technology (MIT), USA, in 2022. He has been an Academic Staff with ABU for the past ten years. He is currently a Postdoctoral Researcher with the SDAIA-KFUPM Joint Research Center for Artificial Intelligence (JRC-AI), King Fahd University of Petroleum and Minerals, Saudi Arabia.

OLARIK SURINTA received the Ph.D. degree in artificial intelligence from the University of Groningen, The Netherlands, in 2016. He is currently with the Department of Information Technology, Faculty of Informatics, Mahasarakham University, Thailand. He is also a Research Member of the Multi-Agent Intelligent Simulation Laboratory (MISL) Research Unit. His research interests include historical document analysis and recognition, pattern recognition, deep learning, machine learning, image and video classification, and image captioning.